

AD-A124 518

HEDGSIIM ROUTINES FOR LEADTIME VARIABILITY INVENTORY
POLICY RESEARCH(U) DECISION SYSTEMS BEAVERCREEK OH
W S DENNY SEP 81 WP-81-02 F33600-80-C-0530

1/1

UNCLASSIFIED

F/G 9/2

NL

RESEARCH

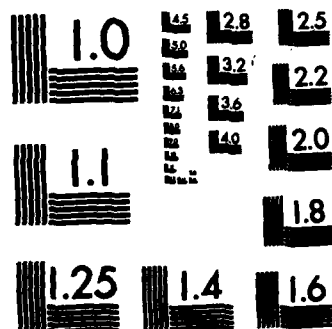
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

END
FILMED
IN
DTIC

ENC

F M MED

D11C



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Hedgesim Routines
for
Leadtime Variability
Inventory Policy Research
Demmy

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DTIC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

2

DOD LOGISTICS STUDY SUMMARY		1. LD NO.		2. DATE OF SUMMARY	
3. TITLE AND ACRONYM <i>WEDGESIM ROUTINES FOR LEADTIME VARIABILITY INVENTORY POLICY RESEARCH</i>					
4. STATUS <i>COMPLETED</i>		5. STARTING DATE <i>06/80</i>		6. COMPLETION DATE <i>09/81</i>	
7. ORGANIZATION REPORT NO. <i>WP-81-02</i>		8. DLSIE SEARCH NO.		9. CONTRACT NO. <i>F83600-80-C-053</i>	
10. TYPE STUDY <i>CONTRACT</i>		11. TIME FRAME		12. COST <i>35,000</i>	
13. SECURITY CLASS <i>UNCLASS</i>		14. DISTRIBUTION LIMITATION <i>UNLIMITED</i>		15. MAN-YEAR EFFORT <i>1 YR</i>	
16. STUDY SPONSOR RESPONSIBLE INDIVIDUAL AUTOVON COMMERCIAL NO.		17. PERFORMING ORGANIZATION <i>DECISION SYSTEMS 2125 CRYSTAL MARIE DR. BEAVER CREEK, OH 45431</i> RESPONSIBLE INDIVIDUAL <i>N. STEVEN DERRY</i> AUTOVON COMMERCIAL NO. <i>1-513-426-8515</i>			
18. ABSTRACT (UNCLASSIFIED) <i>THIS PAPER DOCUMENTS FORTRAN SOURCE CODE FOR SIMULATING GAMMA LEADTIMES AND NEGATIVE BEHEMOTH REQUISITION SIZES IN THE WEDGESIM LONG SUPPLY SIMULATION MODEL.</i>					
19. CONCLUSIONS (UNCLASSIFIED) DTIC FEB 16 1983					
20. RECOMMENDATIONS (UNCLASSIFIED) H					
21. IMPLEMENTING ACTIONS (UNCLASSIFIED) DISTRIBUTION STATEMENT 1 Approved for public release Distribution Unlimited					

HEDGSIM ROUTINES
for
Leadtime Variability
Inventory Policy Research

by
W. Steven Demmy

September 1981

WP-81-02
Decision Systems
2125 Crystal Marie Drive
Beavercreek, Ohio 45431
(513) 426-8515

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER	
	AD-A124518		
4. TITLE (and Subtitle) HEDGSIM Routines for Leadtime Variability Inventory Policy Research		5. TYPE OF REPORT & PERIOD COVERED INTERIM	
		6. PERFORMING ORG. REPORT NUMBER WP-81-02	
7. AUTHOR(s) W. Steven Demmy		8. CONTRACT OR GRANT NUMBER(s) F33600-80-C-0530	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Decision Systems 2125 Crystal Marie Drive Beavercreek, Ohio 45431		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS Directorate of Management Science Hq. AFLC/XRS Wright-Patterson Air Force Base, Ohio		12. REPORT DATE September 1981	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES	
		15. SECURITY CLASS. (of this report)	
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report)			
DISTRIBUTION STATEMENT A Approved for public release; Distribution Unlimited			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
18. SUPPLEMENTARY NOTES			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This paper documents Fortran Source Code for simulating Gamma Leadtimes and Negative Binomial requisition sizes in the HEDGSIM Long Supply Simulation Model.			

Table of Contents

I. Introduction

Appendix A. Job Control Language Programs

INV.OC.H
INVR.A

Appendix B. Fortran Source Listings

B.1. Forecasting and Levels Calculation Routines

FOR576
LEVELN
LPGFOR
SNBROP

B.2. Negative Binomial Requisition Size Routines

DEMPAR
NEGBIN
NEGBN1
RANDEM

B.3 Modified HEDGSIM Routines

INITEM
INORD
INVRSM
ENTERB



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	23 GP

Introduction

This paper documents a series of subroutines developed in support of inventory policy research concerning the impacts of lead time variability upon D062 Economic Order Quantity Items. The subroutines are designed to operate within the framework of the HEDGSIM Long Supply Simulation Model. Appendix A contains Job Control Language (JCL) statements required to utilize these new routines with other members of the HEDGSIM simulation program library. On the other hand, Appendix B presents the Fortran source program listings of new routines developed for this study.

The new routines documented in this paper may be assigned to one of three different categories. These are: (a) routines for proposed forecasting or safety level calculations, (b) routines for simulating negative binomial requisition sizes, and (c) modified HEDGSIM routines required to implement the above programs.

Let us now consider each of these categories.

Routines for Proposed Forecasting and Safety Level Calculations

This category includes subroutines FOR576, LEVELN, LPGFOR, and SNBROP. Subroutine FOR576 is the same as the original HEDGSIM routine with the addition of an outliers test when the forecasting

code ICDFOR = 3. When ICDFOR = 3, the subroutine checks if the largest observed demand exceeds the average demand plus four times the mean absolute deviation (computed using the remaining seven observations). If so, FOR576 assumes the large demand is an "outlier" that is not expected to occur again in future quarters. In this case, the forecasting calculations exclude the outlier data. Subroutines LPGFOR and SNBROP compute optimum reorder levels using the Logarithmic-Poisson-Gamma (LPG) and scaled negative binomial models, respectively. Subroutine LEVELN calls these subroutines and also provides other safety level computation options within the LEVELN logic. As noted above, listings of these programs may be found in Appendix B.

Routines for Negative Binomial Requisition Size Simulation

Subroutines in this category include DEMPARG, NEGBIN, NEGBN1, and RANDEM. These routines implement the generation of negative binomial requisition sizes using techniques described in Reference 2. Subroutine NEGBIN utilizes the twelve most recent observations of actual D062 demand histories to estimate the parameters of a negative binomial distribution of requisition size. It also constructs the cumulative distribution function (CDF) of this distribution. In turn, subroutine NEGBN1 utilizes this CDF to determine the specific requisition size associated with a randomly chosen percentile of the requisition size distribution. Subroutine RANDEM provides a uniform (0,1) random number stream which is

used exclusively in the demand generation process. This guarantees that exactly the same sequence of requisitions is generated for a given random number seed regardless of the management policy being evaluated. Finally, subroutine DEMPARG calls subroutines NEGBM1 and RANDEM as needed to generate the specific requisition streams associated with a given demand history. Subroutine NEGBIN is called to initialize the negative binomial requisition size distribution by subroutine INITEM, while the random number generator RANDEM is initialized by the MAIN program at the beginning of a simulation run. Subroutine DEMPARG also includes new logic to insure that very high activity items do not cause the future events list to overflow. With the new logic, after 450 requisitions have been generated, all remaining units of demand associated with a given quarter are placed on one (last) requisition. Only extremely high activity items are impacted by this change.

Modified HEDGSIM Routines

Subroutines included in this category are INITEM, INORD, INVRSM, and ENTERB. As noted above, these routines contain very slight modifications to the original HEDGSIM programs. Subroutine INITEM now contains a call to subroutine NEGBIN to initialize the negative binomial requisition size simulation procedure. Subroutine INORD has been modified to generate GAMMA distributed lead times with a coefficient variation equal to .353. Data

collected by Hayya (1980) indicates that such a distribution describes a number of D062 items (See Tables I-1 and I-2). The program INVRSM is essentially identical to the HEDGSIM MAIN program, excepted it now contains a call to initialize subroutine RANDEM.

When simulating high activity items with average requisition sizes that are close to one, very large numbers of requisitions may be generated within the simulation model. If policies are being simulated which offer very low levels of support, this can result in very large numbers of back orders. In turn this may cause the HEDGSIM backorder file to overflow. Subroutine ENTERB prevents this from happening by cancelling all new backorders once the backorder file is filled. With the present coding, once five hundred requisitions have been backordered, all subsequent requisitions are cancelled.

Table I-1

Coefficients of Variation of Lead Times
Observed by Hayya (1980)
for 62 High Activity Items

Coefficient of Variation	Number of Items
.1	8
.2	13
.3	15
.4	11
.5	6
.6	2
.7	2
.8	1
.9	3
1.0	<u>1</u>

N = 62

Median = .36

Table I-2

Gamma Probabilities for
Mean = 1 and Coefficient of Variation = .353

<u>x</u>	<u>P(x)</u>	<u>P(X ≤ x)</u>
.100	.000	.000
.200	.001	.001
.300	.007	.007
.400	.022	.030
.500	.048	.077
.600	.077	.154
.700	.101	.255
.800	.116	.371
.900	.119	.490
1.000	.112	.602
1.100	.098	.700
1.200	.081	.781
1.300	.054	.844
1.400	.048	.892
1.500	.035	.927
1.600	.025	.952
1.700	.017	.969
1.800	.011	.980
1.900	.007	.987
2.000	.005	.992
2.100	.003	.995
2.200	.002	.997
2.300	.001	.998
2.400	.001	.999
2.500	.000	.999
2.600	.000	1.000
2.700	.000	1.000

Note: Underlined values have been incorporated into Subroutine INORD.

References

1. Demmy, W. Steven, HEDGSIM: The Long Supply Simulation Model: Volume II, Program Listing and Narratives, Working Paper 80-10, Decision Systems, 2125 Crystal Marie Drive, Beavercreek, Oh 45431, Dec 1980, 97 pp.
2. Demmy, W. Steven, Modeling the Probability Distribution for Depot-Level Requisition Sizes, Working Paper 80-07, Decision Systems, 2125 Crystal Marie Drive, Beavercreek, Oh 45431, Oct 1980, 160 pp.
3. Hayya, Jack C., Lead Time Variability in Inventory Requirements Projections, Air Force Contract 33615-79-C-5143, Item 0004, Phase 3, Technical Report and Summary, 1962 Norwood Lane, State College, Pa, 16801, June 30, 1980, 71 pp.

APPENDIX A
JOB CONTROL LANGUAGE PROGRAMS

INV.OC.H

INVR.A

```

20      $      IDENT  UP1640,XRS-DENNY      INV.DC.H
30      $      LIMITS 70,,,10K
40      $      NOTE   *****INSH.JCL DATA FILE FOLLOWS THIS LINE
50      $      OPTION  FORTRAN,NONAP
60      $      SELECT  INVR/OBJ/INVRSM.O
65      $      SELECT  INVR/OBJ/SNBROP.O
70      $      SELECT  HEDG/OBJ/SIMULA.O
80      $      SELECT  HEDG/OBJ/REQ.O
90      $      SELECT  HEDG/OBJ/CUM.O
100     $      SELECT  HEDG/OBJ/CUMB.O
110     $      SELECT  HEDG/OBJ/DENPR3.O
120     $      SELECT  HEDG/OBJ/ENTR3.O
130     $      SELECT  HEDG/OBJ/FILLB0.O
140     $      SELECT  HEDG/OBJ/FOR573.O
150     $      SELECT  HEDG/OBJ/INITAL.O
160     $      SELECT  HEDG/OBJ/INITM3.O
170     $      SELECT  HEDG/OBJ/INORD3.O
180     $      SELECT  HEDG/OBJ/LEVLN3.O
190     $      SELECT  HEDG/OBJ/LONGSP.O
200     $      SELECT  HEDG/OBJ/ORDER.O
210     $      SELECT  HEDG/OBJ/OUTREP.O
220     $      SELECT  HEDG/OBJ/PFAC.O
230     $      SELECT  HEDG/OBJ/PLOTR.O
240     $      SELECT  HEDG/OBJ/RECEIV.O
250     $      SELECT  HEDG/OBJ/RET.O
260     $      SELECT  HEDG/OBJ/STATN2.O
270     $      SELECT  HEDG/OBJ/SSTAT2.O
280     $      SELECT  HEDG/OBJ/ZERO2.O
290     $      NOTE   *****PRED ROUTINES FOLLOW-----
300     $      NOTE   *****REQS ROUTINES FOLLOW THIS LINE
310     $      SELECT  REQS/STATUS.O
320     $      SELECT  REQS/FORUPD.O
330     $      SELECT  REQS/LEVEL.O
340     $      SELECT  REQS/ENTER.O
350     $      SELECT  REQS/REMOVE.O
360     $      SELECT  REQS/URIFEL.O
370     $      SELECT  REQS/INFEL.O
380     $      SELECT  REQS/RANDU.O
390     $      SELECT  REQS/8P.O
400     $      EXECUTE
410     $      LIMITS 70,40K,,10K
420     $      PRMFL  07/X1S,R,S,INVR/D0620C.H
430     $      FILE   08,X2S
440     $      FILE   09,A3S
450     $      NOTE   *****SELECTA HEDG/PFAC.O FOR PROGRAM FACTORS
460     $      DATA   04
470     $      SELECTA HEDG/PFAC.O

```

INV.DC.H

```

480      DATA      05
490      8060
500      0 1 0 1
510      0 0 0 0 0 0 0
520      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
530      7 2 38
540      2 1 2 8 1 2 1
550      6 36 0 99
560      .20 300. 500 19500 .5 1 C7
570      1 30 1 500 8 0 SIMULATE 500 ITEMS FOR 30 QTRS
580      2000
590      $ NOTE *****RINE/G0/PUNCH.E1
600      $ CONVER NSPIN
610      $ INPUT NMEDIA
620      $ OUTPUT MODBCD
630      $ FILE IN,A3S
640      $ PUNCH OT
650      $ ENBJOB

```

INV.OC.H

```

20      $ IDENT      WP1596,XRS-DEHMY      INVR.A
30      $ LIMITS    15,,,10K
40      $ NOTE      *****INSM.JCL DATA FILE FOLLOWS THIS LINE
50      $ OPTION    FORTRAN,NOMAP
60      $ SELECT    INVR/OBJ/INVRSM.O
65      $ SELECT    INVR/OBJ/SNBROP.O
70      $ SELECT    HEDG/OBJ/SIMULA.O
80      $ SELECT    HEDG/OBJ/REQ.O
90      $ SELECT    HEDG/OBJ/CUM.O
100     $ SELECT    HEDG/OBJ/CUMB.O
110     $ SELECT    HEDG/OBJ/DEHPR3.O
120     $ SELECT    HEDG/OBJ/ENTR3.O
130     $ SELECT    HEDG/OBJ/FILLB0.O
140     $ SELECT    HEDG/OBJ/FOR573.O
150     $ SELECT    HEDG/OBJ/INITAL.O
160     $ SELECT    HEDG/OBJ/INITM3.O
170     $ SELECT    HEDG/OBJ/INORD3.O
180     $ SELECT    HEDG/OBJ/LEVLN3.O
190     $ SELECT    HEDG/OBJ/LONGSP.O
200     $ SELECT    HEDG/OBJ/ORDER.O
210     $ SELECT    HEDG/OBJ/OUTREP.O
220     $ SELECT    HEDG/OBJ/PFAC.O
230     $ SELECT    HEDG/OBJ/PLOTR.O
240     $ SELECT    HEDG/OBJ/RECEIV.O
250     $ SELECT    HEDG/OBJ/RET.O
260     $ SELECT    HEDG/OBJ/STATN2.O
270     $ SELECT    HEDG/OBJ/SSTAT2.O
280     $ SELECT    HEDG/OBJ/ZERO2.O
290     $ NOTE      *****PRED ROUTINES FOLLOW-----
300     $ NOTE      *****REQS ROUTINES FOLLOW THIS LINE
310     $ SELECT    REQS/STATUS.O
320     $ SELECT    REQS/FORUPD.O
330     $ SELECT    REQS/LEVEL.O
340     $ SELECT    REQS/ENTER.O
350     $ SELECT    REQS/REMOVE.O
360     $ SELECT    REQS/WRIFEL.O
370     $ SELECT    REQS/INFEL.O
380     $ SELECT    REQS/RANDU.O
390     $ SELECT    REQS/GP.O
400     $ EXECUTE
410     $ LIMITS    15,40K,,,10K
420     $ PRNFL     07/X1S,R,S,INVR/D062SM.H
430     $ FILE      08,X2S
440     $ FILE      09,A3S
450     $ NOTE      *****SELECTA HEDG/PFAC.D FOR PROGRAM FACTORS
460     $ DATA     04
470     $ SELECTA   HEDG/PFAC.D1

```

INVR.A

```

480      DATA      05
490      9030
500      0 1 0 0
510      0 0 1 0 1      0 0
520      0 0 0 0 0      0 0 0 0 1      0 0 0 0 0      0 0 0 0 0
530      7 2 38
540      2 1 2 9 1      2 1
550      6 36      -99 99
560      .20 300. 500 19500 .5 1 C7
570      1 2 1 30 8 0      RUN 2 QTRS, 30 ITEMS
580      25
650      *      ENDJOB

```

INVR.A

FORECASTING AND LEVELS CALCULATIONS ROUTINES

FOR576

LEVELN

LPGFOR

SNBROP

CATALOG/FILE DESCRIPTION= HEDG/FOR573.S

```

10*#RUN=:HEDG/OBJ/FOR573.00BCD,N000)
20*FOR573.S
30      SUBROUTINE FOR576(N)
40C*****
50      PARAMETER N000=38
60C*****
70      COMMON/IF3 IG/IFBUG
80      COMMON/ITSMGC/ITSMGC
90      COMMON/ITIME/ITIME
100     COMMON/ITQTR/ITQTR
110     COMMON/NDEMAC/NDEMAC(1)
120     COMMON/NRETAC/NRETAC(1)
130     COMMON/NREQAC/NREQAC(1)
140     COMMON/NDEMND/NDEMND(1,N000)
150     COMMON/REQSIZ/REQSIZ(1)
160     COMMON/NRETUR/NRETUR(1,N000)
170     COMMON/NREQ/NREQ(1,N000)
180     COMMON/ADR/ADR(1)
190     COMMON/RSIGLT/RSIGLT(1)
200     COMMON/NDEMT/NDENT(1)
210     COMMON/LTADM/LTADM(1)
220     COMMON/LTPROD/LTPROD(1)
230     COMMON/UCOST/UCOST(1)
240     COMMON/IDBUG/IDBUG
250     COMMON/ICDFOR/ICDFOR
260     COMMON/ICDSIG/ICDSIG
270     LW=0
280     IDUM=0
290     IFUM=0
300     IRUM=0
310     KK=NDENT(N)
320     IF(KK.GT.8)KK=8
330     KL=KK-1
340     DO 1 I=1,KL
350     IDUM=IDUM + NDEMND(N,I)
360     IFUM=IFUM + NREQ(N,I)
370     1 IRUM=IRUM + NRETUR(N,I)
380C
390     IT=MOD(ITIME,ITQTR)
400     R=FLOAT(IT)/FLOAT(ITQTR)
410     RM=1.-R
420     DUM=R*FLOAT(NDEMAC(N))+FLOAT(IDUM)+RM*FLOAT(NDEMND(N,KK))
430     FUM=R*FLOAT(NREQAC(N))+FLOAT(IFUM)+RM*FLOAT(NREQ(N,KK))
440     RUM=R*FLOAT(NRETAC(N))+FLOAT(IRUM)+RM*FLOAT(NRETUR(N,KK))
450C
460C
470*
480***COMPUTE LEVELS BASED ON NET DEMANDS, WHERE NET IS GROSS
490***    DEMAND MINUS SERVICEABLE RETURNS FOR THE PAST 24
500***    MONTHS.
510***RETURNS ARE ADDITIONS TO ON-HAND ASSETS.

```

FOR573.S
FOR576

```

5300*
5310 FOR=0.01
5320 IF (IISMGC.GE. 3)DE=0.4-DE
5330 IF (DE.LE.0.)GO TO 1000
5340 FORCST=FORZELGAT(KK)
5350 ADRCTD=FORCST*4.
5360
5370**COMPUTE MAD OF QUARTERLY DEMAND
5380
5390 QMA=0.
5400
5410 IF IISMGC=>3.
5420     FORCST AND QMA ARE BASED ON NET QUARTLY DEMAND RATE
5430     OTHERWISE, BOTH ARE BASED ON GROSS DEMANDS
5440
5450 KZERO=0
5460 DMAX=-9999999.
5470 DO 20 I=1, KK
5480     DEM=NDEMAC(I)
5490     IF (IISMGC.GE. 3)DEA=DEM-NPETUR(N,I)
5500     IF (DEA.GT.0)KZERO=KZERO+1
5510     IF (DEA.LE.0)MAX=DEA
5520     DMAX=DEA
5530     IDMAX=I
5540 10 CONTINUE
5550     ABSDEV=ABS(DEA-FORCST)
5560     QMA=QMA+ABSDEV
5570 20 CONTINUE
5580
5590 DE=NDEMAC(1)
5600 IF (IISMGC.GE. 3)DE=DE-NPETUR(N,1)
5610 QMA=ABS(DEA-FORCST)*R
5620 QMA=(QMA+QMA)/(FLOAT(KK)+R)
5630 IF (QMA.LE. 0.01) QMA=0.01
5640
5650 IF FORECAST CODE=3, BACK OUT THE LARGEST
5660     DEMAND THAT EXCEEDS 4 MADs.
5670
5680 IF (ICDFOR.NE. 3)GO TO 40
5690
5700 CHECK IF FILTER IS TO BE USED
5710
5720 IF (KZERO.LE. 3)GO TO 40
5730 IF (QMA.LE. 1.)GO TO 40
5740
5750 IF DMAX>TFOR+1*MAD.
5760     THEN BACK OUT BIGGEST DEMAND.
5770
5780
5790 TFOR=(3.*FORCST-IDMAX)/7.
5800
5810
5820

```

```

1030C      COMPUTE 7-PI MAD
1040C
1050      TMAD=0.
1060      DO 30 I=1, KK
1070          IF(I.EQ.1)DMAX GO TO 30
1080          DEM=NDEMAND(N,I)
1090          IF(IISMGZ.GE. 3) DEM=DEM-NRETUR(N,I)
1100          TMAD=TMAD + ABS( DEM -TFOR)
1110      30 CONTINUE
1120      TMAD = TMAD / I.
1130C
1140C
1150C      IS DMAX AN OUTLIER
1160C
1170      IF(DMAX.LE.(TFOR+4.*TMAD))GO TO 40
1180C
1190C      USE BACKED-OUT VALUES
1200C
1210      FORCST=TFOR
1220      ADR(N) =4.*FORCST
1230      QMAD=TMAD
1240      IF(IFBUG.GE.1)WRITE(6,33)DMAX
1250      33 FORMAT('****FOR576--BACK OUT OUTLIER OF ',F8.1,' UNITS')
1260C
1270      40 CONTINUE
1280C
1290C**COMPUTE STANDARD DEVIATION OF LEADTIME DEMAND
1300C
1310      RLT=FLOAT(LTADM(N)+LTPMOD(N))
1320      RSIGLT(N)=0.5945*QMAD*(0.823/5+0.42625*RLT)
1330      IF(RSIGLT(N).LT. 0.01) RSIGLT(N)= 0.01
1340C
1350C**COMPUTE AVERAGE REQUISITION SIZE
1360C
1370      IF(FUM.LT.1)FUM=1
1380      R=DUM/FUM
1390      IF(R.LT.1.)R=1.
1400      REQSIZ(N)=R
1410      GO TO 2000
1420C

```

```

1430C          TREATMENT FOR ZERO-DEMAND ITEMS
1440C
1450 1000 RSIGLT(N)=.5
1460        ADR(N)=0. )
1470        REQSIZ(N)=1.
1480        GO TO 200 )
1490 2000 IF (IFBUG.LT. 1) RETURN
1500        WRITE(LW,100)
1510        WRITE(LW,200) N,NDENT(N),ICDFOR,ICDSIG,LTADM(N),LTPROD(N)
1520        WRITE(LW,300)
1530        WRITE(LW,200)NDEMAC(N),(NDEMND(N,J),J=1,KK)
1540        WRITE(LW,200)NRETAC(N),(NRETR(N,J),J=1,KK)
1550        WRITE(LW,200)NREQAC(N),(NREQ(N,J),J=1,KK)
1560        WRITE(LW,400)
1570        WRITE(LW,500) FORCST,ADR(N),REQSIZ(N),UCOST(N),QDAD,RSIGLT(N)
1580 100        FORMAT(1H,4X,'*****FOR573--N,NDENT,ICDFOR,ICDSIG',
15902        ' LTADM,LTPROD')
1600 200 FORMAT(1H,4X,12I7/12I7)
1610 300        FORMAT(1H,9X,'DEMAND, RETURNS, AND FREQ.',
16202        ' IN CURRENT HISTORY FILE')
1630 400        FORMAT(1H,' FORCST          ADR    AVE-REQ    UCOST',
16402        ' QDAD    RSIGLT')
1650 500        FORMAT(1H,6F10.4)
1660        RETURN
1670        END

```

CATALOG/FILE DESCRIPTION= HEDGZ/LEVELN3.S

10*#RUN=HEDGZ/OBJ/LEVELN3.006CL,N0000

20*LEVELN3.S

30 SUBROUTINE LEVELN(N)

400

500 THIS ROUTINE COMPUTES REORDER, STOCK OBJECTIVE, RETENTION,
600 TERMINATION, AND SUPPORT LEVELS.

700

80 PARAMETER N000=33

900

1000

110 COMMON/THBUG/THBUG

120 COMMON/ICDEFOR/ICDEFOR

130 COMMON/COSHRT/COSHRT

140 COMMON/COSHLN/COSHLN

150 COMMON/COSORD/COSORD(3)

160 COMMON/CSTBRK/CSTBRK

170 COMMON/GSULF/GSULF

180 COMMON/GRLF/GRLF

190 COMMON/GSLF/GSLF

200 COMMON/GTLF/GTLF

210 COMMON/ITINV/ITINV

220 COMMON/ITLEVL/ITLEVL

230 COMMON/IDLEVL/IDLEVL

240 COMMON/NITEM/NITEM

250 COMMON/NDHIS/NDHIS

260 COMMON/POLICY/ICDEQ,ICDSL,EQMAX,EQMIN,SLMAX,SLMIN,RLF,TLF,SUL

270 COMMON/ICDSL/ICDSL

280 COMMON/ADR/ADR(1)

290 COMMON/LTADM/LTADM(1)

300 COMMON/LTPROD/LTPROD(1)

310 COMMON/IQTY/IQTY(1)

320 COMMON/IRL/IRL(1)

330 COMMON/IROL/IROL(1)

340 COMMON/ISUL/ISUL(1)

350 COMMON/ISLEVL/ISLEVL(1)

360 COMMON/ITL/ITL(1)

370 COMMON/RSIGLT/RSIGLT(1)

380 COMMON/REQSIZ/REQSIZ(1)

390 COMMON/UCOST/UCOST(1)

400 COMMON/PE/PE(NQ00),PFA(NQ00,30),ACODE(30),NCODE,AIRCRF

410 COMMON/IISMGC/IISMGC

420 DATA Z/1.0/

430 DATA ITLMIN,ITLMIN/99999999,99999999/

4400

4500 COMPUTE PLANNING FACTORS

4600

470 AMDA=COSHRT

480 RLTM=LTADM(N)+LTPROD(N)

490 IF(RLTM.LT.1.)RLTM=0.5

LEVELN3.S
LEVELN

```

500C
510      AD=ADP(ND)
520      SIG = RSIGLTCND)
530C
540C      DETERMINE PROGRAM FACTOR PFNOW
550C
560      IOTR=IITNV
570      PFNOW=1.00
580C
590C      IF IOTR FOR .EQ. 2, ADJUST FOR FORECAST PROGRAM CHANGE
600C
610      IF(IOTR FOR .NE. 2) GO TO 20
620C
630      PFNOW = PFC(IOTR + NDRIS )
640C
650      AD = PFNOW * AD
660      SIG=SIG * (PFNOW ** 0.85)
670C
680C
690      20 CONTINUE
700C
710      RLT=AD*RLTM/12.
720      RMR=AD/12.
730      UC=UCOST(1)
740      ADDR=UC*AD
750C
760C      SET COR EQUAL TO SMALL PURCHASE ORDER COST
770      COR=COSORD(1)
780C
790C      BRANCH BY FOR FORMULA      CODE
800C
810      GO TO (110,120,130,140,150,160,170,180,190),ICDEF00
820C
830C      2 FEB      /O GUIDELINE LTR POLICY
840C
850      110      IF(ADDR.GE.1000.) GO TO 62
860      Q=12.*RMR
870      GO TO 200
880      62      IF(ADDR.GE.5000.) GO TO 64
890      Q=6.*RMR
900      GO TO 200
910      64      Q=3.*RMR
920      GO TO 200
930C
940C      AFLC 57-6 POLICY---DEC 1976
950C
960      120      Q=SQRT(2.*COR*AD/(COSHLD*UC))
970      ACOST=Q*UC
980      IF(ACOST.LT.CSTBRK) GO TO 200
990      COR=COSORD(2)
1000      Q=SQRT(2.*COR*AD/(COSHLD*UC))
1010      GO TO 200

```

LEVLN3.S
 LEVELN

```

1020 130      CONTINUE
1030 140      CONTINUE
1040 150      CONTINUE
1050 160      CONTINUE
1060 170      CONTINUE
1070 180      CONTINUE
1080 190      CONTINUE
1090          WRITE(6,8195)ICDF00,ICDSL
1100 8195     FORMAT('***LEVELN--UNDEFINED FORMULA CODE....',
1110         ' ICDF00=',I3,' ICDSL=',I3)
1120          STOP
1130C
1140C          CHECK      ORDER SIZE LIMITS
1150 200      CONTINUE
1160          EMX=EQQMAX*RMR
1170          IF(O.GT.EMX) Q=EMX
1180          EMX=EQQMIN*RMR
1190          IF(O.LT.FMX) Q=FMX
1200          IF(O.LT.1.)Q= 1.
1210C
1220C          BRANCH BY SAFETY LEVEL FORMULA CODE ICDSL
1230C
1240          GO TO (310,320,330,340,350,360,370,380,390),ICDSL
1250C
1260C          USE 12 MONTH SUPPLY
1270C
1280 310      CONTINUE
1290          Q= 12.*RMR
1300C
1310          GO TO 500
1320C
1330C          23 AUG 68 GUIDELINE LTR POLICY
1340C
1350 320      SL=0.25*RLT
1360          GO TO 500
1370C
1380C          AFLCM      57-6 POLICY
1390C
1400 330          Z=SQRT(REQSIZ(N))
1410          GO TO 358
1420C
1430C          PT-FORMULA TO MINIMIZE UNITS BACKORDERED
1440C
1450 340          Z=1.
1460          GO TO 358
1470C

```

LEVLN3.S
 LEVELN

```

1480C          PT-FORMULA TO MINIMIZE REQUISITIONS BACKORDERED
1490C
1500  350      Z=REFSIZ(N)
1510C
1520C
1530  358  CONTINUE
1540      IF (SIG.LE.0.)SIG=,0001
1550      QSIG=-1.414*Q/SIG
1560      RNUM=AMDA*SIG
1570      IF ((QSIG.LT.0.).AND.(QSIG.GT.-30.))
1580R      RNUM=AMDA*SIG*(1.-EXP(QSIG))
1590      DNOM=2.*COSHLN*UC*/1.414*Q
1600      RK=0.707*ALOG(RNUM/DNOM)
1610      SL=RK*SIG
1620      GO TO 500
1630C
1640C
1650C          MODIFIED STD.DEV. FOR ACTUAL GAMMA LEAD TIME VARIABILITY
1660C
1670C          ASSUME COEF OF VARIATION =.353. AND 1 PERIOD =EXPECTED LEA
1680C
1690  360  CONTINUE
1700      RMEANL=1.
1710      SIGL=.353*RMEANL
1720      VARL=SIGL**2
1730      VARD=SIG**2
1740      SIGOLD=SIG
1750C
1760C          REVISE SIG FOR LT VARIABILITY
1770C
1780      SIG=SQRT(RMEANL*VARD+(RLT**2)*VARL)
1790      IF (IHBUG.GT.0)WRITE(6,363)SIGOLD,SIG,SIG/SIGOLD
1800C
1810  363  FORMAT(T20,"SIGOLD=",F10.1,"    SIG LTD=",F10.1,"    RATIO=",F10
1820C
1830C          NOW USE D062 SQRT(Z)FORMULA
1840C
1850      GO TO 330
1860C
1870*****
1880C
1890C          SCALED NEGATIVE BINOMIAL MODEL
1900C
1910  370  CONTINUE
1920C

```

```

1850      IROP = 0
1860      IF(AD.LE.0.) GO TO 377
1870      IF(AMDA.LE.0.) GO TO 377
1880
1890      COMPUTE CRITICAL STOCK STOCK OUT PROBABILITY
1900      POUT=(1*CSHLD*UC)/(AMDA*AD)
1910
1920      FIND CRITICAL FILL PROBABILITY
1930      FCRIT=1.-POUT
1940      IF(FCRIT.LE.0.) GO TO 377
1950
1960      AVERAGE REQUISITION SIZE
1970      AVEREQ=REQSIZ(N)
1980
1990      CUSTOMERS PER PERIOD
2000      ED=RMR
2010
2020      E(LT)
2030      ELT=FLOAT(LTADM(N) + LTPROD(N) )
2040
2050      ASSUME COEF. OF VAR. OF LT=.353
2060      CVLT=.353
2070
2080      FIND SCALED NEG. BIN. REORDER POINT
2090      CALL SNBROP(AVEREQ,ED,ELT,CVLT,FCRIT,IROP)
2100
2110      BACK INTO THE SAFETY LEVEL
2120
2130      377 CONTINUE
2140      SL=IROP-RLT
2150
2160
2170      SKIP SL LIMITS
2180
2190      GO TO 600
2200*FILE EXPFIT.S
2210-----
2220      EXPONENTIAL APPROXIMATION MODEL
2230
2240      380 CONTINUE
2250      IF(AD.LE.0.) GO TO 387
2260      IF(AMDA.LE.0.) GO TO 387
2270
2280
2290
2300
2310
2320
2330
2340
2350
2360
2370
2380
2390
2400
2410
2420

```

LEVLN3.S
LEVELN

```

24300      COMPUTE OPTIMAL STOCKOUT PROBABILITY
24400
24500      POUT = (Q*COSHLD*UC)/(AMDA*AD)
24600
24700      LIMIT OPTIMUM FILL PROBABILITY
24800
24900      PFILL = 1.-POUT
25000
25100      IF(PFILL.LE.0.) GO TO 337
25200      IF(PFILL.GT. 0.9999) PFILL = 0.9999
25300
25400      ASSUME P(Z <= ZT) = F(Z) = 1.-.331 *EXP( B*ZT)  FOR ZT >
25500
25600      ZT = 0.
25700      IF(PFILL .GT. 0.9999) ZT = ALOGC( (1.-PFILL)/.331 )/(-.463)
25800
25900      COMPUTE QMAD, AND SAFETY LEVEL FROM MAD
26000
26100      SINCE RESULT = .5245*QMAD*(.8235+.42625*RLTM), WE HAVE
26200
26300      QMAD = SLGZ(.5245*(.82375+.42625*RLTM) )
26400
26500      NOW COMPUTE SAFETY LEVEL
26600
26700      SL = ZT * QMAD * SQRT (RLTM)
26800
26900
27000      SKIP THE LIMIT CALCULATION
27100
27200      GO TO 600
27300
27400      SET SL TO ZERO
27500
27600 387 CONTINUE
27700      SL = 0.
27800      GO TO 600
27900
28000-----
28100 390      CONTINUE
28200
28300      USE SPLIT-EXPONENTIAL-GAMMA MODEL
28400-----
28500      IF(AD.LE.0.) GO TO 397
28600      IF(AMDA.LE.0.) GO TO 397
28700
28800      COMPUTE OPTIMAL STOCKOUT PROBABILITY
28900
29000      POUT = (Q*COSHLD*UC)/(AMDA*AD)
29100

```

LEVLN3.S
 LEVELN

```

2920C          LIMIT OPTIMUM FILL PROBABILITY
2930C
2940C          PFILL = 1.-POUT
2950C
2960C          IF(PFILL.LE.0.) GO TO 397
2970C          IF(PFILL.GT. 0.9999) PFILL = 0.9999
2980C
2990C
3000C          ESTIMATE QUARTERLY DEMAND RATE, MAD, AND LEADTIME
3010C
3020C          SINCE RSIGLT = .5945*QDAD*(.8235+.42625*RLTM), WE HAVE
3030C
3040C          QDAD = SIG/ (.5945*(.8235+.42625*RLTM) )
3050C          CRATE = AD/4.
3060C          QTRLT = RLTM/3.
3070C
3080C          NOW COMPUTE REORDER LEVEL
3090C
3100C          CALL EXPDAM(CRATE,QDAD,QTRLT,PFILL,IROP)
3110C
3120C          IF(IROP.LE.0) IROP=0
3130C          SL=IROP-RLT
3140C
3150C
3160C          SKIP THE LIMIT CALCULATION
3170C
3180C          GO TO 600
3190C
3200C          SET ROP TO ZERO
3210C
3220C          397 CONTINUE
3230C          IROP=0
3240C          SL=IROP-RLT
3250C          GO TO 600
3260C
3270C*****
3280C*****
3290C 500          CONTINUE
3300C
3310C          BRANCH BY SAFETY LEVEL LIMIT CODE, ICDSLL
3320C
3330C          GO TO (510,520),ICDSLL
3340C
3350C          DO62 LIMITS, AS OF JUNE 77
3360C
3370C 510          CONTINUE
3380C

```

LEVLN3.S
 LEVELN

```

3390C          LIMIT SAFETY LEVEL
3400C
3410C
3420C          LIMITS SL TO LESS THAN LEADTIME DEMAND
3430C
3440C          SLM=RL1
3450C          IF(SL.GT. SLM)SL=SLM
3460C
3470C          LIMIT SL TO LESS THAN 3*SIG
3480C
3490C          SLM=3.*SIG
3500C          IF(SL.GT. SLM)SL=SLM
3510**
3520C
3530C          LIMITS BASED ON MONTHS OF SUPPLY
3540C
3550C  520      CONTINUE
3560C          LOWER LIMIT
3570C          SLM=SLM1*RMR
3580C          IF(SL.LT. SLM)SL=SLM
3590C          UPPER LIMIT
3600C          SLM=SLMAX*RMR
3610C          IF(SL.GT. SLM)SL=SLM
3620C
3630C
3640C          GO TO 600
3650C *****
3660C
3670C  600      CONTINUE
3680C
3690C          COMPUTE LEVELS
3700C
3710C          ISLEVL(N)=SL+0.5
3720C          IROTY(N)=Q+0.5
3730C          IROL(N)=SL+RLT+0.5
3740C          IF(IROL(N).LT.0) IROL(N) = 0
3750C          ITL(N) =SLMAX*RMR+RLT+GTLF*RMR+0.5
3760C          IF(ITL(N).LT.0) ITL(N)=0
3770C          IRL(N)=FLOAT(ITL(N))+GRLF*RMR+0.5
3780C          IF(ITL(N).LE.ITLMIN) ITL(N)=ITLMIN
3790C          IF(IRL(N).LE.IRLMIN) IRL(N)=IRLMIN
3800C          ISUL(N)=GSULF*RMR+0.5
3810C  3000      CONTINUE
3820C          ROL=FLOAT(IROL(N))
3830C          IF(IHBUG.LT.1) RETURN
3840C          RK=0.
3850C          IF(SIG.GT.0.) RK = SL/SIG
3860C          WRITE(6,8903)N, IROTY(N), IROL(N), ITL(N), IRL(N), ISUL(N), RK,
3870C          SL, PFNOW, AD, PFILL
3880C  8903      FORMAT(4X, '****LEVELN--N=', I5, ' IROTY=', I5, ' IROL=', I5,
3890C          ' ITL=', I5, ' IRL=', I5, ' ISUL=', I5, ' RK=', F9.4,
3900C          ', T50, ' SL=', F9.1, ' PFNOW=', F6.3, ' AD=', F9.1
3910C          ' PFILL =', F6.3 )
3920C
3930C          RETURN
3940C          END

```

LEVELN3.S
LEVELN

```

4750 SUBROUTINE EXPLTD(X,R,QMAD,OTRLT,CUMPX)
4760
4770 COMMON/INT/INT(20)
4780
4790 SET WRITE FLAGS
4800
4810 IDEFL = INT(10)
4820 IPNTSZ = INT(11)
4830 CUMPT=0.
4840 CUMPX=0.
4850
4860 INITIALIZE PDF PARAMETERS
4870
4880 A1= 0.331
4890 B1 = -0.463
4900
4910 A2= 0.069
4920 B2= 0.7979
4930
4940 GAMMA CONSTANT FOR MEAN=1 AND CV=.353
4950 C1= 0.0015373
4960
4970
4980 -----
4990
5000
5010 INITIALIZE FOR T INTEGRATION
5020
5030 DT = .1
5040 T = DT
5050 CUMPT=0.
5060 CUMPX=0.
5070
5080 BEGIN "T" INTEGRATION LOOP
5090
5100 DO 100 I=1,100
5110
5120 COMPUTE STANDARDIZED ERROR ZT
5130
5140 TOTR= T*OTRLT
5150 ZT = (X - R*TOTR) / (QMAD*SQRT(TOTR) )
5160

```

LEVLN3.S
EXPLTD

```

5170      COMPUTE P( T )
5180
5190      GT = C1*(B.*T)**/* ( EXP(-B.*T) * DT
5200
5210      COMPUTE P( Z <= ZT & T )
5220
5230      IF( ZT.LE.0.) PZ = A2*EXP(B2*ZT)
5240      IF( ZT.GT.0.) PZ = 1. - A1*EXP( B1*ZT)
5250
5260      COMPUTE P( X <= X & T ) P( T )
5270
5280      PXT = PZ*GT
5290
5300      UPDATE CUMULATIVE PROBABILITIES
5310
5320      CUMPT = CUMPT + GT
5330      CUMPX = CUMPX + PXT
5340
5350      IPRNT=0
5360      IF( (IPNTS2.GT.0) .AND. (MOD(I,IPNTS2) .EQ. 0) )
5370          IPRNT=1
5380      IF(IDEFL.LE.0) IPRNT=0
5390      IF(IPRNT.GT.0) WRITE(6,63) X,T,GT,PXT,CUMPT,CUMPX,ZT,PZ
5400  63 FORMAT(2F3.2,6F10.4)
5410
5420
5430      INCREMENT T
5440
5450      T = T+DT
5460
5470      STOP IF CUMPT > .999
5480
5490      IF(CUMPT.GT. 0.999) GO TO 120
5500
5510 -----END OF "T" LOOP-----
5520 100 CONTINUE
5530
5540 120 CONTINUE
5550      RETURN
5560      END

```

LEVLN3.S
 EXPLTD

```

3950 SUBROUTINE EXPGAM(ORATE,QMAD,TOTR,PFILL,IROP)
3960
3970 COMPUTE A RE-ORDER POINT (IROP) TO GIVE
3980 A FILL RATE OF PFILL.
3990
4000 COMMON/IWT/IWT(20)
4010
4020 SET LOW POINT FOR P.D.F
4030
4040 QMADLT = QMAD*SQRT(TOTR)
4050 XLOW = ORATE*TOTR - 5.*QMADLT
4060 XLOW = IFIX(XLOW + 0.5)
4070 IF(XLOW.LT.0.) XLOW = 0.
4080 CALL EXPLT(XLOW,ORATE,QMAD,TOTR,CUMPX)
4090
4100 IF(PFILL.GT.CUMPX) GO TO 500
4110
4120 VERY LOW SUPPORT NEEDED.
4130 SET REORDER POINT TO XLOW
4140
4150 IROP = XLOW
4160 RETURN
4170
4180 SET LOW POINT FOR SEARCH
4190
4200 500 CONTINUE
4210 PLOW= CUMPX
4220
4230 COMPUTE DELTA-X
4240
4250 DX = QMADLT
4260 IF(DX.LT.1.) DX = 1.
4270 IF(DX.GT.1.) DX = IFIX(DX + 0.5)
4280
4290 INITILIZE HIGH VALUES FOR SEARCH
4300
4310 X= XLOW + DX
4320
4330 FIND P(DLT <= X)
4340
4350 510 CONTINUE

```

```

4360 CALL EXPLD(X,DRATE,QMAD,TOTD,CUMPX)
4370 IF (INT(10).GT.0) WRITE(6,523)X,CUMPX,XLOW,PLOW
4380 523 FORMAT("---EXPGAM. X=",F8.1," CUMPX=",F8.4,
4390 " XLOW =",F8.1," PLOW =",F8.4)
4400
4410
4420 HAVE WE BOUNDED THE DESIRED FILL RATE
4430 IF SO, GO TO 600 AND INTERPOLATE FOR ROP.
4440
4450 IF(PFILL.LE. CUMPX ) GO TO 600
4460
4470 IF CUMPX > .99, STOP ANYWAY.
4480
4490 IF(CUMPX.GT. 0.99) GO TO 580
4500
4510 NO, WE HAVE TO KEEP TRYING. RESET FOR SEARCH
4520
4530 XLOW = X
4540 PLOW = CUMPX
4550
4560 X= X + DX
4570 GO TO 510
4580
4590 SET IROP = THIS X
4600
4610 580 CONTINUE
4620 IROP = X+ 0.5
4630 RETURN
4640
4650 THE ROP IS BOUNDED. NOW INTERPOLATE.
4660
4670 600 CONTINUE
4680
4690  $X = XLOW + (PFILL - PLOW) * (X - XLOW) / (CUMPX - PLOW)$ 
4700
4710 IROP = IFIX( X + 0.5)
4720
4730 RETURN
4740 END

```

LEVLN3.S
EXPGAM

CATALOG/FILE DESCRIPTION= INVR/LPGFOR.S

5* FILE LPGFOR.S

06C

7C

FORTTRAN VERSION OF LPGFOR.S

08C

10C

15

CHARACTER ADOLR*1

16C

20C

22C

25

CHARACTER ADOLR*1

30

COMMON/LPGPRM/T1,RL1,A1,B1,ID7,ID8

40C

50C

T(K+1) = T(X,K) FOR THE CURRENT X

60C

T2(K+1) = T(X-1,K) FROM PREVIOUS X CALCULATION

70C

80C

90C

READ(5,13) PARAMETERS

91C

92 90 CONTINUE

92

PRINT , "IDY, ID8 "

93

READ(5,13) ID8, ID7

94C

95 13 FORMAT (V)

100

CALL LPGO

110C

DO LPG RECURSION

120

CALL LPGREC

130C

DO SCALED NEG BINOMIAL CALCULATIONS

140

CALL LPG3

142C

143C

CALL NAHMIAS EXACT LPG ROUTINE

144C

145

CALL LPG1

146C

150

WRITE(6,13) "CONTINUE (Y OR N)"

155

READ(5,13) ADOLR

160

IF(ADOLR.EQ."Y")GO TO 90

170

STOP

171

END

172C

173C

174

SUBROUTINE LPGO

175

COMMON/LPGPRM/T1,RL1,A1,B1,ID7,ID8

177 13 FORMAT(V)

178C

179

DIMENSION T(200),T2(200)

180C

181

WRITE(6,13)

182

WRITE(6,13)

190

WRITE(6,13) "LPGO.S--EXACT LPG PROB CALCULATIONS USING RECURSION"

191

WRITE(6,13)

192

WRITE(6,13)

```

200C
210 WRITE(6,13) "HEAD(5,13) AVE REQ, ECD, ECLD, C.OF.V OF LI "
220 HEAD(5,13) RO,DI,EI,C
230 WRITE (3,13) RO,DI,EI,C
240C
250C SOLVE FOR THETA = T1
260 CALL BSRCB (RO,0)
270 T1=0
280C
290C ESTIMATE PARAMETERS FOR LPG
300C S = STD DEV, B = VAR TO MEAN RATIO
310 S = C*EI
320 B = S*S/EI
330 U1= DI*EI
340C
350 B1=1/B
360 A1 =B1*EI
370 R1=-T1/( (1-T1) *ALOG(1-T1) )
380 RL1 = U1/(EI*R1)
390C
400 WRITE(6,13) "AVE REQ SIZE =",R1
410 WRITE(6,13)
420 WRITE(6,13) "THEIA =",T1,"LAMBDA =",RL1
430 WRITE(6,13) "ALPHA=", A1,"BETA =",B1
440C
450C
460 RM1=-T1/((1-T1)*ALOG(1-T1)*RL1*A1/B1)
470 WRITE(6,13)
480 WRITE(6,13) "MEAN =",RM1
490 WRITE(6,13)
500C ESTIMATE THE FIRST FOUR MOMENTS RM1,RM2,RM3,RM4 OF THE LPG DISTRIB
510 D1 = B1*(1-T1)
520 C4 = -RL1*T1/(ALOG(1.-T1))
530C
540C
550 RM1 = A1*C4/D1
560C
570 RM2 = A1*C4*(B1+C4)/D1**2
580C
590 RM3 = A1*C4*((B1**2)*(1+T1) +2*C4**2)/D1**3
600C
610 RM4 = B1**3*(1+4*T1+T1**2) + B1**2*C4*(3.*A1 + 1.)
620 RM4 = RM4 + 6*B1*C4*C4*A1 +C4**3*(3*A1+6.)
630 RM4 = A1*C4*RM4/D1**4
640C
650 WRITE(6,13) "MOMENTS"
660 WRITE(6,13) "RM1 RM2 RM3 RM4"
670 WRITE(6,13) RM1,RM2,RM3,RM4
680C
690C COMPUTE STANDARDIZED MOMENTS
700 WRITE(6,13)
710 S = SQR(RM2)
720 WRITE(6,13) "COEF OF VAR =", S/RM1
730 WRITE(6,13) "RM3/S**3= ", RM3/S**3
740 WRITE(6,13) "RM4/S**4= ", RM4/S**4
750C
760C OUTPUT LPG PARAMETERS TO FILE
770C
780 WRITE (8,13) RM1,S/RM1,RM3/S**3,RM4/S**4
790 WRITE (8,13) T1,RL1,A1,B1
800 RETURN
801 END

```

```

802      SUBROUTINE LPGREC
810C      *****
820C      LPG RECURSION CALCULATION
830C      *****
831C
833      COMMON/LPGP27/F1,RL1,A1,B1,T1,I07,I08
834      DIMENSION T(200),I2(200)
840C
850C      REDEFINE C FOR PROBABILITY CALCULATIONS
860C
865      WRITE(6,13) " ***LPGREC, F1,RL1,A1,B1 ",T1,RL1,A1,B1
870      C= -RL1/ALOG(1.-T1)
880      C1 = C/(RL1 + B1)
890C
900C      SET LIMIT ON X = 200
910C
920      RL2=200
930C
940      C2=0
950C
960C      SET CONSTANTS FOR USE IN RECURSION
970C
980      S7=1
990      H1=(B1/(RL1+B1))**A1
1000     H1=S7*H1
1010C      S7 = SCALE FACTOR
1020      H2=(C/(RL1+B1))
1030C
1040     WRITE(6,13)"C,C1,H1,H2",C,C1,H1,H2
1041     13 FORMAT(V)
1050C
1060C      EVALUATE H(X=0)
1070C
1080     T(1) = 0
1090     T2(1) = H1
1100     S1 = .H1
1110     C2 = C2 + S1/S7
1120C
1130     WRITE(6,13)
1131     WRITE(8,13) 0,S1,C2
1140C
1141     WRITE(6,13)
1150     WRITE(6,13) "          X          H(X)          F(X)"
1151     WRITE(6,13)
1160     WRITE(6,13) 0,S1,C2
1170C
1180C
1190C      EVALUATE H(X) FOR X .GT. 0
1200     1200 CONTINUE
1201     L2=RL2
1202     DO 1050 IX=1,L2
1203     X=IX
1204     IXP1=IX+1
1220     S1 = 0
1230     T(1)=0

```

LPGFOR.S
LPGREC

```

1240C
1250      T(IX+1)=(T1*H2*(A1+X-1)/X)*T2(IX)
1260C
1270      IF( X .LT. 2 )GO TO 1380
1280C
1290C
1300C
1310      DO 1360 K=1,IX-1
1320      T(K+1)=(T1/X) * (H2*(A1+K-1)*T2( K) +(X-1)*T2(K+1))
1330      S1=S1+T(K+1)
1340      IF( ID8 .LE. 0 )GO TO 1360
1350      WRITE(6,13) "X,K,T(K),S1 ",X,K,T(K),S1
1360 1360 CONTINUE
1370C      PICK UP T(X,X) TERM IN SUM
1375 1380 CONTINUE
1380      S1=S1 +T(IX+1),
1390C
1400C      WRITE(6,13) TOTALS FOR H(X)
1410C
1420      C2 = C2 + S1/S7
1430      WRITE(6,13) IX,S1,C2
1440      WRITE(8,13) IX,S1,C2
1450C
1460C      IF( CUM PROB EXCEEDS .99, STOP
1470C
1480      IF( C2 .GT. .99 )GO TO 1670
1490C
1500C      WRITE(6,13) T(K) TERMS FOR DEBUGGING
1510C
1520      IF( ID8 .LE. 0 )GO TO 1560
1530      DO 1550 KK=1,IXP1
1535      K=KK-1
1540      WRITE(6,13) "X,K,T(X,K) =>",X,K,T(KK)
1550 1550 CONTINUE
1560 1560 CONTINUE
1570C
1580C      RECORD T(K) VALUES FOR USE IN NEXT PASS
1590C
1600      DO 1620 KK=1,IXP1
1610      T2(KK)=T(KK)
1620 1620 CONTINUE
1630C
1640C      -----END OF X LOOP
1650 1650 CONTINUE
1660C
1665 1670 CONTINUE
1670      WRITE (8,13) "-99,-99,-99,"      END OF LPG"
1680      RETURN
1683      END
1690C      *****END OF LPG RECURSION

```

LPGFOR.S
LPGREC

```

1700 SUBROUTINE BSRCH(R0, D)
1710 COMMON/BSRCH/ K, R0, R, Q, I5, I15, I100
1720 ----- AVE. REQ. SIZE EQUATION
1730 ----- FUNCTION FNR(Q) = -1/2 * (1-Q) * ALN(1-Q)
1740 -----
1750 SEE 2182-2190 FOR FNR(Q) DEFINITION.
1760 -----
1770 ----- BINARY SEARCH ROUTINE
1780 -----
1790
1800 SOLVE FOR THE VALUE OF Q WHICH GIVES AND
1810 AVE. REQUISITION SIZE OF R0
1820
1830 SET UP END POINTS FOR SEARCH
1840
1850 K=0
1860 Q9=.99
1870 R9=FNR(Q9)
1880 Q1=.001
1890 R1=FNR(Q1)
1900 GOTO 2080
1910
1920 CHECK IF( R0 .LE. R
1930
1935 1940 CONTINUE
1940 IF( R0 .GT. R )GO TO 2000
1950
1960 RE-SET TOP OF INTERVAL
1970 R9=R
1980 Q9=Q
1990 GOTO 2080
1995
1992 2000 CONTINUE
2000
2010 RE-SET BOTTOM OF INTERVAL
2020 R1=R
2030 Q1=Q
2040 GOTO 2080
2050
2060 HALVE THE INTERVAL, AND RE-EVALUATE THE FUNCTION
2070
2075 2080 CONTINUE
2080 K=K+1
2090 Q=(Q1+Q9)/2
2100 R=FNR(Q)
2110
2120
2130 IF( R IS WITHIN .001 OF R0, )GO TO RETURN
2140
2141 IF( I08.GT.0 ) WRITE(6,23) K,R0,R,Q
2142 23 FORMAT(" ***BSRCH--K,R0,R,Q",I5,3F8.3)
2150 IF( ABS(R-R0) .LT. .001 )GO TO 2180
2160 IF( K .GT. 15 )GO TO 2180
2170 GOTO 1940
2175 2180 CONTINUE
2180 RETURN
2181 END

```

LPGFOR.S
 BSRCH

2182 FUNCTION FNR(Q)
2183 FNR=-Q/((1-Q)*ALOG(1-Q))
2184 RETURN
2185 END

LPGFOR.S
BSRCH

```

2380     SUBROUTINE LPG1
2400     DIMENSION Y(100,100)
2410     COMMON/LPGPPM/T1,RL1,A1,B1,ID7,ID8
2411     WRITE(6,13) "THIS PROGRAM COMPUTES EXACT PROB FOR THE LPG DIST"
2412     13 FORMAT(V)
2420     WRITE(6,13) "THETA =",T1,"LAMBDA =",RL1,"ALPHA =",A1,"BETA =",B1
2430     WRITE(6,13) 1,2,3,4
2440     WRITE(6,13) T1,RL1,A1,B1
2450     C = - RL1 /ALOG(1-T1)
2460     RM1 = -T1/((1-T1)*ALOG(1-T1))*RL1*A1/B1
2470     C1 =C/(RL1 + B1)
2480     L2 = IFIX(100*RM1)
2490     C2 =0
2500     DO 2790 IXP1=1,L2+1
2504         X=IXP1-1
2506         IX=X
2510         Y(IXP1,1)=0
2520         F1=1.
2521C
2522C         COMPUTE (X-1)
2523C
2530     IF( X.LT.2.)GO TO 2570
2540         DO 2560 K=1,IX-1
2550             F1 = F1*K
2560             2560 CONTINUE
2565             2570 CONTINUE
2570         Y (IXP1,IXP1)=1./F1
2580         F1 = 1
2590         IF( X.GT. 0. )GO TO 2630
2600         S1 =(B1/(RL1 +B1))**A1
2610         GO TO 2740
2620C
2625     2630 CONTINUE
2630         S1=0.
2640         DO 2720 KK=1,IX
2641             KPI=KK+1
2650             IF( IX .EQ. 1 )GO TO 2670
2660             Y(IXP1,KPI)=Y(IXP1-1,KPI-1)/(IXP1-1) + Y(IXP1-1,KPI)
2665             2670 CONTINUE
2670             F1=1.
2680             DO 2700 JJ=1,KK
2685                 J=JJ-1
2690                 F1 = F1*(A1 + J)*C1
2700             2700 CONTINUE
2710             S1 = S1 + Y(IXP1,KPI)*F1
2720             2720 CONTINUE
2730             S1 = S1*(B1/(RL1 + B1))**A1*T1**X/X
2735             2740 CONTINUE

```

LPGFOR.S
LPG1

2740 C2 = C2 + S1
2750 IF(ID8 .LE.0)GO TO 2730
2760 WRITE(6,13) X,S1,C2
2770 WRITE (8,13) X,S1,C2
2771 2780 CONTINUE
2780 IF(C2 .GT. .9) GO TO 2800
2790 2790 CONTINUE
2795 2800 CONTINUE
2800 WRITE (8,13) -99,-99,-99
2810 RETURN
2815 END

LPGFOR.S
BSRCH

```

2820C
2830C *****
2840 SUBROUTINE LP2
2845 COMMON/LPGPRM/T1,RL1,A1,B1,ID7,ID8
2850 WRITE(6,13) "SCALED POISSON"
2855 13 FORMAT(V)
2860C
2870 WRITE(6,13) "THETA =",T1,"LAMBDA =",RL1,"ALPHA =",A1,"BETA =",B1
2880 C = -RL1/ALOG(1-T1)
2890 C3= 1./(1.-T1)
2900 K1 = T1*C
2910 C4 = (B1/(K1+B1))*A1
2920 W = K1/(K1+B1)
2930 C2 = 0
2940 DO 3020 MM=1,1001
2945 M=MM-1
2950 P1 = 1
2960 IF( M.NE.0 )GO TO 3000
2970 P1 = C4
2980 GOTO 3040
2990C
2995 3000 CONTINUE
3000 DO 3020 II=1,M
3005 I=II-1
3010 P1 = P1*((A1+M-I-1)/(M-I)*W)
3020 3020 CONTINUE
3030 P1 = P1*C4
3035 3040 CONTINUE
3040 C2 = C2 + P1
3050 IF( ID8.LE.0 )GO TO 3080
3060 WRITE(6,13) M,C3*M,P1,C2
3070 WRITE (8,13) C3*M,P1,C2
3075 3080 CONTINUE
3080 IF( C2.GT..99 )GO TO 3100
3090 3090 CONTINUE
3095 3100 CONTINUE
3100 RETURN
3101 END

```

LPGFOR.S
LPG2

```

3120C *****
3130C
3140 SUBROUTINE LPG3
3142C
3144C NEGATIVE BINOMIAL APPROXIMATION.
3146C
3147 COMMON/LPGPRM/T1,RL1,A1,B1,ID7,ID8
3150 WRITE(6,13) "THIS PROGRAM COMPUTES LPG USING SCALED BIN"
3155 13 FORMAT(V)
3161 WRITE(6,13)
3170 WRITE(6,13) "THETA =",T1,"LAMBDA =",RL1,"ALPHA =",A1,"BETA =",B1
3180 WRITE(6,13)
3190 WRITE(6,13) "NEG BINOMIAL PROBABILITIES"
3200 WRITE(6,13)
3210 WRITE(6,13) " X P F"
3220 WRITE(6,13)
3230 C = 1/(1.-T1)
3240 RK1 = -T1*RL1/ALOG(1-T1)
3250 C4 = (B1/(RK1+B1))**A1
3260 W = RK1/(RK1+B1)
3270 C2 = 0
3280 DO 3550 MM=1,1001
3282 M=MM-1
3290 P1 = 1
3300 IF( M.NE.0 )GO TO 3370
3310 P1 = C4
3320 A = -1
3330 B = IFIX(C/2 +0.5)
3340 S2 = P1/(C/2+1)
3350 GOTO 3440
3360C
3365 3370 CONTINUE
3370 DO 3390 II=1,M
3375 I=II-1
3380 P1 = P1*((A1+M-I-1)/(M-I)*W)
3390 3390 CONTINUE
3400 P1 = P1*C4
3410 IA = IFIX((2*M-1)*C/2+0.5)
3411 RA=IA
3420 IB = IFIX((2*M+1)*C/2+0.5)
3422 RB=IB
3430 S2 = P1/C
3435 3440 CONTINUE
3440 RL3 = C2
3450 DO 3520 IX=IA+1,IB
3455 X=IX
3460 F3 = C2 + S2*(X-RA)
3470 F2 = F3 - RL3
3480 RL3 = F3
3490C
3500 WRITE(6,13) IX,F2,F3
3510 WRITE (8,13) IX,F2,F3
3520 3520 CONTINUE
3530 C2 = C2 + P1
3540 IF( C2 .GT. .99 )GO TO 3560
3550 3550 CONTINUE
3555 3560 CONTINUE
3560 WRITE (8,13) -99,-99,-99,"END OF NB"
3570 RETURN
3575 END

```

LPGFOR.S
LPG3

```

3580C *****LPG4
3590 SUBROUTINE LPG4
3600C LPG4.S
3610C COMPUTE REQUISITION SIZE R VS THETA TABLE
3620C
3630 DO 3670 I=1,100
3635 T1=I*.01
3640 B = -(1-T1)*ALOG(1-T1)
3650 R = T1/B
3660 WRITE(6,13) T1,R
3665 13 FORMAT(V)
3670 3670 CONTINUE
3680 RETURN

```

DEFINITIONS IN NAHMIAS' PROGRAM LPG1.S

3700C	LAMB	RL1
3710C	ALPH	A1
3720C	BET	B1
3730C	MEAN	RM1
3740C	CNST	C1
3750C	LIM	RL2
3760C	CUM	C2
3770C	FACT	F1
3780C	SUM	S1
3790C	KN	RK1
3800C	PROD	P1
3810C	CN	C3
3820C	CONST	C4
3830C	LAST	L3
3840C	FX	F2
3850C	FFX	F3
3860C	SL	S2
3870C		
3880C		
3890C		
3900 FND		

LPGFOR.S
LPG4

CATALOG/FILE DESCRIPTION= INVRZ/SNBROP.S

```

10*#R01=1 INVRZOBJZ/SNBROP.OCBCD,NO300
20*SNBROP.S--CALCULATE SCALED NEGATIVE BINOMIAL CRITICAL POINTS
300
40      SUBROUTINE SNBROP(AVEREQ,D1,ELT,CVLT,FCRIT,IROP)
500
60      COMMON/INT/INT(20)
700
800      ESTIMATE LPO PARAMETERS
900
100     INT6 = INT(5)
1100
120     IF(INT6.GT.0) WRITE(6,13)AVEREQ,D1,ELT,CVLT,FCRIT
1300
1400      CHECK FOR NO DEMAND OR NO PENALTY FOR SHORTAGES CASE
1500      IN THIS EVENT, SET RE-ORDER POINT TO ZERO
1600
170     IF(D1.LE.0.) GO TO 601
180     IF(FCRIT.LE.0.) GO TO 601
1900
2000
210     IF(ELT.LE..1)ELT=.1
220     IF(D1.LE.0.1)D1=0.1
230     IF(CVLT.LE.0.01)CVLT=0.01
240     CALL BSRCH(AVEREQ,0)
250     T1=0
260     S=CVLT*ELT
270     B=S*S/ELT
280     U1=D1*ELT
2900
300     B1=1./B
310     A1=B1*ELT
320     R1=-T1/((1.-T1)*ALOG(1.-T1))
330     RL1=U1/(ELT*R1)
3400
350     IF(INT6.GT.0) WRITE(6,13) "      B1      A1",
3600      "      R1      RL1      C      RK1"
370 13  FORMAT(V)
3800
3900      ESTIMATE SNB PARAMETERS
4000
410     C=1./(1.-T1)
420     RK1=-T1*RL1/(ALOG(1.-T1))
430     IF(INT6.GT.0) WRITE(6,03)B1,A1,R1,RL1,C,RK1
440 03  FORMAT(8F9.3)
4500
4600      (P(X=0.)-P0
4700
480     P0=(B1/(RK1+B1))**A1
490     W=RK1/(RK1+B1)
5000

```

SNBROP.S

```

510C      SET UP INITIAL PROBABILITIES
520C
530      X=0.
540      XL=0.
550      FXL=0.
560      PX=P0
570      IF(PX.LE.0.) GO TO 701
580      XN=C/2.
590      FXN=P0
600C
610C      IS COM>FCRIT
620C      IF SO, EXIT LOOP.
630C
640C
650C
660      IF(INT0.GT.0)WRITE(6,13) "      X      PX",
670      "      FX      XL      FXL"
680C
690 220 CONTINUE
700      IF((INT0.GT.0).AND.(X.LE.20.))WRITE(6,93)XN,PX,FXN,XL,FXL
710 93 FORMAT(1X,2(F10.2,2F10.5))
720      IF(FXN.GE.FCRIT) GO TO 410
730      IF(FXN.GT. .999)GO TO 510
740C
750C      COMPUTE NEXT NEG. BIN PROB.
760C
770C
780C      GENERATE SNB PROB TILL X*
790C      INTERVAL IS FOUND
800C
810C
820      X = X+1.
830      XL=XN
840      FXL=FXN
850      XN=XN+C
860      PX = PX * ( (A1+ X-1.) / X ) * W
870      IF(PX.LE.0.) GO TO 701
880      FXN=FXN+PX
890      GO TO 220
900C
910C      INTERPOLATE TO FIND X*
920C
930 410 CONTINUE
940      XCRIT=XL+(FCRIT-FXL)*(XN-XL)/(FXN -FXL)
950C
960C      ROUND TO GET FINAL VALUE
970C
980      IROP=(XCRIT+.5)
990      RETURN
1000C
1010C      HIGH PROTECTION LIMIT. SET ROP TO UPPER LIMIT OF X.
1020C
1030 510 IROP=(XN+.5)
1040      RETURN
1050C
1060C      DEMAND OR FCRIT IS ZERO. SET ROP=0
1070C
1080 601 CONTINUE

```

```

1090C
1100      IROP=0
1110      RETURN
1120C
1130C      PS =0, SO USE DEFAULT CALCULATION
1140C
1150 701 CONTINUE
1160      IROP = AVEREQ*DI*ELT*(1.+3.*CVLT)
1170      WRITE(6,723)IROP
1180 723 FORMAT(" *****SNBROP. PX=0, SSO SET ROP=",I3)
1190C
1200      RETURN
1210      END
1220C
1230      SUBROUTINE BSRCH(RO,Q)
1240C
1250          COMMON/IWT/IWT(20)
1260          ID7=IWT(6)
1270          ID8=IWT(6)
1280C      -----AVE REQ SIZE EQUATION
1290C----- FUNCTION FNR(Q) = -Q/( (1-Q)*ALOG(1-Q) )
1300C
1310C      SEE 2182-2190 FOR FNR(Q) DEFINITION.
1320C
1330C
1340C      -----
1350C          BINARY SEARCH ROUTINE
1360C      -----
1370C
1380C      SOLVE FOR THE VALUE OF Q WHICH GIVES AND
1390C      AVE. REQUISITION SIZE OF RO
1400C
1410C      SET UP END POINTS FOR SEARCH
1420C
1430      K= 0
1440      Q9=.999
1450      R9=FNR(Q9)
1460      Q1=.001
1470      R1=FNR(Q1)
1480      GOTO 2080

```

SNBROP.S

```

1490C
1500C          CHECK IF( R0 .LE. R
1510C
1520 1940 CONTINUE
1530          IF( R0 .GT. R )GO TO 2000
1540C
1550C          RE-SET TOP OF INTERVAL
1560      R9=R
1570      Q9=Q
1580      GOTO 2080
1590C
1600 2000 CONTINUE
1610C
1620C          RE-SET BOTTOM OF INTERVAL
1630      R1=R
1640      Q1=Q
1650      GOTO 2080
1660C
1670C          HALVE THE INTERVAL, AND RE-EVALUATE THE FUNCTION
1680C
1690 2080 CONTINUE
1700      K=K+1
1710      Q=(Q1+Q9)/2
1720      R=FNR(Q)
1730C
1740C          IF( R IS WITHIN .001 OF R0, )GO TO RETURN
1750C
1760      IF( ID3.GT.0 ) WRITE(6,23) K,R0,R,Q
1770 23  FORMAT(" ***BSRCH--K,R0,R,Q",I5,3F8.3)
1780      IF( ABS(R-R0) .LT. .001 )GO TO 2180
1790      IF( K .GT.25 )GO TO 2180
1800      GOTO 1940
1810 2180 CONTINUE
1820      RETURN
1830      END
1840      FUNCTION FNR(Q)
1850          FNR=-Q/((1-Q)*ALOG(1-Q))
1860      RETURN
1870      END

```

NEGATIVE BINOMIAL REQUISITION SIZE ROUTINES

DEMPAR

NEGBIN.

NEGBN1

RANDEM

CATALOG FILE DESCRIPTION = HEDGZDEMPR3.S

```

10*#RUN=HEDGZ03.1ZDEMPR3.0(CBD,NOC0)
20*DEMPR3.S
30      SUBROUTINE DEMPAR(IDPER,IP4,IP5)
40C *****
50      PARAMETER NO00=33
60C *****
70      COMMON/IGBUG/IGBUG
80      COMMON/ITIME/ITIME
90      COMMON/ITWEEK/ITWEEK
100     COMMON/ITMNTH/ITMNTH
110     COMMON/ITQTR/ITQTR
120     COMMON/ITYEAR/ITYEAR
130     COMMON/NENTRY/NENTRY
140     COMMON/NFEMAX/NFEMAX
150     COMMON/NITEM/NITEM
160     COMMON/PRI2F/PRI2F
170     COMMON/NDENT/NDENT(1)
180     COMMON/IDEMND/IDEMND(1,N000)
190     COMMON/IRETRN/IRETRN(1,N000)
200     COMMON/IREQ/IREQ(1,N000)
210     IF(IGBUG.NE.1) GO TO 25
220     IY=ITIME/ITYEAR +1
230     IQT=ITIME/ITQTR +1
240     ITW=(ITIME-(IY-1)*ITYEAR)/ITWEEK +1
250     WRITE(6,103)IY,IQT,ITW
260 103 FORMAT(" DEMPAR-----",5X,"YEAR NO.",13,5X,"QTR NO.",13,5X,
270X      "WEEK NO.",13)
280 25 CONTINUE
290C
300C      INCREMENT PERIOD COUNTERS
310C
320      IDPER=IDPER+1
330      DO 100 N=1,NITEM
340      IQTY=IDEMND(N,IDPER)
350      IRET=IRETRN(N,IDPER)
360      IREQ=IREQ(N,IDPER)
370      IF(IGBUG.EQ.1)WRITE(6,8013)N,NDENT(N),IDPER,IQTY,IRET,IREQ
380 8013      FORMAT(4X,"****DEMPAR--N=",15," NDENT=",15," IDPER=",15,
390X      " IDEMND=",15," IRETRN=",15," IREQ=",15)
400      IF(NDENT(N).LE.0) GO TO 100
410C
420C      GENERATE SERVICEABLE RETURNS, IF ANY
430C
440      IF(IRET.LE.0)GO TO 30
450      IT=ITIME+ITMNTH
460      CALL ENTER(IT,4,N,IRET,0)
470 30      CONTINUE
480C
490C      REFLECT      REQUISITION FOR ITEM N
500C
510C
520C      SET REQUISITION COUNTER KNTREQ TO ZERO
530C

```

DEMPR3.S
DEMPAR

```

510 KNTREQ=0
520 IF (CITF(.05,0) GO TO 100)
530 ITOT=0
540
550 OBTAIN A REQUISITION SIZE IR BY MONTE CARLO
560
570 50 R=RAIDEM(.2)
580 CALL NEGONTCR,1000
590 ITEMP=ITOT+1.0
600 IF=1.0
610 IF (ITEMP.LE.10TY) GO TO 60
620
630 LIMIT LAST REQUISITION SO THAT TOTAL DEMAND IN QTR
640 EQUALS 10TY.
650
660 IT=10TY-ITEMP
670 50 CONTINUE
680
690 KNTREQ= KNTREQ + 1
700
710 IF (NFEMAX=50) REQUISITIONS HAVE BEEN
720 GENERATED, PUT ALL REMAINING DEMAND ON THIS REQUISITION.
730
740 IF (KNTREQ .LT. (NFEMAX - 50)) GO TO 80
750
760 IR = 10TY-ITOT
770 WRITE(6,73)ITIME,IR,10TY
780 73 FORMAT(4X,'****DEMPAR--ITIME=',13,' F.F.L ',
790 ' OVERFLOW REQ=',15,' OF ',15)
800 WRITE(6,8013)N,NDEF(N),IDPER,10TY,IPR,IRQ
810
820 DETERMINE PRIORITY OF THIS REQUISITION
830 ASSUMING 50% ARE PRIORITY 1
840
850 80 CONTINUE
860 IPR=1
870 IF(RANDEM(.2).LE. PRI2F) IPR=2
880
890 DETERMINE ARRIVAL TIME FOR THIS REQUISITION
900
910 LIMIT ARRIVAL TIME TO NO LATER THAN .5 WEEKS
920 BEFORE THE END OF THE QUARTER
930
940 LM=ITOTR-0.5*ITWEEK
950 IT=ITIME+RANDEM(.2)*FLOAT(LM)
960 IF (ISBUG.EQ.1)WRITE(6,8023)IT,N,R,IRQ,IR,IPR
970 8023 FORMAT(4X,'****DEMPAR--IT=',110,' N=',14,' R=',F5.3,
980 ' INITIAL REQ-SIZE=',14,' FINAL REQ-SIZE=',14,
990 ' PRIORITY=',14)
1000
1010 PUT REQUISITION OF F.F.L.

```

DEMPR3.S
DEMPAR

```

1020      CALL ENTER (CI, I, N, IR, 100)
1030      ITOT=ITOT+1
1040      IF (ITOT.LT.100) GO TO 50
1050C
1060C *****CREATE NEXT DEMP PAR EVENT
1070C
1080      100      CONTINUE
1090      IT=ITIME+ITOTR
1100      CALL ENTER (IT, IP, IDPER, 0,0)
1110      RETURN
1120      END
1130C      NOTE * GETREQ IS NO LONGER USED.
1140C      RATHER, NEGBN1 GENERATES NEGBN REQ SIZES.
1150C

```

```

1160 SUBROUTINE GETREQR,IQTY,IR)
1170C THIS ROUTINE DETERMINES A REQUISITION SIZE
1180C CORRESPONDING TO A CUM PROB. OF R.
1190C
1200 DIMENSION IUCAT(9),CPROB(10,8),IRQ(10,8)
1210 DATA IUCAT(1,3,10,31,100,316,1000,3162/
1220 DATA CPROB(10*1,
1230C .40,.82,8*1.0,
1240C .27,.52,.65,.73,.86,.91,.92,.95,.96,1.0,
1250C .23,.53,.73,.73,.82,.90,.92,.97,.99,1.00,
1260C .06,.39,.60,.73,.81,.85,.89,.93,.98,1.00,
1270C .015,.23,.42,.56,.65,.70,.76,.92,.97,1.00,
1280C .012,.17,.31,.37,.48,.55,.60,.78,.87,1.00,
1290C .01,.03,.32,.50,.60,.66,.86,.96,1.00,1.00/
1300 DATA IRQ(10*1,
1310C 1, 2, 3, 7*3,
1320C 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
1330C 1, 4, 6, 8, 10, 13, 15, 20, 25, 30,
1340C 1, 5, 10, 15, 20, 25, 30, 40, 70, 100,
1350C 1, 5, 10, 15, 20, 25, 30, 80, 150, 300,
1360C 1, 5, 10, 15, 20, 25, 30, 80, 200, 600,
1370C 1, 20, 40, 60, 80, 100, 200, 300, 600, 700/
1380C
1390C
1400C ESTABLISH DEMAND RATE CATEGORY
1410C
1420 DO 10 I=1,8
1430 IM=I
1440 M=IUCAT(I)
1450 IF(IQTY.LE.M) GO TO 30
1460 10 CONTINUE
1470C
1480C DETERMINE REQUISITION SIZE
1490C
1500 30 CONTINUE
1510 DO 40 I=1,10
1520 IF(CPROB(I,IM).GE.R) GO TO 60
1530 40 CONTINUE
1540 60 CONTINUE
1550C
1560C SET REQUISITION SIZE
1570C
1580 IF(I.GT.1)GO TO 80
1590 IR=1
1600 RETURN
1610 80 CONTINUE
1620 IF(I.LT.10)GO TO 100
1630 IR=IRQ(10,IM)
1640 RETURN
1650 100 CONTINUE
1660 DPROB=CPROB(I,IM)-CPROB(I-1,IM)
1670 RPROB=R-CPROB(I-1,IM)
1680 DRQ=IRQ(I,IM)-IRQ(I-1,IM)
1690 IR=IRQ(I-1,IM)+IFIX(RPROB*DRQ/DPROB+.5)
1700 RETURN
1710 END

```

DEMPR3.S
GETREQ

```

2270C          COMPUTE GRAND MEAN  X9
2280C
2290 50 CONTINUE
2300    X9=T2/C9
2310C
2320C          IF THERE IS ONLY 1 OBSERVATION, ASSUME R=.0439
2330C
2340    IF (CPT .GT. 1) GO TO 59
2350C
2360C          USE AVERAGE R9 VALUE,
2370C
2380 51 R9=.0439
2390    P9=(X9-1)/R9
2400    Q9=P9+1
2410C
2420C          IF P9 =0, USE AVE ITEM VALUES
2430C
2440    IF(P9.LE.0.) GO TO 41
2450    GO TO 170
2460C
2470C          COMPUTE VAR(X) ESTIMATE
2480C
2490 59 CONTINUE
2500    V=0.
2510    DO 60 I=1,NPT
2520      V=V+C(I)*(X(I)-X9)**2
2530 60 CONTINUE
2540    V=V/(FLOAT(NPT-1))
2550    S=SQRT(V)
2560C
2570C          IF VARIANCE > (X9-1), CONTINUE TO 100. OTHERWISE,
2580C          A NEGATIVE BINOMIAL MODEL DOESN'T FIT.
2590C
2600    IF(IWT(3).GE.2) WRITE(6,73)NPT,X9,V,S,S/X9,(X(I),I=1,NPT)
2610 73 FORMAT(T20,'NPT=',T30,I10/
2620      T20,'MEAN=',T30,F13.2/
2630      T20,'VAR=',T30,F13.2/
2640      T20,'STD DEV=',T30,F13.2/
2650      T20,'COEF OF V=',T30,F13.2/
2660      T20,'X(1)=',T30,20F6.2)
2670    IF(V.GT.(X9-1))GO TO 100
2675    IF(IWT(3).GE.1)
2680      WRITE(6,63)NPT,X9,V,S,S/X9,(X(I),I=1,NPT)
2690 63 FORMAT(4('***'),'NEGBIN. VAR(X) <= MEAN.',
2710      T20,'NPT,MEAN,VAR,STD DEV,C.OF.V =',I4,4F13.2/
2720      T20,'X(1)=',20F6.2)
2730C
2740C          CHECK FOR CONSTANT REQ SIZE. IF NOT , GO TO 51
2750C
2760    IF( (V.GT. 0.05) .OR. (NPT.LT.4) ) GO TO 51
2770C
2780C          USE CONSTANT REQ SIZE
2790C

```



```

2800 IREQS = 1
2810 PREQS(1) = 1.00
2820 IREQS(1) = X9 + .5
2830 GO TO 210
2840C
2850C
2860C
2870C COMPUTE NEGATIVE BINOMIAL PARAMETERS
2880C
2890 100 Q9=V/(X9-1)
2900 P9=Q9-1
2910 R9=(X9-1)/P9
2920 R2=(X9-1)**2/(V-X9+1)
2930C
2940C LET
2950C IREQS(1)=ITH REQUISITION SIZE
2960C PREQS(1)=PROBABILITY REQUISITION SIZE <=IREQS(1)
2970C NREQS=NUMBER OF POINTS IN REQUISITION SIZE C.D.F. APP
2980C
2990C R9,P9,Q9=NEGATIVE BINOMIAL PARAMETERS
3000C
3010C COMPUTE NEGATIVE BINOMIAL PROBABILITIES
3020C USING R9 AND Q9
3030C
3040C INITIALIZE VARIABLES
3050 170 CONTINUE
3060 IF(IWT(3).GE.2)WRITE(6,173)R9,P9,Q9
3070 173 FORMAT(//NEG. BIN PARAMETERS//
3080 & ' R9=',F8.3,' P9=',F8.3,' Q9=',F8.3)
3090C
3100 Y=0.
3110 P=1/(Q9**R9)
3120 F=P
3130 AVE= 1-P
3140 P8=P9/Q9
3150 IREQS(1)=1
3160 PREQS(1)=F
3170 NREQS=1
3180C
3190C COMPUTE NEGATIVE BINOMIAL PROBABILITIES
3200C
3210C MAX REQUISITION SIZE ALLOWED = 1000 UNITS
3220 DO 190 I=1,1000
3230 IPRNT=1
3240 IF(I.GT.20)IPRNT=0
3250 IF(MOD(I,10).EQ.0)IPRNT=1
3260 IF( (IWT(3).GE.3).AND.(IPRNT.EQ.1))WRITE(6,183)I,Y,P,F,AVE
3270 183 FORMAT(' I,Y,P,F,AVE=',I10,F10.0,3F10.4)
3280 Y=1-I
3290 P=((Y+R9)/(Y+1))*P8*P
3300 F=F+P
3310 AVE=AVE + (1-F)
3320C
3330C RECORD APPROXIMATION TO F(X) AT LEAST EVERY .01 INOR

```

DEMPR3.S
NEGBIN.S

```

3870 350 R=RANDB4(,2)
3880 CALL NEGBIN(R,IR)
3890 ITOT=ITOT+IR
3900 KNT=KNT+1
3910 IF(ITOT.LT.IOTY)GO TO 350
3920
3930 SET NUMBER OF REQUISITIONS EQUAL TO KNT.
3940
3950 370 IREQ(1)=KNT
3960
3970 WRITE DEBUG MESSAGE.
3980
3990 IF(INT(3).GE.2)WRITE(6,373)I,IDEVND(1),IREQ(1)
4000 373 FORMAT(3110)
4010
4020 380 CONTINUE
4030 RETURN
4040 END

```

DEMPR3.S
 NEGBIN.S

```

4070      SUBROUTINE NEGBN1(R,IR)
4080C*****
4090      PARAMETER NQ00=38
4100C*****
4110C
4120      COMMON/IR1/IRF(20)
4130      COMMON/IRFAD/IDEMND(NQ00)
4140      COMMON/IRFQ/IRFQ(NQ00)
4150      DIMENSION C(10),CC(10),DC(10)
4160      COMMON/NEGBN1/ NREQS, IREQS(100),PREQS(100)
4170C          GIVE R, A U(0,1) PSEUDO-RANDOM NUMBER,
4180C          DETERMINE THE CORRESPONDING REQUISITION SIZE IR.
4190C
4200      DO420 I=1, IREQS
4210      IF(R.LE.PREQS(1))GO TO 430
4220 420 CONTINUE
4230 430 CONTINUE
4240C
4250C          SET THE REQUISITION SIZE IR
4260C
4270      IF(1.GT.1)GO TO 460
4280      IR=IREQS(1)
4290      RETURN
4300C
4310 460 CONTINUE
4320      IF(1.LT.NREQS)GO TO 480
4330      IR=IREQS(NREQS)
4340      RETURN
4350C
4360 480 CONTINUE
4370      IM1=I-1
4380      IRI=IREQS(I)
4390      IRI1=IREQS(IM1)
4400      IRDIF=IRI-IRI1
4410      IF(IRDIF.GT.1)GO TO 510
4420      IR=IRI
4430      RETURN
4440C
4450C          INTERPOLATE TO DETERMINE REQUISITION
4460C
4470 510 CONTINUE
4480      DPR0B=PREQS(I)-PREQS(IM1)
4490      RPR0B=R-PREQS(IM1)
4500      IR=IRI1+IFIX((RPR0B*IRI)/DPR0B)+0.5)
4510      RETURN
4520C
4525      END

```

```

4530C-----RANDEN--U(0,1) RANDOM NUMBER GENERATOR.
4540C          RANDEN--U(0,1) RANDOM NUMBER GENERATOR.
4560      FUNCTION RANDEN(X)
4570C          A CALL WITH X < 0. INITIALIZES THE RANDOM NUMBER STREAM.
4580      IF(X) 10,20,20
4590      20 RN=RND*RANDEN
4600      RN1=AMOD(RN,BN)
4605      RANDEN=RN1/BN
4610      RETURN
4620      10 RHO=/.0**13
4630      BN=10.0**10
4640      RANDEN=-X
4650      GO TO 20
4660      END

```

CATALOG FILE DESCRIPTIONS INITM3.S

```

10*FILE=HIDGZO3/INITM3.OO(3D, 1030)
20*INITM3.S
30 SUBROUTINE INITM
40 -----
50 PARAMETER NLOC=38
60 -----
70 CHARACTER ALC,FMT,UM,NOUN,MGTCD
80 COMMON/FSN/ALC,FMT(4),UM,NOUN(2),MGTCD(4),IOH,IOR,IPI,IPIPR
90 COMMON/ISLEZC/LE
100 COMMON/IDBUG/IDBUG
110 COMMON/ITAY/ITAY
120 COMMON/ITPTH/ITPTH
130 COMMON/ITEM/ITEM
140 COMMON/NDEM/NDEM
150 COMMON/NDHIS/NDHIS
160 COMMON/INL/INL
170 COMMON/INTYPE/INTYPE
180 COMMON/IFBUG/IFBUG
190 COMMON/IBOP/IBOP(3),IBOPOR(3)
200 COMMON/IFEND/IFEND(1,NQ00)
210 COMMON/RMREQS/RMREQS(1)
220 COMMON/INACT/INACT(1)
230 COMMON/NORDPT/NORDPT(1)
240 COMMON/IDEMAC/IDEMAC(1)
250 COMMON/IRETAC/IRETAC(1)
260 COMMON/NREQAC/NREQAC(1)
270 COMMON/NDEMND/NDEMND(1,NQ00)
280 COMMON/NRETUR/NRETUR(1,NQ00)
290 COMMON/NREQ/NREQ(1,NQ00)
300 COMMON/NIDENT/NIDENT(1)
310 COMMON/INVDUE/INVDUE(1)
320 COMMON/NBOPT/NBOPT(1)
330 COMMON/NBOTU/NBOTU(1)
340 COMMON/NBOIU/NBOIU(1)
350 COMMON/NBOIR/NBOIR(1)
360 COMMON/NBOTR/NBOTR(1)
370 COMMON/REQSI/REQSI(1)
380 COMMON/REQIAD/REQIAD(1)
390 COMMON/LTPROD/LTPROD(1)
400 COMMON/LTADM/LTADM(1)
410 COMMON/JCOST/JCOST(1)
420 COMMON/ADR/ADR(1)
430 COMMON/ISUL/ISUL(1)
440 COMMON/IREQ/IREQ(1,NQ00)
450 COMMON/IRETUR/IRETUR(1,NQ00)
460 COMMON/IRL/IRL(1)
470 COMMON/ITL/ITL(1)
480 COMMON/IROL/IROL(1)
490 COMMON/IROTY/IROTY(1)
500 COMMON/RMTBR/RMTBR(1)
510 COMMON/RMEAN/RMEAN(1)
520 COMMON/RTREND/RTREND(1)
530 COMMON/RMAD/RMAD(1)
540 COMMON/REPSUM/REPSUM(1)

```

INITM3.S

```

550 COMMON/KNT /KNT(1)
560 COMMON/GROSS/GROSS(3)
570 COMMON/ZGROSS/ZGROSS(3)
580 COMMON/ICDEFOR/ICDEFOR
590C
600 ENTRY INIT41
610C
620C SET NUMBER OF PERIODS OF DATA FROM
630C
640 IDPER=NDPM
650C
660C READ DEMAND DATA FOR ITEM N FROM LOGICAL UNIT LR
670 LR=INLU
680 IKNT=IKNT+NITEM
690 DO 100 N=1,NITEM
700 10 CONTINUE
710C
720C READ ITEM DATA INPUT FROM FILE LR
730C
740 IF(INTYPE.F).2) GO TO 20
750C
760C READ BCD INPUT
770C
780C
790 READ(LR,8000,FID=200) ALC,FSN,UM,UCOST(N),NOUN,MGTCD,IOH,IOR,
800 LTADM(N),LTPROD(N),IPPL,IPPP
810C
820 RIPPPR=FLOAT(IPPPR)/100.
830 IF(IEBUG.EQ.1)WRITE(6,8010)IKNT,ALC,FSN,UM,UCOST(N),NOUN,MGTCD,
840 IOH,IOR,LTADM(N),LTPROD(N),IPPL,RIPPPR
850 IF(IEBUG.EQ.1)WRITE(6,8015)(I,I=1,10)
860C
870C READ DEMAND, RETURNS, AND REQ-FREQUENCY
880C
890 READ(LR,8000)(IDEMND(N,J),J=1,IDPER)
900 IF(IEBUG.EQ.1)WRITE(6,8020)(IDEMND(N,J),J=1,IDPER)
910 READ(LR,8000)(IRETUR(N,J),J=1,IDPER)
920 IF(IEBUG.EQ.1)WRITE(6,8030)(IRETUR(N,J),J=1,IDPER)
930 READ(LR,8000)(IREQ(N,J),J=1,IDPER)
940 IF(IEBUG.EQ.1)WRITE(6,8040)(IREQ(N,J),J=1,IDPER)
950 GO TO 50
960C
970C READ BINARY DATA
980C
990 20 CONTINUE

```

INITM3.S

```

1300      IKNT=IKNT+1
1310      READ(17)ALC,FSN,IR,UM,UCOST(N),NOUN,MGICD,IOH,IOR,
1320      LTADAC(N),LEPROD(N),IPPL,PIPPR
1330      IF (IEBUG.EQ.1)WRITE(6,3010)IKNT,ALC,FSN,UM,UCOST(N),NOUN,
1340      MGICD,IOH,IOR,LTADAC(N),LEPROD(N),IPPL,PIPPR
1350      READ(17)ALC,FSN,IR,DEMAND
1360      IF (IEBUG.EQ.1)WRITE(6,8020)(DEMAND(N,J),J=1,IDPER)
1370      READ(17)ALC,FSN,IR,IRETUR
1380      IF (IEBUG.EQ.1)WRITE(6,8030)(IRETUR(N,J),J=1,IDPER)
1390      READ(17)ALC,FSN,IR,IREQ
1400      IF (IEBUG.EQ.1)WRITE(6,8040)(IREQ(N,J),J=1,IDPER)
1410C      WRITE INPUT DATA TO FILE 09
1420C
1430C
1440C
1450 8000      FORMAT(V)
1460 8010      FORMAT(/I5,IX,A2,IX,A2,A4,A6,A3,IX,A2,F11.2,IX,A6,A4,
1470      T53,2A1,A4,A2,2X,2I7,3I6,F5.2)
1480 8015      FORMAT(T21,10I10)
1490 8020      FORMAT('DEMAND DEMAND/QTR',(T21,10I10))
1500 8030      FORMAT('IRETUR RETURN/QTR',(T21,10I10))
1510 8040      FORMAT('IREQ REQ /QTR',(T21,10I10))
1520C
1530 50      CONTINUE
1540C
1550C      DETERMINE PARAMETERS FOR NEGATIVE BINOMIAL
1560C      REQUISITION SIZE GENERATION
1570C
1580      CALL NEGBIN
1590C
1600 100 CONTINUE
1610      RETURN
1620C
1630C      INITIALIZE ITEM ARRAYS FOR A NEW SIM. RUN
1640C
1650      ENTRY INIT42
1660C
1670      DO 210 N=1,NITEM
1680C
1690C          INDICATE ITEM HAS NDHIS PERIODS OF DEMAND HISTORY
1700C
1710      NDENT(N)=NDHIS
1720C
1730C      ZERO DEMAND HISTORY RECORDS
1740      NRETAC(N)=0
1750      NDEMAC(N)=0
1760      NREQAC(N)=0
1770C

```

INITM3.S

```

1480C          LOAD DEMAND HISTORY ARRAY
1490C
1500C          KK=NDPTS
1510C          DO 1 I=1, NDPTS
1520C              NITEM=N(N, KK)=IDEMND(N, I)
1530C              NRETRN(N, KK)=LRETRN(N, I)
1540C              NREOCH(N, KK)=IREOCH(N, I)
1550C              KK=KK-1
1560C          1 CONTINUE
1570C
1580C          SET INVENTORY DUE-IN TO ZERO
1590C          TO INVDUE(N)=0
1600C          NORDET(N)=0
1610C
1620C          SET INITIAL BACKORDER COUNTERS TO ZERO
1630C
1640C          NBOTD(N)=0
1650C          NBOLD(N)=0
1660C          NBQIR(N)=0
1670C          NBQTR(N)=0
1680C          NBQPT(N)=0
1690C
1700C          COMPUTE DEMAND RATES
1710C
1720C          NN=N
1730C          CALL FOR5/6(NN)
1740C
1750C          SET INITIAL INVENTORY ON-HAND EQUAL TO LEAD-TIME DEMAND
1760C
1770C          RL=(LTADM(N)+LTPROD(N))*ADR(N)/12.
1780C          INVACT(N) = RL +0.5
1790C
1800C
1810C          210 CONTINUE
1820C
1830C          INITIALIZE GROSS ON-HAND AND ON-ORDER STATISTICS
1840C
1850C          110 CONTINUE
1860C              DO 130 N=1, NITEM
1870C                  IF(NDENT(N).LT.0) GO TO 130
1880C                  IF(INVACT(N).LE.0) GO TO 120
1890C                  IBOPDH(1)=IBOPDH(1)+1
1900C                  IBOPDH(2)=IBOPDH(2)+INVACT(N)
1910C                  IBOPDH(3)=IBOPDH(3)+FIX(UCOST(N)*FLOAT(INVACT(N)))
1920C              120 CONTINUE
1930C                  IF(INVDUE(N).LE.0) GO TO 130
1940C                  IBOPOR(1)=IBOPOR(1)+1
1950C                  IBOPOR(2)=IBOPOR(2)+INVDUE(N)
1960C                  IBOPOR(3)=IBOPOR(3)+FIX(UCOST(N)*FLOAT(INVDUE(N)))
1970C              130 CONTINUE
1980C              140 CONTINUE
1990C              RETURN
2000C          200 WRITE(6,8200)
2010C          8200 FORMAT(1H1,////20X,'END OF FILE READING.....'////)
2020C          CALL OUT
2030C          CALL OUTCST
2040C          CALL PLOTR
2050C          STOP
2060C          END

```

INITM3.S

*#RUN=:HEDG/OBJ/INORD3.0(BCD,NOGO)

*INORD3.S

SUBROUTINE INORD(N,IOH,INIOR,IHO,IROL,IRQTY)

C*****

PARAMETER NQQQ=38

C*****

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

THIS SUBROUTINE DETERMINES THE RECEIPT TIME FOR ASSETS
ON-ORDER AT THE BEGINNING OF THE SIMULATION. IT IS
CALLED AT THE BEGINNING OF EACH REPLICATION AND ITS
OUTPUT IS STORED IN TWO TABLES (IDUE & IORQ) TO BE USED
FOR EACH SHORTAGE FACTOR WITHIN THE REPLICATION. WHEN
THE ON-ORDER QUANTITY IS ONE AND A HALF TIMES THE
EOQ OR LESS, THE TOTAL ON-ORDER ASSETS ARE ASSUMED
TO BE THE RESULT OF ONE PROCUREMENT ACTION AND
ALL DUE IN AT ONE TIME. WHEN ON-ORDER ASSEST
EXCEED ONE AND A HALF EOQ'S, THIS SUBROUTINE THEN
COMPUTES A DJE-IN DATE FOR EACH REMAINING EOQ SO
LONG AS THE COMPUTED DUE-IN DATE IS GREATER THAN
IFIVE (IE,500). WHEN THE DUE-IN DATE IS LESS
THAN IFIVE, THE REMAINING ON ORDER IS SET AS DUE-IN
AT IFIVE.

COMMON/IDBUG/IDBUG

COMMON/IEBUG/IEBUG

COMMON/NDHIS/NDHIS

COMMON/NDEMND/NDEMND(1,NQQQ)

COMMON/NRETUR/NRETUR(1,NQQQ)

COMMON/ITDAY/ITDAY

COMMON/ITMNTN/ITMNTN

COMMON/LTPROD/LTPROD(1)

COMMON/LTADM/LTADM(1)

COMMON/IDUE/IDUE(NQQQ)

COMMON/IORQ/IORQ(NQQQ)

COMMON/JCTR/JCTR

C

C

C

WRITE DEBUG MESSAGE

IF(IEBUG.EQ.1) WRITE(6,901)IOH,INIOR,IHO,IROL,IRQTY
901 FORMAT("..INORD.S.. IOH=",I6," INIOR=",I6," IHO=",I6," IROL=",
&I6," IRQTY=",I6)

C

C

C

C

ZERO ON-ORDER ARRAYS

C

DO 5 I=1,NQQQ

IDUE(I)=0

IORQ(I)=0

5 CONTINUE

C

C

C

LOGIC TO TEST LEAD-TIME SENSITIVITY

INORD3.S

```

IOPT=1
IF(IOPT.NE.2) GO TO 7
C
C      SET ALL ON-ORDER STOCK DUE-IN ONE LEAD-TIME IN
C      THE FUTURE.
C      THIS IS A VERY PESSIMISTIC ASSUMPTION.
C
JCTR=0
IF(INIOR.LE.0) RETURN
JCTR=1
IDUE(1)=(LTADM(N)+LTPROD(N))*ITMNTN
IORQ(1)=INIOR
RETURN
7 CONTINUE
C-----END OF SENSITIVITY LOGIC
C
C      LET
C
C      ISTKOBJ=STOCKING OBJECTIVE
C      NETASSTS=VET AVAILABLE STOCK
C      IFIVE=CLOCK TIME 5 DAYS INTO SIMULATION
C      CHECKIOR=MAX SIZE FOR ONE REQUISITION
C      IHOLDIOR=WORKING VAR. FOR REMAINING ON-ORDER STOCK
C      NREQQS00=MAX REQUISITION ON-ORDER (INTEGER)
C
C      IOR=INIOR
C      ISTKOBJ=IORL+IRQTY
C      NETASSTS=IOH+IOR-IBO
C      IFIVE=5*ITDAY
C      CHECKIOR=FLOAT(IRQTY)*1.5
C      IHOLDIOR=IOR
C
C      IF NREQQS00 > NDHIS , SET EQUAL TO NDHIS TO AVOID ARRAY OV
C
C      KQTR=1
C      COMPUTE MAXIMUM NO OF OUTSTANDING ORDERS
NREQQS00=IOR/IRQTY +1
IF(NREQQS00.GT. NDHIS )NREQQS00= NDHIS
NDMDS=0
DO 50 J=1,NREQQS00
DO 10 K=KQTR,8
NDMDS=NDMDS+NDEMND(N,K)-NRETUR(N,K)
IAVAIL=NETASSTS+NDMDS
IF(IEBUG.EQ.1) WRITE(6,903)J,K,NDMDS,IAVAIL
903 FORMAT("      ORDER NO=",I2,"      QTR=",I2,"      NET DEMAND=",I6,
&      "      IAVAIL=",I6)
IF(IAVAIL.GT.ISTKOBJ)GOTO 20
10 CONTINUE
C
CALL INTIME(N,RLT)
XMNTN=(RLT-24.)
IDUE(J)=XMNTN*FLOAT(ITMNTN)
IF(IDUE(J).LT.IFIVE)IDUE(J)=IFIVE

```

```

      GOTO 30
C
C      COMPUTE RANDOM TIME WITHIN QTR FOR ORDER RECEIPT.
C      SET IDUE(J)=DUE-IN TIME FOR ORDER NO.J.
C
20  T=RANDU(.2)
    KQTR=K+1
    ORDQTR=K
    CALL INTIME(N,RLT)
    XMNTH = RLT + 3.*(T-ORDQTR)
    XIDUE=XMNTH*FLOAT(ITMNTH)
    IDUE(J)=IFIX(XIDUE)
C
C      LIMIT DUE-IN TIME TO AT LEAST 5 DAYS IN FUTURE.
C
    IF(IDUE(J).GT.IFIVE)GOTO 25
    IDUE(J)=IFIVE
    GOTO 30
C
C      SET IORQ(J)=QUANTITY FOR ORDER J.
C
25  IF(IOR.LE.CHECKIOR)GOTO 30
    IORQ(J)=IRQTY
    GO TO 35
30  CONTINUE
    IORQ(J)=IOR
35  CONTINUE
    IF(J.EQ. NDHIS )IORQ(J)=IHOLDIOR
    IF(IEBUG.EQ.1) WRITE(6,904)T,XMNTH,IDUE(J),IORQ(J),IHOLDIOR
904  FORMAT("      T=",F10.8," XMNTH=",F10.6," IDUE(J)=",I7,
&" IORQ(J)=",I7," ON-ORDER BEFORE PLACING THIS ORDER =",I7)
C
C      UPDATE COUNTER VARIABLES
C
    IF(IHOLDIOR-IORQ(J))200,100,40
40  IHOLDIOR=IHOLDIOR-IORQ(J)
    ISTKOBJ=ISTKOBJ+IRQTY
    IOR=IHOLDIOR
50  CONTINUE
100 JCTR=J
    RETURN
C
C      PRINT ERROR MESSAGE
C
200  CONTINUE
    WRITE(6,213)
213  FORMAT("*** **INORD---ERROR---")
    IDBUG=1
    RETURN
END

```

INORD3.S

*INTIME.S

```
C
SUBROUTINE INTIME(N,RLT)
C   THIS ROUTINE COMPUTES THE ACTUAL LEADTIME, RLT,
C   FOR A GIVEN REPLENISHMENT ORDER FOR ITEM N,
C
C   DEFINITIONS
C   CP(I) = PROB( ACT LT TO PRED LT <= RATIO(I))
C   RATIO(I) = RATIO OF ACTUAL LEADTIME TO PREDICTED LEADTIME
C               ASSOCIATED WITH CLASS I
C
COMMON/IWT/IWT(20)
COMMON/LTPROD/LTPROD(1)
COMMON/LTADM /LTADM(1)
COMMON/IRNDLT/IRNDLT
COMMON/ITIME/ITIME
C
DIMENSION CP(10),RATIO(10)
C
C   THE CP(I) DATA IS FOR A GAMMA PDF WITH MEAN = 1
C   AND COEF. OF VAR. = .353. >  $1/\sqrt{\text{ALPHA}}$ 
C   HENCE, ALPHA = 8, BETA = 1/8.
C   THIS IS THE MEDIAN COEF OF VAR. FOR THE 62 ITEMS
C   REPORTED IN APPENDIX D OF HAYYA(1980).
C
DATA CP/.000,.077,.255,.490,.700,
&      .844,.927,.969,.987,1.00/
DATA RATIO/0.3,.5,.7,    .9, 1.10,
&      1.3, 1.5, 1.7, 1.9, 2.70/
C
IDBUG=IWT(1)
C
C   IF IRNDLT = 1, MONTE CARLO TO DETERMINE LEADTIME.
C   OTHERWISE, SET LEADTIME = PREDICTED VALUE.
C
IF(IRNDLT.EQ.1) GO TO 17
C
C   SET LEADTIME TO PREDICTED VALUE
C
RLT = LTPROD(N) + LTADM(N)
RETURN
C
C   OBTAIN A U(0,1) RANDOM NUMBER R
C
17 CONTINUE
R=RANDU(.2)
C
C   FIND PROB. CLASS THAT INCLUDES R
C
DO 20 I=2,10
  IF(R.LE.CP(I)) GO TO 40
20 CONTINUE
```

INORD3.S
INTIME

40 CONTINUE

C
C
C

INTERPOLATE FOR EXACT RATIO V

IL=I-1
RDELTRATIO(I)-RATIO(IL)
CPDELTRATIO(I)-RATIO(IL)

C
C

V=RATIO(IL) + (RDELTRATIO/CPDELTRATIO)*(R-RATIO(IL))

C

LEADTM= LTPROD(N) + LTADM(N)
RLT=V*LEADTM

C

IF (IDBUG.EQ.1)WRITE(6,113)ITIME,N,R,LEADTM,V,RLT
113 FORMAT(" ---INTIME-ITIME=",I8," N=",I4," R=",F4.2,
" PLANNED LEADTIME=",I3," V=",F4.2,
" RANDOM LEADTIME (MNTHS)=",F5.2)

&
&
C

IF(RLT.LT..1) RLT=.1

C

RETURN
END

INORD3.S
INTIME

*#RUN=;HEDG/OBJ/ENTR3.S(BCD,NOGO)

*#ENTR3.S

```
      SUBROUTINE ENTERB(N,IQTY,IPRI,JTIME)
C      THIS ROUTINE BACKORDERS REQUISITIONS FOR ITEM N
C      IQTY=QUANTITY PLACED ON BACKORDER
C      IPRI=1 HIGH-PRIORITY REQUISITION
C      IPRI=2 OTHERWISE
C      JTIME= CLOCK TIME RFQ WAS RECEIVED
      COMMON/IDBUG/IDBUG
      COMMON/NBMAX/NBMAX
      COMMON/NLOCBK/NLOCBK
      COMMON/NBOIU/NBOIU(1)
      COMMON/NBOIR/NBOIR(1)
      COMMON/NBOTR/NBOTR(1)
      COMMON/NBOTU/NBOTU(1)
      COMMON/NBOPT/NBOPT(1)
      COMMON/IBACPT/IBACPT(1)
      COMMON/ICANCL/ICANCL(1)
      COMMON/IDFSNB/IDFSNB(1)
      COMMON/ILOCBK/ILOCBK(1)
      COMMON/IPRIOR/IPRIOR(1)
      COMMON/IQTYB/IQTYB(1)
      COMMON/ITIME/ITIME
      COMMON/ITMBAC/ITMBAC(1)
C      RESERVE A STORAGE LOCATION FOR THIS INFORMATION
      IPT=ILOCBK(NLOCBK)
C
C      IF BO FILE IS FULL, CANCEL THIS REQUISITION
C      OTHERWISE, GO TO 5 AND RECORD THIS BO.
      IF(NLOCBK.GE.1) GO TO 5
C      THERE'S NO ROOM. CANCEL IT.
      IF(IDBUG.GE.1) WRITE(6,7) ITIME,IQTY,IPRI
7  FORMAT(4X,'***ENTERB--ITIME=',I8,' ',
      'BO FILE IS FULL. CANCEL ',
      ' REQ FOR ',I8,' UNITS, PRI=',I5)
C
      CALL CUM(ICANCL,IQTY,N)
      RETURN
C
5  CONTINUE
C      UPDATE BACKORDER COUNTERS
C
      NBOTR(N)=NBOTR(N)+1
      NBOTU(N)=NBOTU(N)+IQTY
      IF(IPRI.NE.1)GO TO 10
      NBOIR(N)=NBOIR(N)+1
      NBOIU(N)=NBOIU(N)+IQTY
10  CONTINUE
      IF(IDBUG.NE.1)GO TO 15
      WRITE(6,13)N,IQTY,IPRI,NBOIU(N),NBOTU(N),NBOIR(N),NBOTR(N),IPT
13  FORMAT(4X,'***ENTERB--N=',I5,' IQTY=',I5,' IPRI=',I5,
      &' NBOIU=',I5,' NBOTU=',I5,' NBOIR=',I5,' NBOTR=',I5,
```

ENTR3.S

```

      8 IPT=,15)
15  CONTINUE
C    DID THIS REQUISITION CAUSE THE BACKORDER FILE TO OVERFLOW
      NLOCBK=NLOCBK-1
      IF(NLOCBK.GE.0)GO TO 20
C    WRITE ERROR MESSAGE
      WRITE(6,91)
91  FORMAT(1HC,2)X,"ERROR--BACKORDER FILE OVERFLOW,FILE DUMP ON",
      & " NEXT PAGE")
      WRITE(6,82)
82  FORMAT(1H1,1)X,23H**BACKORDER FILE DUMP**)
      DO 83 K=1,NBMAX
83  WRITE(6,84) K,ITMBAC(K),IDFSNB(K),IPRIOR(K),IQTYB(K),IBACPT(K)
84  FORMAT(1H,3X,7HREC NO=,13,5X,7HITMBAC=,17,5X,7HIDFSNB=,110,5X,7HI-
      & PRIOR=,11,5X,5HIQTYB=,17,5X,7HIBACPT=,17)
      RETURN
C    RECORD QUANTITY,PRIORITY,FSN ID,AND TIME DATA FOR THIS BO REQ
20  ITMBAC(IPT)=JTIME
      IDFSNB(IPT)=N
      IPRIOR(IPT)=IPRI
      IQTYB(IPT)=IQTY
C    ARE ANY OTHER BACKORDERS OUTSTANDING ON ITEM N
      IF(NBOPT(N).GT.0) GO TO 40
C    RECORD POINTER DATA
      NBOPT(N)=IPT
      IBACPT(IPT)=0
      RETURN
C    IS THE NEW BO A PRIORITY 1 REQUISITION
40  IF(IPRI.EQ.1) GO TO 60
C    NOTE-- NEW LOW PRIORITY BACKORDERS ARE INSERTED LAST ON THE
C    BACKORDER CHAIN. THE REMAINING STEPS IN THIS PORTION
C    OF THE SUBROUTINE ACCOMPLISH THIS OBJECTIVE
C    SET JPT EQUAL TO THE FILE LOCATION NO OF THE FIRST BACKORDERED
C    REQUISITION IN THE CHAIN
      JPT=NBOPT(N)
C    IS JPT THE LAST LINK IN THE CHAIN
49  IF(IBACPT(JPT).EQ.0) GO TO 50
      KPT=IBACPT(JPT)
      JPT=KPT
      GO TO 49
C    RECORD NEW POINTERS
50  IBACPT(JPT)=IPT
      IBACPT(IPT)=0
      RETURN
C    SET JPT EQUAL TO LOCATION NO OF FIRST BO ON CHAIN
60  JPT=NBOPT(N)
C    IS BACKORDER JPT A HIGH PRI BO
61  IF(IPRIOR(JPT).NE.1) GO TO 80
C    IS JPT THE LAST BACKORDER ON THE CHAIN
      IF(IBACPT(JPT).EQ.0) GO TO 62
      KPT=JPT
      JPT=IBACPT(KPT)

```

ENTRB3,S

```

      GO TO 61
C      INSERT NEW 30 AS LAST LINK ON CHAIN
62 IBACPT(JPT)=IPT
   IBACPT(IPT)=J
   RETURN
C      IS JPT THE ONLY HO ON CHAIN
80 IF(JPT.NE.NBOPT(N)) GO TO 81
C      INSERT NEW 30 AS FIRST LINK ON CHAIN
   NBOPT(N)=IPT
   IBACPT(IPT)=JPT
   RETURN
C      INSERT NEW 30 AS LINK BETWEEN KPT AND JPT
81 IBACPT(KPT)=IPT
   IBACPT(IPT)=JPT
   RETURN
END

```

ENTRB3,S

CATALOG FILE DESCRIPTION= HEDG/DATAB3.S

```

10*ZRUJ=HEDGZOBJZDATAB3.OOCHD,UIIRDFOSZRANDU.O,R
20*# HEDGZOBJZDF-SPR3,OWHEDSZID62DATA"03",R
30*DATAB3.S
40*
50C      THIS ROUTINE SELECTS INSIM RECORDS WITH AVERAGE ANNUAL
60C      DEMANDS IN CTR 1-3 THAT LIE IN THE RANGE RLB TO RUB.
70C      REQUISITION COUNTERS ARE GENERATED BY MONTE CARLO TO BE
80C      CONSISTENT WITH UNIT DEMAND RECORDS.
90C
100C     THIS ROUTINE ASSUMES INPUT RECORDS ARE BINARY FORMAT AS
110C     DEFINED IN INSIM VOL. 1, APPENDIX A.
120C
130     PARAMETER NDOQ=33
140     DIMENSION FSN(4),NOUN(2),IDEM(NDOQ),IRET(NDOQ),IREQ(NDOQ),MGTC(4)
150     CHARACTER MGTC(4)
160     CALL FPARAM(1,132)
170     READ(5,3) LOIT,LPRNT,LSKIP
180     3    FORMAT(V)
190     PRINT, "      LOIT =",LOIT,"      LPRNT =",LPRNT,"      LSKIP =",LSKIP
200     READ(5,3)RLB,RUB
210     PRINT,"      LOWER BOUND =",RLB,"      UPPER BOUND =",RUB
220C
230C     INITIALIZE RANDOM NUMBER SEED
240C
250     R=RANDU(-.1)
260C
270C     INITIALIZE COUNTERS
280C
290     KNT=0
300     IOU=0
310C
320     ISKIP=0
330     10    CONTINUE
340         ISKIP = ISKIP + 1
350     15    CONTINUE
360         KNT = KNT + 1
370         IF(MOD(KNT,10)EQ.0) WRITE(6,3)"INPUT ITEM # ",KNT
380         READ(8,END=200)ALC,FSN,IR,UM,UCOST,NOUN,MGTC,IOH,IOR,
390         LTADM,LTPRX),IPP,RIPP
400     13    FORMAT(A2,A2,A4,A6,A3,11,A2,F9.2,A6,A4,2A1,A4,A2,
410         2I7,3I2,F4.2)
420     23    FORMAT(17,1X,2A3,A4,1X,A6,A3,1X,12,A3,F10.2,1X,A6,A4,1X,
430         2A1,A4,A2,2I7,2X,3I3,F5.2)
440     33    FORMAT(A2,A2,A4,A6,A3,11,24I7)
450     43    FORMAT(3(/10X,8I7),T95,110,T120,F11.0)
460     53    FORMAT(3(/10X,8I7),T105,110)
470     63    FORMAT(3(/10X,8I7),T115,110)
480     READ(8)ALC,FSN,IR,IDEM
490     READ(8)ALC,FSN,IR,IRET
500     READ(8)ALC,FSN,IR,IREQ

```

```

5100C
5200C
5300C
5400C
5500C
5600C
5700C
5800C
5900C
6000C
6100C
6200C
6300C
6400C
6500C
6600C
6700C
6800C
6900C
7000C
7100C
7200C
7300C
7400C
7500C
7600C
7700C
7800C
7900C
8000C
8100C
8200C
8300C
8400C
8500C
8600C
8700C
8800C
8900C
9000C
9100C
9200C
9300C
9400C
9500C
9600C
9700C
9800C
9900C
10000C
10100C
10200C
10300C
10400C

      REJECT THIS ITEM IF TOTAL DEMANDS OR RETURNS EXCEED
      20 TIMES THE UPPER-BOUND BOUND RATE.
      IF (IDEM(3).GT."4000") GO TO 15
      IF (IRFET(3).GT."4000") GO TO 15

      SELECT ITEMS WITHIN THE RANGE REL-LEAD-RUB
      ID=0
      IR=0
      DO 80 I=1,3
      ID=ID+IDEM(I)
      IR=IR+IRFET(I)
      60 CONTINUE
      TD=UCOST*FLOAT(ID-IR)/2.
      IF (TD.LT.RLB) GO TO 15
      IF (TD.GT.RUB) GO TO 15
      KDEM=0
      KRFET=0
      KREQ=0
      DO 110 I=1,NQ00
      KDEM=KDEM+IDEM(I)
      KRFET=KRFET+IRFET(I)
      KREQ=KREQ+IRFREQ(I)
      IF (IDEM(I).GT.10000) GO TO 15
      IF (IRFET(I).GT. 2000) GO TO 15
      110 CONTINUE
      PRINT THIS ITEM
      REJECT THIS ITEM IF TOTAL DEMANDS OR RETURNS EXCEED
      20 TIMES THE UPPER-BOUND BOUND RATE.
      RMAX=20.*RUB
      RATE=UCOST*FLOAT(KDEM)*4./FLOAT(NQ00)
      IF (RATE.GT.RMAX) GO TO 15
      RATE=UCOST*FLOAT(KRFET)*4./FLOAT(NQ00)
      IF (RATE.GT.RMAX) GO TO 15
      IF (ISKIP.LT.LSKIP) GO TO 10
      ISKIP=0
      GENERATE REQUISITION COUNTS FOR FIRST 8 QUARTERS
      DO 100 N=1,NQ00

```

```

1000 IOUT=IOUT+1
1010 IOUTL=I
1020 IOUTF=I
1030 IF(IOUTF,15,0000) GO TO 31
1040
1050
1060
1070
1080
1090
1100
1110
1120
1130 GO TO 100
1140
1150
1160
1170
1180
1190
1200
1210
1220
1230
1240
1250
1260
1270
1280
1290
1300
1310
1320
1330
1340
1350
1360
1370
1380
1390
1400
1410
1420
1430
1440
1450
1460
1470
1480
1490
1500
1510
1520
1530
1540
1550
1560
1570
1580

```

IREQ=IREQ+1
 IOUTL=I
 IOUTF=I
 IF(IOUTF,15,0000) GO TO 31

 GIVEN TOTAL DEMAND (IQTY) IN THE QUARTER, OBTAIN
 THE NUMBER OF REQUISITIONS BY MONTH CARD.

 GO TO 100
 R=RANDM(.2)
 CALL GETR-OCR, IOUT, IREQ
 IREQ=IREQ+1
 IOUTL=IOUTL+1
 IF (IOUTL,15,IQTY) GO TO 60

 IREQCH=IREQ
 GO TO 100
 IF IREQCH=0
 100 CONTINUE
 TDOLE=UCOST*IOUTL*(KDEM)
 WRITE(6,23) IREQ,ALC,FSN,IQ,UM,UCOST,NOUN,MGTCD,
 ICH,IOR,LTADM,LTPROD,IPP,PIPP
 WRITE(6,73) ICH,KREF,KREF,TDOLE
 73 FORMAT(25,3I8,F10.0)
 IF(IOUT.GT.IPRINT) GO TO 200
 WRITE(6,77) IDEM
 WRITE(6,77) IRET
 WRITE(6,77) IREQ
 77 FORMAT(10X,16I7)
 1350
 1360
 1370
 200 CONTINUE
 1390
 1400

OUTPUT THIS ITEM TO FILE 9

WRITE(9)ALC,FSN,1,UM,UCOST,NOUN,MGTCD,ICH,IOR,LTADM,
 LTPROD,IPP,PIPP
 83 FORMAT(17,1X,A2,A2,A4,A6,A3,11,A2,F9.2,A6,A4,2A1,A4,A2,
 2I7,3I2,F4.2)
 93 FORMAT(5X,8I7)
 WRITE(9)ALC,FSN,2,IDEM
 WRITE(9)ALC,FSN,3,IRET
 WRITE(9)ALC,FSN,4,IREQ
 IOUT=IOUT+1
 300 CONTINUE
 IF(IOUT.GT.IOUT) GO TO 900
 GO TO 10
 900 CONTINUE
 WRITE(6,903) KNI, IOUT
 903 FORMAT(77777T40,'RECORDS IN =',I5/T40,'RECORDS OUT =',I5)
 STOP
 END

APPENDIX B

FORTRAN SOURCE LISTINGS

```

C---BEGIN LAGRANGIAN LOOP
C
  DO 200 MRUN=1,NMUN
    IRUN=MRUN
    COSHRT=CSHORT(MRUN)
    IF(IDBUG.EQ.1)WRITE(6,8090)COSHRT
C
    8090 FORMAT(//10('*****'),'BEGIN SIMULATION ',
  & WITH COSHRT =',F10.4,10('*****')//)
C
    REWIND ITEM INPUT FILE INLU
    REWIND INLU
C
C---RECORD RANDOM NUMBER SEED
C
C
C---INITIALIZE RANDOM NUMBER STREAM
  INITIALIZE IEMPR3 RANDOM NUMBER STREAM
  IF(IISEED.EQ.0) RNLAST = RANDEN(-.2)
  IF(IISEED.NE.0) RNLAST = RANDEN(-ABS(FLOAT(IISEED)))
C
  INITIALIZE GENERAL RANDOM NUMBER STREAM
C
  RNLAST = IISEED
  IF(IISEED.EQ.0) RNLAST = RANDU(-.1)
  IF(IISEED.NE.0) RNLAST = RANDU(-ABS(RNLAST))
C
C
  CALL ZERO
C

```

NEW HEDGS
MAIN
PROGRAM
CODE

INVRSM.S