U.S. DEPARTMENT OF COMMERCE
National Technical Information Service
PB-250 439

Hypercube Queuing Model Program Description

New York City-Rand Inst.

Prepared For
Department of Housing & Urban Development

July 1975

KEEP UP TO DATE

Between the time you ordered this report—which is only one of the hundreds of thousands in the NTIS information collection available to you—and the time you are reading this message, several *new* reports relevant to your interests probably have entered the collection.

Subscribe to the **Weekly Government Abstracts** series that will bring you summaries of new reports *as soon as they are received by NTIS* from the originators of the research. The WGA's are an NTIS weekly newsletter service covering the most recent research findings in 25 areas of industrial, technological, and sociological interest—invaluable information for executives and professionals who must keep up to date.

The executive and professional information service provided by NTIS in the **Weekly Government Abstracts** newsletters will give you thorough and comprehensive coverage of government-conducted or sponsored research activities. And you'll get this important information within two weeks of the time it's released by originating agencies.

WGA newsletters are computer produced and electronically photocomposed to slash the time gap between the release of a report and its availability. You can learn about technical innovations immediately—and use them in the most meaningful and productive ways possible for your organization. Please request NTIS-PR-205/PCW for more information.

The weekly newsletter series will keep you current. But *learn what you have missed in the past* by ordering a computer **NTISearch** of all the research reports in your area of interest, dating as far back as 1964, if you wish. Please request NTIS-PR-186/PCN for more information.

WRITE: Managing Editor
5285 Port Royal Road
Springfield, VA 22161

Keep Up To Date With SRIM

SRIM (Selected Research in Microfiche) provides you with regular, automatic distribution of the complete texts of NTIS research reports *only* in the subject areas you select. SRIM covers almost all Government research reports by subject area and/or the originating Federal or local government agency. You may subscribe by any category or subcategory of our WGA **(Weekly Government Abstracts)** or **Government Reports Announcements and Index** categories, or to the reports issued by a particular agency such as the Department of Defense, Federal Energy Administration, or Environmental Protection Agency. Other options that will give you greater selectivity are available on request.

The cost of SRIM service is only 45¢ domestic (60¢ foreign) for each complete microfiched report. Your SRIM service begins as soon as your order is received and processed and you will receive biweekly shipments thereafter. If you wish, your service will be backdated to furnish you microfiche of reports issued earlier.

Because of contractual arrangements with several Special Technology Groups, not all NTIS reports are distributed in the SRIM program. You will receive a notice in your microfiche shipments identifying the exceptionally priced reports not available through SRIM.

A deposit account with NTIS is required before this service can be initiated. If you have specific questions concerning this service, please call (703) 451-1558, or write NTIS, attention SRIM Product Manager.

This information product distributed by

NTIS U.S. DEPARTMENT OF COMMERCE
National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

BIBLIOGRAPHIC DATA SHEET	1. Report No.	2.	PB 250 439

4. Title and Subtitle
Hypercube Queuing Model: Program Description

5. Report Date
July 1975
6. 28

7. Author(s)
Richard C. Larson

8. Performing Organization Rept. No. R-1688/3-HUD

9. Performing Organization Name and Address
The New York City-Rand Institute
545 Madison Avenue
New York, New York 10022

10. Project/Task/Work Unit No.

11. Contract/Grant No.
H-2164

12. Sponsoring Organization Name and Address
U.S. Department of Housing and Urban Development
Office of Policy Development and Research
541 Seventh Street, S.W.
Washington, D.C. 20410

13. Type of Report & Period Covered

14.

15. Supplementary Notes The development of the Hypercube Model was also supported by a grant to the Massachusetts Institute of Technology from the National Science Foundation.

16. Abstracts
The Hypercube Queuing Model is a computer program that calculates selected performance measures of emergency service systems (police, fire and medical). It is especially useful for assisting police and emergency medical agencies in designing response districts for their mobile vehicles. This report provides a listing of the computer program and is published to assist data processing personnel in modifying the program and installing it. A nontechnical introduction to the model is provided in R-1688/1-HUD, and a user's manual is contained in R-1688/2-HUD. Theoretical development of the model is contained in R-1238-HUD and R-1493-HUD.

17. Key Words and Document Analysis. 17c. Descriptors

17b. Identifiers/Open-Ended Terms

2-24-76

17c. COSATI Field/Group P D & R REPORTS CONTROL

18. Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages
	20. Security Class (This Page) UNCLASSIFIED	

This report was prepared as an account of work sponsored by the United States government, Office of Policy Development and Research, Department of Housing and Urban Development (HUD), under Contract H-2164. Neither the United States nor HUD, nor any person acting on behalf of HUD, makes any warranty, expressed or implied, or assumes responsibility for the accuracy or completeness of the information herein, or assumes liability with respect to the use of any information or method herein.

The New York City-Rand Institute is a nonprofit research institution founded, as its Articles of Incorporation state, "... primarily to conduct programs of scientific research and study, and provide reports and recommendations, relevant to the operations, planning or administration of the City of New York." The Institute was established in 1969 by the City of New York and The Rand Corporation as a center for the continuing application of scientific and analytic techniques to problems of urban life and local government. It is governed by a Board of Trustees appointed jointly by the City and Rand.

Published by The Rand Corporation

HYPERCUBE QUEUING MODEL: PROGRAM DESCRIPTION

PREPARED FOR THE OFFICE OF POLICY DEVELOPMENT AND RESEARCH, DEPARTMENT OF HOUSING AND URBAN DEVELOPMENT

RICHARD C. LARSON

R-1688/3-HUD
JULY 1975

THE
NEW YORK CITY
RAND
INSTITUTE

Rand
SANTA MONICA, CA. 90406

PREFACE

The Hypercube Queuing Model is a computer program that calculates selected performance measures of emergency service systems (police, fire, and medical). It is especially useful for assisting police and emergency medical agencies in designing response districts for their mobile vehicles.

The model is completely described in five reports, available from The Rand Corporation. There are two versions of the model--an *exact* model and an *approximate* model--both of which are incorporated in a single computer program. The mathematical formulations of the two models are presented in the following reports:

- Richard C. Larson, *A Hypercube Queuing Model for Facility Location and Redistricting in Urban Emergency Services*, R-1238-HUD.
- Richard C. Larson, *Urban Emergency Service Systems: An Iterative Procedure for Approximating Performance Characteristics*, R-1493-HUD.

The computer program itself is described in three volumes. The first volume is a summary, describing potential applications of the Hypercube Queuing Model, how it works, when it should be used in preference to other models, and the resources needed to use it:

- Jan M. Chaiken, *Hypercube Queuing Model: Executive Summary*, R-1688/1-HUD.

The second volume is a manual for users of the model. It describes and gives examples of applications, describes the procedures to operate the computer program once it has been installed in the user's computer system, and discusses the decisions to be made (such as the dispatching strategy employed), the results, and the costs and requirements for operation:

- Richard C. Larson, *Hypercube Queuing Model: User's Manual*, R-1688/2-HUD.

The present volume (R-1688/3-HUD) comprises a listing of the computer program for the Hypercube Queuing Model, which will be of use to data processing personnel who may wish to modify the program or keypunch it. The program is available on cards or tape and may be obtained by writing to either of the addresses shown in the Appendix.

The author, a consultant to The Rand Corporation, is an Associate Professor of Electrical Engineering and Urban Studies at the Massachusetts Institute of Technology. All work on this model, from design through documentation, has been supported jointly by grants to the Massachusetts Institute of Technology from the National Science Foundation (NSF) and by contracts between The New York City-Rand Institute and the U.S. Department of Housing and Urban Development (HUD). The most recent work has been funded by the Division of Social Systems and Human Resources (Research Applied to National Needs) at NSF and the Office of Policy Development and Research at HUD.

The project funded by HUD has resulted in the development, field testing, and documentation of a number of models for improving the deployment of municipal emergency services. Further information about the models themselves and case studies of applications of the models in several cities can be obtained from Rand or HUD (at the addresses in the Appendix).

CONTENTS

HYPERCUBE QUEUING MODEL: PROGRAM DESCRIPTION

The version of the Hypercube Queuing Model listed in this report is a batch program issued in 1975. We anticipate possible future revisions based on suggestions from users, and an interactive version of the program, which is currently under development at the Massachusetts Institute of Technology, may also be released at a later date. When the program is requested from either of the addresses listed in the Appendix, the latest version will be provided, together with any materials needed to update the information in this report.

The program is provided on cards at a cost of $35 or on tape at a cost of $25, plus $15 if we supply the tape. It is written in the programming language PL/I[1] and can only be operated on computer systems that have a PL/I compiler.

Information required by the program is contained on data cards, which must be prepared in accordance with the instructions in the user's manual.[2] System control cards to compile, link-edit, and execute the program must appear before the source deck listed here, and the source deck is followed directly by the data cards.

If the program is to be operated repeatedly, we recommend creation of an object module containing the compiled program, since compilation costs considerably more than typical runs of the program. The core requirements of the program vary according to the value of the variable ESTSTAT, which is a code specifying either the exact hypercube model or the approximate hypercube model, and the dimensions of the variables M (number of response units), R (number of atoms), and NUM (number of times the program is to be run). See Section 6.4 of the user's manual for further details.

Most runs on an IBM system 370/168 require under 300K bytes of core storage, so we recommend testing the program first with this amount of core and then adjusting subsequent runs in accordance with the actual amount of core needed with the user's data cards on the test run.

The program listing which follows is generated by the compiler and indicates line numbers, the program level for subroutines and blocks,

and the nest level for do-loops. Also included is an attribute and cross-reference table, which describes the characteristics of each program symbol and the line number(s) on which it appears.

```
STMT LEVEL NEST
  1                HYPCUBE:                                                                  00000010
                     PROCEDURE OPTIONS (MAIN);                                               00000020
  2     1            DCL                                                                     00000030
                        (N,                                                                  00000040
                        M,                                                                   00000050
                        R,                                                                   00000060
                        NUM,                                                                 00000070
                        ESTSTAT,                                                             00000080
                        DEBUG,                                                               00000090
                        NCOEFF,                                                              00000100
                        N_BD_STS) FIXED BIN;                                                 00000110
                   /*M=TOTAL NUMBER OF RESPONSE UNITS(E.G.,PATROL CARS) R=TOTAL NUMBER      00000120
                   OF REPORTING AREAS N=TOTAL NUMBER OF BINARY STATES ON HYPERCUBE          00000130
                   NCOEFF=NUMBER OF COEFFICIENTS THAT MUST BE STORED TO SOLVE EQUAS         00000140
                   N_BD_STS=NUMBER OF STATES IN REDUCED BIRTH AND DEATH MODEL NUM=TOTAL     00000150
                   NUMBER OF RUNS*/                                                        00000160
                   /*'STAT' SIGNALS TYPE OF STATISTICAL OUTPUT: 0 - ONLY HYPERCUBE         00000170
                   STATISTICS; 1 - ONLY ESTIMATED STATISTICS; 2 - BOTH */                  00000180
  3     1              DEBUG=0;                                                            00000190
  4     1            LOOP:                                                                 00000200
                        GET DATA (M,R,NUM,ESTSTAT,DEBUG);/*FIRST DATA CARD  */             00000210
  5     1              IF DEBUG>0 THEN                                                     00000220
  6     1                 DO;                                                              00000230
  7     1     1             PUT LIST('START') SKIP;                                       00000240
  8     1     1             CALL PRNT_TIME;                                               00000250
  9     1     1          END;                                                             00000260
 10     1              IF ESTSTAT¬=1 & M>15 THEN                                           00000270
 11     1                 CALL ERR(1);                                                     00000280
 12     1              IF R>200 THEN                                                       00000290
 13     1                 CALL ERR(2);                                                     00000300
 14     1              IF NUM>10 THEN                                                      00000310
 15     1                 NUM=10;                                                          00000320
 16     1              IF ESTSTAT<0|ESTSTAT>2 THEN                                         00000330
 17     1                 CALL ERR(3);                                                     00000340
 18     1              N=1;                                                                00000350
 19     1              IF ESTSTAT¬=1 THEN                                                  00000360
 20     1                 N=2**M;                                                          00000370
 21     1              NCOEFF=1;                                                           00000380
 22     1              IF ESTSTAT¬=1 THEN                                                  00000390
 23     1                 NCOEFF=N + M*N/2;                                                00000400
 24     1              N_BD_STS=M+1;                                                       00000410
 25     1              CALL CALCVAL;                                                       00000420
 26     1              GO TO LOOP;                                                         00000430
 27     1            FINISH:                                                               00000440
                        RETURN;                                                          00000450
 28     1            CALCVAL:                                                            00000460
                      PROC;                                                              00000470
 29     2              IF DEBUG>0 THEN                                                   00000480
 30     2                 PUT LIST('CALCVAL ENTERED') SKIP;                              00000490
 31     2            DCL                                                                 00000500
                        (L(R),                                                           00000510
                        T(M+1,R),                                                        00000520
                        RO,                                                              00000530
                        TR(R,R),                                                         00000540
                        SEC(M,R),                                                        00000550
                        IRO,                                                             00000560
```

```
STMT LEVEL NEST
                    X(R),                                                    00000570
                    Y(R),                                                    00000580
                    XSPEED,                                                  00000590
                    YSPEED,                                                  00000600
                    SERVTM,                                                  00000610
                    T_RATE(NCOLORS),                                         00000620
                    XDUM,                                                    00000630
                    YDUM,                                                    00000640
                    MINVAL,                                                  00000650
                    CAPACITY,                                                00000660
                    C(M+1,R),                                                00000670
                    XUCM(M),                                                 00000680
                    YUCM(M),                                                 00000690
                    XICM(M),                                                 00000700
                    YICM(M),                                                 00000710
                    QUFAC(M),                                                00000720
                    STMI(R),                                                 00000730
                    PAT_SPEED,                                               00000740
                    STPROB(N),                                               00000750
                    CONV_METRIC,                                             00000760
                    EPSILON,                                                 00000770
                    INFINITY,                                                00000780
                    SUM,                                                     00000790
                    XSTPROB(N,NUM),                                          00000800
                    MU(P),                                                   00000810
                    D_SCALE) FLOAT BIN,                                      00000820
                    (DENOM,                                                  00000830
                    BDPROB(N_BD_STS+1),                                      00000840
                    XBDPROB(N_BD_STS+1,NUM)) FLOAT DECIMAL(12),              00000850
                    (RUNNUM,                                                 00000860
                    ONE,                                                     00000870
                    TWO,                                                     00000880
                    KB,                                                      00000890
                    KC,                                                      00000900
                    I,                                                       00000910
                    J,                                                       00000920
                    JJ,                                                      00000930
                    K,                                                       00000940
                    B(N),                                                    00000950
                    SCALE1,                                                  00000960
                    SCALE2,                                                  00000970
                    IDUM,                                                    00000980
                    SCALE3,                                                  00000990
                    W(N),                                                    00001000
                    MAP(N),                                                  00001010
                    I_T_ORD(M,R),                                            00001020
                    NCHANGE,                                                 00001030
                    NTIE,                                                    00001040
                    ZERO,                                                    00001050
                    KK,                                                      00001060
                    RR,                                                      00001070
                    TIECAR(M),                                               00001080
                    ADDRESS,                                                 00001090
                    NS(N_BD_STS),                                            00001100
                    ITERATE,                                                 00001110
```

```
STMT LEVEL NEST

                    NSELECT,                                                                    00001120
                    NPREF,                                                                      00001130
                    NCAR,                                                                       00001140
                    MAXBOUND,                                                                   00001150
                    SIM(R+1),                                                                   00001160
                    ISIM(R+1),                                                                  00001170
                    FLAG(R),                                                                    00001180
                    FRST_PREF,                                                                  00001190
                    DIS_METHOD,                                                                 00001200
                    PAT_FLAG,                                                                   00001210
                    OVERRIDE,                                                                   00001220
                    PRNT_TR,                                                                    00001230
                    TFLAG,                                                                      00001240
                    NO_UNIT(M),                                                                 00001250
                    NO_DIST(M),                                                                 00001260
                    PRNT_AT_FLAG,                                                               00001270
                    A_MAP(R)) FIXED BIN,                                                        00001280
                    FACT(16) FIXED(13) DECIMAL INITIAL(1,1,2,6,24,120,720,5040, 40320,00001290
                     362880,3628800,39916800,479001600,6227020800, 87178291200,                 00001300
                1307674368000),                                                                 00001310
                    (ALPHA,                                                                     00001320
                    BETA,                                                                       00001330
                    MASK(M),                                                                    00001340
                    DUMBIT,                                                                     00001350
                    LASTVER) BIT(15) ALIGNED,                                                   00001360
                    COMM CHAR(10),                                                              00001370
                    CBIT BIT(1) ALIGNED,                                                        00001380
                    (I_RETURN,                                                                  00001390
                    K_RETURN,                                                                   00001400
                    ALT_RETURN) LABEL,                                                          00001410
                    TITLE CHAR(50),                                                             00001420
                    (R_DIST,                                                                    00001430
                    NM_UNIT(M),                                                                 00001440
                    NM_DIST(M),                                                                 00001450
                    ATOM) CHAR(8),                                                              00001460
                    (R_UNIT,                                                                    00001470
                    T_COST,                                                                     00001480
                    CFS ) CHAR(18);                                                             00001490
32      2           DCL                                                                         00001500
                    ((MWL,                                                                      00001510
                    MISC,                                                                       00001520
                    TAV,                                                                        00001530
                    VAR,                                                                        00001540
                    SWL,                                                                        00001550
                    UBWL,                                                                       00001560
                    DINPUT(M),                                                                  00001570
                    PINPUT(M)) FLOAT BIN,                                                       00001580
                    (PD(M,R),                                                                   00001590
                    PD2(M,R),                                                                   00001600
                    WORKLOAD(M),                                                                00001610
                    SATPROB) FLOAT DECIMAL(12),                                                 00001620
                    (TCAR(M),                                                                   00001630
                    TSEC(M),                                                                    00001640
                    TREP(R),                                                                    00001650
                    TQUE,                                                                       00001660
```

```
STMT LEVEL NEST

                    DCAR(M),                                                    00001670
                    DSEC(M),                                                    00001680
                    DREP(R),                                                    00001690
                    DWL(M),                                                     00001700
                    PWL(M),                                                     00001710
                    DISCO(M),                                                   00001720
                    PISCO(M),                                                   00001730
                    DISCI(M),                                                   00001740
                    PISDI(M),                                                   00001750
                    INPUT(M),                                                   00001760
                    DTSEC(M),                                                   00001770
                    PTSEC(M),                                                   00001780
                    GWKLD(M),                                                   00001790
                    ISOC(M),                                                    00001800
                    ISDI(M),                                                    00001810
                    PINSEC(M),                                                  00001820
                    MOX,                                                        00001830
                    MON,                                                        00001840
                    PATFREQ(P+1),                                               00001850
                    PINT(M)) FLOAT BIN) CONTROLLED;                             00001860
                /****  INITIALIZE CONSTANTS  *****/                           00001870
 33     2           ONE=1;                                                      00001880
 34     2           TWO=2;                                                      00001890
 35     2           INFINITY=1.0E+7;                                            00001900
 36     2           ZERO=0;                                                     00001910
 37     2           MAXBOUND=40;                                                00001920
 38     2           EPSILON=.001/N;                                             00001930
 39     2           MINVAL=.001/N;                                              00001940
 40     2           CAPACITY=0E0;                  /*DEFAULT ASSUMES ZERO-LINE  00001950
                                                   CAPACITY QUEUE*/            00001960
 41     2           XSPEED=10./60.;                                             00001970
 42     2           YSPEED=10./60.;                /*DEFAULT:TRAV SP =10MPH */  00001980
 43     2           D_SCALE=1./52.8;               /*DEFAULT: CONVERTS 100FT UNITS 00001990
                                                   INTO MILES*/                00002000
 44     2           SERVTM=30.E0;                                               00002010
 45     2           TFLAG=0;                                                    00002020
 46     2           STMI=1E0;                                                   00002030
 47     2           PAT_FLAG=0;                                                 00002040
 48     2           PAT_SPEED=7.5;                                              00002050
 49     2           PRNT_AT_FLAG=0;                                             00002060
 50     2           XSTPROB=0;                                                  00002070
 51     2           OVERRIDE=0;                                                 00002080
 52     2           PRNT_TP=0;                                                  00002090
 53     2           MU=INFINITY;                   /*THIS SERVES DUAL PURPOSE   00002100
                                                   LATER*/                     00002110
 54     2           FRST_PREF=0;                   /*DEFAULT: DISTRICT UNITS NOT 00002120
                                                   GIVEN 1ST PREF*/            00002130
                /*DISPATCH METHODS: 1:SCM, 2:MCM, 3:EX SCM, 4:EX MCM DEFAULT ASSUMES 00002140
                EX MCM*/                                                        00002150
 55     2           DIS_METHOD=4;                                               00002160
                /**   SET UP GLOSSARY....DEFAULT VERSION   **/                 00002170
 56     2           R_DIST='DISTRICT';                                          00002180
 57     2           R_UNIT='RESPONSE_UNIT';                                     00002190
 58     2           T_COST='TRAVEL TIME';                                       00002200
 59     2           CFS='CALLS FOR SERVICE';                                    00002210
```

```
STMT LEVEL NEST

 60     2              ATOM='ATOM';                                                    00002220
 61     2              DO I=1 TO M;                    /*SET UP UNIT AND RESPONSE AREA 00002230
                                                       DEFAULTS*/                     00002240
 62     2    1            NO_UNIT(I)=I;                                             00002250
 63     2    1            NO_DIST(I)=I;                                             00002260
 64     2    1            NM_UNIT(I)=' UNIT';                                       00002270
 65     2    1            NM_DIST(I)='DIST';                                        00002280
 66     2    1        END;                            /*LOOP ON I*/                 00002290
 67     2              DO I=1 TO R;                                                    00002300
 68     2    1            A_MAP(I)=I;                                               00002310
 69     2    1        END;                            /* LOOP ON I */               00002320
 70     2          START:                                                            00002330
                        K_RETURN=INI;                                                00002340
 71     2              IF DEBUG>0 THEN                                               00002350
 72     2                 PUT LIST('READATA CALLED') SKIP;                           00002360
 73     2              GO TO READATA;                                                00002370
 74     2          INI:                                                              00002380
                        K_RETURN=PROG_LOOP;                                          00002390
 75     2              RUNNUM=0;                                                     00002400
 76     2              IF DEBUG>0 THEN                                               00002410
 77     2                 PUT LIST('INITIALIZE CALLED') SKIP;                        00002420
 78     2              GO TO INITIALIZE;                                             00002430
 79     2          PROG_LOOP:                                                        00002440
                        RUNNUM=RUNNUM+1;                                             00002450
 80     2              IF RUNNUM>NUM THEN                                            00002460
 81     2                 GO TO CCOUT;                                               00002470
 82     2              K_RETURN=UP_RO;                                               00002480
 83     2              IF DEBUG>0 THEN                                               00002490
 84     2                 DO;                                                        00002500
 85     2    1               PUT LIST('SOLV_EQ CALLED') SKIP;                       00002510
 86     2    1               CALL PRNT_TIME;                                        00002520
 87     2    1            END;                                                      00002530
 88     2              GO TO SOLV_EQ;                                                00002540
 89     2          UP_RO:                                                            00002550
                        RO=RO+IRO;                                                   00002560
 90     2              GO TO PROG_LOOP;                                              00002570
 91     2          CCOUT:                                                            00002580
                        K_RETURN=START;                                              00002590
 92     2              IF DEBUG>0 THEN                                               00002600
 93     2                 DO;                                                        00002610
 94     2    1               PUT LIST('CALCOUT CALLED') SKIP;                       00002620
 95     2    1               CALL PRNT_TIME;                                        00002630
 96     2    1            END;                                                      00002640
 97     2              GO TO CALCOUT;                                                00002650
 98     2          RET:                                                              00002660
                        RETURN;                                                      00002670
 99     2          READATA:                                                          00002680
                        GET LIST(COMM);                                              00002690
100     2              IF COMM='GLOSSARY' THEN                                       00002700
101     2                 GET DATA;                                                  00002710
102     2              PUT                                                           00002720
                       LIST('NSF/RANN-HUD SPATIALLY DISTRIBUTED QUEUING MODEL OF AN') 00002730
                       PAGE;                                                         00002740
103     2              PUT LIST('URBAN EMERGENCY SERVICE SYSTEM') SKIP;              00002750
104     2              PUT LIST('RESPONSE UNIT= ',R_UNIT,'   TOTAL NUMBER= ',M) SKIP; 00002760
```

```
STMT LEVEL NEST

105     2             PUT LIST('RESPONSE AREA= ',R_DIST,'   TOTAL NUMBER= ',M) SKIP;     00002770
106     2             PUT LIST('GEOGRAPHICAL ATOM= ',ATOM,'   TOTAL NUMBER= ',R) SKIP;   00002780
107     2             GO TO LAAP2;                                                       00002790
108     2         LAAP:                                                                  00002800
                      GET LIST(COMM);                    /*THIS IS BASIC INSTRUCTION   00002810
                                                         USED TO READ IN COMMS*/        00002820
109     2         LAAP2: /*TWO OPTIONS FOR STARTING EXECUTION AFTER READING IN OR       00002830
                  CHANGING DATA RUN ASSUMES A CHANGED GEOGRAPHY (& CALLS TRAVTM) RERUN  00002840
                  DOES NOT*/                                                           00002850
                      IF COMM='RUN' THEN                                               00002860
110     2                 DO;                                                          00002870
111     2     1               GET LIST(XDUM,YDUM);         /*XDUM=#CALLS/HOUR,       00002880
                                                           YDUM=INCREMENT*/            00002890
112     2     1               RO=XDUM*SERVTM/60.;          /*RO=#CALLS/SERVICE TIME UNIT*/ 00002900
113     2     1               IRO=YDUM*SERVTM/60.;                                     00002910
114     2     1               CALL TRAVTM;                                             00002920
115     2     1               PUT EDIT(R_UNIT,' SPATIAL ALLOCATION, WHILE AVAILABLE') (  00002930
                               COLUMN(5),A,A) PAGE;                                    00002940
116     2     1                DO J=1 TO M BY 10;                                      00002950
117     2     2                    IF J+9<M THEN KK=J+9;                               00002960
119     2     2                    ELSE KK=M;                                          00002970
120     2     2               PUT EDIT(ATOM,'ID OF ',R_UNIT)(A,COLUMN(16),A,A) SKIP(2);  00002980
121     2     2               PUT EDIT('NO.',(NM_UNIT(I) DO I=J TO KK)) (                00002990
                               A,COLUMN(10),(M)(A(8),X(1))) SKIP;                      00003000
122     2     2               PUT EDIT((NO_UNIT(I) DO I=J TO KK))(                       00003010
                               COLUMN(11),(M)(F(3),X(6))) SKIP;                        00003020
123     2     2               DO K=1 TO R;                                             00003030
124     2     3                  PUT EDIT(A_MAP(K),(SEC(I,K) DO I=J TO KK)) (          00003040
                                  COLUMN(2),F(3),COLUMN(9),(M)(F(6,3),X(3)));          00003050
125     2     3               END;                                                     00003060
126     2     2              END;  /*LOOP ON J*/                                       00003070
127     2     1               GO TO K_RETURN;                                          00003080
128     2     1             END;                                                       00003090
129     2             IF COMM='RERUN' THEN                                             00003100
130     2             DO;                                                              00003110
131     2     1          GET LIST(XDUM,YDUM);                                        00003120
132     2     1          RO=XDUM*SERVTM/60.;                                         00003130
133     2     1          IRO=YDUM*SERVTM/60.;                                        00003140
134     2     1          PUT EDIT('GEOGRAPHY UNCHANGED FROM PREVIOUS RUN') (         00003150
                          COLUMN(5),A) PAGE;                                        00003160
135     2     1          GO TO K_RETURN;                                             00003170
136     2     1        END;                                                          00003180
                  /*FINISHES PROCESSING ON ONE GEOGRAPHICAL CONFIGURATION*/          00003190
137     2             IF COMM='JOB' THEN                                             00003200
138     2             GO TO RET;                                                     00003210
                  /*TITLE OF RUN: 50 CHARACTERS OR LESS*/                          00003220
139     2             IF COMM='TITLE' THEN                                           00003230
140     2             GET LIST(TITLE);                                               00003240
                  /*L(J)=FRACTION OF CALL FOR SERVICE WORKLOAD GENERATED IN ATOM J READ 00003250
                  IN NUMBERS THAT ARE PROPORTIONAL TO L'S AND THEY ARE NORMALIZE*/  00003260
141     2             IF COMM='LAM' THEN                                             00003270
142     2             DO;                                                            00003280
143     2     1          GET LIST(L);                                                00003290
144     2     1          XDUM=SUM(L);                                                00003300
145     2     1          L=L/XDUM;                                                   00003310
```

```
STMT LEVEL NEST

146     2    1              PUT EDIT(CFS,' DISTRIBUTION, BY ',ATOM) (COLUMN(5),A,A,A)   00003320
                            SKIP(3);                                                    00003330
147     2    1              DO K=1 TO R;                                                00003340
148     2    2                 PUT EDIT(A_MAP(K),L(K)) (                                00003350
                                COLUMN(7),F(3),COLUMN(18),F(7,5)) SKIP;                 00003360
149     2    2              END;                                                        00003370
150     2    1           END;                              /*END OF 'LAM' INSTRUCTIONS*/ 00003380
                     /*SEC(I,J)¬=0E0 IF ATOM J IS IN SECTOR I;0E0 OTHERWISE 2 OPTIONS: 1) 00003390
                     CONSTRUCT WITH 'S', ASSUMING LOCATION DEFAULT 2) READ IN DIRECTLY   00003400
                     USING 'SS', WHILE DEFINING DISTRS*/                                00003410
151     2         IF COMM='VAR_SER_TM' THEN                                             00003420
152     2            DO;                                                                00003430
153     2    1          GET LIST(MU);                 /*1ST READ IN AS SERVICE TIMES*/ 00003440
154     2    1          PUT EDIT('SERVICE TIME FOR EACH ',R_UNIT)(A,A) SKIP(4);        00003450
155     2    1          DO I=1 TO M;                                                    00003460
156     2    2             PUT EDIT(NM_UNIT(I),NO_UNIT(I),MU(I),' MINUTES') (           00003470
                             A(8),X(3),F(3),X(5),F(6,2),A) SKIP;                        00003480
157     2    2          END;                                                           00003490
158     2    1          MU=1/MU;                      /*CONVERT TO RATES*/             00003500
159     2    1          SERVTM=M/SUM(MU);             /*NEW SERVICE TIME UNIT*/        00003510
160     2    1          MU=MU*SERVTM;                 /*RELATIVE SERVICE RATES PER     00003520
                                                      SERVICE TIME UNIT*/               00003530
161     2    1       END;                             /*END VARIABLE SERV TIME INPUT*/ 00003540
162     2         IF COMM='SERVTM' THEN                                                 00003550
163     2            GET LIST(SERVTM);                                                  00003560
                     /*DETERMINE ATOMS IN UNIT I'S DISTRICT (OPTION 1 ABOVE) CARD: UNIT # 00003570
                     TOT # ATOMS THEIR #'S THIS IS NOT USED IF SEC ARRAY IS READ IN      00003580
                     DIRECTLY  */                                                       00003590
164     2         IF COMM='S' THEN                                                      00003600
165     2            DO;                                                                00003610
166     2    1          GET LIST (I,H);                                                00003620
167     2    1          SEC(I,*)=0E0;                                                  00003630
168     2    1          DO K=1 TO H;                                                   00003640
169     2    2             GET LIST(J);                                                00003650
170     2    2             SEC(I,J)=1E0;                                               00003660
171     2    2          END;                             /* LOOP ON K   */             00003670
172     2    1          CALL CALCSEC(I);                                               00003680
173     2    1       END;                                                              00003690
                     /*OPTION 2: DEFINE DISTRICTS AND RELATIVE ALLOCATIONS CARD: UNIT #   00003700
                     TOT # ATOMS ATOM # 1 REL FRAC TIME IN # 1 ATOM # 2 REL FRAC TIME IN # 00003710
                     2, ...ETC. EXAMPLE: 2 2 4 1E0 7 3.2E0 */                           00003720
174     2         IF COMM='SS' THEN                                                     00003730
175     2            DO;                                                                00003740
176     2    1          GET LIST(I,H);                                                 00003750
177     2    1          SEC(I,*)=0E0;                                                  00003760
178     2    1          SOM=0E0;                                                       00003770
                     /*SEC(I,J)<0E0 IMPLIES ATOM J "IN" DIST I BUT UNIT I SPENDS NO TM TH*/ 00003780
179     2    1          DO K=1 TO H;                                                   00003790
180     2    2             GET LIST(J,XDUM);                                           00003800
181     2    2             SEC(I,J)=XDUM;                                              00003810
182     2    2             IF XDUM>0E0 THEN                                            00003820
183     2    2                SOM=SOM+XDUM;                                            00003830
184     2    2             ELSE                                                        00003840
184     2    2                SEC(I,J)=-1.E0;                                          00003850
185     2    2          END;                             /*LOOP ON K*/                00003860
```

```
STMT LEVEL NEST

186      2    1              SEC(I,*)=SEC(I,*)/SOM;                                          00003870
187      2    1           END;                                                               00003880
                    /*TR(I,J)=MEAN TRAVEL TIME FROM ATOM I TO ATOM J*/                      00003890
188      2           IF COMM='TR' THEN                                                      00003900
189      2              DO;                                                                 00003910
190      2    1            GET LIST(TR);                                                    00003920
191      2    1            TFLAG=1;                                                         00003930
192      2    1           END;                                                               00003940
                    /*EFFECTIVE SPEEDS OF RESPONSE*/                                       00003950
193      2           IF COMM='SPEED' THEN                                                   00003960
194      2              DO;                                                                 00003970
195      2    1            GET LIST(XSPEED);                                                00003980
196      2    1            XSPEED=XSPEED/60.;                                               00003990
197      2    1            YSPEED=XSPEED;                                                   00004000
198      2    1           END;                              /*BOTH SPEEDS ASSUMED EQUAL IF  00004010
                                                           ONLY ONE READ IN*/              00004020
199      2           IF COMM='XSPEED' THEN                                                  00004030
200      2              DO;                                                                 00004040
201      2    1            GET LIST(XSPEED);                                                00004050
202      2    1            XSPEED=XSPEED/60.;                                               00004060
203      2    1           END;                                                              00004070
204      2           IF COMM='YSPEED' THEN                                                  00004080
205      2              DO;                                                                 00004090
206      2    1            GET LIST(YSPEED);                                                00004100
207      2    1            YSPEED=YSPEED/60.;                                               00004110
208      2    1           END;                                                              00004120
                    /*DEFAULT OPTION FOR TRAVEL TIMES:ASSUMES RIGHT-ANGLE DISTANCE NEED    00004130
                    TO READ IN SPEEDS FIRST*/                                             00004140
209      2           IF COMM='TX' THEN                                                      00004150
210      2              DO;                                                                 00004160
211      2    1            DO I=1 TO R;                                                     00004170
212      2    2               GET LIST(X(I),Y(I));                                          00004180
213      2    2            END;                                                             00004190
214      2    1            X=X*D_SCALE;                                                     00004200
215      2    1            Y=Y*D_SCALE;                                                     00004210
216      2    1            DO I=1 TO R;                                                     00004220
217      2    2               DO J=1 TO R;                                                  00004230
218      2    3                  TR(I,J)=(ABS(X(I)-X(J)))/XSPEED + (ABS(Y(I)-Y(J)))/     00004240
                                 YSPEED;                                                  00004250
219      2    3               END;                                                          00004260
220      2    2            END;                              /*LOOP ON J, THEN I*/        00004270
221      2    1           END;                                                              00004280
                    /*AN OPTIONAL PART OF THE TRAVEL TIME DEFAULT CORRECTS INTRA-ATOM      00004290
                    TRAVEL TIMES MUST BE READ IN AFTER 'TX'*/                             00004300
222      2           IF COMM='CORTM' THEN                                                   00004310
223      2              DO;                                                                 00004320
224      2    1            GET LIST(YDUM);                                                  00004330
225      2    1            DO I=1 TO R;                                                     00004340
226      2    2               GET LIST(XDUM);                                               00004350
227      2    2               TR(I,I)=YDUM*SQRT(XDUM/(XSPEED*YSPEED));                      00004360
228      2    2            END;                                                             00004370
229      2    1           END;                              /*LOOP ON I, THEN FINISH       00004380
                                                           CONDITIONAL*/                   00004390
230      2           IF COMM='TX_OV' THEN                                                   00004400
231      2              DO;                               /*SELECTIVE OVERRIDE OF RIGHT   00004410
```

```
STMT LEVEL NEST

                                                        ANGLE DISTS*/                                    00004420
232      2    1                 GET DATA;                                                                00004430
233      2    1                 TFLAG=2;                                                                 00004440
234      2    1              END;                                                                        00004450
235      2              IF COMM='D_SCALE' THEN                                                          00004460
236      2                 GET LIST(D_SCALE);                                                           00004470
                      /*A 'CAP' CARD SIGNALS INFINITE CAPACITY QUEUE;ELSE ZERO LINE CAPAC.*/           00004480
237      2              IF COMM='CAP' THEN                                                              00004490
238      2                 CAPACITY=INFINITY;                                                           00004500
                      /*FOUR POSSIBLE DISPATCH OPTIONS*/                                              00004510
239      2              IF COMM='SCM' THEN                                                              00004520
240      2                 DIS_METHOD=1;                                                                00004530
241      2              IF COMM='MCM' THEN                                                              00004540
242      2                 DIS_METHOD=2;                                                                00004550
243      2              IF COMM='ESCM' THEN                                                             00004560
244      2                 DIS_METHOD=3;                                                                00004570
245      2              IF COMM='EMCM' THEN                                                             00004580
246      2                 DIS_METHOD=4;                                                                00004590
                      /*A  'FRST' CARD SIGNALS 1ST PREF FOR DISTRICT UNIT*/                           00004600
247      2              IF COMM= 'FRST' THEN                                                            00004610
248      2                 FRST_PREF=1;                                                                 00004620
249      2              IF COMM='PATROL' THEN                                                           00004630
250      2                 DO;                                                                          00004640
251      2    1               GET LIST(PAT_SPEED);                                                     00004650
252      2    1               GET LIST(STMI);                                                          00004660
253      2    1               PUT EDIT('PATROL STREET MILES PER ',ATOM)(A,A) PAGE;                     00004670
254      2    1               DO J=1 TO R;                                                             00004680
255      2    2                  PUT EDIT(A_MAP(J),STMI(J))(COLUMN(5),F(3),X(5),F(7,2))                00004690
                                 SKIP;                                                                00004700
256      2    2               END;                                                                     00004710
257      2    1               PAT_FLAG=1;                                                              00004720
258      2    1            END;                                                                        00004730
259      2              IF COMM='DISP_OV_RD' THEN                                                       00004740
260      2                 OVERRIDE=1;                                                                  00004750
261      2              IF COMM='PRNT_TR' THEN                                                          00004760
262      2                 PRNT_TR=1;                                                                   00004770
263      2              IF COMM='NO_PRNT_AT' THEN                                                       00004780
264      2                 PRNT_AT_FLAG=1;                                                              00004790
265      2              IF COMM='ATOM_NO' THEN                                                          00004800
266      2                 GET LIST (A_MAP);                                                            00004810
267      2              GO TO LAAP;                                                                     00004820
                      /*CALCSEC IS A CONVENIENT DEFAULT FOR DETERMINING SPATIAL POSITION OF           00004830
                      RESPONSE UNIT WHILE IDLE; HOWEVER IT IS NOT GENERAL AND ONE MAY WISH           00004840
                      TO READ IN THE SPATIAL DIST (SEC) DIRECTLY SEC(I,J) IS MANIPULATED TO          00004850
                      EQUAL FRAC OF IDLE TIME UNIT I SPENDS IN ATOM J*/                              00004860
268      2           CALCSEC:                                                                         00004870
                       PROC(I);                                                                        00004880
269      3             DCL                                                                            00004890
                          (I,                                                                         00004900
                          J) FIXED BIN;                                                               00004910
270      3             SOM=0E0;                                                                       00004920
271      3             DO J=1 TO R;                                                                   00004930
272      3    1           IF SEC(I,J)> 0E0 THEN                                                      00004940
273      3    1              SOM=SOM+L(J);                                                           00004950
274      3    1          END;                                  /* LOOP ON J  */                      00004960
```

```
STMT LEVEL NEST

275     3              DO J=1 TO R;                                                    00004970
276     3     1           IF SEC(I,J)> 0E0 THEN                                      00004980
277     3     1              SEC(I,J)=L(J)/SOM;                                      00004990
278     3     1           END;                              /* LOOP ON J  */         00005000
279     3            END CALCSEC;                                                    00005010
                   /*TRAVTM COMPUTES ESTIMATED TRAVEL TIMES FROM DISTRICTS TO ATOMS 00005020
                   COMPUTATION DEPENDS ON TYPE OF DISPATCH METHOD(DIS_METHOD) ALSO  00005030
                   COMPUTES T(I,J)=EXACT EXP TRAV TIME FROM DISTRICT I TO ATOM J*/  00005040
280     2          TRAVTM:                                                         00005050
                     PROC;                                                         00005060
281     3              C=-1.E0;                                                    00005070
282     3              DO I=1 TO M;                                                00005080
283     3     1           DO J=1 TO R;                                           00005090
284     3     2              SOM=0E0;                                            00005100
285     3     2              DO K=1 TO R;                                        00005110
286     3     3                 XDUM=SEC(I,K);                                   00005120
287     3     3                 IF XDUM>0E0 THEN                                 00005130
288     3     3                    SOM=SOM+XDUM*TR(K,J);                         00005140
289     3     3              END;                              /* LOOP ON K   */  00005150
290     3     2              T(I,J)=SOM;                                         00005160
291     3     2           END;                                                   00005170
292     3     1        END;                                 /*LOOP ON J, THEN I*/  00005180
                   /****DEVICE FOR STATE '111...11' IN 'TOUR'*******/             00005190
293     3              T(M+1,*)=INFINITY;                                        00005200
294     3              IF DIS_METHOD=4 THEN                                      00005210
295     3                 C=T;                           /*FINISHED IF EX MCM .  00005220
                                                        DISPATCHING*/            00005230
296     3              C(M+1,*)=INFINITY;                                        00005240
297     3              IF DIS_METHOD<3 THEN                                      00005250
298     3                 DO;                            /*FIND CENTER-OF-MASS POSITIONS  00005260
                                                        OF UNITS*/               00005270
299     3     1              DO I=1 TO M;                                        00005280
300     3     2                 XDUM=0E0;                                        00005290
301     3     2                 YDUM=0E0;                                        00005300
302     3     2                 DO J=1 TO R;                                     00005310
303     3     3                    IF SEC(I,J)>0E0 THEN                          00005320
304     3     3                       DO;                                        00005330
305     3     4                        XDUM=XDUM+X(J)*SEC(I,J);                  00005340
306     3     4                        YDUM=YDUM+Y(J)*SEC(I,J);                  00005350
307     3     4                       END;                                       00005360
308     3     3                 END;                           /*LOOP ON J*/     00005370
309     3     2                 XUCM(I)=XDUM;                                    00005380
310     3     2                 YUCM(I)=YDUM;                                    00005390
311     3     2                 IF TFLAG>0 THEN                                  00005400
312     3     2                    DO;                         /*CORRECTION:PLACE C.M. ON CENT  00005410
                                                                OF */            00005420
313     3     3                       XDUM=INFINITY;           /****   NEAREST ATOM   ****/  00005430
314     3     3                       DO J=1 TO R;                              00005440
315     3     4                          YDUM=(ABS(XUCM(I)-X(J) ))/XSPEED + (   00005450
                                          ABS(YUCM(I)-Y(J) ))/YSPEED;            00005460
316     3     4                          IF YDUM<XDUM THEN                      00005470
317     3     4                             DO;                                 00005480
318     3     5                                XDUM=YDUM;                       00005490
319     3     5                                K=J;                             00005500
320     3     5                             END;                                00005510
```

```
STMT LEVEL NEST

321     3    4                    END;                /*LOOP ON J*/                    00005520
322     3    3                    XUCM(I)=K;          /***IMPORTANT: HERE XUCM IS     00005530
                                                      USED TO INDEX ATOM # CLOSEST TO 00005540
                                                      STATIS C.M.*/                  00005550
323     3    3                 END;                   /*END DO THAT CORRECTS FOR C.M.  00005560
                                                      LOCATIONS*/                    00005570
324     3    2              END;                                                      00005580
325     3    1           END;                        /*LOOP ON I AND END OF           00005590
                                                      CONDITIONAL*/                  00005600
326     3              IF CIS_METHOD=1 THEN                                          00005610
327     3              DO;                                                           00005620
                    /*COMPUTE CENTER-OF-MASS POSITIONS OF INCIDENTS POTENTIAL PROBLEMS 00005630
                    HERE FOR OVERLAPPING DISTRICTS*/                               00005640
328     3    1              DO I=1 TO M;                                           00005650
329     3    2                 XDUM=0E0;                                           00005660
330     3    2                 YDUM=0E0;                                           00005670
331     3    2                 SOM=0E0;                                            00005680
332     3    2                 DO J=1 TO R;                                        00005690
333     3    3                    IF SEC(I,J)¬=0E0 THEN                            00005700
334     3    3                       DO;              /*"¬=" CORRECT HERE*/         00005710
335     3    4                         XDUM=XDUM+X(J)*L(J);                        00005720
336     3    4                         YDUM=YDUM+Y(J)*L(J);                        00005730
337     3    4                         SOM=SOM+L(J);                               00005740
338     3    4                       END;                                          00005750
339     3    3                 END;                  /*END CONDITIONAL AND LOOP ON J*/00005760
340     3    2                 IF SOM=0E0 THEN                                     00005770
341     3    2                    GO TO NO_SEC;                                    00005780
342     3    2                 XICM(I)=XDUM/SOM;                                   00005790
343     3    2                 YICM(I)=YDUM/SOM;                                   00005800
344     3    2                 IF TFLAG>0 THEN                                     00005810
345     3    2                    DO;                                              00005820
346     3    3                       XDUM=INFINITY;                                00005830
347     3    3                       DO J=1 TO R;                                  00005840
348     3    4                          YDUM=(ABS(XICM(I)-X(J)))/XSPEED + (        00005850
                                         ABS(YICM(I)-Y(J)))/YSPEED;                00005860
349     3    4                          IF YDUM<XDUM THEN                          00005870
350     3    4                             DO;                                     00005880
351     3    5                                XDUM=YDUM;                           00005890
352     3    5                                K=J;                                 00005900
353     3    5                             END;                                    00005910
354     3    4                       END;            /*LOOP ON J*/                 00005920
355     3    3                       XICM(I)=K;      /*SAME SITUATION AS XUCM ABOVE*/ 00005930
356     3    3                    END;               /*END TFLAG COND*/            00005940
357     3    2    NO_SEC:                                                          00005950
                 END;                               /*LOOP ON I*/                 00005960
358     3    1           DO I=1 TO M;                                             00005970
359     3    2              DO J=1 TO R;                                          00005980
360     3    3                 DO K=1 TO M;         /*NOTE IMPLICIT PREF FOR LOWER 00005990
                                                      #'ED UNS*/                     00006000
361     3    4                    IF SEC(K,J)¬=0E0 THEN                            00006010
362     3    4                       DO;                                           00006020
363     3    5                          IF C(I,J)<0E0 THEN                         00006030
364     3    5                             DO;                                     00006040
365     3    6                                IF TFLAG=0 THEN                      00006050
366     3    6                                   C(I,J)=ABS(XUCM(I)-XICM(K))/XSPEED +00006060
```

```
STMT LEVEL NEST

                                                    ABS(YUCM(I)-YICM(K))/YSPEED;          00006070
367     3     6                                   ELSE                                    00006080
367     3     6                                      DO;                                  00006090
368     3     7                                         JJ=XUCM(I);                       00006100
369     3     7                                         KK=XICM(K);                       00006110
370     3     7                                         C(I,J)=TR(JJ,KK);                 00006120
371     3     7                                      END;                                 00006130
372     3     6                                   K=M+1;                                  00006140
373     3     6                                END;                                       00006150
374     3     5                          END;           /*TERMINATE 2 CONDITIONALS*/      00006160
375     3     4                    END;                                                  00006170
376     3     3                 END;                                                     00006180
377     3     2              END;                      /*LOOP ON K,J,I*/                 00006190
378     3     1           END;                         /*FINISHED WITH DIS_METHOD=1      00006200
                                                       CONDITIONAL*/                    00006210
379     3              IF DIS_METHOD=2 THEN                                             00006220
380     3              DO;                                                              00006230
381     3     1           DO I=1 TO M;                                                  00006240
382     3     2              DO J=1 TO R;                                               00006250
383     3     3                 IF TFLAG=0 THEN                                         00006260
384     3     3                    C(I,J)=ABS(XUCM(I)-X(J))/XSPEED +                    00006270
                                   ABS(YUCM(I)-Y(J))/YSPEED;                           00006280
385     3     3                 ELSE                                                    00006290
386     3     3                    DO;                                                  00006300
386     3     4                       JJ=XUCM(I);                                       00006310
387     3     4                       C(I,J)=TR(JJ,J);                                  00006320
388     3     4                    END;                                                 00006330
389     3     3                 END;                                                    00006340
390     3     2              END;                                                       00006350
391     3     1           END;                         /*LOOP ON J AND I, THEN END       00006360
                                                       COND. FOR MCM*/                  00006370
392     3              IF DIS_METHOD=3 THEN                                             00006380
393     3              DO;                                                              00006390
                    /*I=UNIT #;  K=# OF DISTRICT OF INCIDENT*/                         00006400
394     3     1           DO I=1 TO M;                                                  00006410
395     3     2              DO K=1 TO M;                                               00006420
396     3     3                 XDUM=0E0;                                               00006430
397     3     3                 SOM=0E0;                                                00006440
398     3     3                 DO J=1 TO R;                                            00006450
399     3     4                    IF SEC(I,J)>0E0 THEN                                 00006460
400     3     4                       DO JJ=1 TO R;                                     00006470
                    /*SCREENING THRU ATOMS TO TEST FOR UNIT I*/                        00006480
401     3     5                        IF SEC(K,JJ)¬=0E0 THEN                          00006490
402     3     5                           DO;       /*DIST K TEST*/                   00006500
403     3     6                             XDUM=XDUM+TR(J,JJ)*SEC(I,J)*L(JJ);        00006510
404     3     6                             SOM=SOM+L(JJ)*SEC(I,J);                   00006520
405     3     6                           END;                                        00006530
406     3     5                     END;                                              00006540
407     3     4                 END;                  /*LOOP ON JJ,END COND., LOOP ON  00006550
                                                       J*/                              00006560
408     3     3                 XDUM=XDUM/SOM;                                          00006570
409     3     3                 DO JJ=1 TO R;                                           00006580
410     3     4                    IF SEC(K,JJ)¬=0E0 THEN                               00006590
411     3     4                       DO;                                               00006600
412     3     5                          IF C(I,JJ)<0E0 THEN                            00006610
```

```
STMT LEVEL NEST

413     3    5                              C(I,JJ)=XDUM;                                    00006620
414     3    5                          END;                                                 00006630
415     3    4                      END;                    /*END COND. AND LOOP ON JJ*/     00006640
416     3    3                  END;                                                         00006650
417     3    2              END;                                                             00006660
418     3    1          END;                                /*LOOP ON K AND I AND END OF     00006670
                                                            COND.*/                         00006680
419     3            IF FRST_PREF=1 THEN                                                    00006690
420     3               DO;                                                                 00006700
                   /*DISTRICT UNIT GETS 1ST PREF? IF SO, PREF GIVEN TO LOWER # UNITS*/     00006710
421     3    1          DO J=1 TO R;                                                        00006720
422     3    2             DO I=1 TO M;                                                     00006730
423     3    3                IF SEC(I,J)¬=OEO THEN                                         00006740
424     3    3                   DO;                                                       00006750
425     3    4                      C(I,J)=OEO;                                            00006760
426     3    4                      I=M+1;                                                 00006770
427     3    4                   END;                                                      00006780
428     3    3             END;                                                           00006790
429     3    2          END;                                                              00006800
430     3    1      END;                                /*LOOP ON I,J AND END OF COND.*/ 00006810
431     3            IF OVERRIDE=1 THEN                                                     00006820
432     3               DO;                              /*ASSUMES THESE COMMANDS FOLLOW*/00006830
433     3    1  OV:                                                                        00006840
                      GET LIST (COMM);                                                    00006850
434     3    1          IF COMM='FRONT' THEN                                              00006860
435     3    1             DO;                                                            00006870
436     3    2                GET LIST(I,H);             /*I=UNIT #, H=# ATOMS INVOLVED*/ 00006880
437     3    2                DO K=1 TO H;                                               00006890
438     3    3                   GET LIST(J);                                            00006900
439     3    3                   C(I,J)=OEO;             /*NOTE TIE IN CASE OF USE OF     00006910
                                                         FRST_PREF=1*/                   00006920
440     3    3                END;                       /*LOOP ON K*/                   00006930
441     3    2                GO TO OV;                                                  00006940
442     3    2             END;                          /*END DO*/                      00006950
443     3    1          IF COMM='BACK' THEN                                               00006960
444     3    1             DO;                                                            00006970
445     3    2                GET LIST(I,H);                                             00006980
446     3    2                DO K=1 TO H;                                               00006990
447     3    3                   GET LIST(J);                                            00007000
448     3    3                   C(I,J)=999.;            /*NOTE POSSIBILITY OF TIES HERE  00007010
                                                         ALSO*/                          00007020
449     3    2                END;                       /*LOOP ON K*/                   00007030
450     3    2                GO TO OV;                                                  00007040
451     3    2             END;                          /*END DO*/                      00007050
452     3    1          IF COMM='MIDDLE' THEN                                             00007060
453     3    1             DO;                                                            00007070
454     3    2                GET LIST(I,XDUM,H);                                        00007080
455     3    2                DO K=1 TO H;                                               00007090
456     3    3                   GET LIST(J);                                            00007100
457     3    3                   C(I,J)=XDUM;                                            00007110
458     3    3                END;                       /*LOOP ON K*/                   00007120
459     3    2                GO TO OV;                                                  00007130
460     3    2             END;                          /*END DO*/                      00007140
                   /*'END_OV_RD' CARD IMPLICITLY ASSUMED HERE*/                         00007150
461     3    1          END;                            /*END BIG DO FOR OVERRIDE*/     00007160
```

```
STMT LEVEL NEST

462    3          IF PRNT_TR=1 THEN                                                    00007170
463    3             DO;                                                               00007180
464    3    1           PUT EDIT('TRAVEL TIME MATRIX:  INTER-',ATOM) (                 00007190
                        COLUMN(10),A,A) PAGE;                                        00007200
465    3    1           DO K=1 TO R BY 10;                                            00007210
466    3    2              IF K+9<R THEN                                              00007220
467    3    2                 KK=K+9;                                                 00007230
468    3    2              ELSE                                                       00007240
469    3    2                 KK=R;                                                   00007250
                           PUT                                                       00007260
                           EDIT(ATOM,' NUMBER: ORIGIN',ATOM,' NUMBER:DESTINATION' ) 00007270
                           (                                                         00007280
                            A,A,X(6),A,A) SKIP(2);                                   00007290
470    3    2              PUT EDIT((A_MAP(I) DO I=K TO KK))(                        00007300
                            COLUMN(31),(10)(F(3), X(5))) SKIP;                       00007310
471    3    2              DO J=1 TO R;                                              00007320
472    3    3                 PUT EDIT(A_MAP(J),(TR(J,I) DO I=K TO KK)) (            00007330
                               COLUMN(16),F(3),COLUMN(29),(10)(F(6,2),X(2)));        00007340
473    3    3              END;                          /*LOOP ON J*/               00007350
474    3    2           END;                             /*LOOP ON K*/               00007360
475    3    1        END;                                /*END PRNT_TR*/             00007370
476    3             PUT EDIT('MEAN TRAVEL TIMES FOR EACH ',R_UNIT)(COLUMN(10),A,A)   00007380
                      PAGE;                                                          00007390
477    3             PUT EDIT('TO EACH ',ATOM)(COLUMN(10),A,A) SKIP;                  00007400
478    3             PUT EDIT(ATOM,'ID OF ',R_UNIT)(A,X(6),A,A) SKIP(2);              00007410
479    3             PUT EDIT('ID')(A) SKIP;                                          00007420
480    3             PUT EDIT('NO',(NM_UNIT(I) DO I=1 TO M))(                         00007430
                      A,COLUMN(6),(M)(A(8),X(1))) SKIP;                              00007440
481    3             PUT EDIT((NO_UNIT(I) DO I=1 TO M))(COLUMN(7),(M)(F(3),X(6)));    00007450
482    3             DO K=1 TO R;                                                     00007460
483    3    1           PUT EDIT(A_MAP(K),(T(I,K) DO I=1 TO M))(                      00007470
                        F(3),X(1),(M)(F(6,2), X(3))) SKIP;                           00007480
484    3    1        END;                                /*LOOP ON K*/               00007490
485    3             IF DIS_METHOD=1 THEN                                             00007500
486    3                PUT LIST('STRICT CENTER-OF-MASS DISPATCHING') PAGE;           00007510
487    3             IF DIS_METHOD=2 THEN                                             00007520
488    3                PUT LIST('MODIFIED CENTER-OF-MASS DISPATCHING') PAGE;         00007530
489    3             IF DIS_METHOD=3 THEN                                             00007540
490    3                PUT LIST('EXPECTED SCM DISPATCHING') PAGE;                    00007550
491    3             IF DIS_METHOD=4 THEN                                             00007560
492    3                PUT LIST('EXPECTED MCM DISPATCHING') PAGE;                    00007570
493    3             IF FRST_PREF=1 THEN                                              00007580
494    3                DO;                                                           00007590
495    3    1              PUT EDIT('FIRST PREFERENCE ASSIGNED TO ',R_UNIT)(A,A)      00007600
                           SKIP(2);                                                  00007610
496    3    1              PUT EDIT('ASSOCIATED WITH EACH ',R_DIST)(A,A);             00007620
497    3    1           END;                                                         00007630
498    3             IF TFLAG=1 THEN                                                  00007640
499    3                PUT EDIT('ARBITRARY (EMPIRICAL) TRAVEL TIMES USED') (A)      00007650
                        SKIP(2);                                                     00007660
500    3             IF TFLAG=2 THEN                                                  00007670
501    3                PUT EDIT('SELECTIVE OVERRIDE OF RIGHT-ANGLE DISTANCE USED')( 00007680
                         A) SKIP(2);                                                 00007690
502    3             IF TFLAG>0 & DIS_METHOD<3 THEN                                   00007700
503    3                PUT                                                          00007710
```

```
STMT LEVEL NEST

                     EDIT('CENTER-OF-MASS POSITIONS OF UNITS AND/OR INCIDENTS',      00007720
                      ' MOVED TO NEAREST CENTER OF ',ATOM)(A) SKIP;                 00007730
504     3          PUT EDIT('ESTIMATED "COST" OF DISPATCHING I_TH ',R_UNIT)(A,A)     00007740
                   SKIP(2);                                                         00007750
505     3          PUT EDIT('TO J_TH ',ATOM)(COLUMN(6),A,A);                        00007760
506     3          PUT EDIT(ATOM,'ID OF ',R_UNIT)(A,X(6),A,A) SKIP(2);              00007770
507     3          PUT EDIT('ID')(A) SKIP;                                          00007780
508     3          PUT EDIT('NO',(NM_UNIT(I) DO I=1 TO M))(                         00007790
                    A,COLUMN(6),(M)(A(8),X(1))) SKIP;                             00007800
509     3          PUT EDIT((NO_UNIT(I) DO I=1 TO M))(COLUMN(7),(M)(F(3),X(6)));    00007810
510     3          DO K=1 TO R;                                                     00007820
511     3    1        PUT EDIT(A_MAP(K),(C(I,K) DO I=1 TO M))(                     00007830
                       F(3),X(1),(M)(F(6,2), X(3))) SKIP;                         00007840
512     3    1     END;                                                            00007850
513     3       END TRAVTM;                                                       00007860
              /*COMPUTE MATRIX OF ORDERED DISPATCH PREFERENCES C(I,J)="COST" OF   00007870
              ASSIGNING UNIT I TO ATOM J...TIES ALLOWED I_T_ORD(I,J)=NUMBER OF ITH 00007880
              BEST UNIT TO ASSIGN TO ATOM J*/                                     00007890
514     2       INITIALIZE:                                                       00007900
                   IF DEBUG>0 THEN                                              00007910
515     2             DO;                                                        00007920
516     2    1           PUT LIST('TIMING I_T_ORD') SKIP;                       00007930
517     2    1           CALL PRNT_TIME;                                        00007940
518     2    1        END;                                                       00007950
519     2             DO J=1 TO R;                                               00007960
520     2    1           IF J=1 THEN                                            00007970
521     2    1              DO;                          /*INITIAL DUMMY GUESS OF 00007980
                                                         ORDERINGS*/            00007990
522     2    2                DO I=1 TO M;                                       00008000
523     2    3                   I_T_ORD(I,1)=I;                                 00008010
524     2    3                END;                      /*LOOP ON I*/           00008020
525     2    2              END;                        /*END OF FIRST CONDITIONAL*/ 00008030
              /*SINCE ADJACENT ATOMS IN LIST HAVE SIMILAR CHARACTERISTICS, USE LAST 00008040
              ORDERING AS GOOD FIRST GUESS*/                                      00008050
526     2    1           ELSE                                                    00008060
526     2    1              DO;                                                  00008070
527     2    2                DO I=1 TO M;                                       00008080
528     2    3                   I_T_ORD(I,J)=I_T_ORD(I,J-1);                    00008090
529     2    3                END;                      /*LOOP ON I*/           00008100
530     2    2              END;                        /*END OF SECOND CONDITIONAL*/ 00008110
531     2    1           NCHANGE=1;                                             00008120
532     2    1           DO I=1 TO M+1 WHILE (NCHANGE>0);/*"M+1" IS A DUMMY STOPPING 00008130
                                                         PT*/                   00008140
533     2    2              NCHANGE=0;                                          00008150
534     2    2              DO K=1 TO M-1;              /*TIES DO NOT GENERATE  00008160
                                                         RANDOMNESS IN ORDERING*/ 00008170
535     2    3                IF C(I_T_ORD(K+1,J),J)<C(I_T_ORD(K,J),J) THEN     00008180
536     2    3                   GO TO SWITCH;                                   00008190
537     2    3                IF C(I_T_ORD(K+1,J),J)=C(I_T_ORD(K,J),J) THEN     00008200
538     2    3                   DO;                                             00008210
539     2    4                      IF I_T_ORD(K+1,J)<I_T_ORD(K,J) THEN          00008220
540     2    4                         GO TO SWITCH;                             00008230
541     2    4                   END;                  /*END CHECK FOR TIES*/    00008240
542     2    3                GO TO END_SWITCH;                                  00008250
543     2    3     SWITCH:                                                       00008260
```

```
STMT LEVEL NEST

                        IDUM=I_T_ORD(K,J);                                      00008270
544     2    3             I_T_ORD(K,J)=I_T_ORD(K+1,J);                         00008280
545     2    3             I_T_ORD(K+1,J)=IDUM;                                 00008290
546     2    3             NCHANGE=NCHANGE+1;                                   00008300
547     2    3    END_SWITCH:                                                   00008310
                     END;                          /*LOOP ON K*/              00008320
548     2    2          END;                       /*LOOP ON I*/              00008330
549     2    1          IF I=M+2 THEN                                          00008340
550     2    1             CALL ERR(4);                                        00008350
551     2    1       END;                          /*LOOP ON J*/              00008360
552     2            IF DEBUG>0 THEN                                          00008370
553     2               DO;                                                   00008380
554     2    1             PUT LIST('I_T_ORD FINISHED') SKIP;                  00008390
555     2    1             CALL PRNT_TIME;                                     00008400
556     2    1          END;                                                   00008410
                  /*SET UP TABLE IDENTIFYING SIMILAR ATOMS RR=TOTAL NUMBER OF SETS OF  00008420
                  SIMILAR ATOMS ISIM(J)=LOC IN SIM OF 1ST OF A SET OF ATOM NO.S       00008430
                  BELONGING TO JTH SIMILARITY SET; J=1,2,...,RR ISIM(RR+1)=R+1 BY     00008440
                  DEFINITION*/                                                        00008450
557     2            RR=0;                                                          00008460
558     2            FLAG=0;                       /*FLAG IS A BINARY VECTOR*/    00008470
559     2            ISIM(1)=1;                                                     00008480
560     2            DO I=1 TO R;                                                   00008490
561     2    1          IF FLAG(I)=1 THEN                                          00008500
562     2    1             GO TO END_SIM_LP;       /*IS ATOM I ALREADY IN A */    00008510
563     2    1          RR=RR+1;                   /*ONE MORE SET IDENTIFIED      00008520
                                                    SIMILARITY SET? */           00008530
564     2    1          ISIM(RR+1)=ISIM(RR)+1;                                     00008540
565     2    1          SIM(ISIM(RR))=I;                                           00008550
                  /*NOW SEARCH FOR OTHER MEMBERS OF THIS SET*/                    00008560
566     2    1          DO J=I+1 TO R;                                             00008570
567     2    2             IF FLAG(J)=1 THEN                                       00008580
568     2    2                GO TO J_LOOP;                                        00008590
                  /*OK, ATOM J NOT YET IN A SIMILARITY SET NOW TEST FOR SIMILARITY*/  00008600
569     2    2             DO K=1 TO M;                                            00008610
570     2    3                IF I_T_ORD(K,I)¬=I_T_ORD(K,J) THEN                   00008620
571     2    3                   GO TO J_LOOP;                                     00008630
572     2    3             END;                       /*LOOP ON K*/              00008640
573     2    2             DO K=1 TO M-1;             /*CHECK THAT ANY TIES ARE  00008650
                                                       IDENTICAL*/               00008660
574     2    3                IF C(I_T_ORD(K+1,I),I)-C(I_T_ORD(K,I),I)=0E0 THEN   00008670
575     2    3                   DO;                                               00008680
576     2    4                      IF C(I_T_ORD(K+1,J),J)-C(I_T_ORD(K,J),J)¬=0E0  00008690
                                    THEN                                         00008700
577     2    4                         GO TO J_LOOP;                             00008710
578     2    4                      GO TO K_LOOP;     /*THERE IS A TIE FOR BOTH ATOMS*/  00008720
579     2    4                   END;                 /*END OF CONDITIONAL*/      00008730
580     2    3                IF C(I_T_ORD(K+1,J),J)-C(I_T_ORD(K,J),J)=0E0 THEN   00008740
581     2    3                   GO TO J_LOOP;                                     00008750
582     2    3    K_LOOP:                                                        00008760
                     END;                          /*LOOP ON K*/              00008770
                  /*ATOM J IS FOUND TO BE SIMILAR TO ATOM I:     */             00008780
583     2    2             FLAG(J)=1;                                             00008790
584     2    2             SIM(ISIM(RR+1))=J;                                     00008800
585     2    2             ISIM(RR+1)=ISIM(RR+1)+1;                               00008810
```

```
STMT LEVEL NEST

586      2    2    J_LOOP:                                                                  00008820
                       END;                              /*LOOP ON J*/                    00008830
587      2    1    END_SIM_LP:                                                              00008840
                  END;                                   /*LOOP ON I*/                    00008850
588      2         IF ISIM(RR+1)¬=R+1 THEN                                                  00008860
589      2            CALL ERR(5);                                                          00008870
590      2         PUT                                                                      00008880
                   EDIT('THE FOLLOWING ',ATOM,' GROUPS HAVE BEEN FORMED BECAUSE') (       00008890
                    A,A,A) SKIP(4);                                                       00008900
591      2         PUT EDIT('OF IDENTICAL DISPATCH PREFERENCES')(A) SKIP;                   00008910
592      2         DO I=1 TO RR;                                                           00008920
593      2    1       IDUM=ISIM(I+1)-ISIM(I);                                             00008930
594      2    1       IF IDUM>1 THEN                                                      00008940
595      2    1          DO;                                                              00008950
596      2    2             PUT EDIT((A_MAP(SIM(ISIM(I)+J)) DO J=0 TO IDUM-1)) (         00008960
                            COLUMN(5),(IDUM)(F(3),X(3))) SKIP;                            00008970
597      2    2          END;                             /*END CONDITIONAL*/             00008980
598      2    1    END;                                   /*LOOP ON I*/                    00008990
599      2         IF ESTSTAT=1 THEN                                                       00009000
600      2            GO TO K_RETURN;                                                      00009010
                  /*ALGORITHM FOR DETERMINING UNIT-STEP TOUR OF HYPERCUBE VERTICES       00009020
                  B(I)=I_TH VERTEX IN TOUR */                                             00009030
601      2         B(1)=0B;                                                                00009040
602      2         B(2)=1B;                                                                00009050
603      2         SCALE2=TWO;                                                             00009060
604      2         DO J=2 TO M;                                                            00009070
605      2    1       SCALE1=SCALE2;                                                      00009080
606      2    1       SCALE2=SCALE2*TWO;                                                  00009090
607      2    1       DO I=SCALE1 TO SCALE2-1;                                            00009100
608      2    2          B(I+1)=SCALE1 + B(SCALE2-I);                                     00009110
609      2    2       END;                                                                00009120
610      2    1    END;                                   /*LOOP ON I THEN J*/             00009130
                  /*DETERMINE WEIGHTS OF BINARY NUMBERS W(K)=0,1,2,...,M; W(K) IS        00009140
                  WEIGHT OF NUMBER WHOSE VALUE IS K-1 USE UNIT-STEP PROPERTY OF ABOVE  00009150
                  SEQUENCE */                                                             00009160
611      2         W(1)=ZERO;                                                              00009170
612      2         DO K=2 TO N;                                                            00009180
613      2    1       IF B(K)>B(K-1) THEN                                                 00009190
614      2    1          W(B(K)+1)=W(B(K-1)+1)+1;                                         00009200
615      2    1       ELSE                                                                00009210
615      2    1          W(B(K)+1)=W(B(K-1)+1)-1;                                         00009220
616      2    1    END;                                   /*LOOP ON K*/                    00009230
                  /*DETERMINE NUMBER OF STATES IN EACH DIAGONAL HYPERPLANE NS(J)=#       00009240
                  STATES IN J-1ST HYPERPLANE; J=1,2,...,M+1*/                             00009250
617      2         DO I=0 TO M;                           /*RECALL:                      00009260
                                                          FACT(I)=(I-1)FACTORIAL*/        00009270
618      2    1       NS(I+1)=FACT(M+1)/(FACT(M+1-I)*FACT(I+1));                          00009280
619      2    1    END;                                                                    00009290
                  /*SET UP MAP FOR PACKED MATRIX OF "UPWARD" AND "ON-DIAGONAL" COEF      00009300
                  TOTAL NUMBER OF MATRIX ENTRIES= 2**M + ((2**M)*M)/2 */                  00009310
620      2         MAP(1)=1;                                                               00009320
621      2         DO K=2 TO N;                                                            00009330
                  /*'+1' FOR ON-DIAG TERM, 'W(K-1)' FOR EACH BUSY UNIT*/                  00009340
622      2    1       MAP(K)=MAP(K-1)+1+W(K-1);                                           00009350
623      2    1    END;                                                                    00009360
```

```
STMT LEVEL NEST

                    /*GENERATE BINARY MASKS; MASK IS A 15-DIGIT BINARY STRING*/                    00009370
624      2              MASK(1)='0'B;                                                             00009380
625      2              DO I=2 TO M;                                                              00009390
626      2    1            IDUM=2**(I-1) - 1;                                                    00009400
627      2    1            MASK(I)=IDUM;                  /*EX:MASK(4)=000000000001118*/        00009410
628      2    1         END;                              /*LOOP ON I*/                          00009420
                    /*FILL IN MATRIX OF COEFFICIENTS*/                                           00009430
629      2              T_RATE=0E0;                       /*INITIALIZE*/                         00009440
630      2              I_RETURN=MATRIX_INPUT;                                                    00009450
631      2              ALT_RETURN=HERE_IF_ONES;                                                  00009460
632      2              KK=0;                                                                     00009470
633      2         L1:                                                                            00009480
                        KK=KK+1;                                                                00009490
634      2              IF KK>RR THEN                                                             00009500
635      2                 GO TO MATRET;                                                         00009510
636      2              J=SIM(ISIM(KK));                  /*1ST (& ARBITRARY) MEMBER OF          00009520
                                                          KKTH SIM SET*/                        00009530
637      2              K=0;                                                                      00009540
638      2         L2:                                                                            00009550
                        K=K+1;                                                                  00009560
639      2              IF K>N THEN                                                               00009570
640      2                 GO TO L1;                                                             00009580
641      2              ALPHA=B(K);                       /*BIT STRING REPRESENTATION OF         00009590
                                                          STATE B(K)*/                          00009600
642      2              GO TO TOUR;                                                               00009610
                    /*NOW INPUT INTO MATRIX,  "OFF-DIAGONAL TERMS" HERE*/                        00009620
643      2         MATRIX_INPUT:                                                                  00009630
                        XDUM=0E0;                         /*AGGREGATE CALL RATES FROM SIM        00009640
                                                          ATOMS*/                               00009650
644      2              DO JJ=ISIM(KK) TO ISIM(KK+1)-1;                                           00009660
645      2    1            XDUM=XDUM+L(SIM(JJ));                                                00009670
646      2    1         END;                              /*LOOP ON JJ*/                         00009680
647      2              XDUM=XDUM/(NTIE+1);               /*INCREMENTAL LAMBDA FOR EACH          00009690
                                                          OPTIMAL UNIT*/                        00009700
648      2              DO I=1 TO NTIE+1;                                                         00009710
649      2    1            BETA=ALPHA;                                                          00009720
650      2    1            SUBSTR(BETA,16-TIECAR(I),1)='1'B;/*GET ADJACENT ST*/                 00009730
651      2    1            KB=BETA+1;                    /*NUMERICAL VALUE OF ADJACENT          00009740
                                                          STATE*/                               00009750
                    /*FIND NO OF BUSY UNITS IN STATE BETA WHOSE UNIT NO DESIGNATIONS ARE       00009760
                    LESS THAN TIECAR(I)...THIS GIVES US CORRECT ADDRESS IN PACKED MATRIX*/      00009770
652      2    1            DUMBIT=BETA&MASK(TIECAR(I));                                         00009780
653      2    1            KC=DUMBIT+1;                                                         00009790
654      2    1            ADDRESS=MAP(KB)+W(KB)-W(KC);                                         00009800
                    /*NOTE THAT THIS FILLING OF T_RATE IS RO INDEPEND*/                          00009810
655      2    1            T_RATE(ADDRESS)=T_RATE(ADDRESS)+XDUM;                                00009820
656      2    1         END;                              /*LOOP ON I*/                          00009830
657      2         HERE_IF_ONES:                                                                  00009840
                        GO TO L2;                                                               00009850
658      2         MATRET:                                                                        00009860
                        GO TO K_RETURN;                                                         00009870
                    /*SOLVE EQUATIONS OF DETAILED BALANCE ON HYPERCUBE FIRST COMPUTE            00009880
                    BIRTH AND DEATH PROBABILITIES*/                                             00009890
659      2         SOLV_EQ:                                                                       00009900
                        BDPROB=0E0;                                                             00009910
```

```
STMT LEVEL NEST

                    /*ABOVE MATRIX INIT. IMPORTANT FOR BDPROB(M+2)=PROB OF A QUEUE OF +     00009920
                    LENGTH BDPROB(I)=PROB OF BEING IN B&D STATE I-1 FOR I=1,2,...,M+1      00009930
                    'CAPACITY'>0 SIGNALS INFINITE CAPACITY QUEUE*/                         00009940
660       2               BDPROB(1)=1E0;                                                   00009950
661       2               DENOM=1E0;                                                       00009960
662       2               DO I=1 TO M;                                                     00009970
663       2     1            BDPROB(I+1)=BDPROB(I)*RO/I;    /*=RO**I/I */                 00009980
664       2     1            DENOM=DENOM+BDPROB(I+1);                                      00009990
665       2     1         END;                              /*LOOP ON I*/                  00010000
666       2               IF CAPACITY>ZERO THEN                                            00010010
667       2                  DO;                                                           00010020
668       2     1               IF RO>=M THEN                                              00010030
669       2     1                  CALL ERR(6);            /*TEST STILL APPLIES WITH DIFF 00010040
                                                          MU'S*/                           00010050
670       2     1               BDPROB(M+2)=BDPROB(M+1)*RO/(M-RO);                         00010060
671       2     1               DENOM=DENOM+BDPROB(M+2);                                   00010070
672       2     1            END;                          /*END OPS FOR INFINITE CAPACITY 00010080
                                                          QUEUE*/                          00010090
673       2               DO I=1 TO M+2;                                                   00010100
674       2     1            BDPROB(I)=BDPROB(I)/DENOM;    /*NORMALIZE*/                 00010110
675       2     1         END;                                                            00010120
676       2               XDUM=SUM(BDPROB);                                              00010130
677       2               IF ABS(XDUM-1E0)>.001 THEN                                     00010140
678       2                  PUT LIST('ROUNDING ERR.,RUN PROCEEDING') SKIP;              00010150
                    /*POSSIBLE CODE HERE TO CORRECT ROUNDING ERRORS*****************/    00010160
679       2               XBDPROB(*,RUNNUM)=BDPROB;      /*STORE RESULTS FOR OUTPUT     00010170
                                                          CALCULATIONS*/                   00010180
680       2               IF FSTSTAT=1 THEN                                              00010190
681       2                  GO TO K_RETURN;                                             00010200
                    /*INSERT CURRENT VALUE OF RO IN MATRIX OF COEFFICIENTS*/             00010210
682       2               IF RUNNUM=1 THEN                                               00010220
683       2                  DO;                                                         00010230
684       2     1               T_RATE=RO*T_RATE;          /*MATRIX MULTIPLICATION*/    00010240
685       2     1            END;                                                        00010250
686       2               ELSE                                                           00010260
686       2                  DO;                                                         00010270
687       2     1               XDUM=RO/(RO-1RO);                                        00010280
688       2     1               T_RATE=T_RATE*XDUM;        /*SCALES UP FROM LAST VALUE*/ 00010290
689       2     1            END;                                                        00010300
                    /*NOW FILL IN "ON-DIAGONAL" TERMS*/                                  00010310
690       2                  IF MU(1)=INFINITY THEN                                      00010320
691       2               DO K=1 TO N;                                                   00010330
692       2     1            T_RATE(MAP(K))=RO +W(K);                                    00010340
693       2     1         END;                              /*LOOP ON K*/                  00010350
694       2               ELSE DO;                                                       00010360
695       2     1         T_RATE(MAP(B(1)+1))=RO;                                       00010370
696       2     1         T_RATE(MAP(B(2)+1))=MU(1)+RO;                                 00010380
697       2     1         SCALE2=TWO;                                                   00010390
698       2     1         DO J=2 TO M;                                                  00010400
699       2     2         SCALE1=SCALE2;                                                00010410
700       2     2         SCALE2=SCALE2*TWO;                                            00010420
701       2     2         DO I=SCALE1 TO SCALE2-1;                                      00010430
702       2     3         T_RATE(MAP(B(I+1)+1))=MU(J)+T_RATE(MAP(B(SCALE2-I)+1));       00010440
703       2     3         END;                                                          00010450
704       2     2         END;                                                          00010460
```

```
STMT LEVEL NEST

705     2    1        END;   /*END ELSE DO COND*/                                          00010470
706     2             IF CAPACITY=0E0 THEN T_RATE(MAP(N))=T_RATE(MAP(N))-R0;             00010480
                   /*COMPUTE INITIAL STATE PROBABILITY GUESSES FOR STATES IN EVEN-       00010490
                   NUMBERED HYPERPLANES:0,2,4,6,...*/                                   00010500
708     2             STPROB=0;                                                          00010510
709     2             DO J=1 TO N BY 2;              /*USE ODD-EVEN HYPERPLANE PROP     00010520
                                                     OF UNIT-STEP TOUR*/                00010530
710     2    1           I=B(J)+1;                   /*INDEX UP ONE FOR ARRAY           00010540
                                                     CORRESPONDENCE*/                   00010550
711     2    1           IDUM=W(I)+1;                                                   00010560
712     2    1           STPROB(I)=BDPROB(IDUM)/NS(IDUM);                               00010570
713     2    1        END;                           /*LOOP ON J*/                      00010580
                   /*ITERATE FOR SOLUTIONS TO EQUATIONS*/                             00010590
                   /*FOR SOME STATE KA, THE EQUATION OF DETAILED BALANCE IS           00010600
                   STPROB(KA)*T_RATE(MAP(KA))= SUM OF M TERMS. IF UNIT I IS BUSY IN    00010610
                   STATE KA, THEN TERM I (COUNTING FROM THE RIGHT) IS                 00010620
                   STPROB(KB)*T_RATE(MAP(KA) + W(KA) - INDEX) WHERE KB=STATE          00010630
                   UNIT-HAMMING DISTANCE FROM KA, WITH DIGIT I A ZERO INDEX=# OTHER BUSY 00010640
                   SERVERS IN STATE KA TO THE RIGHT OF UNIT I ELSE TERM I IS          00010650
                   STPROB(KB)*MU(K) */                                              00010660
714     2             IF DEBUG>0 THEN                                                00010670
715     2                DO;                                                         00010680
716     2    1              PUT LIST('TIMING ITERATIONS') SKIP;                      00010690
717     2    1              CALL PRNT_TIME;                                          00010700
718     2    1           END;                                                        00010710
719     2             CONV_METRIC=1E0;                                               00010720
720     2             IDUM=2;                        /*START ITERATING ON               00010730
                                                     ODD-NUMBERED HYPERPLANES*/         00010740
721     2             DO ITERATE=1 TO MAXBOUND WHILE (CONV_METRIC>EPSILON);/*DOUB      00010750
                                                     COUNT*/                            00010760
722     2    1           CONV_METRIC=0E0;                                            00010770
723     2    1           DO J=IDUM TO N BY 2;                                        00010780
724     2    2              ALPHA=B(J);                                              00010790
725     2    2              I=ALPHA+1;                                               00010800
726     2    2              XDUM=STPROB(I);                                          00010810
727     2    2              STPROB(I)=0E0;                                           00010820
728     2    2              ADDRESS=MAP(I)+W(I)+1;                                   00010830
                   /*ONE LESS MULT IN CRIT LOOP IF ALL SERV TIMES ARE IDENT*/         00010840
729     2    2              IF MU(1)=INFINITY THEN                                   00010850
730     2    2                 GO TO NO_MULT;                                        00010860
731     2    2              DO K=1 TO M;                                             00010870
732     2    3                 BETA=ALPHA;                                           00010880
                   /*READING BINARY DIGITS FROM RIGHT TO LEFT*/                       00010890
733     2    3                 CBIT=¬SUBSTR(ALPHA,16-K,1);                           00010900
734     2    3                 SUBSTR(BETA,16-K,1)=CBIT;                             00010910
735     2    3                 KB=BETA+1;                                            00010920
736     2    3                 IF CBIT='1'B THEN                                     00010930
737     2    3                    STPROB(I)=STPROB(I)+MU(K)*STPROB(KB);              00010940
                   /*ABOVE WAS TRANSITIONING FROM A '1' TO A '0' BELOW IS REVERSE     00010950
                   (HARDER CASE) */                                                 00010960
738     2    3                 ELSE                                                  00010970
738     2    3                    DO;                                                00010980
739     2    4                       ADDRESS=ADDRESS-1;                             00010990
740     2    4                       STPROB(I)=STPROB(I)+STPROB(KB)*                00011000
                                     T_RATE(ADDRESS);                               00011010
```

```
STMT LEVEL NEST

741     2    4                    END;                                                    00011020
742     2    3                END;                          /*LOOP ON K*/                00011030
743     2    2                GO TO DIAG_DIV;                                           00011040
744     2    2    NO_MULT:                                                             00011050
                  DO K=1 TO M;                                                         00011060
745     2    3       BETA=ALPHA;                                                       00011070
746     2    3       CBIT=¬SUBSTR(ALPHA,16-K,1);                                       00011080
747     2    3       SUBSTR(BETA,16-K,1)=CBIT;                                         00011090
748     2    3       KB=BETA+1;                                                        00011100
749     2    3       IF CBIT='1'B THEN                                                 00011110
750     2    3          STPROB(I)=STPROB(I)+STPROB(KB);                                00011120
                  /*A COEFF OF MU=1 IS ASSUMED*/                                       00011130
751     2    3       ELSE                                                              00011140
751     2    3          DO;                                                            00011150
752     2    4             ADDRESS=ADDRESS-1;                                          00011160
753     2    4             STPROB(I)=STPROB(I)+STPROB(KB)*                             00011170
                           T_RATE(ADDRESS);                                           00011180
754     2    4          END;                                                           00011190
755     2    3       END;                          /*LOOP ON K*/                      00011200
756     2    2    DIAG_DIV:                                                            00011210
                  STPROB(I)=STPROB(I)/T_RATE(MAP(I));/*DIVIDE BY ON-DIAG */          00011220
757     2    2       IF I=N & CAPACITY>0E0 THEN                                        00011230
758     2    2          DO;                                                            00011240
759     2    3             STPROB(N)=STPROB(N)+(M-RO)*BDPROB(M+2)/(M+RO);              00011250
760     2    3             IF MU(1)<INFINITY THEN                                      00011260
761     2    3                BDPROB(M+2)=STPROB(N)*RO/(M-RO);                         00011270
762     2    3          END;                          /*END TEST FOR BOUNDARY STATE*/  00011280
763     2    2       IF STPROB(I)>MINVAL THEN                                          00011290
764     2    2          DO;                                                            00011300
765     2    3             XDUM=ABS(XDUM-STPROB(I));                                   00011310
766     2    3             CONV_METRIC=MAX(CONV_METRIC,XDUM);                          00011320
767     2    3          END;                          /*END CONVERGENCE TEST*/         00011330
768     2    2    END;                               /*LOOP ON J*/                    00011340
769     2    1    IF DEBUG>0 THEN                                                     00011350
770     2    1       PUT DATA(CONV_METRIC) SKIP;                                     00011360
771     2    1    IF IDUM=2 THEN                                                      00011370
772     2    1       IDUM=1;                                                         00011380
773     2    1    ELSE                                                              00011390
773     2    1       IDUM=2;                                                         00011400
                  /*ONE ITERATION COMPLETED**********/                              00011410
774     2    1  END;                                  /*ITERATE ON 'ITERATE'*/       00011420
775     2       ITERATE=ITERATE/2;                    /*ELIMINATES THE DOUBLE        00011430
                                                      COUNTING*/                     00011440
776     2       IF CAPACITY=0E0 THEN                                                00011450
777     2          DO;                                                              00011460
778     2    1        DENOM=SUM(STPROB);                                           00011470
779     2    1        STPROB=STPROB/DENOM;                                         00011480
780     2    1     END;                                                            00011490
781     2       ELSE                                                             00011500
781     2          DO;                                                           00011510
782     2    1        DENOM=SUM(STPROB) + STPROB(N)*RO/(M-RO);                     00011520
783     2    1        STPROB=STPROB/DENOM;                                         00011530
784     2    1.       IF MU(1)¬=INFINITY THEN                                      00011540
785     2    1           DO;                          /*MUST NORMALIZE THESE IN GEN*/ 00011550
786     2    2              BDPROB(M+2)=BDPROB(M+2)/DENOM;                         00011560
```

```
STMT LEVEL NEST

787     2    2                  BDPROB(M+1)=STPROB(N);/*CORRECTS FOR APPROX PROC- IF  00011570
                                                      REQ*/                          00011580
788     2    2                END;                     /*END COND FOR DIFF SERVICE    00011590
                                                      TIMES*/                        00011600
789     2    1            END;                                                       00011610
790     2                 XSTPROB(*,RUNNUM)=STPROB;                                  00011620
791     2                 IF DEBUG>0 THEN                                            00011630
792     2                    DO;                                                     00011640
793     2    1                  PUT LIST('ITERATION COMPLETED') SKIP;                00011650
794     2    1                  PUT DATA(ITERATE) SKIP;                              00011660
795     2    1                  CALL PRNT_TIME;                                      00011670
796     2    1               END;                                                    00011680
797     2                 GO TO K_RETURN;                                            00011690
798     2            TOUR:                                                          00011700
                          IF K=1 THEN                                               00011710
799     2                    DO;                                                     00011720
800     2    1                  NPREF=1;                 /*NPREF=PREF. # OF UNIT     00011730
                                                      DISPATCHED TO J*/              00011740
                     /*NSELECT=# UNIT DESIGNATED BY NPREF*/                         00011750
801     2    1                  NSELECT=I_T_ORD(1,J);                               00011760
802     2    1                  TIECAR(1)=NSELECT;                                  00011770
803     2    1                  NTIE=0;                                             00011780
804     2    1                  DO I=2 TO M;             /*REMOTE POSS. OF TIES,EVEN 00011790
                                                      WHEN ALL AV*/                  00011800
805     2    2                     IF C(I_T_ORD(I,J),J)>C(I_T_ORD(1,J),J) THEN      00011810
806     2    2                        GO TO I_RETURN;                               00011820
807     2    2                     NTIE=NTIE+1;                                     00011830
808     2    2                     TIECAR(I)=I_T_ORD(I,J);                          00011840
809     2    2                  END;                     /*LOOP ON I*/              00011850
810     2    1                  GO TO I_RETURN;                                     00011860
811     2    1               END;                        /*FINISHED HERE IF K=1*/    00011870
812     2                 LASTVER=B(K-1);               /*BIT STRING*/              00011880
                     /*DUMBIT=BINARY STRING WITH ALL ZERO"S EXCEPT A ONE IN POSITION 00011890
                     CORRESPONDING TO A CHANGED STATUS*/                            00011900
813     2                 DUMBIT=(LASTVER&¬ALPHA)|((¬LASTVER)&ALPHA);                00011910
814     2                 KC=DUMBIT;                    /*KC=1,2,4,8,...*/          00011920
815     2                 NCAR=LOG2(KC)+1;              /*THIS IS 1ST INSTRUCTION THAT 00011930
                                                      MAPS CAR*/                     00011940
                     /*#'S ONTO BINARY STATES; CAR 1 STATUS IN RIGHT-MOST DIGIT, CAR 2 IN 00011950
                     2ND, ETC.*/                                                    00011960
816     2                 IF B(K)<B(K-1) THEN                                        00011970
817     2                    DO;                        /*IMPLIES UNIT NCAR NOW AVAIL*/ 00011980
818     2    1                  IF C(NCAR,J)<C(NSELECT,J) THEN                      00011990
819     2    1                     DO;                                              00012000
                     /*CAR NCAR IS NOW THE OPTIMAL UNIT*/                           00012010
820     2    2                        NSELECT=NCAR;                                 00012020
821     2    2                        NTIE=0;                                       00012030
822     2    2                        TIECAR(1)=NSELECT;                            00012040
823     2    2                        DO I=1 TO M;        /*GET PREFERENCE NUMBER*/ 00012050
824     2    3                           IF I_T_ORD(I,J)=NCAR THEN                  00012060
825     2    3                              DO;                                     00012070
826     2    4                                 NPREF=I;                             00012080
827     2    4                                 I=M+1;                               00012090
828     2    4                              END;          /*END CONDITIONAL*/       00012100
829     2    3                        END;                /*LOOP ON I*/             00012110
```

```
STMT LEVEL NEST

830     2    2                  GO TO I_RETURN;                                          00012120
831     2    2                END;                        /*FINISHED,IF NEW UNIT IS     00012130
                                                          OPTIMAL*/                    00012140
832     2    1             IF C(NCAR,J)=C(NSELECT,J) THEN                              00012150
833     2    1                DO;                         /*A TIE*/                     00012160
834     2    2                   NTIE=NTIE+1;                                          00012170
835     2    2                   TIECAR(NTIE+1)=NCAR;                                  00012180
836     2    2                   GO TO I_RETURN;                                       00012190
837     2    2                END;                        /*FINISHED, IF TIE*/          00012200
838     2    1          END;                              /*END CASE FOR WHICH NCAR NOW 00012210
                                                          AVAILABLE*/                  00012220
839     2               ELSE                                                          00012230
839     2                  DO;                           /*IMPLIES UNIT NCAR NOW       00012240
                                                          UNAVAILABLE*/                00012250
                     /*WAS UNIT NCAR AN OPTIMAL CHOICE?*/                            00012260
840     2    1             IF C(NCAR,J)=C(NSELECT,J) THEN                              00012270
841     2    1                DO;                                                     00012280
                     /*IF SO, WAS THERE A TIE?*/                                     00012290
842     2    2                   IF NTIE¬=0 THEN                                       00012300
843     2    2                      DO;                                                00012310
                     /*IF SO, REMOVE NCAR FROM TIE LIST*/                            00012320
844     2    3                         DO I=1 TO NTIE;                               00012330
845     2    4                            IF TIECAR(I)=NCAR THEN                     00012340
846     2    4                               DO;                                     00012350
                     /*THIS LOOP DOES NOT ERASE TIECAR( NTIE+1). DOES NOT MATTER.*/  00012360
847     2    5                                  DO JJ=I TO NTIE;                     00012370
848     2    6                                     TIECAR(JJ)=TIECAR(JJ+1);          00012380
849     2    6                                  END;  /*LOOP ON JJ*/                 00012390
850     2    5                                  I=NTIE+1;                            00012400
851     2    5                               END;     /*END OF COND. */              00012410
852     2    4                         END;           /*LOOP ON I*/                  00012420
853     2    3                         NTIE=NTIE-1;                                  00012430
854     2    3                         NSELECT=TIECAR(1);                            00012440
855     2    3                         GO TO I_RETURN;                               00012450
856     2    3                      END;              /*FINISH OPERATIONS FOR THE    00012460
                                                          CASE OF A TIE*/              00012470
857     2    2                   ELSE                                                  00012480
857     2    2                      DO;                                                00012490
                     /*NO TIE AND NCAR WAS OPTIMAL FIND NEXT PREFERRABLE AVAILABLE UNIT*/ 00012500
858     2    3                         DO I=NPREF+1 TO M;                            00012510
                     /*IS THE NEXT PREFERRED UNIT AVAILABLE?*/                       00012520
859     2    4                            IF SUBSTR(ALPHA,16-I_T_ORD(I,J),1)='0'B    00012530
                                          THEN                                       00012540
860     2    4                               DO;                                     00012550
861     2    5                                  NPREF=I;                             00012560
862     2    5                                  NSELECT=I_T_ORD(I,J);                00012570
863     2    5                                  TIECAR(1)=NSELECT;                   00012580
                     /*CHECK FOR POSSIBLE TIE*/                                     00012590
864     2    5                                  DO JJ=I+1 TO M;                      00012600
865     2    6                                     IF C(NSELECT,J)=C(I_T_ORD(JJ,J),J) 00012610
                                                   THEN                            00012620
866     2    6                                        DO;                           00012630
867     2    7                                        IF                            00012640
                                                      SUBSTR(ALPHA,16-I_T_ORD(JJ,J),1)= 00012650
                                                                                    00012660
```

```
STMT LEVEL NEST

                                                    '0'B THEN                                  00012670
868     2    7                                      DO;                                        00012680
869     2    8                                      NTIE=NTIE+1;                               00012690
870     2    8                                      TIECAR(NTIE+1)=                            00012700
                                                    I_T_ORD(JJ,J);                             00012710
871     2    8                                      END;/*FINISH AV CK*/                       00012720
872     2    7                                      ELSE                                       00012730
872     2    7                                      DO;                                        00012740
873     2    8                                      END;/*DUMMY*/                              00012750
874     2    7                                      END;                                       00012760
875     2    6                                  ELSE                                           00012770
875     2    6                                      DO;                                        00012780
                 /*NO TIES, OR AT LEAST, NO MORE TIES*/                                       00012790
876     2    7                                      GO TO I_RETURN;                            00012800
877     2    7                                      END;/*END TEST */                          00012810
878     2    6                                  END;  /*LOOP ON JJ*/                           00012820
879     2    5                                  GO TO I_RETURN;                                00012830
                 /*FINALLY, END INSTRUC INVOKED ONCE OPTIMAL UNIT FOUND*/                     00012840
880     2    5                                  END;                                           00012850
881     2    4                          END;                /*LOOP ON I*/                     00012860
                 /*ARTIF DEVICE FOR STATE '111...11'*/                                        00012870
882     2    3                          NSELECT=M+1;                                           00012880
883     2    3                          GO TO ALT_RETURN;                                      00012890
                 /*END CONDITIONAL FOR WHICH NCAR WAS AN OP UN*/                              00012900
884     2    3                      END;                                                       00012910
                 /*****IF NCAR WAS NOT AN OPTIMAL UNIT THEN NOTHING HAPPENS*******/           00012920
885     2    2                 END;                     /*END OF INSTR FOR                    00012930
                                                        T(NCAR,J)=T(NSELECT,J)*/               00012940
886     2    1            END;                         /*END OF INSTR FOR UNIT NCAR          00012950
                                                        NOW UNAVAILABLE*/                      00012960
887     2              GO TO I_RETURN;                                                         00012970
888     2         CALCOUT:                                                                   00012980
                       ALLOCATE PD,PD2,MWL,MISD,TAV,VAR,SWL,UBWL,DINPUT,PINPUT,             00012990
                       INPUT, WORKLOAD,TCAR,TSEC,TREP,TQUE, DCAR,DSEC,DREP,DWL,PWL,DISDO,00013000
                       PISCO, DISDI,PISDI,DTSEC,PTSEC,ISDO,ISDI,PINSEC,MOX,MON,PINT,        00013010
                       SATPROB,GWKLD,PATFREQ;                                               00013020
889     2                RUNNUM=0;                                                          00013030
890     2         OUTLOOP:                                                                   00013040
                       RUNNUM=RUNNUM+1;                                                     00013050
891     2                IF RUNNUM>NUM THEN                                                 00013060
892     2                   GO TO OUTLOOP_FINI;                                             00013070
893     2                BDPROB=XBDPROB(*,RUNNUM);                                          00013080
894     2                STPROB=XSTPROB(*,RUNNUM);                                          00013090
895     2                SATPROB=BDPROB(M+1);                                               00013100
896     2                PD2=0E0;                                                           00013110
897     2                IF CAPACITY>0E0 THEN                                               00013120
898     2                   DO;                                                             00013130
                 /*COMPUTE PD2(I,J)=FRAC OF ALL DISPATCHES THAT (1)ARE 'UNIT I TO ATOM    00013140
                 J' AND (2) INCUR A POSITIVE QUEUE DELAY */                                 00013150
899     2    1              YDUM=SUM(MU);                /*2_ND FUNCTION OF MU(I)'S*/      00013160
900     2    1              SATPROB=SATPROB + BDPROB(M+2);                                  00013170
901     2    1              DO I=1 TO M;                                                    00013180
902     2    2                 XDUM=SATPROB*MU(I)/YDUM;/*EVEN WORKS WHEN                    00013190
                                                        MU(I)=INFINITY.. FOR                00013200
                                                        SUFFICIENTLY SMALL INFINITY*/       00013210
```

```
STMT LEVEL NEST

903      2    2                 DO J=1 TO R;                                                    00013220
904      2    3                    PQ2(I,J)=XDUM*L(J);                                          00013230
905      2    3                 END;                          /*LOOP ON J*/                    00013240
906      2    2              END;                             /*LOOP ON I*/                    00013250
907      2    1           END;                                /*END OF CONDITIONAL FOR         00013260
                                                             INFINITE CAPACITY QUEUE*/         00013270
908      2             ESTSTAT=ABS(ESTSTAT);                                                   00013280
909      2             IF ESTSTAT¬=1 THEN                                                      00013290
910      2                GO TO PD_CAL;                                                        00013300
                    /*COMPUTE ESTIMATED WORKLOADS*/                                          00013310
                    /*XDUM=AVERAGE LAMBDA.    MWL=MEAN WORKLOAD*/                          00013320
911      2          APPROX:                                                                 00013330
                       XDUM=(RO-(NUM+1-RUNNUM)*IRO)/M;                                      00013340
912      2             MWL=XDUM;                                                            00013350
913      2             IF CAPACITY=0E0 THEN                                                 00013360
914      2                MWL=MWL*(1E0-BDPROB(M+1));                                        00013370
                    /*FIRST COMPUTE QUEUE SCALE FACTORS DUE TO RANDOM INCID ON BUSY UNITS*/00013380
915      2             QUFAC=0E0;                                                           00013390
916      2             MON=1E0;                                                             00013400
917      2             IF CAPACITY=0E0 THEN                                                 00013410
918      2                MON=(1E0-XDUM)/(1E0-XDUM+BDPROB(M+1)*XDUM);                       00013420
919      2             DO J=2 TO M;                          /*J=PREF # OF 1ST AVAIL UNIT*/  00013430
920      2    1           YDUM=0E0;                                                         00013440
921      2    1           DO I= 0 TO M-J;                                                   00013450
922      2    2              YDUM=YDUM+(M-I-J+1)*(M*XDUM)**I/FACT(I+1);                     00013460
923      2    2           END;                             /*LOOP ON I*/                    00013470
924      2    1           QUFAC(J)=YDUM*FACT(M-J+1)*M**(J-1);                               00013480
925      2    1           IF CAPACITY=0E0 THEN                                              00013490
926      2    1              DO;                                                            00013500
927      2    2                 MON=MON*(1E0/(1E0-BDPROB(M+1)));                            00013510
928      2    2                 QUFAC(J)=QUFAC(J)*MON;                                      00013520
929      2    2              END;                             /*END COND ON ZERO QUEUE*/       00013530
930      2    1        END;                                  /*LOOP ON J*/                    00013540
931      2             YDUM=BDPROB(1)/(FACT(M+1)*(1E0-XDUM));                               00013550
932      2             QUFAC=QUFAC*YDUM;                                                    00013560
933      2             QUFAC(1)=1E0;                                                        00013570
934      2             IF DEBUG>0 THEN                                                      00013580
935      2                PUT DATA(QUFAC) SKIP(5);                                          00013590
                    /*NOW COMPUTE EITHER WORKLOADS OR FRAC DISPS THAT ARE UN I TO AT J*/   00013600
936      2             GWKLD=MWL;                                                           00013610
                    /* IDUM =0 IMPLIES CALCULATING WKLDS;=1 IMPLIES PD CALC*/              00013620
937      2             IDUM=0;                                                              00013630
938      2             PD=0E0;                                                              00013640
939      2             ITERATE=1;                            /*COUNTS # ITERATIONS THRU MAIN 00013650
                                                             LOOP*/                           00013660
940      2          W_LOOP:                                                                 00013670
                       WORKLOAD=1E0;                         /*MAIN LOOP FOR WORKLOAD       00013680
                                                             CALCULATIONS*/                   00013690
941      2             IF CAPACITY>0E0 THEN                                                 00013700
942      2                WORKLOAD=WORKLOAD+XDUM*(BDPROB(M+1)+BDPROB(M+2))/(                00013710
                           1E0-MWL);                                                        00013720
943      2          W_LOOP2:                                                                00013730
                       DO J=1 TO R;                          /*MAIN LOOP FOR PD CALCULATIONS*/00013740
944      2    1           IF IDUM =0 THEN                                                   00013750
945      2    1              MOX=L(J)*XDUM*M;                /*CALLS/SERV TM UNIT FR AT J*/  00013760
```

```
STMT LEVEL NEST

946     2    1            ELSE                                                                  00013770
946     2    1               MOX=L(J);                                                          00013780
947     2    1            DO I=1 TO M;                    /*CENTRAL SUBLOOP...IMPLEMENTS      00013790
                                                          APPROX EQUATION*/                    00013800
948     2    2               IF I¬=M THEN                                                       00013810
949     2    2                  IF C(I_T_ORD(I,J),J)=C(I_T_ORD(I+1,J),J) THEN                   00013820
950     2    2                     GO TO W_TIES;                                                00013830
951     2    2                  ELSE                                                            00013840
951     2    2                     ;                                                            00013850
952     2    2               ELSE                                                               00013860
952     2    2                  ;                                                               00013870
953     2    2               IF IDUM=0 THEN                                                     00013880
                        /*TOTAL WKLD OF I_TH PREFERRED UNIT FOR ATOM J IS INCREMENTED BY THE  00013890
                        RELATIVE FREQUENCY WITH WHICH IT IS DISPATCHED TO ATOM J */          00013900
954     2    2                  WORKLOAD(I_T_ORD(I,J))=WORKLOAD(I_T_ORD(I,J)) +                 00013910
                                MOX*QUFAC(I);                                                 00013920
955     2    2               ELSE                                                               00013930
955     2    2                  PD(I_T_ORD(I,J),J)=PD(I_T_ORD(I,J),J)+MOX*(                     00013940
                                1E0- WORKLOAD(I_T_ORD(I,J)))*QUFAC(I);                        00013950
956     2    2               MOX=MOX*GWKLD(I_T_ORD(I,J));                                       00013960
957     2    2               GO TO END_W_LOOP;           /*SKIP OVER TIES*/                     00013970
958     2    2        W_TIES:                                                                   00013980
                        NTIE=1;                                                               00013990
959     2    2               DO K=I+1 TO M-1;            /*HOW MANY TIES?*/                     00014000
960     2    3                  IF C(I_T_ORD(K,J),J)=C(I_T_ORD(K+1,J),J) THEN                   00014010
961     2    3                     NTIE=NTIE+1;                                                 00014020
962     2    3                  ELSE                                                            00014030
962     2    3                     K=M+1;                                                       00014040
963     2    3               END;                        /*LOOP ON K*/                          00014050
964     2    2               YDUM=0E0;                                                          00014060
965     2    2               DO K=0 TO NTIE;                                                    00014070
966     2    3                  YDUM=YDUM+GWKLD(I_T_ORD(I+K,J));                                00014080
967     2    3               END;                        /*LOOP ON K*/                          00014090
968     2    2               YDUM=YDUM/(NTIE+1);                                                00014100
969     2    2               SWL=MOX/(NTIE+1);                                                  00014110
970     2    2               DO K=0 TO NTIE;                                                    00014120
971     2    3                  DO KK=0 TO NTIE;                                                00014130
972     2    4                     IF IDUM=0 THEN                                               00014140
973     2    4                        WORKLOAD(I_T_ORD(I+KK,J))=                                00014150
                                      WORKLOAD(I_T_ORD(I+KK,J)) +SWL*YDUM**K*QUFAC(I+K);      00014160
974     2    4                     ELSE                                                         00014170
974     2    4                        PD(I_T_ORD(I+KK,J),J)=PD(I_T_ORD(I+KK,J),J)+ SWL*(       00014180
                                      1E0-YDUM)*YDUM**K*QUFAC(I+ K);                          00014190
975     2    4                  END;                     /*LOOP ON KK*/                        00014200
976     2    3               END;                        /*LOOP ON K*/                          00014210
977     2    2               DO K=0 TO NTIE;                                                    00014220
978     2    3                  MOX=MOX*GWKLD(I_T_ORD(I+K,J));                                  00014230
979     2    3               END;                        /*LOOP ON K*/                          00014240
980     2    2               I=I+NTIE;                   /*END PROCESSING OF TIES*/             00014250
981     2    2        END_W_LOOP:                                                               00014260
                        END;                            /*LOOP ON I*/                          00014270
982     2    1         END;                             /*LOOP ON J*/                          00014280
983     2              IF IDUM=1 THEN                                                          00014290
984     2                 DO;                           /*CONDITIONS NEXT 40 OR SO             00014300
                                                        INSTRUCTIONS*/                        00014310
```

```
STMT LEVEL NEST

 985     2    1              IF CAPACITY=0E0 THEN                                          00014320
 986     2    1                 PD=PD/(1E0-SATPROB);                                       00014330
 987     2    1              MOX=1E0;                    /*USED AS MULT IN NORMALIZING    00014340
                                                         PROC*/                          00014350
 988     2    1              IF CAPACITY>0E0 THEN                                          00014360
 989     2    1                 MOX=1E0-SATPROB;                                           00014370
 990     2    1              DO J=1 TO R;                                                  00014380
 991     2    2                 IF L(J)=0E0 THEN                                           00014390
 992     2    2                    GO TO SKIP_J;                                         00014400
 993     2    2                 K=0;                                                       00014410
                    /*THIS LOOP NORMALIZES PD(*,J) TO KNOWN VALUE BY USE OF EXPON DAMP*/ 00014420
 994     2    2    RECYCLE:                                                              00014430
                       MON=SUM(PD(*,J));                                                 00014440
 995     2    2       YDUM=MON-L(J)*MOX;          /*ABS AMT OF ERROR*/                  00014450
 996     2    2       XDUM=YDUM/(L(J)*MOX);                                             00014460
 997     2    2       ;                                                                 00014470
 998     2    2       XDUM=ABS(XDUM);                                                   00014480
                    /*STRINGENT CONV CRITERION FOR REL ERROR*/                          00014490
 999     2    2       IF XDUM<.001E0 THEN                                               00014500
1000     2    2          GO TO SKIP_J;                                                  00014510
1001     2    2       K=K+1;                                                            00014520
1002     2    2       IF K>30 THEN                                                      00014530
1003     2    2          GO TO SKIP_JJ;                                                 00014540
1004     2    2       MON=L(J)*MOX;                                                     00014550
1005     2    2       YDUM=MON/(MON+YDUM);        /*EXPONENTIAL CORRECTION FACTOR*/     00014560
1006     2    2       YDUM=SQRT(YDUM);            /*MADE SMALLER TO AVOID              00014570
                                                   OSCILLATION*/                         00014580
1007     2    2       ;                                                                 00014590
1008     2    2       DO I=2 TO M;                /*EXECUTE EXPONENTIAL DAMPING*/       00014600
1009     2    3          JJ=I_T_ORD(I,J);                                               00014610
1010     2    3          IF I=M THEN                                                    00014620
1011     2    3             DO;                                                         00014630
1012     2    4                PD(JJ,J)=PD(JJ,J)*(YDUM**(I-1));                         00014640
1013     2    4                GOTO END_I;                                              00014650
1014     2    4             END;                                                        00014660
1015     2    3          IF C(JJ,J)¬=C(I_T_ORD(I+1,J),J) THEN                           00014670
1016     2    3             PD(JJ,J)=PD(JJ,J)*(YDUM**(I-1));                            00014680
1017     2    3          ELSE                                                           00014690
1017     2    3             DO;                                                         00014700
1018     2    4                NTIE=1;                                                  00014710
1019     2    4                DO JJ=I+1 TO M-1;                                        00014720
1020     2    5                   IF C(I_T_ORD(JJ,J),J)=C(I_T_ORD(JJ+1,J),J)           00014730
                                  THEN                                                00014740
1021     2    5                      NTIE=NTIE+1;                                     00014750
1022     2    5                   ELSE                                                00014760
1022     2    5                      JJ=M+1;                                          00014770
1023     2    5                   ;                                                   00014780
1024     2    5                END;              /*LOOP ON JJ*/                       00014790
1025     2    4                DO JJ=0 TO NTIE;                                       00014800
1026     2    5                   MON=0E0;                                            00014810
1027     2    5                   DO KK=0 TO NTIE;                                    00014820
1028     2    6                      MON=PD(I_T_ORD(I+JJ,J),J)*(                      00014830
                                     YDUM**(I+KK-1)) + MON;                           00014840
1029     2    6                      ;                                                00014850
1030     2    6                   END;              /*LOOP ON KK*/                     00014860
```

```
STMT LEVEL NEST
1031     2    5                          PD(I_T_ORD(I+JJ,J),J)=MON/(NTIE+1);          00014870
1032     2    5                          ;                                            00014880
1033     2    5                       END;                                            00014890
1034     2    4                       I=I+NTIE;                                       00014900
1035     2    4                    END;                  /*LOOP ON JJ,   END OF DO*/  00014910
1036     2    3    END_I:                                                             00014920
                   END;                  /*LOOP ON I*/                              00014930
1037     2    2     GO TO RECYCLE;                                                   00014940
1038     2    2    SKIP_JJ:                                                          00014950
                   PUT LIST('TOO MANY ITERATIONS, RUN PROCEEDING') SKIP;            00014960
1039     2    2    SKIP_J:                                                           00014970
                END;                     /*LOOP ON J*/                             00014980
1040     2    1     YDUM=SUM(PD);                                                    00014990
1041     2    1     IF ABS(YDUM-MOX)>.001 * SQRT(R) THEN                             00015000
1042     2    1        PUT LIST('PD SUM ERROR, RUN PROCEEDING') SKIP;                00015010
                /*EXIT FROM APPROXIMATION PROCEDURE TO OUTPUT STATISTICS*/          00015020
1043     2    1        GO TO T_CAL;                                                  00015030
1044     2    1     END;                     /*END COND FOR  IDUM =1*/               00015040
                /*MANIPULATING AS IN DER OF EQ FOR APPROX WKLDS*/                   00015050
1045     2          WORKLOAD=1E0-1E0/WORKLOAD;                                       00015060
1046     2          MON=SUM(WORKLOAD);                                              00015070
1047     2          MON=MON/M;                                                      00015080
1048     2          MON=MON/MWL;                                                    00015090
1049     2          WORKLOAD=WORKLOAD/MON;           /*NORMALIZE*/                 00015100
1050     2          UBWL=0E0;                                                       00015110
1051     2          DO I=1 TO M;                                                    00015120
1052     2    1        VAR=ABS(WORKLOAD(I)-GWKLD(I));/*CMPTG LARGEST CHNG  IN WKLDS*/ 00015130
1053     2    1        IF VAR>UBWL THEN                                            00015140
1054     2    1           UBWL=VAR;                                                00015150
1055     2    1        END;                        /*LOOP ON I*/                   00015160
1056     2          IF UBWL>.001/M THEN                                             00015170
1057     2             DO;                        /*CONVERGENCE TEST*/            00015180
1058     2    1           GWKLD=WORKLOAD;                                          00015190
1059     2    1           ITERATE=ITERATE+1;                                       00015200
1060     2    1           GO TO W_LOOP;                                            00015210
1061     2    1        END;                                                        00015220
1062     2          ELSE                                                          00015230
1062     2             DO;                                                        00015240
1063     2    1           IDUM=1;                  /*FINISHED WITH WORKLOADS, NOW 00015250
                                                  ON TO PD*/                       00015260
1064     2    1           GWKLD=WORKLOAD;                                          00015270
1065     2    1           GO TO W_LOOP2;           /*GO TO 2_ND MAIN LOOP*/       00015280
1066     2    1        END;                                                        00015290
                /*CALCULATE PD(I,J)=FRACTION OF ALL DISPATCHES THAT (1)ARE 'UNIT I TO 00015300
                ATOM J' AND (2) INCUR NO QUEUE DELAY*/                           00015310
1067     2          PD_CAL:                                                       00015320
                   PD=0E0;                  /*MATRIX INITIALIZATION*/            00015330
1068     2          I_RETURN=PD_CALC;                                             00015340
1069     2          ALT_RETURN=ALL_BUSY;                                          00015350
1070     2          KK=0;                                                         00015360
1071     2          L11:                                                          00015370
                   KK=KK+1;                                                      00015380
1072     2          IF KK>RR THEN                                                 00015390
1073     2             GO TO RETOUR_FINI;                                         00015400
1074     2          J=SIM(ISIM(KK));                                              00015410
```

```
 STMT LEVEL NEST

1075     2              K=0;                                                                    00015420
1076     2          L22:                                                                        00015430
                        K=K+1;                                                                 00015440
1077     2              IF K>N THEN                                                             00015450
1078     2                 GO TO L11;                                                           00015460
1079     2              ALPHA=B(K);                                                             00015470
1080     2              GO TO TOUR;                                                             00015480
1081     2          PD_CALC:                                                                    00015490
                        DO JJ=ISIM(KK) TO ISIM(KK+1)-1;                                        00015500
1082     2    1            J=SIM(JJ);                                                          00015510
1083     2    1            XDUM=L(J)/(NTIE+1);                                                 00015520
1084     2    1            DO I=1 TO NTIE+1;                                                   00015530
1085     2    2               PD(TIECAR(I),J)=PD(TIECAR(I),J)+STPROB(B(K)+1)*XDUM;             00015540
1086     2    2            END;                                                                00015550
1087     2    1         END;                                 /*LOOP ON I, THEN JJ*/          00015560
1088     2          ALL_BUSY:                                                                   00015570
                        GO TO L22;                                                             00015580
1089     2          RETOUR_FINI:                                                                00015590
                        IF CAPACITY =0E0 THEN                                                  00015600
1090     2              PD=PD/(1E0-STPROB(N));                                                  00015610
                   /*COMPUTE WORKLOADS. THIS IS DONE BY SUMMING CERTAIN STATE PROBS IN        00015620
                   THE ORDER DETERMINED BY THE HYPERCUBE TOUR, AS INDICATED BY B(I), ...     00015630
                   # THE STRICT PERIODICITIES ASSOCIATED WITH EACH UNIT ARE EXPLOITED. */     00015640
1091     2              SCALE1=1;                            /*POINTER TO LAST "0" BEFORE     00015650
                                                            FIRST "1" FOR UNIT I*/             00015660
                   /*EACH UNIT WORKS THE TIME THAT A QUEUE EXISTS*/                           00015670
1092     2              WORKLOAD=BDPROB(M+2);                /*MATRIX OP;= 0 FOR 0-LINE       00015680
                                                            CAPACITY CASE*/                    00015690
1093     2              DO I=1 TO M;                                                            00015700
1094     2    1            IF I¬=M THEN                                                        00015710
1095     2    1               SCALE2=SCALE1*TWO;            /*NUMBER OF SUCCESSIVE ENTRIE*/  00015720
1096     2    1            SCALE3=TWO*SCALE2;               /*S HAVING '1' IN ITH DIGIT*/   00015730
1097     2    1            DO J=SCALE1 BY SCALE3 WHILE(J<N);/*SKIP TO NEXT SET OF STATES     00015740
                                                            WITH SUCCESSIVE '1''S*/            00015750
1098     2    2               DO K=1 TO SCALE2;             /*PICK UP SUCCESSIVE ENTRIES*/  00015760
1099     2    3                  WORKLOAD(I)=WORKLOAD(I)+STPROB(B(J+K)+1);                    00015770
1100     2    3               END;                          /*LOOP ON K*/                   00015780
1101     2    2            END;                             /*LOOP ON J*/                   00015790
1102     2    1            SCALE1=SCALE1*TWO;                                                  00015800
1103     2    1         END;                                 /*LOOP ON I*/                   00015810
                   /*CALCULATE TRAVEL TIMES*/                                                00015820
                   /*TQUE=MEAN TRAVEL TIME TO CALLS THAT INCUR A POSITIVE QUEUE DELAY*/      00015830
1104     2          T_CAL:                                                                      00015840
                        TQUE=0E0;                                                              00015850
1105     2              IF CAPACITY>0E0 THEN                                                    00015860
1106     2                 DO;                                                                  00015870
1107     2    1               DO J=1 TO R;                                                      00015880
1108     2    2                  DO K=1 TO R;                                                   00015890
1109     2    3                     TQUE=TQUE+L(J)*L(K)*TR(J,K);                              00015900
                   /***ABOVE ONLY APPROX FOR CASE OF UNEQ SERV TIMES*/                       00015910
1110     2    3                  END;                                                          00015920
1111     2    2               END;                          /*LOOP ON K, THEN J*/           00015930
1112     2    1            END;                             /*TQUE NOW CALCULATED FOR        00015940
                                                            INFINITE CAPACITY CASE*/           00015950
1113     2              TAV=TQUE*SATPROB;                    /*NOTE THIS IS ZERO FOR          00015960
```

```
STMT LEVEL NEST
                                                        ZERO-CAPACITY CASE*/                    00015970
1114    2           DO I=1 TO M;                                                                00015980
1115    2    1         SOM=0E0;                                                                 00015990
1116    2    1         DENOM=0E0;                                                               00016000
1117    2    1         DO J=1 TO R;                                                             00016010
1118    2    2           SOM=SOM+PD(I,J)*T(I,J);                                                00016020
1119    2    2           DENOM=DENOM+PD(I,J);                                                   00016030
1120    2    2         END;                          /*LOOP ON J*/                              00016040
1121    2    1         TAV=TAV+SOM;                  /*COMPUTING COMMAND-WIDE MEAN              00016050
                                                     TRAVEL TIME*/                              00016060
                  /*THE FOLLOWING 2 INSTRUCTIONS ONLY APPROXIMATE THE MEAN TRAVEL TIME         00016070
                  FOR UNIT I FOR QUEUED CALLS*/                                               00016080
1122    2    1         SOM=SOM + TQUE*SATPROB/M;                                              00016090
1123    2    1         IF CAPACITY>0E0 THEN                                                   00016100
1124    2    1           DENOM=DENOM + SATPROB/M;                                             00016110
1125    2    1         DCAR(I)=DENOM;                /*=FRACTION OF ALL DISPATCHES            00016120
                                                     WHICH ARE TO UN I*/                        00016130
1126    2    1         TCAR(I)=SOM/DENOM;            /*COND MEAN TRAVEL TIME OF UNIT          00016140
                                                     I*/                                        00016150
1127    2    1       END;                            /*LOOP ON I*/                            00016160
1128    2           DO J=1 TO R;                    /*NOW COMPUTE COND MEAN TRAVEL             00016170
                                                     TIMES TC REP AREAS*/                       00016180
1129    2    1         SOM=0E0;                                                               00016190
1130    2    1         DENOM=0E0;                                                             00016200
1131    2    1         DO I=1 TO M;                                                           00016210
1132    2    2           SOM=SOM+PD(I,J)*T(I,J);                                              00016220
1133    2    2           DENOM=DENOM+PD(I,J);                                                 00016230
1134    2    2         END;                          /*LOOP ON I*/                            00016240
1135    2    1         IF CAPACITY>0E0 THEN                                                   00016250
1136    2    1           DO;                                                                  00016260
                  /*FOLLOWING INSTRUCTIONS INVOLVE NO APPROXIMATIONS;ASSUMG = SERV TMS*/      00016270
1137    2    2            DENOM=DENOM+L(J)*SATPROB;                                           00016280
1138    2    2            XDUM=0E0;                                                           00016290
1139    2    2            DO K=1 TO R;                                                        00016300
1140    2    3              XDUM=XDUM+L(K)*TR(K,J);                                           00016310
1141    2    3            END;                        /*LOOP ON K*/                          00016320
1142    2    2            SOM=SOM+XDUM*SATPROB*L(J);                                        00016330
1143    2    2          END;                          /*END CONDITIONAL FOR INFINITE           00016340
                                                     CAPACITY QUEUE CASE*/                      00016350
1144    2    1         DREP(J)=DENOM;                /*=FRACTION OF TOTAL DISPATCHES          00016360
                                                     FROM REP AR J*/                            00016370
1145    2    1         TREP(J)=SOM/DENOM;            /*CONDITIONAL MEAN TRAVEL TIME           00016380
                                                     TO REP A J*/                               00016390
1146    2    1       END;                            /*LOOP ON J*/                            00016400
1147    2           DO I=1 TO M;                    /*COMPUTE CONDITIONAL MEAN                 00016410
                                                     TRAVEL TIMES TO SECTORS*/                  00016420
1148    2    1         SOM=0E0;                                                               00016430
1149    2    1         DENOM=0E0;                                                             00016440
1150    2    1         DO J=1 TO R;                                                           00016450
1151    2    2           IF SEC(I,J)^=0E0 THEN                                                00016460
1152    2    2             DO;                                                                00016470
1153    2    3               SOM=SOM+TREP(J)*DREP(J);                                         00016480
1154    2    3               DENOM=DENOM+DREP(J);                                             00016490
1155    2    3             END;                                                               00016500
1156    2    2           END;                          /*LOOP ON J*/                          00016510
```

```
 STMT LEVEL NEST

1157     2    1            CSEC(I)=DENOM;                  /*=FRACTION OF ALL DISPATCHES     00016520
                                                           FROM SECTOR I*/                   00016530
1158     2    1            TSEC(I)=SOM/DENOM;              /*DESIRED COND TRAVEL TIME*/      00016540
1159     2    1         END;                               /*LOOP ON I*/                     00016550
                    /*MEAN WORKLOAD AND UNBALANCE*/                                         00016560
1160     2              MWL=0E0;                                                             00016570
1161     2              SOM=0E0;                                                             00016580
1162     2              XDUM=INFINITY;                                                       00016590
1163     2              UBWL=0E0;                       /*FIX LATER WITH MAX AND MIN*/     00016600
1164     2              DO I=1 TO M;                                                         00016610
1165     2    1            MWL=MWL+WORKLOAD(I);                                           00016620
1166     2    1            SOM=SOM+WORKLOAD(I)*WORKLOAD(I);                               00016630
1167     2    1            IF WORKLOAD(I)<XDUM THEN                                       00016640
1168     2    1               XDUM=WORKLOAD(I);                                           00016650
1169     2    1            IF WORKLOAD(I)>UBWL THEN                                       00016660
1170     2    1               UBWL=WORKLOAD(I);                                           00016670
1171     2    1         END;                               /*LOOP ON I*/                     00016680
1172     2              MWL=MWL/M;                                                           00016690
1173     2              VAR=(SOM-M*MWL*MWL)/(M-1);                                           00016700
1174     2              SWL=SQRT(VAR);                                                       00016710
1175     2              UBWL=UBWL - XDUM;                                                    00016720
                    /*AGGREGATE INPUT CALLS*/                                             00016730
1176     2              INPUT=0E0;                                                           00016740
1177     2              DO I=1 TO M;                                                         00016750
1178     2    1            DO J=1 TO R;                                                   00016760
1179     2    2               IF SEC(I,J)¬=0E0 THEN                                       00016770
1180     2    2                  INPUT(I)=INPUT(I)+L(J);                                  00016780
1181     2    2            END;                                                           00016790
1182     2    1         END;                               /*LOOP ON J, THEN I*/             00016800
                    /*FIND % DISPATCHES THAT ARE INTERSECTOR*/                            00016810
                    /*LOOP BY RESPONSE UNIT IN ONE CASE, BY DISTRICT IN ANOTHER*/         00016820
1183     2              DO I=1 TO M;                                                         00016830
1184     2    1            PINSEC(I)=0E0;                                                 00016840
1185     2    1            PINT(I)=0E0;                                                   00016850
                    /*PINSEC(I)=FRAC OF ALL DISPATCHES THAT ARE INTO DISTRICT I AND ARE   00016860
                    INTRA DISTRICT PINT(I)=FRAC OF ALL DISPATCHES THAT (1)ORIGINATE FROM  00016870
                    ATOMS CONTAINED IN DISTRICT I AND (2) ASSIGN UNIT I */              00016880
1186     2    1            J=0;                                                           00016890
1187     2    1    TROUBLE:                                                               00016900
                       J=J+1;                                                            00016910
1188     2    1            IF J>R THEN                                                    00016920
1189     2    1               GO TO TROUBLE_OUT;                                          00016930
1190     2    1            IF SEC(I,J)¬=0E0 THEN                                          00016940
1191     2    1               DO;                          /*IS ATOM J IN DIST I?*/      00016950
1192     2    2                  PINT(I)=PINT(I)+PD(I,J)+PD2(I,J);                       00016960
1193     2    2                  DO K=1 TO M;                                            00016970
                    /*DOES UNIT K'S DIST OVERLAP WITH THIS PART OF DIST I? IF SO, ALL     00016980
                    DISPS OF UNIT K TO ATOM J MUST BE CONSIDERED INTRADISTRICT FOR      00016990
                    DISTRICT I*/                                                        00017000
1194     2    3                     IF SEC(K,J)¬=0E0 THEN                                00017010
1195     2    3                        PINSEC(I)=PINSEC(I)+PD(K,J) + PD2(K,J);           00017020
1196     2    3                  END;                                                    00017030
1197     2    2               END;                        /*LOOP ON K AND END OF        00017040
                                                           CONDITIONAL*/                  00017050
1198     2    1            GO TO TROUBLE;                                                 00017060
```

```
 STMT LEVEL NEST

1199     2    1    TROUBLE_OUT:                                                                    00017070
                       END;                                  /*LOOP ON I*/                        00017080
1200     2                SOM=0E0;                                                                 00017090
1201     2                DO I=1 TO M;                                                             00017100
1202     2    1              ISDO(I)=1E0-PINT(I)/DCAR(I);    /*INTERSEC DISPATCHES FOR UNIT      00017110
                                                             I*/                                  00017120
1203     2    1              ISDI(I)=1E0-PINSEC(I)/DSEC(I);/*DISPATCHES INTO SECTOR I */         00017130
1204     2    1              SOM=SOM+PINT(I);               /*THAT ARE INTERSECTOR*/             00017140
1205     2    1           END;                              /*LOOP ON I*/                        00017150
1206     2                MISD=1E0-SOM;                     /*TOTAL FRAC OF ALL DISP THAT        00017160
                                                             ARE INTRASECTOR*/                    00017170
                       /*COMPUTE PATROL FREQUENCIES*/                                           00017180
1207     2                IF PAT_FLAG=1 THEN DO;                                                   00017190
1209     2    1           PATFREQ(R+1)=0E0;     /*REGION-WIDE PATROL FREQUENCY*/                   00017200
1210     2    1           DO J=1 TO R;                                                             00017210
1211     2    2           YDUM=0E0;                                                                00017220
1212     2    2           DO I=1 TO M;                                                             00017230
1213     2    3           IF SEC(I,J)>0E0 THEN                                                     00017240
1214     2    3               YDUM=YDUM+SEC(I,J)*(1.-WORKLOAD(I));                                 00017250
1215     2    3           END;     /*LOOP ON I*/                                                  00017260
1216     2    2           IF STMI(J)>0E0 THEN                                                      00017261
1217     2    2           PATFREQ(J)=YDUM*PAT_SPEED/STMI(J);                                       00017262
1218     2    2           ELSE                                                                     00017263
1218     2    2           PATFREQ(J)=0E0;                                                          00017264
1219     2    2           PATFREQ(R+1)=PATFREQ(R+1)+L(J)*PATFREQ(J);                               00017265
1220     2    2           END;    /*LOOP ON J*/                                                  00017266
1221     2    1           END;     /*END PATROL CONDITIONAL*/                                     00017267
                       /*CALCULATE DIFFERENCE AND % DIFF FOR OUTPUT*/                            00017270
1222     2                DO I=1 TO M;                                                             00017280
1223     2    1              DWL(I)=WORKLOAD(I)-MWL;        /*WORKLOAD DIFFERENCE FROM MEAN*/   00017290
1224     2    1              PWL(I)=WORKLOAD(I)/MWL;        /*% OF MAEN*/                       00017300
1225     2    1              CISDO(I)=ISDO(I)-MISD;         /*DIFF FROM MEAN OF INTERSEC        00017310
                                                             DISP FOR*/                           00017320
1226     2    1              PISDO(I)= ISDO(I)/MISD;        /*% OF MEAN **************UNIT       00017330
                                                             I*/                                  00017340
1227     2    1              CISDI(I)=ISDI(I)-MISD;         /*DIFF FROM MEAN OF INTERSEC        00017350
                                                             DISP FOR*/                           00017360
1228     2    1              PISDI(I)=ISDI(I)/MISD;         /*% OF                              00017370
                                                             MEAN**************DISTRICT I*/       00017380
1229     2    1              CTSEC(I)=TSEC(I)-TAV;          /*DIFF FROM MEAN OF MEAN TRAV       00017390
                                                             TM TO S I*/                          00017400
1230     2    1              PTSEC(I)=TSEC(I)/TAV;          /*% OF MEAN*/                       00017410
1231     2    1              DINPUT(I)=INPUT(I)-1E0/M;      /*SECTOR'S DIFF FROM MEAN           00017420
                                                             LAMBDA*/                             00017430
1232     2    1              PINPUT(I)= INPUT(I)*M;         /*% OF MEAN*/                       00017440
1233     2    1           END;                              /*LOOP ON I*/                        00017450
                       /* PRINT OUTPUT */                                                       00017460
1234     2                PUT                                                                      00017470
                         EDIT('NSF/RANN - HUD SPATIALLY DISTRIBUTED QUEUING MODEL OF AN') (00017480
                          A) PAGE;                                                               00017490
1235     2                PUT                                                                      00017500
                         EDIT('URBAN SERVICE SYSTEM:  COMPUTED PERFORMANCE MEASURES') (A)       00017510
                         SKIP;                                                                  00017520
1236     2                PUT EDIT('PROBLEM TITLE: ',TITLE)(A,A) SKIP;                             00017530
1237     2                IF ESTSTAT=1 | ESTSTAT=-2 THEN                                           00017540
```

```
 STMT LEVEL NEST

1238     2                  DO;                                                                    00017550
1239     2    1                PUT EDIT('***ITERATIVE APPROXIMATION METHOD USED***')(            00017560
                               COLUMN(5),A) SKIP(2);                                             00017570
1240     2    1                PUT EDIT('NUMBER OF ITERATIONS REQUIRED: ',ITERATE)(              00017580
                               COLUMN(5),A,A);                                                   00017590
1241     2    1             END;                                                                  00017600
1242     2               IF CAPACITY>0E0 THEN                                                     00017610
1243     2                  PUT                                                                   00017620
                           EDIT(                                                                 00017630
                    'UNLIMITED CAPACITY QUEUE WITH 1_ST-COME 1_ST-SERVED QUEUE DISCIPLINE'       00017640
                           )(                                                                    00017650
                           A) SKIP;                                                              00017660
1244     2               ELSE                                                                     00017670
1244     2                  PUT                                                                   00017680
                           EDIT(                                                                 00017690
                           'NO QUEUING ALLOWED, BACK-UP SYSTEM FOR OVERFLOWS ASSUMED')(A)        00017700
                           SKIP;                                                                 00017710
1245     2               IF CAPACITY>0E0 & MU(1)¬=INFINITY THEN                                   00017720
1246     2                  PUT                                                                   00017730
                           EDIT(                                                                 00017740
                    'IN-QUEUE TRAVEL TIMES  ONLY APPROXIMATE DUE TO UNEQUAL SERVICE TIMES'       00017750
                           )(                                                                    00017760
                           A) SKIP;                                                              00017770
1247     2               PUT EDIT('RUN NUMBER: ',RUNNUM)(A,F(2)) SKIP;                            00017780
1248     2               PUT EDIT(R_UNIT,'...TOTAL NUMBER OF = ',M)(A,X(2),A,F(2)) SKIP;          00017790
1249     2               PUT EDIT(ATOM,'...TOTAL NUMBER OF = ',R)(A,X(2),A,F(3)) SKIP;            00017800
1250     2               XDUM=RO-(NUM+1-RUNNUM)*IRO;                                              00017810
1251     2               PUT EDIT('AVERAGE SERVICE TIME= ',SERVTM,' MINUTES')(                    00017820
                         A,F(6,2),A) SKIP;                                                       00017830
1252     2               PUT                                                                      00017840
                         EDIT('AVERAGE NUMBER PER HOUR OF ',CFS,' = ',XDUM*60./SERVTM) (          00017850
                         A,A,A,F(7,3)) SKIP;                                                     00017860
1253     2               PUT                                                                      00017870
                         EDIT('AVERAGE NUMBER PER ',SERVTM,' MINUTES OF ',CFS,' = ',XDUM) (       00017880
                         A,F(6,2),A,A,A,F(7,3)) SKIP;                                            00017890
1254     2               IF PAT_FLAG=1 THEN                                                       00017900
1255     2                  PUT EDIT('SPEED OF PATROL= ',PAT_SPEED,' MPH')(A,F(6,2),A)            00017910
                           SKIP;                                                                 00017920
1256     2               PUT                                                                      00017930
                         EDIT(                                                                   00017940
                    'AVERAGE UTILIZATION FACTOR (IN THE CASE OF UNLIMITED LINE CAPACITY)= '      00017950
                         ,                                                                       00017960
                         XDUM/M)(A,F(7,3)) SKIP;                                                 00017970
1257     2               PUT EDIT('REGION-WIDE AVERAGE TRAVEL TIME= ',TAV,' MINUTES') (           00017980
                         A,F(7,3),A) SKIP(3);                                                    00017990
1258     2               IF CAPACITY>0E0 THEN                                                     00018000
1259     2                  PUT                                                                   00018010
                           EDIT('AVERAGE TRAVEL TIME FOR QUEUED CALLS= ',TQUE,' MINUTES')        00018020
                           (                                                                     00018030
                            A,F(7,3),A) SKIP(3);                                                 00018040
1260     2               PUT EDIT('PROBABILITY OF SATURATION= ',SATPROB)(A,F(7,5)) SKIP;          00018050
1261     2               PUT EDIT('REGION-WIDE AVERAGE WORKLOAD (% TIME BUSY)= ',MWL) (           00018060
                         A,F(7,5)) SKIP;                                                         00018070
1262     2               PUT EDIT('STANDARD DEVIATION OF WORKLOAD= ',SWL)(A,F(7,3)) SKIP;         00018080
1263     2               PUT EDIT('MAXIMUM WORKLOAD IMBALANCE= ',UBWL)(A,F(7,5)) SKIP;            00018090
```

```
STMT LEVEL NEST
1264     2          PUT                                                                    00018100
                    EDIT('FRACTION OF DISPATCHES THAT ARE INTER-',R_DIST,' = ',MISD) (00018110
                     A,A,A,F(7,5)) SKIP(3);                                              00018120
1265     2          IF PAT_FLAG=1 THEN                                                     00018130
1266     2             PUT                                                                 00018140
                      EDIT('REGION-WIDE AVERAGE PATROL FREQUENCY= ',                     00018150
                       PATFREQ(R+1),' PASSES PER HOUR')(                                 00018151
                       A,F(8,3),A) SKIP(3);                                              00018170
1267     2          PUT                                                                    00018180
                    EDIT('PERFORMANCE MEASURES THAT ARE SPECIFIC TO EACH ',R_UNIT) (    00018190
                     A,A) PAGE;                                                          00018200
1268     2          PUT EDIT('ID OF')(COLUMN(4),A) SKIP(2);                              00018210
1269     2          PUT EDIT(R_UNIT,'FRACTION OF')(COLUMN(4),A,COLUMN(34),A) SKIP;       00018220
1270     2          PUT EDIT('WORKLOAD','% OF','DISPATCHES','% OF','AVERAGE') (          00018230
                     COLUMN(16),A,COLUMN(28),A,COLUMN(34),A,COLUMN(50),A, COLUMN(58),   00018240
                     A) SKIP;                                                            00018250
1271     2          PUT                                                                    00018260
                    EDIT('NAME','NO  OF UNIT','MEAN','OUT OF ',R_DIST,                   00018270
                     ' MEAN  TRAVEL TIME')(                                              00018280
                     A,COLUMN(12),A,COLUMN(28),A,COLUMN(34),A,A,A) SKIP;                 00018290
1272     2          DO I=1 TO M;                                                           00018300
1273     2     1       PUT                                                              00018310
                       EDIT(NM_UNIT(I),NO_UNIT(I),WORKLOAD(I),PWL(I)*100.,ISDO(I),      00018320
                        PISDO(I)*100.,TCAR(I))(                                         00018330
                        A,COLUMN(11),F(3),COLUMN(17),F(5,3),CCLUMN(27),F(5,1),          00018340
                        COLUMN(37),F(5,4),COLUMN(49),F(5,1),COLUMN(58),F(7,3)) SKIP;    00018350
1274     2     1    END;                                  /*LOOP ON I*/                  00018360
1275     2          PUT                                                                    00018370
                    EDIT('PERFORMANCE MEASURES THAT ARE SPECIFIC TO EACH ',R_DIST) (     00018380
                     A,A) SKIP(5);                                                       00018390
1276     2          PUT EDIT('ID OF')(COLUMN(4),A) SKIP(2);                              00018400
1277     2          PUT EDIT(R_DIST,'FRACTION OF')(COLUMN(3),A,COLUMN(34),A) SKIP;       00018410
1278     2          PUT EDIT('WORKLOAD','% OF','DISPATCHES','% OF','AVERAGE') (          00018420
                     COLUMN(16),A,COLUMN(28),A,COLUMN(34),A,COLUMN(50),A,COLUMN(58),    00018430
                    A);                                                                 00018440
1279     2          PUT                                                                    00018450
                    EDIT('NAME','NO  OF ',R_DIST,' MEAN  INTER-',R_DIST,                 00018460
                     'MEAN  TRAVEL TIME')(A,COLUMN(12),A,A,A,A,COLUMN(50),A) SKIP;       00018470
1280     2          DO I=1 TO M;                                                           00018480
1281     2     1       PUT                                                              00018490
                       EDIT(NM_DIST(I),NO_DIST(I),INPUT(I)*XDUM,PINPUT(I)*100.,         00018500
                        ISDI(I), 100.*ISDI(I)/MISD,TSEC(I))(                            00018510
                        A,COLUMN(11),F(3),COLUMN(17),F(5,3), COLUMN(27),F(5,1),         00018520
                        COLUMN(37),F(5,4),COLUMN(49),F(5,1),COLUMN(58), F(7,3)) SKIP;   00018530
1282     2     1    END;                                  /*LOOP ON I*/                  00018540
1283     2          IF PRNT_AT_FLAG=1 THEN                                                 00018550
1284     2             GO TO AT_SKIP;                                                      00018560
1285     2          PUT                                                                    00018570
                    EDIT('PERFORMANCE MEASURES THAT ARE SPECIFIC TO EACH ',ATOM) (A,A)00018580
                                                                                        00018590
                    PAGE;                                                               00018600
1286     2          PUT EDIT('ID #      WORKLOAD','AVE   FRACTION OF ',CFS) (            00018610
                     A,COLUMN(23),A,A) SKIP(2);                                         00018620
1287     2          IDUM=29+MAX(31,5*M);                                                   00018630
1288     2          IF PAT_FLAG=1 THEN                                                     00018640
```

```
 STMT LEVEL NEST

1289     2                PUT EDIT('FREQUENCY OF')(COLUMN(IDUM),A) SKIP(0);                00018650
1290     2             PUT EDIT(ATOM,'OF','TRAV  FROM ',ATOM)(                             00018660
                        A,COLUMN(13),A,COLUMN(23),A,A) SKIP;                             00018670
1291     2             IF PAT_FLAG=1 THEN                                                  00018680
1292     2                PUT EDIT('PREVENTIVE PATROL')(COLUMN(IDUM),A) SKIP(0);           00018690
1293     2             PUT EDIT(ATOM,'TIME  SERVICED BY UNIT NUMBER:')(                    00018700
                        COLUMN(12),A, COLUMN(23),A) SKIP;                                00018710
1294     2             IF PAT_FLAG=1 THEN                                                  00018720
1295     2                PUT EDIT('PASSINGS(#/HOUR)')(COLUMN(IDUM),A) SKIP(0);            00018730
1296     2             PUT EDIT('(#CALLS/100HR)',(NO_UNIT(I) DO I=1 TO M)) (               00018740
                        COLUMN(9),A,COLUMN(29),(M)(F(3),X(2))) SKIP;                     00018750
1297     2             XDUM=XDUM*6000./SERVTM;                                             00018760
1298     2             DO J=1 TO R;                                                        00018770
1299     2     1          YDUM=L(J);                                                       00018780
1300     2     1          IF YDUM=0E0 THEN                                                 00018790
1301     2     1             YDUM=INFINITY;                                                00018800
1302     2     1          K=A_MAP(J);                                                      00018810
1303     2     1          PUT                                                              00018820
                          EDIT(K,L(J)*XDUM,TREP(J),(                                     00018830
                           ((PD1(I,J)+PD2(I,J))/YDUM) DO I=1 TO M)) (                     00018840
                           COLUMN(3),F(3),COLUMN(11),F(8,2),COLUMN(20),F(7,3),           00018850
                          COLUMN(28), (M)(F(4,2),X(1))) SKIP;                            00018860
1304     2     1          IF PAT_FLAG=1 THEN                                               00018870
1305     2     1           PUT EDIT(PATFREQ(J))(COLUMN(IDUM+2),F(6,2)) SKIP(0);           00018871
1306     2     1       END;                                  /*LOOP ON J*/                00019000
1307     2          AT_SKIP:                                                              00019010
                       ESTSTAT=-ESTSTAT;                                                 00019020
1308     2             IF ESTSTAT=-2 THEN                                                  00019030
1309     2                GO TO APPROX;                                                    00019040
1310     2             GO TO OUTLOOP;                                                      00019050
1311     2          OUTLOOP_FINI:                                                         00019060
                       FREE PD,PD2,MWL,MISD,TAV,VAR,SWL,UBWL,DINPUT,PINPUT,INPUT,        00019070
                       WORKLOAD,TCAR,TSEC,TREP,TQUE, DCAR,DSEC,DREP,DWL,PWL,DISDO,PISDO, 00019080
                       DISDI,PISDI,DTSEC,PTSEC,ISDO,ISDI,PINSEC,MOX,MON,PINT,SATPROB,    00019090
                       GWKLD,PATFREQ;                                                    00019100
1312     2             IF DEBUG>0 THEN                                                     00019110
1313     2                DO;                                                              00019120
1314     2     1             PUT LIST('TIME AT COMPLETION OF RUN') SKIP(5);               00019130
1315     2     1             CALL PRNT_TIME;                                              00019140
1316     2     1          END;                                                            00019150
1317     2             GO TO K_RETURN;                                                     00019160
1318     2          END CALCVAL;                                                          00019170
1319     1          PRNT_TIME:                                                            00019180
                       PROC;                                                             00019190
1320     2          DCL                                                                   00019200
                          TIME BUILTIN;                                                  00019210
1321     2          DCL                                                                   00019220
                          ATIME CHAR(9),                                                 00019230
                          HR CHAR(2),                                                    00019240
                          MN CHAR(2),                                                    00019250
                          SC CHAR(2),                                                    00019260
                          MLS CHAR(3);                                                   00019270
1322     2             ATIME=TIME;                                                         00019280
1323     2             HR=SUBSTR(ATIME,1,2);                                               00019290
1324     2             MN=SUBSTR(ATIME,3,2);                                               00019300
```

```
 STMT LEVEL NEST

1325     2              SC=SUBSTR(ATIME,5,2);                                          00019310
1326     2              MLS=SUBSTR(ATIME,7,3);                                         00019320
1327     2              PUT                                                            00019330
                        EDIT('CURRENT TIME= ',HR,' HR ',MN,' MIN ',SC, ' SEC ',MLS,   00019340
                         ' MILLISEC')(A,A,A,A,A,A,A,A,A) SKIP;                         00019350
1328     2              PUT SKIP(2);                                                   00019360
1329     2            END PRNT_TIME;                                                   00019370
1330     1          ERR:                                                             00019380
                      PROC(I);                                                       00019390
1331     2            DCL                                                            00019400
                        I FIXED DECIMAL(2);                                          00019410
1332     2              IF I=1 THEN                                                  00019420
1333     2                 PUT LIST('TOO MANY UNITS FOR EXACT MODEL') SKIP;          00019430
1334     2              IF I=2 THEN                                                  00019440
1335     2                 PUT LIST('TOO MANY GEOGRAPHICAL ATOMS') SKIP;             00019450
1336     2              IF I=3 THEN                                                  00019460
1337     2                 PUT LIST('INCORRECT SPECIFICATION FOR ESTSTAT') SKIP;     00019470
1338     2              IF I=4 THEN                                                  00019480
1339     2                 PUT LIST('ORDERING LOOP ERROR') SKIP;                     00019490
1340     2              IF I=5 THEN                                                  00019500
1341     2                 PUT LIST('SIMILARITY SET ERROR') SKIP;                    00019510
1342     2              IF I=6 THEN                                                  00019520
1343     2                 PUT LIST('QUEUE SATURATED') SKIP;                         00019530
1344     2              PUT LIST('RUN TERMINATED') SKIP;                             00019540
1345     2              STOP;                                                        00019550
1346     2            END ERR;                                                       00019560
1347     1          END HYPCUBE;                                                     00019570
```

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
31	*********	A_MAP	(*)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 68,124,148,255,266,470,472,483,511,596,1302
		ABS	GENERIC,BUILT-IN FUNCTION 218,218,315,315,348,348,366,366,384,384,677,765,908,998,1041,1052
31	*********	ADDRESS	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 654,655,655,728,739,739,740,752,752,753
1088		ALL_BUSY	STATEMENT LABEL CONSTANT 1069
31		ALPHA	AUTOMATIC,ALIGNED,STRING(15),BIT 641,649,724,725,732,733,745,746,813,813,859,867,1079
31		ALT_RETURN	AUTOMATIC,LABEL 631,883,1069
911		APPROX	STATEMENT LABEL CONSTANT 1309
1307		AT_SKIP	STATEMENT LABEL CONSTANT 1284
1321		ATIME	AUTOMATIC,UNALIGNED,STRING(9),CHARACTER 1322,1323,1324,1325,1326
31		ATOM	AUTOMATIC,UNALIGNED,STRING(8),CHARACTER 60,106,120,146,253,464,469,469,477,478,503,505,506,590,1249,1285 1290,1290,1293
31	*********	B	(*)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 601,602,608,608,613,613,614,614,615,615,641,695,696,702,702,710,724 812,816,816,1079,1085,1099
31		BDPROB	(*)AUTOMATIC,ALIGNED,DECIMAL,FLOAT(DOUBLE) 659,660,663,663,664,670,670,671,674,674,676,679,712,759,761,786,786 787,893,895,900,914,918,927,931,942,942,1092
31		BETA	AUTOMATIC,ALIGNED,STRING(15),BIT 649,650,651,652,732,734,735,745,747,748
31		C	(*,*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 281,295,296,363,366,370,384,387,412,413,425,439,448,457,511,535,535 537,537,574,574,576,576,580,580,805,805,818,818,832,832,840,840,865 865,949,949,960,960,1015,1015,1020,1020
888		CALCOUT	STATEMENT LABEL CONSTANT 97
268		CALCSFC	ENTRY,DECIMAL,FLOAT(SINGLE)

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
			172
28		CALCVAL	ENTRY,DECIMAL,FLOAT(SINGLE) 25
31		CAPACITY	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 40,238,666,706,757,776,897,913,917,925,941,985,988,1089,1105,1123 1135,1242,1245,1258
31		CBIT	AUTOMATIC,ALIGNED,STRING(1),BIT 733,734,736,746,747,749
91		CCOUT	STATEMENT LABEL CONSTANT 81
31		CFS	AUTOMATIC,UNALIGNED,STRING(18),CHARACTER 59,146,1252,1253,1286
31		COMM	AUTOMATIC,UNALIGNED,STRING(10),CHARACTER 99,100,108,109,129,137,139,141,151,162,164,174,188,193,199,204,209 222,230,235,237,239,241,243,245,247,249,259,261,263,265,433,434,443 452
31		CONV_METRIC	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 719,721,722,766,766,770
31		C_SCALE	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 43,214,215,236
32		CCAP	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1125,1202,1311
2	*********	DEBUG	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 3,4,5,29,71,76,83,92,514,552,714,769,791,934,1312
31		DENOM	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(DOUBLE) 661,664,664,671,671,674,778,779,782,783,786,1116,1119,1119,1124,1124 1125,1126,1130,1133,1133,1137,1137,1144,1145,1149,1154,1154,1157 1158
756		DIAG_DIV	STATEMENT LABEL CONSTANT 743
32		DINPUT	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1231,1311
31	*********	DIS_METHOD	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 55,240,242,244,246,294,297,326,379,392,485,487,489,491,502
32		DISDI	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1227,1311
32		DISDO	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1225,1311

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
32		OREP	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1144,1153,1154,1311
32		OSEC	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1157,1203,1311
32		OTSEC	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1229,1311
31		OUMBIT	AUTOMATIC,ALIGNED,STRING(15),BIT 652,653,813,814
32		OWL	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1223,1311
1036		END_I	STATEMENT LABEL CONSTANT 1013
587		END_SIM_LP	STATEMENT LABEL CONSTANT 562
547		END_SWITCH	STATEMENT LABEL CONSTANT 542
981		END_W_LOOP	STATEMENT LABEL CONSTANT 957
31		EPSILON	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 38,721
1330		ERR	ENTRY,DECIMAL,FLOAT(SINGLE) 11,13,17,550,589,669
2	*********	ESTSTAT	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 4,10,16,16,19,22,599,680,908,908,909,1237,1237,1307,1307,1308
31		FACT	(16)AUTOMATIC,ALIGNED,INITIAL,DECIMAL,FIXED(13,0) 618,618,618,922,924,931
27		FINISH	STATEMENT LABEL CONSTANT
31	*********	FLAG	(*)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 558,561,567,583
31	*********	FRST_PREF	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 54,248,419,493
32		GWKLD	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,936,956,966,978,1052,1058,1064,1311
		H	AUTOMATIC,ALIGNED,DECIMAL,FLOAT(SINGLE) 166,168,176,179,436,437,445,446,454,455
657		HERE_IF_ONES	STATEMENT LABEL CONSTANT 631

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
1321		HR	AUTOMATIC,UNALIGNED,STRING(2),CHARACTER 1323,1327
1		HYPCUBE	ENTRY,DECIMAL,FLOAT(SINGLE)
1331		I	PARAMETER,ALIGNED,DECIMAL,FIXED(2,0) 1330,1332,1334,1336,1338,1340,1342
269	*********	I	PARAMETER,ALIGNED,BINARY,FIXED(15,0) 268,272,276,277
31	*********	I	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 61,62,62,63,63,64,65,67,68,68,121,121,122,122,124,124,155,156,156 156,166,167,170,172,176,177,181,184,186,186,211,212,212,216,218,218 218,225,227,227,282,286,290,299,303,305,306,309,310,315,315,322,328 333,342,343,348,348,355,358,363,366,366,366,368,370,381,384,384,384 386,387,394,399,403,404,412,413,422,423,425,426,436,439,445,448,454 457,470,470,472,472,480,480,481,481,483,483,508,508,509,509,511,511 522,523,523,527,528,528,532,549,560,561,565,566,570,574,574,574,574 592,593,593,596,607,608,608,617,618,618,618,625,626,627,648,650,652 662,663,663,663,664,673,674,674,701,702,702,710,711,712,725,726,727 728,728,737,737,740,740,750,750,753,753,756,756,756,757,763,765,804 805,808,808,823,824,826,827,844,845,847,850,858,859,861,862,864,901 902,904,921,922,922,922,947,948,949,949,954,954,954,955,955,955,955 956,959,966,973,973,973,974,974,974,978,980,980,1008,1009,1010,1012 1015,1016,1019,1028,1028,1031,1034,1034,1051,1052,1052,1084,1085 1085,1093,1094,1099,1099,1114,1118,1118,1119,1125,1126,1131,1132 1132,1133,1147,1151,1157,1158,1164,1165,1166,1166,1167,1168,1169 1170,1177,1179,1180,1180,1183,1184,1185,1190,1192,1192,1192,1192 1195,1195,1201,1202,1202,1202,1203,1203,1203,1204,1212,1213,1214 1214,1222,1223,1223,1224,1224,1225,1225,1226,1226,1227,1227,1228 1228,1229,1229,1230,1230,1231,1231,1232,1232,1272,1273,1273,1273 1273,1273,1273,1273,1280,1281,1281,1281,1281,1281,1281,1281,1296 1296,1303,1303,1303
31		I_RETURN	AUTOMATIC,LABEL 630,806,810,830,836,855,876,879,887,1068
31	*********	I_T_ORD	(*,*)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 523,528,528,535,535,537,537,539,539,543,544,544,545,570,570,574,574 576,576,580,580,801,805,805,808,824,859,862,865,867,870,949,949,954 954,955,955,955,956,960,960,966,973,973,974,974,978,1009,1015,1020 1020,1028,1031
31	*********	IDUM	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 543,545,593,594,596,596,626,627,711,712,712,720,723,771,772,773,937 944,953,972,983,1063,1287,1289,1292,1295,1305
31		INFINITY	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 35,53,238,293,296,313,346,690,729,760,784,1162,1245,1301
74		INI	STATEMENT LABEL CONSTANT 70

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
514		INITIALIZE	STATEMENT LABEL CONSTANT 78
32		INPUT	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1176,1180,1180,1231,1232,1281,1311
31		IRO	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 89,113,133,687,911,1250
32		ISDI	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1203,1227,1228,1281,1281,1311
32		ISDO	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1202,1225,1226,1273,1311
31	*********	ISIM	(*)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 559,564,564,565,584,585,585,588,593,593,596,636,644,644,1074,1081 1081
31	*********	ITERATE	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 721,775,775,794,939,1059,1059,1240
269	*********	J	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 271,272,273,275,276,277,277
31	*********	J	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 116,117,118,121,122,124,169,170,180,181,184,217,218,218,218,254,255 255,283,288,290,302,303,305,305,306,306,314,315,315,319,332,333,335 335,336,336,337,347,348,348,352,359,361,363,366,370,382,384,384,384 387,387,398,399,403,403,404,421,423,425,438,439,447,448,456,457,471 472,472,519,520,528,528,535,535,535,535,537,537,537,537,539,539,543 544,544,545,566,567,570,576,576,576,576,580,580,580,580,583,584,596 596,604,636,698,702,709,710,723,724,801,805,805,805,805,808,818,818 824,832,832,840,840,859,862,865,865,865,867,870,903,904,904,919,921 922,924,924,924,928,928,943,945,946,949,949,949,949,954,954,955,955 955,955,955,956,960,960,960,960,966,973,973,974,974,974,974,978,990 991,994,995,996,1004,1009,1012,1012,1015,1015,1015,1016,1016,1020 1020,1020,1020,1028,1028,1031,1031,1074,1082,1083,1085,1085,1097 1097,1099,1107,1109,1109,1117,1118,1118,1119,1128,1132,1132,1133 1137,1140,1142,1144,1145,1150,1151,1153,1153,1154,1178,1179,1180 1186,1187,1187,1188,1190,1192,1192,1194,1195,1195,1210,1213,1214 1216,1217,1217,1218,1219,1219,1298,1299,1302,1303,1303,1303,1303 1305
586		J_LOOP	STATEMENT LABEL CONSTANT 568,571,577,581
31	*********	JJ	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 368,370,386,387,400,401,403,403,404,409,410,412,413,644,645,847,848 848,864,865,867,870,1009,1012,1012,1015,1016,1016,1019,1020,1020 1022,1025,1028,1031,1081,1082
31	*********	K	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 123,124,124,147,148,148,168,179,285,286,288,319,322,352,355,360,361 366,366,369,372,395,401,410,437,446,455,465,466,467,470,472,482,483

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		483,510,511,511,534,535,535,537,537,539,539,543,544,544,545,569,570 570,573,574,574,576,576,580,580,612,613,613,614,614,615,615,621,622 622,622,637,638,638,639,641,691,692,692,731,733,734,737,744,746,747 798,812,816,816,959,960,960,962,965,966,970,973,973,974,974,977,978 993,1001,1001,1002,1075,1076,1076,1077,1079,1085,1098,1099,1108,1109 1109,1139,1140,1140,1193,1194,1195,1195,1302,1303
582	K_LOOP	STATEMENT LABEL CONSTANT 578
31	K_RETURN	AUTOMATIC,LABEL 70,74,82,91,127,135,600,658,681,797,1317
31 *********	KB	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 651,654,654,735,737,740,748,750,753
31 *********	KC	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 653,654,814,815
31 *********	KK	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 118,119,121,122,124,369,370,467,468,470,472,632,633,633,634,636,644 644,971,973,973,974,974,1027,1028,1070,1071,1071,1072,1074,1081,1081
31	L	(*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 143,144,145,145,148,273,277,335,336,337,403,404,645,904,945,946,991 995,996,1004,1083,1109,1109,1137,1140,1142,1180,1219,1299,1303
633	L1	STATEMENT LABEL CONSTANT 640
1071	L11	STATEMENT LABEL CONSTANT 1078
638	L2	STATEMENT LABEL CONSTANT 657
1076	L22	STATEMENT LABEL CONSTANT 1088
108	LAAP	STATEMENT LABEL CONSTANT 267
109	LAAP2	STATEMENT LABEL CONSTANT 107
31	LASTVER	AUTOMATIC,ALIGNED,STRING(15),BIT 812,813,813
	LOG2	GENERIC,BUILT-IN FUNCTION 815
4	LOOP	STATEMENT LABEL CONSTANT 26
2 *********	M	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
			4,10,20,23,24,31,31,31,31,31,31,31,31,31,31,31,31,31,31,31,61,104 105,116,117,119,121,122,124,155,159,282,293,296,299,328,358,360,372 381,394,395,422,426,480,480,481,481,483,483,508,508,509,509,511,511 522,527,532,534,549,569,573,604,617,618,618,625,662,668,670,670,670 671,673,698,731,744,759,759,759,761,761,782,786,786,787,804,823,827 858,864,882,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32 32,32,32,32,895,900,901,911,914,918,919,921,922,922,924,924,927,931 942,942,945,947,948,959,962,1008,1010,1019,1022,1047,1051,1056,1092 1093,1094,1114,1122,1124,1131,1147,1164,1172,1173,1173,1177,1183 1193,1201,1212,1222,1231,1232,1248,1256,1272,1280,1287,1296,1296 1303,1303
31	*********	MAP	(*)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 620,622,622,654,692,695,696,702,702,707,707,728,756
31		MASK	(*)AUTOMATIC,ALIGNED,STRING(15),BIT 624,627,652
658		MATRET	STATEMENT LABEL CONSTANT 635
643		MATRIX_INPUT	STATEMENT LABEL CONSTANT 630
		MAX	GENERIC,BUILT-IN FUNCTION 766,1287
31	*********	MAXBOUND	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 37,721
31		MINVAL	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 39,763
32		MISD	CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1206,1225,1226,1227,1228,1264,1281,1311
1321		MLS	AUTOMATIC,UNALIGNED,STRING(3),CHARACTER 1326,1327
1321		MN	AUTOMATIC,UNALIGNED,STRING(2),CHARACTER 1324,1327
32		MON	CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,916,918,927,927,928,994,995,1004,1005,1005,1026,1028,1028,1031 1046,1047,1047,1048,1048,1049,1311
32		MOX	CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,945,946,954,955,956,956,969,978,978,987,989,995,996,1004,1041 1311
31		MU	(*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 53,153,156,158,158,159,160,160,690,696,702,729,737,760,784,899,902 1245
32		MWL	CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE)

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		888,912,914,914,936,942,1048,1160,1165,1165,1172,1172,1173,1173,1223 1224,1261,1311
2	********* N	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 18,20,23,23,31,31,31,31,31,38,39,612,621,639,691,707,707,709,723,757 759,759,761,782,787,1077,1090,1097
2	********* N_BD_STS	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 24,31,31,31
31	********* NCAR	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 815,818,820,824,832,835,840,845
31	********* NCHANGE	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 531,532,533,546,546
2	********* NCOEFF	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 21,23,31
31	NM_DIST	(*)AUTOMATIC,UNALIGNED,STRING(8),CHARACTER 65,1281
31	NM_UNIT	(*)AUTOMATIC,UNALIGNED,STRING(8),CHARACTER 64,121,156,480,508,1273
31	********* NO_DIST	(*)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 63,1281
744	NO_MULT	STATEMENT LABEL CONSTANT 730
357	NO_SEC	STATEMENT LABEL CONSTANT 341
31	********* NO_UNIT	(*)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 62,122,156,481,509,1273,1296
31	********* NPREF	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 800,826,858,861
31	********* NS	(*)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 618,712
31	********* NSELECT	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 801,802,818,820,822,832,840,854,862,863,865,882
31	********* NTIE	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 647,648,803,807,807,821,834,834,835,842,844,847,850,853,859,869,869 870,958,961,961,965,968,969,970,971,977,980,1018,1021,1021,1025,1027 1031,1034,1083,1084
2	********* NUM	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 4,14,15,31,31,80,891,911,1250
31	********* ONE	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0)

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
			33
890		CUTLOOP	STATEMENT LABEL CONSTANT 1310
1311		CUTLOOP_FINI	STATEMENT LABEL CONSTANT 892
433		CV	STATEMENT LABEL CONSTANT 441,450,459
31	*********	OVERRIDE	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 51,200,431
31	*********	PAT_FLAG	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 47,257,1207,1254,1265,1288,1291,1294,1304
31		PAT_SPEED	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 48,251,1217,1255
32		PATFREQ	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1209,1217,1218,1219,1219,1219,1266,1305,1311
32		PD	(*,*)CONTROLLED,ALIGNED,DECIMAL,FLOAT(DOUBLE) 888,938,955,955,974,974,986,986,994,1012,1012,1016,1016,1028,1031 1040,1067,1085,1085,1090,1090,1118,1119,1132,1133,1192,1195,1303 1311
32		PD2	(*,*)CONTROLLED,ALIGNED,DECIMAL,FLOAT(DOUBLE) 888,896,904,1192,1195,1303,1311
1067		PD_CAL	STATEMENT LABEL CONSTANT 910
1081		PD_CALC	STATEMENT LABEL CONSTANT 1068
32		PINPUT	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1232,1281,1311
32		PINSEC	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1184,1195,1195,1203,1311
32		PINT	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1185,1192,1192,1202,1204,1311
32		PISDI	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1228,1311
32		PISDO	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1226,1273,1311
31	*********	PRNT_AT_FLAG	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 49,264,1283

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
1319		PRNT_TIME	ENTRY,DECIMAL,FLOAT(SINGLE) 8,86,95,517,555,717,795,1315
31	*********	PRNT_TR	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 52,262,462
79		PROG_LOOP	STATEMENT LABEL CONSTANT 74,90
32		PTSEC	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1230,1311
32		PWL	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1224,1273,1311
31		QUFAC	(*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 915,924,928,928,932,932,933,935,954,955,973,974
2	*********	R	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 4,12,31,31,31,31,31,31,31,31,31,31,31,31,67,106,123,147,211 216,217,225,254,271,275,283,285,302,314,332,347,359,382,398,400,409 421,465,466,468,471,482,510,519,560,566,588,32,32,32,32,32,903,943 990,1041,1107,1108,1117,1128,1139,1150,1178,1188,1209,1210,1219,1219 1249,1266,1298
31		R_DIST	AUTOMATIC,UNALIGNED,STRING(8),CHARACTER 56,105,496,1264,1271,1275,1277,1279,1279
31		R_UNIT	AUTOMATIC,UNALIGNED,STRING(18),CHARACTER 57,104,115,120,154,476,478,495,504,506,1248,1267,1269
99		READATA	STATEMENT LABEL CONSTANT 73
994		RECYCLE	STATEMENT LABEL CONSTANT 1037
98		RET	STATEMENT LABEL CONSTANT 138
1089		RETOUR_FINI	STATEMENT LABEL CONSTANT 1073
31		RO	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 89,89,112,132,663,668,670,670,684,687,687,692,695,696,707,759,759 761,761,782,782,911,1250
31	*********	RR	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 557,563,563,564,564,565,584,585,585,588,592,634,1072
31	*********	RUNNUM	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 75,79,79,80,679,682,790,889,890,890,891,893,894,911,1247,1250
32		SATPROB	CONTROLLED,ALIGNED,DECIMAL,FLOAT(DOUBLE) 888,895,900,900,902,986,989,1113,1122,1124,1137,1142,1260,1311

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
1321		SC	AUTOMATIC,UNALIGNED,STRING(2),CHARACTER 1325,1327
31	*********	SCALE1	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 605,607,608,699,701,1091,1095,1097,1102,1102
31	*********	SCALE2	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 603,605,606,606,607,608,697,699,700,700,701,702,1095,1096,1098
31	*********	SCALE3	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 1096,1097
31		SEC	(*,*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 124,167,170,177,181,184,186,186,272,276,277,286,303,305,306,333,361 399,401,403,404,410,423,1151,1179,1190,1194,1213,1214
31		SERVTM	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 44,112,113,132,133,159,160,163,1251,1252,1253,1297
31	*********	SIM	(*)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 565,584,596,636,645,1074,1082
1039		SKIP_J	STATEMENT LABEL CONSTANT 992,1000
1038		SKIP_JJ	STATEMENT LABEL CONSTANT 1003
659		SOLV_EQ	STATEMENT LABEL CONSTANT 88
31		SOM	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 178,183,183,186,270,273,273,277,284,288,288,290,331,337,337,340,342 343,397,404,404,408,1115,1118,1118,1121,1122,1122,1126,1129,1132 1132,1142,1142,1145,1148,1153,1153,1158,1161,1166,1166,1173,1200 1204,1204,1206
		SQRT	GENERIC,BUILT-IN FUNCTION 227,1006,1041,1174
70		START	STATEMENT LABEL CONSTANT 91
31		STMI	(*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 46,252,255,1216,1217
31		STPROB	(*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 708,712,726,727,737,737,737,740,740,740,750,750,750,753,753,753,756 756,759,759,761,763,765,778,779,779,782,782,783,783,787,790,894,1085 1090,1099
		SUBSTR	GENERIC,BUILT-IN FUNCTION 650,733,734,746,747,859,867,1323,1324,1325,1326

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
		SUM	GENERIC,BUILT-IN FUNCTION 144,159,676,778,782,899,994,1040,1046
543		SWITCH	STATEMENT LABEL CONSTANT 536,540
32		SWL	CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,969,973,974,1174,1262,1311
		SYSIN	FILE,EXTERNAL 4,99,101,108,111,131,140,143,153,163,166,169,176,180,190,195,201,206 212,224,226,232,236,251,252,266,433,436,438,445,447,454,456
		SYSPRINT	FILE,EXTERNAL 7,30,72,77,85,94,102,103,104,105,106,115,120,121,122,124,134,146,148 154,156,253,255,464,469,470,472,476,477,478,479,480,481,483,486,488 490,492,495,496,499,501,503,504,505,506,507,508,509,511,516,554,590 591,596,678,716,770,793,794,935,1038,1042,1234,1235,1236,1239,1240 1243,1244,1246,1247,1248,1249,1251,1252,1253,1255,1256,1257,1259 1260,1261,1262,1263,1264,1266,1267,1268,1269,1270,1271,1273,1275 1276,1277,1278,1279,1281,1285,1286,1289,1290,1292,1293,1295,1296 1303,1305,1314,1327,1328,1333,1335,1337,1339,1341,1343,1344
31		T	(*,*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 290,293,295,483,1118,1132
1104		T_CAL	STATEMENT LABEL CONSTANT 1043
31		T_COST	AUTOMATIC,UNALIGNED,STRING(18),CHARACTER 58
31		T_RATE	(*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 629,655,655,684,684,688,688,692,695,696,702,702,707,707,740,753,756
32		TAV	CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1113,1121,1121,1229,1230,1257,1311
32		TCAR	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1126,1273,1311
31	*********	TFLAG	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 45,191,233,311,344,365,383,498,500,502
31	*********	TIECAR	(*)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 650,652,802,808,822,835,845,848,848,854,863,870,1085,1085
1320		TIME	BUILT-IN FUNCTION 1322
31		TITLE	AUTOMATIC,UNALIGNED,STRING(50),CHARACTER 140,1236
798		TOUR	STATEMENT LABEL CONSTANT 642,1080

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
32		TQUE	CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1104,1109,1109,1113,1122,1259,1311
31		TR	(*,*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 190,218,227,288,370,387,403,472,1109,1140
280		TRAVTM	ENTRY,DECIMAL,FLOAT(SINGLE) 114
32		TREP	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1145,1153,1303,1311
1187		TROUBLE	STATEMENT LABEL CONSTANT 1198
1199		TROUBLE_OUT	STATEMENT LABEL CONSTANT 1189
32		TSEC	(*)CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1158,1229,1230,1281,1311
31	*********	TWO	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 34,603,606,697,700,1095,1096,1102
32		UBWL	CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1050,1053,1054,1056,1163,1169,1170,1175,1175,1263,1311
89		UP_RO	STATEMENT LABEL CONSTANT 82
32		VAR	CONTROLLED,ALIGNED,BINARY,FLOAT(SINGLE) 888,1052,1053,1054,1173,1174,1311
31	*********	W	(*)AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 611,614,614,615,615,622,654,654,692,711,728
940		W_LOOP	STATEMENT LABEL CONSTANT 1060
943		W_LOOP2	STATEMENT LABEL CONSTANT 1065
958		W_TIES	STATEMENT LABEL CONSTANT 950
32		WORKLOAD	(*)CONTROLLED,ALIGNED,DECIMAL,FLOAT(DOUBLE) 888,940,942,942,954,954,955,973,973,1045,1045,1046,1049,1049,1052 1058,1064,1092,1099,1099,1165,1166,1166,1167,1168,1169,1170,1214 1223,1224,1273,1311
31		X	(*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 212,214,214,218,218,305,315,335,348,384
31		XBDPROB	(*,*)AUTOMATIC,ALIGNED,DECIMAL,FLOAT(DOUBLE)

DCL NO.		IDENTIFIER	ATTRIBUTES AND REFERENCES
			679,893
31		XDUM	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 111,112,131,132,144,145,180,181,182,183,226,227,286,287,288,300,305 305,309,313,316,318,329,335,335,342,346,349,351,396,403,403,408,408 413,454,457,643,645,645,647,647,655,676,677,687,688,726,765,765,766 902,904,911,912,918,918,918,922,931,942,945,996,998,998,999,1083 1085,1138,1140,1140,1142,1162,1167,1168,1175,1250,1252,1253,1256 1281,1297,1297,1303
31		XICM	(*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 342,348,355,366,369
31		XSPEED	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 41,195,196,196,197,201,202,202,218,227,315,348,366,384
31		XSTPROB	(*,*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 50,790,894
31		XUCM	(*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 309,315,322,366,368,384,386
31		Y	(*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 212,215,215,218,218,306,315,336,348,384
31		YDUM	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 111,113,131,133,224,227,301,306,306,310,315,316,318,330,336,336,343 348,349,351,895,902,920,922,922,924,931,932,964,966,966,968,968,973 974,974,995,996,1005,1005,1006,1006,1012,1016,1028,1040,1041,1211 1214,1214,1217,1299,1300,1301,1303
31		YICM	(*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 343,348,366
31		YSPEED	AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 42,197,206,207,207,218,227,315,348,366,384
31		YUCM	(*)AUTOMATIC,ALIGNED,BINARY,FLOAT(SINGLE) 310,315,366,384
31	*********	ZERO	AUTOMATIC,ALIGNED,BINARY,FIXED(15,0) 36,611,666

Appendix

ADDRESSES FOR FURTHER INFORMATION

1. For documentation of the Hypercube Queuing Model, copies of the program on cards or tape, or answers to questions about the program:

Dr. Jan Chaiken
The Rand Corporation
1700 Main Street
Santa Monica, California 90406
(213) 393-0411

Professor Richard Larson
Room 4--209
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
(617) 253-1358

2. Research sponsor:

U.S. Department of Housing and Urban Development

Alan Siegel, Director
Hartley Campbell Fitts, Program Manager

Office of Policy Development and Research
Community Development and Management
Research Division
451 Seventh Street, S.W.
Washington, D.C. 20410
(202) 755-6970

REFERENCES

1. Bates, Frank, and Mary L. Douglas, *Programming Language/One*, Prentice-Hall, Englewood Cliffs, New Jersey, 1967.

2. Larson, Richard C., *Hypercube Queuing Model: User's Manual*, The Rand Corporation, R-1688/2-HUD, 1975.

END