UC Berkeley
Working Papers

Title

Studies of Vehicle Collisions - A Documentation of the Simulation Codes: SMAC (Simulation Model of Automobile Collisions) [Diskette Included]

Permalink

https://escholarship.org/uc/item/5q26873h

Author

Chan, Ching-Yao

Publication Date

1998-08-01

eScholarship.org Powered by the California Digital Library
University of California

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Studies of Vehicle Collisions - A Documentation of the Simulation Codes: SMAC (Simulation Model of Automobile Collisions)

Ching-Yao Chan

California PATH Working Paper
UCB-ITS-PWP-98-16

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department Transportation, Federal Highway Administration.

The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California. This report does not constitute a standard, specification, or regulation.

Report for MOU 324

August 1998

ISSN 1055-1417

Studies of Vehicle Collisions – A Documentation of the Simulation Codes: SMAC (Simulation Model of Automobile Collisions)

Ching-Yao Chan
California PATH
Institute of Transportation Studies
University of California, Berkeley

ABSTRACT

This document describes part of the work conducted under MOU252 and MOU324, related to the studies of vehicle collisions in vehicle-following operations.

This working paper is a detailed documentation of a computer program that is the core element of the simulation tools for vehicle collision dynamics. The program, SMAC (Simulation Model of Automobile Collisions), and its PC-platform version EDSMAC have been used extensively in recent work at PATH to investigate the consequences of vehicle collisions and the effects of vehicle-following parameters on collisions.

A copy of the source codes of SMAC was obtained from the University of Michigan Transportation Research Institute (UMTRI). Revisions to the program were made to insert the option of exercising feedback control in collision situations. After these revisions, control algorithms can now be tested in crash scenarios to examine the feasibility and effectiveness of vehicle control in emergency conditions.

The structure of the program was outlined in this report with descriptions of the major subroutines. The added options for implementing user-specified steering inputs and feedback controllers were explained in details. An example input file for running the computer program was also provided to illustrate the formats and contents of input parameters. This working report provides a concise and essential documentation for the computer program.

KEY WORDS

Vehicle Collisions
Simulation of Vehicle Crashes
Vehicle Control in Collisions
Advanced Vehicle Control Systems

EXECUTIVE SUMMARY

This working paper is a detailed documentation of a computer program that is the core element of the simulation tools for vehicle collision dynamics. The program, SMAC (Simulation Model of Automobile Collisions), and its PC-platform version EDSMAC have been used extensively in previous studies at PATH to investigate the consequences of vehicle collisions and the effects of vehicle-following parameters on collisions.

SMAC analyzes the longitudinal and lateral movements of vehicles as well as the rotational motion about the vertical axis of vehicles on a horizontal plane. If a contact between vehicles is detected, the collision phase is analyzed. The external forces can be applied either at the tire/road interface or between the vehicles.

A copy of the source codes of SMAC was obtained from the University of Michigan Transportation Research Institute (UMTRI). The source code was written in FORTRAN. Revisions to the program, described in this paper, were made by the author to insert the option of exercising feedback control in collision situations. After these revisions, control algorithms can now be tested in crash scenarios to examine the feasibility and effectiveness of vehicle control in emergency conditions.

The structure of the program was outlined in this report with descriptions of the major subroutines. The added options for implementing user-specified steering inputs and feedback controllers were explained in details. An example input file for running the computer program was also provided to illustrate the formats and contents of input parameters.

With the completion of the revisions to the program, the simulation tool is now available for the studies of advanced vehicle control in collisions. This working report provides a concise and essential documentation for the computer program. It will facilitate the use of this program as a validation model or combine it with other simulation tools.

1.0 INTRODUCTION

This progress report is generated from the work conducted under MOU-324. In this project, which is an extension of MOU-252, a simulation program was used to study the phenomena of vehicle dynamics in vehicle-following collisions. As part of the proposed work, the source codes of this simulation program was obtained and modified so that additional features can be incorporated into the codes to serve the purposes of research activities.

In this report, the source codes are reviewed and documented for future references. A floppy disk containing the source codes is attached with this report.

2.0 BACKGROUND INFORMATION ON SMAC

The analysis of vehicle collisions in the work related to MOU-252 is conducted with a simulation program developed by Engineering Dynamics Corporation (EDC) for PC-DOS platforms. The software package, EDSMAC (Engineering Dynamics Corporation Simulation Model of Automobile Collisions), is used for the analysis of two-vehicle collisions. It is based on a program called SMAC [1-3] initially developed by Calspan Corporation for National Highway Traffic Safety Administration (NHTSA) and subsequently improved by EDC [4-7]. EDSMAC uses a set of assumed or estimated parameters, including vehicle and roadway properties to predict the outcome of a collision. Engineers have been using this simulation program to analyze vehicle dynamics and the damage resulting from crashes. Researchers have found that the program yields reasonable results with sound input data [8-13].

EDSMAC analyzes the longitudinal and lateral movements of vehicles as well as the rotational motion about the vertical axis of vehicles on a horizontal plane. If a contact between vehicles is detected, the collision phase is analyzed. The external forces can be applied either at the tire/road interface or between the vehicles.

In the collision phase of the simulation model, a force proportional to the amount of crush is exerted as the body of a vehicle is crushed. This is accomplished by dividing the vehicle's perimeter into equally spaced intervals. Each of these intervals forms a pie-shaped wedge having its focus at the center of gravity of the vehicle. The vehicle exterior is assumed to have homogeneous stiffness. By knowing where the vehicles are with respect to each other,

EDSMAC locates the wedges that are in contact and equalizes the force between them. The resulting summation of forces dictates the motion of each vehicle due to the collision. This process continues for each collision time increment until the vehicles are no longer in contact.

EDSMAC allows the direct entry of vehicle data by users or the selection of default values. The vehicles are categorized by their wheelbase lengths into several classes. Class I and II are small passenger cars while Classes III to V are medium to large cars. In this paper, default values of vehicle parameters provided by EDSMAC are used in the simulation during this study. [7]

Attempts were made at the beginning of MOU-324 to acquire the source codes of SMAC directly from NHTSA. Since the program was developed more than 20 years ago, efforts to track down and locate the source codes were not successful. Through a series of contacts, a copy of the source code was finally obtained from the University of Michigan Transportation Research Institute with the help of Charlie Compton and Joel MacWilliams of University of Michigan, Traffic Research Institute (UMTRI). The source code was written in FORTRAN with limited revisions by UMTRI researchers added to the original version.

3.0 STRUCTURE AND FLOW CHART OF SMAC

The SMAC program is separated into several main parts:

1. Input phase: This portion receives the information regarding the integration and output time steps, the vehicle parameters, state variables such as position and speed, steering and tire torque inputs from an input file.
2. Trajectory phase: A subroutine and associated functions calculate the trajectories of the subject vehicles while the vehicles are not in a collision.
3. Collision phase: The collision routines determine if the vehicles are in contact and calculate the contact forces and its direction, which is then used to determine the subsequent motions of the vehicles.
4. Output phase: The states of the vehicles are printed at specified intervals, and also plotted on the terminal display if desired.

The plotting routines in the output phase is removed from the current version since it was originally designed to work with a plotting library on a mainframe computer. The plotting features can be added back if needed for a particular type of operating systems and their graphic libraries.

The main sequence and flowchart of the program is depicted in Figure 1. The major subroutines and their related functions are grouped in the associated blocks.

3.1 Descriptions of Program Subroutines and Sequences

The main portion of the program is separated into several sections: Input, Trajectory, Collision, and Output, as indicated in Figure 1. This section describes the structure and the sequence that these subroutines are related.

3.1.1 Initiation - SMACPATH

The main portion of the original SMAC program is converted into a subroutine SMAC. A short initiation program, SMACPATH, is constructed as the main program. In SMACPATH, the files are open and ready for the program before SMAC is called.

3.1.2 Main Subroutine – SMAC

SMAC called several subroutines in the following sequence:

A. SMAC calls INPUT and CNSTNT to initialize constants and variables.
B. SMAC checks if there is contact between two vehicles.
C. SMAC calls RNGKT1 to integrate the motion of the vehicles for each time step.
D. SMAC calls ACCEL and SAVMAX to update the acceleration and the direction of impact.
E. SMAC calls OUTPUT if it is time to print.
F. SMAC checks the variables and simulation conditions for error handling.
G. SMAC ends the simulation if the final time has been reached or if other conditions have been met.

Some of the major subroutines are explained as follow:

(1) INPUT
INPUT reads input information from an input file, INPUT.DAT. The details of the input file is given in the next section.
(2) ACCEL
ACCEL uses the change in linear and angular velocities to calculate the longitudinal and lateral components of vehicle accelerations.
(3) SAVMAX
In the process of calculation, SAVMAX saves those maxima that are followed by resultant acceleration less than 1.0G and then followed by resultant acceleration greater than 1.0G. At the end of run, SAVMAX arranges in order of decreasing maxima and computes the principal direction of impact. SAVMAX also calls subroutine DELTAV, which computes the velocity change of the vehicle center of gravity (CG) during a collision.
(4) RNGKT1
RNGKT1 is a fourth-order Runge-Kutta integration subroutine. A subroutine, DAUX, is supplied by users to describe the differential equations.
(5) DAUX
DAUX calls TRAJ to compute vehicle trajectories and COLL if vehicles are in a collision.
(6) TRAJ
TRAJ uses wheel inputs to calculate forces and torques applied to the vehicles.
(7) COLL
COLL checks around the profiles of the vehicles to see if contact is made. If so, calculates the relative speeds and forces. COLL calls PSBRAN, TORDER, SETIND. For details of the calculation in (6) and (7), please refer to McHenry, 1971 [1].
(8) OUPUT
OUTPUT produces a formatted output to OUTPUT.DAT. OUPUT calls other subroutines OUT2, RNGDAM, DAMAGE, NCOLDV. RNGDAM in turns calls ADJEND. DAMAGE calls FULSRC, CORSRC, DEFMJ. Most of these subroutines post-process the outcome of the collisions and computes the damage range, sum of velocity change, etc. The range of damage indicates the width, depth, and location of damage occurred to each vehicle.

3.2 Input File

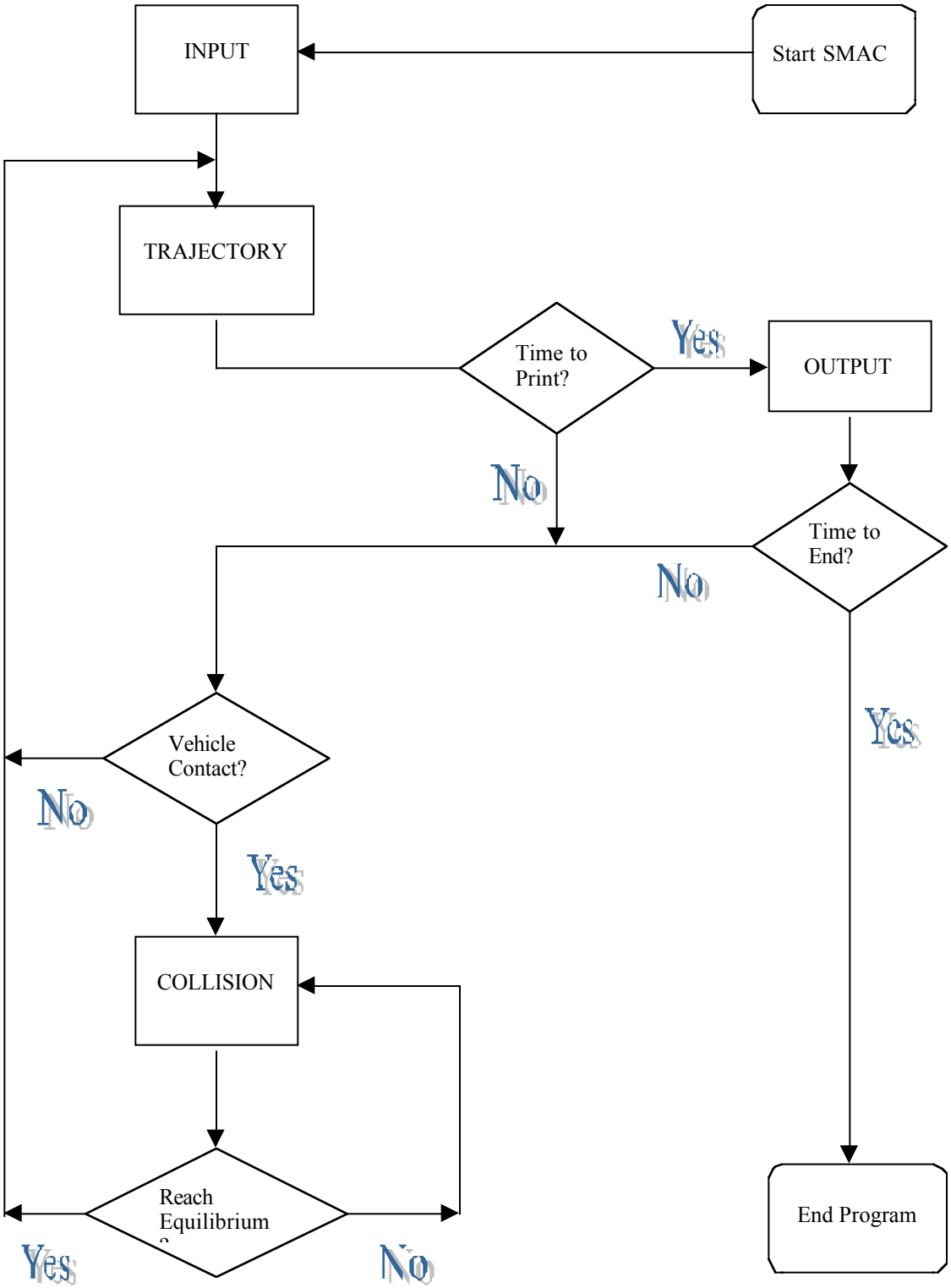
The input file is assigned a name of 'input.dat' in the main program, SMACPATH. A sample file is shown in Figure 2. Except lines 1 and 2, each line in the input file has an integer entry on columns 73-80. The integer indicates the entry row number to the input subroutine so that the inputs are read into the proper parameters and variables. As a result, the lines in the input files can be entered in altered orders as long as the entry row numbers are associated with the correct contents. The descriptions of the contents are as follow:

Line Sequence	Row Entry No.	Contents
1-2	N/A	Text Description of simulation Format: characters up to 80 per row
3	1	T0: begin time; all times in seconds TF: final time DTTRAJ: integration time step for trajectory phase DTCOLL: integration time for collision phase DTCOLT: integration time step following collision DTPRNT: time step for print out UVMIN: minimum velocity to terminate simulation, inch/sec PSIDOT: min. angular velocity to terminate simulation, deg/sec No. of vehicles: 1 or 2 Format: 9 floating numbers with each entry occupying 8 columns
4	2	Vehicle 1: XC10: initial x position, inch YC10: initial y position, inch PSI10: initial yaw angle, deg PSI1D0: initial yaw rate, deg/sec U10: initial longitudinal velocity, inch/sec

		V10: initial lateral velocity, inch/sec Format: 6 floating numbers with each entry occupying 8 columns
5	3	Vehicle 2: similar to the row above (replace 1 by 2 in variable names)
6	4	Vehicle 1: A1: distance CG to front axle, inch B1: distance CG to rear axle, inch TR1: track width, inch I1: yaw inertia, lb-sec-sec-inch M1: mass, lb-sec-sec/inch PSIR10: rear axle angle if damaged, deg XF1: distance CG to front end, inch XR1: distance CG to rear end, inch YS1: distance CG to side, inch Format: 9 floating numbers with each entry occupying 8 columns
7	5	Vehicle 2: similar to the row above (replace 1 by 2 in variable names)
8	6	Vehicle 1: C(1): cornering stiffness for right front tire, lb/radian C(2): cornering stiffness for left front tire C(3): cornering stiffness for right rear tire C(4): cornering stiffness for left rear tire Format: 4 floating numbers with each entry occupying 8 columns
9	7	Vehicle 2: C(5): cornering stiffness for right front tire, lb/radian C(6): cornering stiffness for left front tire C(7): cornering stiffness for right rear tire C(8): cornering stiffness for left rear tire Format: 4 floating numbers with each entry occupying 8 columns
10	8	Vehicle 1: TBTQ1 (initial time for torque input, sec) TETQ1 (final time for torque input, sec) TINCQ1 (time increment for torque input, sec) NTBLQ1 (if not 0.0, do not read table) Format: 4 floating numbers with each entry occupying 8 columns Note: The first 3 variables decide the number of entries in the table.
11	N/A	Following the row above, 7 entries per row for right front wheel with minimum of 3 and maximum up to 201 entries Format: 7 floating numbers with each entry occupying 10 columns
12	N/A	Similar to the row above with entries for left front wheel
13	N/A	Similar to the row above with entries for right rear wheel
14	N/A	Similar to the row above with entries for left rear wheel
15	9	Vehicle 2: similar to row 8 above
16-19	N/A	Similar to row 12-15 above
20	10	Vehicle 1: TBPSF1 (initial time for steer input, sec)

		TEPSF1 (final time for steer input, sec) TINCP1 (time increment for steer input, sec) NTBLP1 (if not 0.0, do not read table) Format: 4 floating numbers with each entry occupying 8 columns Note: The first 3 variables decide the number of entries in the table.
21	N/A	Following the row above, 7 entries per row for right front wheel with minimum of 3 and maximum up to 201 entries Format: 7 floating numbers with each entry occupying 10 columns
22	N/A	Similar to the row above with entries for left front wheel
23	11	Vehicle 2: similar to row 10 above (replace 1 by 2 in variable names)
24-25	N/A	Similar to row 22-23 above
26	12	XBP(1): points defining terrain zones, inch YBP(1) XBP(2) YBP(2) XMU1: friction coefficient in zone 1 at zero speed XMU2: friction coefficient in zone 2 at zero speed CMU: coefficient of linear decrement of friction with tire speed FMOVIE: not used in this version Format: 7 floating numbers with each entry occupying 10 columns
27	13	DELPSI: interval between radial vectors, deg DELRHO: increment change in radius vector, inch LAMBDA: acceptance error in equilibrium, lb/inch ZETAV: minimum relative velocity for friction, inch/sec KV1: load-deflection characteristics, vehicle 1, lb/inch-inch KV2: load-deflection characteristics, vehicle 2, lb/inch-inch MU: inter-vehicle friction coefficient Format: 7 floating numbers with each entry occupying 8 columns
28	14	C0, C1, C2: coefficients of assumed parabolic variation of coefficient of restitu with deflection Format: 3 floating numbers with each entry occupying 8 columns
29	15	PSILIM1: PSIB Range Test, Collision Criteria, default 70 degrees PSILIM2: default 110 degrees PSILIM3: default 250 degrees PSILIM4: default 290 degrees PSILIM5: PSIB1 for RHOB1 Test, default 10 degrees PSILIM6: default 170 degrees PSILIM7: default 190 degrees PSILIM8: default 350 degrees Format: 7 floating numbers with each entry occupying 8 columns
30	16	IOPTION: added features (to be explained in the next section)
31	10001	End of input file

Figure 1. Flow Chart of Simulation Program



```
SMAC INPUT EXAMPLE FILE

MODIFIED FOR PATH ON 11-12-1997

0.0     5.0     0.025   0.001   0.005   0.100   36.0    10.0    2.0     1

0.0     0.0     5.0     0.0     600.0   0.0                             2

500.0   0.0     60.0    0.0     0.0     0.0                             3
52.7    54.8    57.7    22424.  8.51    0.0     85.7    -100.0  35.7    4
57.3    59.7    60.0    29560.  9.86    0.0     94.8    -110.8  38.4    5
-10250  -10250  -10195  -10195                                          6
-10250  -10250  -10195  -10195                                          7
0.0     5.0     1.0     -0.1                                            8
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0     5.0     1.0     0.0                                             9
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0     5.0     1.0     0.0                                             10
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
0.0     5.0     1.0     0.0                                             11
0.0       0.0       0.0       0.0       0.0       0.0
0.0       0.0       0.0       0.0       0.0       0.0
  -100.0  -100.0  -100.0   100.0     0.7     0.7     0.0               12
     2.0     0.2    15.0     5.0    50.0    50.0   0.550     0.0        13
 4606E-517547E-716711E-9                                                14
1                                                                       16

       10001
```

Figure 2. A Sample Input File

3.3 Added Options

In row entry 16, an integer, IOPTION, is used to indicate a few added options of executing the simulation. The options are:

A. ICNTRL = 1
Run a simulation with vehicle 1 and a step input of steering at the front wheels. The program will ask interactively for a steering angle. The step input starts at 0.25 second into the simulation.
B. ICNTRL = 2
Run a simulation with vehicle 1 and a sinusoidal input of steering angle at the front wheels. The program will ask for the frequency and amplitude of the input. The sinusoidal input starts at 0.25 second into the simulation.
C. ICNTRL = 3
Run a simulation with a feedback controller implemented in subroutine CNTRL.
D. ICNTRL = 4
Run a simulation with feedback controllers with time delay, also implemented in subroutine CNTRL.

These added features are implemented for the purpose of ongoing development work at PATH. The addition of these options allows the studies of vehicle dynamics and control algorithms in collisions.

4.0 CONCLUDING REMARKS

This report summarizes the contents of SMAC source codes. A copy of the code is attached to this report. This documentation will allow future developers to utilize this program as a validation model or combine it with other simulation tools.

This work is part of the research project conducted under MOU-324 and an extension of MOU-252. Readers can refer to previous publications [16-21] for findings using this simulation tool. Future work of this project includes the testing of control algorithms in certain collision scenarios.

ACKNOWLEDGMENT

This work was performed as part of the California PATH Program of the University of California in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department of Transportation, Federal Highway Administration.

The contents of this paper reflect the views of the author who is responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California. This report does not constitute a standard, specification, or regulation.

The author would like to express special thanks to Charlie Compton and Joel MacWilliams of University of Michigan, Traffic Research Institute (UMTRI) for their help in providing the source codes of SMAC. I would also like to thank Seymour Stern of National Highway Traffic Safety Administration (NHTSA), who directs me to Joel after a search for the source codes at NHTSA was unsuccessful.

A commercial PC-DOS version of EDSMAC was provided by Engineering Dynamics Corporation (EDC) of Beaverton, Oregon, to PATH at no cost. Thanks should go to Terry Day, who offered the program and provided helpful instructions.

REFERENCE

1. McHenry, R.R., "Development of a Computer Program to Aid the Investigation of Highway Accidents," Calspan Report No. VJ-2979-V-1, DOT HS 800 821, NTIS-PB 208-537, 1971.
2. Solomon, P.L., "The Simulation Model of Automobile Collisions (SMAC) Operator's Manual," US DOT, NHTSA, Accident Investigation Division, 1974.
3. Noga, T., Oppenheim, T., "CRASH3 User's Guide and Technical Manual," U.S. DOT, January, 1981.
4. T.D. Day, R.L. Hargens, "Differences between EDCRASH and CRASH3," SAE Paper No. 850253.
5. Day, T., Hargens, R., "An Overview of the Way EDSMAC Computes Delta-V," SAE Paper No. 880069, Society of Automotive Engineers, 1988.
6. Engineering Dynamics Corporation, EDCRASH, "Reconstruction of Accident Speeds on the Highway," Version 4.5, June 1989.
7. Engineering Dynamics Corporation, EDSMAC, "Simulation Model of Automobile Collisions," Version 2.4, May 1989.
8. Jones, I.S., "Results of Selected Applications to Actual Highway Accidents of the SMAC Reconstruction Program, " SAE Paper No. 741179, 1974.
9. Jones, I.S., "The Application of the SMAC Accident Reconstruction Program to Actual Highway Accidents," Proceedings of the Eighteenth Conference of the American Association of Automotive Medicine, 1974.
10. R.S. Smith, J.T. Noga, "Accuracy and Sensitivity of CRASH," National Center for Statistics and Analysis, NHTSA, January 1982.

11. A.K. Prasad, "CRASH3 Damage Algorithm Reformulation for Front and Rear Collisions," SAE Paper No. 900098.
12. "Collision Deformation Classification," SAE Technical Report J224 MAR80, March 1980.
13. Day, T., Hargens, R., "Application and Misapplication of Computer Programs for Accident Reconstructions," SAE Paper No. 890738, 1989.
14. R.R. McHenry, N.J. Deleys, "Vehicle Dynamics in Single Vehicle Accidents – Validation and Extensions of a Computer Simulation, " Calspan Report No. VJ-2251-V-3, NTIS-PB 182-663, 1968.
15. R.R. McHenry, et al. "Mathematical Reconstruction of Highway Accidents " Calspan Report No. ZM-5096-V-1, DOT HS 053-1-146, NTIS-PB 220-150, 1973.
16. C. Chan, "Studies of Collisions in Vehicle Following Operation by Two-Dimensional Impact Simulations," *Proceedings of the 1996 Annual Meeting of ITS America*, Sixth Annual Meeting, Houston, Texas, April 1996.
17. C. Chan, "Collision Analysis of Vehicle Following Operation in Automated Highway Systems," *Proceedings of the Third World Congress of Intelligent Transport Systems*, Orlando, Florida, October 1996.
18. C. Chan, "Collision Analysis of Vehicle Following Operations by Two-Dimensional Simulation Model: Part I - Effects of Operational Variables," *California PATH Research Report, UCB-ITS-PRR-97-4*, January 1997.
19. C. Chan, "Collision Analysis of Vehicle Following Operations by Two-Dimensional Simulation Model: Part II - Vehicle Trajectories with Follow-Up Maneuvers," *California PATH Research Report, UCB-ITS-PRR-97-5*, January 1997.
20. C. Chan, "Simulation of Vehicle Trajectories and Maneuvers in Vehicle-Following Collisions," *Proceedings of the 1997 Annual Meeting of ITS America*, Seventh Annual Meeting, Washington, D.C., June 1997.
21. C. Chan, "Studies of Vehicle Collisions by EDSMAC," *California PATH Research Report, UCB-ITS-PRR-98-11,* March 1998.

```
      SUBROUTINE COLL
C       SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
      COMMON/INTG/NEQ,T,DT,VAR(12),DER(12)
      EQUIVALENCE (VAR(1),XCP1),(VAR(2),YCP1),(VAR(3),PSI1),
     1            (VAR(4),PSI1D),(VAR(5),U1),(VAR(6),V1),
     2            (VAR(7),XCP2),(VAR(8),YCP2),(VAR(9),PSI2),
     3            (VAR(10),PSI2D),(VAR(11),U2),(VAR(12),V2)
      EQUIVALENCE (DER(1),DXCP1),(DER(2),DYCP1),(DER(3),DPSI1),
     1            (DER(4),DPSI1D),(DER(5),DU1),(DER(6),DV1),
     2            (DER(7),DXCP2),(DER(8),DYCP2),(DER(9),DPSI2),
     3            (DER(10),DPSI2D),(DER(11),DU2),(DER(12),DV2)
      COMMON/INPT/  T0,TF,DTTRA0,DTCOL0,DTCLT0,DTPRN0,UVMIN,PSIDMN,FVEH0
     1             ,XCP10,YCP10,PSI10,PSI1D0,U10,V10,
     2              XCP20,YCP20,PSI20,PSI2D0,U20,V20,
     3              A1,B1,TR1,FIZ1,FM1,PSIR10,XF1,XR1,YS1,
     4              A2,B2,TR2,FIZ2,FM2,PSIR20,XF2,XR2,YS2,
     5              CSTFI(8),TBTQ1,TETQ1,TINCQ1,NTBLQ1,NTQ1,
     6              TBTQ2,TETQ2,TINCQ2,NTBLQ2,NTQ2,TII(8,201),
     7              TBPSF1,TEPSF1,TINCP1,NTBLP1,NPSF1,
     8              TBPSF2,TEPSF2,TINCP2,NTBLP2,NPSF2,PSIFI(4,201),
     9              XBP(2),YBP(2),XMU1,XMU2,CMU
      COMMON/INPT/  DELPS0,DELRO0,ALAMB,ZETAV,AKV(2) ,AMU,C0,C1,C2,
     1              PSLM10,PSLM20,PSLM30,PSLM40,PSLM50,PSLM60,PSLM70,
     2              PSLM80,HED(40),DADE(3),XINPUT(9)
      COMMON/INTGR/JCC(4),JCOR(4),NPSIB,IPSIB0,IPSIB,IBB,IFF,NDTAB,
     1            I9,J9(361,2),  NPSJB,IPSJB,I3,ILAST,IL,IND1,IND2,INDI
     2            ,INDJ,ISTOP,ITZER1,ITZER2,IPZER1,IPZER2,ICOUNT,INDXB,
     3            ICOLLP,ICOLL,IQ,IND,IVEH,LL,ICTVDI,ITCOL,ITFLG1,
     4            ITFLG2,NWRDB1,NWRDE1,NWRDB2,NWRDE2,ICOLAC
C            IN COMMON /COMP/ DO NOT DISTURB THE ORDER OF THE VARIABLES
C                             WHICH ARE EQUIVALENCED TO ARRAYS.
      COMMON/COMP/  DTTRAJ,DTCOLL,DTCOLT,DTPRNT,UVMN2,PSIDMA,TPRINT,DTP,
     1              U1V1SQ,U2V2SQ,DELPSI,DELPST,DELPS2,DELRHO,
     2              PSILM1,PSILM2,PSILM3,PSILM4,PSILM5,PSILM6,PSILM7,
     3              PSILM8,EJJ(4,2),GJJ(4,2),C1OC2,PSCC,PSIR1,PSIR2,
     4            XCMXC1,XCMXC2,YCMYC1,YCMYC2,SPSI1,SPSI2,CPSI1,CPSI2,
     5            XCPI,YCPI,PSII,XCPJ,YCPJ,PSIJ,XCMXC,YCMYC,SPSIJ,
     6            CPSIJ,XFI,XRI,YSI,XFJ,XRJ,YSJ,AKVI,AKVJ,EJ,GJ,
     7            XPJ(4),YPJ(4),ANGYX,SPAN,PSIBPI(4),PSIBPB,PSIBPF,
     8            FNPSIB,PSIBB,PSIB,PSFMPS,SCPSIB,CSPSIB,
     9            RHOBI,RHOBIN(4),PSIBIJ,SPSBIJ,CPSBIJ,XBIJ,YBIJ,PSJTB
      COMMON/COMP/ PSIBPJ,FNPSJB,PSIBJ,RHOBJ,XYJSQR,SPSIB,CPSIB,XBI,YBI,
     1         PRESI,PRESJ,XYSR,DELTA,CRHO,RHOBIC,RHOBMX,RHOBIT,RHOBT,
     2             SPSI21,CPSI21,X2TEM,Y2TEM,XX,YY,PAV,XXYYSR,FN1,SPSF,
     3             CPSF,X1AV,Y1AV,XYAVSR,X2TERM,Y2TERM,SPS21F,CPS21F,
     4             X2AV,Y2AV,VT1AV,VT2AV,VTMVT,FRICT,FNX1,FNY1,FNN1,
     5             FNX2,FNY2,FNN2,SFX1,SFY1,SFN1,SFX2,SFY2,SFN2,
     6              SFX1C,SFX2C,SFY1C,SFY2C,SFN1C,SFN2C,
     7              SFX1T,SFX2T,SFY1T,SFY2T,SFN1T,SFN2T,  TCOL,TEND,
     8              TREST1,TREST2,TCOLS1,TCOLS2,P1TEM,P2TEM,U1V1E,U2V2E,
     9              EXTRA(10)
      COMMON/COMPT/ W(8),XWP(8),YWP(8),FLAGW(8),PSIW(8),TQW(8),
     1             TFX,TFY,TMOM,PSIRR,COEF,COEF1,COEF2,TRD2,TRD21,TRD22,
     2             XBMXB,FNUM,A,B,FIZ,FMASS,XC,YC,U,V,PSIC,PSICD,UD2,
     3             SGNU,SGNI,AOMB,CSPSIC,SNPSIC,TRCPSC,TRSPSC,TRPSCD,
     4             XI,XW,YW,SX,SY,S,XPP,YPP,RHO,RHOP,XMUS,XMU,PSI,XNUM,
     5             XDEN,RATIO,ALFA,F,TIF,FC,FCOS,SMAL,FSMX,FS,BETB,FX,FY
     6             ,CSPSI,SNPSI,XCXC,YCYC,SPJ,CPJ,SPIJ,CPIJ,XFJJ,XRJJ,
     7              YSJJ,EJJI,GJJI,XICORJ,YICORJ,NTQ,TBTQ,TETQ,TINCRQ,
     8             NPSF,TBPSF,TEPSF,TINCRP
```

```
      COMMON/TAB/  TRHOB(361,2),TROBMX(361,2),TPSIB(361,2),
     1             TCPSIB(361,2),TSPSIB(361,2),TXB1(100),TYB1(100),
     2             TPSB1(100),TPRES1(100)
      DIMENSION   XCMXCI(2),YCMYCI(2),SPSII(2),CPSII(2),PSLM0(8),PSLM(8)
      EQUIVALENCE (XCMXCI(1),XCMXC1),(YCMYCI(1),YCMYC1),(SPSII(1),SPSI1)
     1           ,(CPSII(1),CPSI1),(PSLM0(1),PSLM10),(PSLM(1),PSILM1)
C        COMMON VDICON HAS CONSTANTS USED FOR VEHICLE DAMAGE INDEX
      COMMON/VDICON/EXTF1( 9),EXTR1( 9),EXTB1( 9),PSP1(12),
     2              EXTF2( 9),EXTR2( 9),EXTB2( 9),PSP2(12),NCOL51(10),
     2              NCOL52(10),MMM( 9)
      IF(ISTOP.NE. 0) RETURN
C                CLEAR TABLE 3
      I3 = 100
      DO 2   IJ = 1,I3
      TXB1(IJ) = 0.0
      TYB1(IJ) = 0.0
      TPSB1(IJ) = 0.0
    2 TPRES1(IJ)= 0.0
      I3 = 0
      INDI = IND1
      INDJ = IND2
C         SEE RESET AT ST 201
      I = 1
      J = 2
      I1= 1
      I2= 7
    5 XCPI = VAR(I1)
      YCPI = VAR(I1 + 1)
      PSII = VAR(I1 +2)
      XCPJ = VAR(I2)
      YCPJ = VAR(I2 +1)
      PSIJ = VAR(I2 + 2)
      XCMXC = XCMXCI(I)
      YCMYC = YCMYCI(I)
      SPSIJ = SPSII(J)
      CPSIJ = CPSII(J)
      XFI   = EJJ(1,I)
      XRI   = EJJ(3,I)
      YSI   = GJJ(1,I)
      XFJ   = EJJ(1,J)
      XRJ   = EJJ(3,J)
      YSJ   = GJJ(1,J)
      AKVI  = AKV(I)
      AKVJ  = AKV(J)
      XFIPAR = 0.15 * XFI
      YSIPAR = 0.36 * YSI
      XFJPAR = 0.15 * XFJ
      YSJPAR = 0.36 * YSJ
      DO 10 IC=1,4
      EJ  = EJJ(IC,J)
      GJ  = GJJ(IC,J)
      XPJ(IC) = XCPJ + EJ *CPSIJ  - GJ *SPSIJ
   10 YPJ(IC) = YCPJ + EJ *SPSIJ  +GJ *CPSIJ
C     CORNER PTS OF VEHICLE J IN SPACE-FIXED SYSTEM, RF 1,LF 2,RR 3,LR 4
      CALL PSBRAN
   80 IF( PSIB - TWOPI) 82,81,81
   81 PSIB = PSIB - TWOPI
      GO TO 80
   82 IF( ABS(PSIB) - DELPST) 83,83,84
   83 PSIB = 0.0
      IPSIB = 1
      PSFMPS = PSIBPF
```

```
      GO TO 85
   84 IF( ABS(PSIB-TWOPI) - 0.005) 83,83,840
  840 PSFMPS = PSIBPF - PSIB
   85 IF(PSFMPS) 86,86,88
   86 IF(PSFMPS + PI) 88,88,200
C
   88 RHOBI = TRHOB(IPSIB,I)
C                 NEG. RHOBI FROM TABLE MEANS NO ENTRY YET
      I9 = 0
C      I9 = 1  IS INDICATOR FOR TEMPORARY TRHOB ENTRY (REPORT STEP NINE)
      RHOBMX = TROBMX(IPSIB,I)
  881 IF(RHOBMX) 100,100,89
   89 IF(RHOBI) 90,90,120
   90 RHOBI = RHOBMX
      GO TO 115
  100 IF((PSILM1.LE.PSIB.AND.PSIB.LE.PSILM2) .OR.(PSILM3.LE.PSIB
     X    .AND. PSIB.LE.PSILM4)) GO TO 103
      SCPSIB = 1.0/COS(PSIB)
      RHOBIN(1)  = XFI*SCPSIB
      RHOBIN(2)  = XRI*SCPSIB
      IF((PSILM5.LE.PSIB.AND.PSIB.LE.PSILM6) .OR. (PSILM7.LE.PSIB
     X    .AND. PSIB.LE.PSILM8)) GO TO 105
      RHOBIN(3) = -100.
      RHOBIN(4) = -100.
      GO TO 107
  103 RHOBIN(1) = -100.
      RHOBIN(2) = -100.
  105 CSPSIB =  1.0/SIN(PSIB)
      RHOBIN(3) = YSI * CSPSIB
      RHOBIN(4) = - RHOBIN(3)
  107 RHOBIT= 1.E20
      DO 110 IJ =1,4
      IF(RHOBIN(IJ)) 110,108,108
  108 IF(RHOBIN(IJ) - RHOBIT) 109,110,110
  109 RHOBIT= RHOBIN(IJ)
  110 CONTINUE
      IF(RHOBIT)111,111,112
  111 ISTOP=3
      WRITE(NOUT,5111) T,RHOBIT,PSIB
 5111 FORMAT(5H05111,2X,3H T=,F8.4,5X,7HRHOBIT=,E14.6,2X,6H PSIB=,E14.6)
      GO TO 300
  112 TROBMX(IPSIB,I) = RHOBIT
      RHOBMX = RHOBIT
      IF (RHOBI) 114,114,120
  114 RHOBI = RHOBIT
      TPSIB(IPSIB,I) = PSIB
  115 TRHOB(IPSIB,I) = RHOBI
      I9 = 1
C
  120 PSIBIJ = PSIB + PSII - PSIJ
      CPSBIJ = COS(PSIBIJ)
      SPSBIJ = SIN(PSIBIJ)
      XBIJ  =   YCMYC*SPSIJ + XCMXC*CPSIJ + RHOBI*CPSBIJ
      YBIJ  =   YCMYC*CPSIJ - XCMXC*SPSIJ + RHOBI*SPSBIJ
      IF(XRJ.LT.XBIJ .AND. XBIJ.LT.XFJ .AND. (-YSJ).LT.YBIJ .AND.
     X   YBIJ.LT.YSJ ) GO TO 130
      IF(I9)        122,122,121
  121 TRHOB(IPSIB,I) = -10.0
C           MESSAGE 5121
      I9 = 0
C
C          AT ST 122 INCREMENT PSIB AND IPSIB
```

```
  122 IF(IPSIB - IPSIB0 ) 125,123,125
  123 IF(IPSIB0 - NDTAB ) 125,124,124
  124 IPSIB = 1
      PSIB  = 0.0
      PSFMPS = PSIBPF
      GO TO 85
  125 PSIB  = FLOAT(IPSIB) *DELPSI
      IPSIB = IPSIB + 1
      GO TO 80
C
  130 IF(XBIJ) 133,131,133
  131 IF(YBIJ) 133,132,133
  132 PSIBPJ = 0.0
      GO TO 138
  133 PSIBPJ = ATAN2(YBIJ,XBIJ)
  134 IF(PSIBPJ) 135,138,139
  135 PSIBPJ = TWOPI + PSIBPJ
      GO TO 134
  138 NPSJB = 0
      FNPSJB = 0.0
      PSIBJ = 0.0
      IPSJB = 1
C           AT ST 138 PSIBPJ IS ZERO
      GO TO 141
  139 NPSJB = PSIBPJ/DELPSI  + 0.5
      FNPSJB = FLOAT(NPSJB)
      IPSJB = NPSJB + 1
  140 PSIBJ = FNPSJB * DELPSI
 1400 IF(PSIBJ - TWOPI)1402,1401,1401
 1401 PSIBJ = PSIBJ - TWOPI
      GO TO 1400
 1402 IF (ABS(PSIBJ) - DELPST) 1403,1403, 1404
 1403 PSIBJ = 0.0
      IPSJB = 1
      GO TO 141
 1404 IF(ABS(PSIBJ - TWOPI) - 0.005)1403,1403, 141
  141 PSJTB  = TPSIB(IPSJB,J)
      RHOBJ = TRHOB(IPSJB,J)
      XYJSQR = SQRT(XBIJ*XBIJ + YBIJ*YBIJ)
      IPRESZ = 0
      IF(RHOBJ) 149,149,142
  142 IF( ABS(PSJTB -PSIBJ) - DELPST) 145,145,143
  143 ISTOP = 4
      WRITE(NOUT ,5143) T,DELPST,PSIBJ,PSJTB,IPSJB
 5143 FORMAT(7H05143  ,3H T=,F8.4,8H DELPST=,E14.6,7H PSIBJ=,E14.6,
     1   7H PSJTB=,E14.6,7H IPSJB=,I4)
      GO TO 300
  145 TEMP1 = ABS(TSPSIB(IPSJB,J))
      TEMP2 = ABS(TCPSIB(IPSJB,J))
      IF(TEMP1)  1454,1451,1454
 1451 IF(TEMP2)  1454,1452,1454
 1452 ISTOP = 11
      WRITE(NOUT,5145) T,RHOBJ,XYJSQR,PSIBJ,PSJTB,IPSJB
 5145 FORMAT(7H05145  ,3H T=,F8.4,8H RHOBJ =,E14.6,   9H XYJSQR =,E14.6,
     1   7H PSIBJ=,E14.6,   7H PSJTB=,E14.6,  7H IPSJB=,I4)
      GO TO 300
 1454 IF ( TEMP1-TEMP2) 1455,1456,1456
 1455 TEMP3 = (DELRHO + 0.5*DELPS2*RHOBJ*TEMP1 ) / TEMP2
      GO TO 1457
 1456 TEMP3 = (DELRHO + 0.5*DELPS2*RHOBJ*TEMP2 ) / TEMP1
 1457 IF(RHOBJ - XYJSQR + TEMP3) 146,151,151
  146 IF(I9)         148,148,147
```

```
  147 TRHOB(IPSIB,I) = - 10.0
      I9 = 0
      GO TO 122
  148 PRESI = 0.0
      PRESJ = 0.0
      IPRES = 0
      IPRESZ = 1
      GO TO 151
C
  149 IF (I9) 150,150,151
  150 TEMP = PSIBJ - PSIBPJ
      IF(ABS(TEMP) -DELPS2) 1500,1500,1499
 1499 TEMP = TEMP + TWOPI
 1500 IF(XBIJ) 1501,1502,1501
 1501 IF(YBIJ) 1503,1502,1503
 1502 TRHOB(IPSJB,J) = XYJSQR
      GO TO 1508
 1503 TEMPYX = YBIJ/XBIJ
      IF(ABS(TEMPYX) - 0.27) 1507,1504,1504
 1504 IF(ABS(TEMPYX) - 1.0) 1505,1505,1506
 1505 IF( TEMP*TEMPYX) 1507,1507,1506
 1506 TRHOB(IPSJB,J) = XYJSQR*(1.0 - TEMP*XBIJ/YBIJ)
      GO TO 1508
 1507 TRHOB(IPSJB,J) = XYJSQR*(1.0 + TEMP*TEMPYX)
 1508 IF(TCPSIB(IPSJB,J)) 1511,1509,1511
 1509 IF(TSPSIB(IPSJB,J)) 1511,1510,1511
 1510 TPSIB(IPSJB,J) = PSIBJ
      TCPSIB(IPSJB,J) = COS(PSIBJ)
      TSPSIB(IPSJB,J) = SIN(PSIBJ)
 1511 J9(IPSJB,J) = -1
  151 CPSIB =  COS(PSIB)
      SPSIB =  SIN(PSIB)
      XBI   =  RHOBI*CPSIB
      YBI   =  RHOBI*SPSIB
      IF (IPRESZ) 152,152,180
  152 IPRES = 1
  153 TEMP1 =  XFI - XBI   + .0001
      IF(TEMP1)1600,161,154
  154 TEMP2 = XBI-XRI   +.0001
      IF(TEMP2)1600,161,155
  155 IF(TEMP1-TEMP2) 1561,1561,156
  156 TEMP1 = TEMP2
 1561 TEMP2 =  YSI - ABS(YBI) + .0001
      IF(TEMP2)1600,161,1562
 1562 IF(INDI) 160,1572,158
 1572 TEMP1 = TEMP2
      GO TO 160
  158 IF(TEMP1 - XFIPAR) 1582,1582,1581
 1581 IF(TEMP2 - YSIPAR ) 1584, 160,160
 1582 IF( TEMP1-TEMP2)   160, 160,1583
 1583 TEMP1 = TEMP2
      GO TO 160
 1584 TEMP1 = TEMP1 *(TEMP2/YSIPAR)
  160 PRESI = AKVI * TEMP1
      GO TO 162
 1600 WRITE(NOUT,5160) T,XFI,XRI,YSI,XBI,YBI,PSIB,PSIBIJ,IPRES,INDI
 5160 FORMAT(22H05160  PRESI TENDS NEG,2X,3H T=,F8.4,5H XFI=,E14.6,
     X5H XRI=,E14.6,5H YSI=,E14.6/34X,6H XBI =,E14.6,18X,6H YBI =,E14.6/
     X 34X,6H PSIB=,E14.6,8H PSIBIJ=,E14.6,7H IPRES=,I4,6H INDI=,I4/
     X  21H        SET PRESI=0.0)
  161 PRESI = 0.0
  162 TEMP1 = XFJ - XBIJ  + .0001
```

```
      IF(TEMP1) 170,1700,163
  163 TEMP2 = XBIJ - XRJ     + .0001
      IF(TEMP2) 170,1700,164
  164 IF(TEMP1- TEMP2) 1660,1660,165
  165 TEMP1 = TEMP2
 1660 TEMP2 = YSJ - ABS(YBIJ)   + .0001
      IF(TEMP2) 170,1700,1661
 1661 IF(INDJ) 169,1665,167
 1665 TEMP1 = TEMP2
      GO TO 169
  167 IF(TEMP1 - XFJPAR)  1671,1671,1670
 1670 IF(TEMP2 - YSJPAR)  1673, 169,169
 1671 IF(TEMP1-TEMP2)    169, 169,1672
 1672 TEMP1 = TEMP2
      GO TO 169
 1673 TEMP1 = TEMP1 *(TEMP2/YSJPAR)
  169 PRESJ = AKVJ * TEMP1
      GO TO 1701
  170 WRITE(NOUT,5170) T,XFJ,XRJ,YSJ,XBIJ,YBIJ,PRESI,PSIB,PSIBIJ,IPRES,
     X INDJ
 5170 FORMAT(22H05170  PRESJ TENDS NEG,2X,3H T=,F8.4,5H XFJ=,E14.6,
     X5H XRJ=,E14.6,5H YSJ=,E14.6/34X,6H XBIJ=,E14.6,18X,6H YBIJ=,E14.6,
     X 2X,7H PRESI=,E14.6/
     X 34X,6H PSIB=,E14.6,8H PSIBIJ=,E14.6,7H IPRES=,I4,6H INDJ=,I4/
     X  21H        SET PRESJ=0.0)
C       OMIT FORMER STOP
C     GO TO 300
 1700 PRESJ = 0.0
 1701 IF(PRESI + ALAMB - PRESJ) 1704,180,1703
 1703 PRESI = PRESJ
      GO TO 180
 1704 IF((ABS(AKVJ/AKVI) - 1.25) - 0.75) 1705,1705,1706
 1705 TEMP3 = DELRHO
      GO TO 172
 1706 TEMP1 = (PRESJ-PRESI-0.0001*(AKVJ-AKVI))/((AKVI+AKVJ)*DELRHO)
      ITEMP2= TEMP1 - 0.5
C              ROUND TO ONE LESS THAN NEAREST INTEGER
C            TEMP1 SHOULD ALWAYS BE GREATER THAN 0.5
      TEMP2 = FLOAT (ITEMP2)
      TEMP3 = (TEMP2 + 0.5) * DELRHO
  172 XBI = XBI - TEMP3  * CPSIB
      YBI = YBI - TEMP3  * SPSIB
      XBIJ =  XBIJ - TEMP3  * CPSBIJ
      YBIJ =  YBIJ - TEMP3  * SPSBIJ
C     IF(IPRES -150)  174,173,173
      IF(IPRES -200)  174,173,173
  173 ISTOP=7
      WRITE(NOUT,5173) T,PSIB,XBI,YBI,XBIJ,YBIJ,PRESI,PRESJ,IPRES
 5173 FORMAT(5H05173,2X,F8.4,5X,7E14.6,I4)
      GO TO 300
  174 IPRES = IPRES + 1
      GO TO 153
C
C
C             TABLE 3 HAS  FOUR ARRAYS,TXB1,TYB1,TPSB1,TPRES1
  180 I3 = I3 + 1
      IF( I3 - 100) 1801,1801,1800
 1800 ISTOP = 9
      I3P = I3 - 1
      WRITE(NOUT,5030) T,IND1,IND2,    I3P,(TPSB1(IJ),IJ=1,I3P)
 5030 FORMAT(5H05030,1X,3H T=,F8.4,7H, IND1=,I4,7H, IND2=,I4,
     X  6H,  I3=,I4,16H,TPSB1 AS STORED / (1X,10E13.5/))
```

```
      WRITE(NOUT,5031)    (TXB1(IJ),IJ=1,I3P)
 5031 FORMAT(5H 5031,10X, 6H  TXB1/(1X,10E13.5/))
      WRITE(NOUT,5032)    (TYB1(IJ),IJ=1,I3P)
 5032 FORMAT(5H 5032,10X, 6H  TYB1/(1X,10E13.5/))
      WRITE(NOUT,5033)   (TPRES1(IJ),IJ=1,I3P)
 5033 FORMAT(5H 5033,10X, 6HTPRES1/(1X,10E13.5/))
      GO TO 300
 1801 CONTINUE
      IF(I-1) 181,181,182
  181 TXB1(I3) = XBI
      TYB1(I3) = YBI
      TPSB1(I3) = PSIB
      TPRES1(I3) = PRESI
      GO TO 183
  182 TXB1(I3) = XBIJ
      TYB1(I3) = YBIJ
      IF(XBIJ) 1823,1821,1823
 1821 IF(YBIJ) 1823,1822,1823
 1822 PSIBPJ = 0.0
      GO TO 1826
 1823 PSIBPJ = ATAN2(YBIJ,XBIJ)
 1824 IF(PSIBPJ)  1825,1826,1826
 1825 PSIBPJ = TWOPI + PSIBPJ
      GO TO 1824
 1826 TPSB1(I3) = PSIBPJ
      TPRES1(I3) = PRESJ
  183 XYSR =  SQRT(XBI*XBI + YBI*YBI)
      IF (J9(IPSIB,I)) 190,184,184
  184 DELTA = RHOBMX - XYSR
      IF(DELTA) 185,185,186
  185 CRHO = 0.0
      RHOBIC = RHOBMX
      GO TO 192
  186 IF(DELTA-C1OC2)  187,190,190
  187 CRHO = C0 + DELTA*(-C1 + C2*DELTA)
  189 RHOBIC = CRHO * RHOBMX  +  (1.0-CRHO)* XYSR
      GO TO 192
  190 CRHO = 0.0
      RHOBIC = XYSR
  192 IF(RHOBIC) 193,193,194
  193 ISTOP=10
      WRITE(NOUT,5193) T,RHOBIC,PSIB
 5193 FORMAT(5H05193,2X,3H T=,F8.4,5X,7HRHOBIC=,E14.6,2X,6H PSIB=,E14.6)
      GO TO 300
  194 RHOBT = TRHOB(IPSIB,I)
      IF(RHOBT) 197,197,195
  195 IF(RHOBIC - RHOBT) 197,198,198
  197 TRHOB(IPSIB,I) = RHOBIC
  198 I9 = 0
      TCPSIB(IPSIB,I)= CPSIB
      TSPSIB(IPSIB,I)= SPSIB
      GO TO 122
C
C      AT STATEMENT 200 INTERCHANGE VEHICLES I AND J AND RETEST
  200 IF(I-1) 201,201,202
  201 I=2
      J=1
      I1 = 7
      I2 = 1
      INDI = IND2
      INDJ = IND1
      GO TO 5
```

```
C
C
  202 IF (I3) 203 ,240,203
  203 CALL TORDER( I3,IB1,IF1,TPSB1,TXB1,TYB1,TPRES1)
  210 CPSI21 =  CPSI2 * CPSI1  +  SPSI2 * SPSI1
      SPSI21 =  SPSI2 * CPSI1  -  CPSI2 * SPSI1
      X2TEM  = YCMYC1 * SPSI2  + XCMXC1 * CPSI2
      Y2TEM  =  YCMYC1 * CPSI2  - XCMXC1 * SPSI2
      IB = IB1
      IL = I3 - 1
  211 DO 220 IJ = IB,IL
      IF(I3-1) 212,2131,212
  212 IF(IB-I3) 2131,213,2131
  213 IJ1 = 1
      GO TO 2132
 2131 IJ1 = IJ + 1
 2132 XX = TXB1(IJ) -TXB1(IJ1)
      YY = TYB1(IJ)- TYB1(IJ1)
      PAV = 0.5 * (TPRES1(IJ) + TPRES1(IJ1))
      XXYYSR  = SQRT(XX*XX + YY*YY)
      FN1 = PAV * XXYYSR
      CPSF =  XX/XXYYSR
      SPSF =  YY/XXYYSR
C
      X1AV =  0.5 *(TXB1(IJ) + TXB1(IJ1))
      Y1AV =  0.5 *(TYB1(IJ) + TYB1(IJ1))
      X2TERM = X1AV * CPSI21 + Y1AV * SPSI21
      Y2TERM =  Y1AV * CPSI21 - X1AV * SPSI21
      CPS21F =  CPSI21 * CPSF   +  SPSI21 * SPSF
      SPS21F =  SPSI21 * CPSF   -  CPSI21 * SPSF
      X2AV = X2TEM  + X2TERM
      Y2AV = Y2TEM  + Y2TERM
      VT1AV = CPSF *(U1 - Y1AV*PSI1D )  +  SPSF *(V1 + X1AV*PSI1D )
      VT2AV = CPS21F*(U2- Y2AV*PSI2D ) - SPS21F *(V2 + X2AV*PSI2D )
      VTMVT =  VT2AV - VT1AV
      IF(ABS(VTMVT) - ZETAV) 214,215,215
  214 FRICT = AMU * (VTMVT/ZETAV)
      GO TO 216
  215 FRICT = SIGN(AMU,VTMVT)
  216 FNX1 =  FN1  * (SPSF + FRICT*CPSF)
      FNY1 =  FN1  * (FRICT*SPSF - CPSF)
      FNN1 = -FNX1*Y1AV  + FNY1* X1AV
      FNX2 = -FNX1*CPSI21 - FNY1 * SPSI21
      FNY2 =  FNX1*SPSI21 - FNY1 * CPSI21
      FNN2 = -FNX2*Y2AV   + FNY2 * X2AV
C
      SFX1C = SFX1C + FNX1
      SFY1C = SFY1C + FNY1
      SFN1C = SFN1C + FNN1
      SFX2C = SFX2C + FNX2
      SFY2C = SFY2C + FNY2
      SFN2C = SFN2C + FNN2
  220 CONTINUE
      IF(IB-1) 240,240,221
  221 IF(IB-I3) 222,223,222
  222 IB=I3
      IL=I3
      GO TO 211
  223 IB=1
      IL=IF1-1
      GO TO 211
  240 RETURN
```

```
  300 WRITE(NOUT,301) T,ISTOP
  301 FORMAT( 9H0 STOP AT,F9.4,12H SEC,  ISTOP=,I3)
      RETURN
      END
      SUBROUTINE PSBRAN
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
C         RANGE OF CLOCKWISE ANGULAR SWEEP FOR VEHICLE I
C      RETURNS PSIBPB(BEGIN SWEEP) AND PSIBPF(FINISH SWEEP)
C      RETURNS PSIB TO BEGIN AND IPSIB INDEX
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
      COMMON/INTG/NEQ,T,DT,VAR(12),DER(12)
      EQUIVALENCE (VAR(1),XCP1),(VAR(2),YCP1),(VAR(3),PSI1),
     1            (VAR(4),PSI1D),(VAR(5),U1),(VAR(6),V1),
     2            (VAR(7),XCP2),(VAR(8),YCP2),(VAR(9),PSI2),
     3            (VAR(10),PSI2D),(VAR(11),U2),(VAR(12),V2)
      EQUIVALENCE (DER(1),DXCP1),(DER(2),DYCP1),(DER(3),DPSI1),
     1            (DER(4),DPSI1D),(DER(5),DU1),(DER(6),DV1),
     2            (DER(7),DXCP2),(DER(8),DYCP2),(DER(9),DPSI2),
     3            (DER(10),DPSI2D),(DER(11),DU2),(DER(12),DV2)
      COMMON/INPT/  T0,TF,DTTRA0,DTCOL0,DTCLT0,DTPRN0,UVMIN,PSIDMN,FVEH0
     1             ,XCP10,YCP10,PSI10,PSI1D0,U10,V10,
     2              XCP20,YCP20,PSI20,PSI2D0,U20,V20,
     3              A1,B1,TR1,FIZ1,FM1,PSIR10,XF1,XR1,YS1,
     4              A2,B2,TR2,FIZ2,FM2,PSIR20,XF2,XR2,YS2,
     5              CSTFI(8),TBTQ1,TETQ1,TINCQ1,NTBLQ1,NTQ1,
     6              TBTQ2,TETQ2,TINCQ2,NTBLQ2,NTQ2,TII(8,201),
     7              TBPSF1,TEPSF1,TINCP1,NTBLP1,NPSF1,
     8              TBPSF2,TEPSF2,TINCP2,NTBLP2,NPSF2,PSIFI(4,201),
     9              XBP(2),YBP(2),XMU1,XMU2,CMU
      COMMON/INPT/  DELPS0,DELRO0,ALAMB,ZETAV,AKV(2) ,AMU,C0,C1,C2,
     1              PSLM10,PSLM20,PSLM30,PSLM40,PSLM50,PSLM60,PSLM70,
     2              PSLM80,HED(40),DADE(3),XINPUT(9)
      COMMON/INTGR/JCC(4),JCOR(4),NPSIB,IPSIB0,IPSIB,IBB,IFF,NDTAB,
     1            I9,J9(361,2),  NPSJB,IPSJB,I3,ILAST,IL,IND1,IND2,INDI
     2            ,INDJ,ISTOP,ITZER1,ITZER2,IPZER1,IPZER2,ICOUNT,INDXB,
     3            ICOLLP,ICOLL,IQ,IND,IVEH,LL,ICTVDI,ITCOL,ITFLG1,
     4            ITFLG2,NWRDB1,NWRDE1,NWRDB2,NWRDE2,ICOLAC
C             IN COMMON /COMP/ DO NOT DISTURB THE ORDER OF THE VARIABLES
C                              WHICH ARE EQUIVALENCED TO ARRAYS.
      COMMON/COMP/  DTTRAJ,DTCOLL,DTCOLT,DTPRNT,UVMN2,PSIDMA,TPRINT,DTP,
     1              U1V1SQ,U2V2SQ,DELPSI,DELPST,DELPS2,DELRHO,
     2              PSILM1,PSILM2,PSILM3,PSILM4,PSILM5,PSILM6,PSILM7,
     3              PSILM8,EJJ(4,2),GJJ(4,2),C1OC2,PSCC,PSIR1,PSIR2,
     4             XCMXC1,XCMXC2,YCMYC1,YCMYC2,SPSI1,SPSI2,CPSI1,CPSI2,
     5             XCPI,YCPI,PSII,XCPJ,YCPJ,PSIJ,XCMXC,YCMYC,SPSIJ,
     6             CPSIJ,XFI,XRI,YSI,XFJ,XRJ,YSJ,AKVI,AKVJ,EJ,GJ,
     7             XPJ(4),YPJ(4),ANGYX,SPAN,PSIBPI(4),PSIBPB,PSIBPF,
     8             FNPSIB,PSIBB,PSIB,PSFMPS,SCPSIB,CSPSIB,
     9             RHOBI,RHOBIN(4),PSIBIJ,SPSBIJ,CPSBIJ,XBIJ,YBIJ,PSJTB
      COMMON/COMP/ PSIBPJ,FNPSJB,PSIBJ,RHOBJ,XYJSQR,SPSIB,CPSIB,XBI,YBI,
     1         PRESI,PRESJ,XYSR,DELTA,CRHO,RHOBIC,RHOBMX,RHOBIT,RHOBT,
     2             SPSI21,CPSI21,X2TEM,Y2TEM,XX,YY,PAV,XXYYSR,FN1,SPSF,
     3             CPSF,X1AV,Y1AV,XYAVSR,X2TERM,Y2TERM,SPS21F,CPS21F,
     4             X2AV,Y2AV,VT1AV,VT2AV,VTMVT,FRICT,FNX1,FNY1,FNN1,
     5             FNX2,FNY2,FNN2,SFX1,SFY1,SFN1,SFX2,SFY2,SFN2,
     6              SFX1C,SFX2C,SFY1C,SFY2C,SFN1C,SFN2C,
     7              SFX1T,SFX2T,SFY1T,SFY2T,SFN1T,SFN2T,  TCOL,TEND,
     8              TREST1,TREST2,TCOLS1,TCOLS2,P1TEM,P2TEM,U1V1E,U2V2E,
     9              EXTRA(10)
      COMMON/COMPT/ W(8),XWP(8),YWP(8),FLAGW(8),PSIW(8),TQW(8),
     1             TFX,TFY,TMOM,PSIRR,COEF,COEF1,COEF2,TRD2,TRD21,TRD22,
     2             XBMXB,FNUM,A,B,FIZ,FMASS,XC,YC,U,V,PSIC,PSICD,UD2,
```

```
  3             SGNU,SGNI,AOMB,CSPSIC,SNPSIC,TRCPSC,TRSPSC,TRPSCD,
  4             XI,XW,YW,SX,SY,S,XPP,YPP,RHO,RHOP,XMUS,XMU,PSI,XNUM,
  5             XDEN,RATIO,ALFA,F,TIF,FC,FCOS,SMAL,FSMX,FS,BETB,FX,FY
  6             ,CSPSI,SNPSI,XCXC,YCYC,SPJ,CPJ,SPIJ,CPIJ,XFJJ,XRJJ,
  7              YSJJ,EJJI,GJJI,XICORJ,YICORJ,NTQ,TBTQ,TETQ,TINCRQ,
  8             NPSF,TBPSF,TEPSF,TINCRP
   COMMON/TAB/  TRHOB(361,2),TROBMX(361,2),TPSIB(361,2),
  1             TCPSIB(361,2),TSPSIB(361,2),TXB1(100),TYB1(100),
  2             TPSB1(100),TPRES1(100)
   DIMENSION   XCMXCI(2),YCMYCI(2),SPSII(2),CPSII(2),PSLM0(8),PSLM(8)
   EQUIVALENCE (XCMXCI(1),XCMXC1),(YCMYCI(1),YCMYC1),(SPSII(1),SPSI1)
  1           ,(CPSII(1),CPSI1),(PSLM0(1),PSLM10),(PSLM(1),PSILM1)
   DO 22 IC=1,4
   JCC(IC) = 0
   JCOR(IC) = IC
   TEMPX  =  XPJ(IC)- XCPI
   TEMPY  =  YPJ(IC)- YCPI
   IF(TEMPX) 13,11,13
11 IF(TEMPY) 13,12,13
12 ANGYX = 0.0
   JCC(IC) = IC
   GO TO 17
13 ANGYX = ATAN2(TEMPY,TEMPX)
17 TEMPSI = ANGYX - PSII
   IF (TEMPSI) 19,22,20
19 TEMPSI = TWOPI + TEMPSI
   IF (TEMPSI) 19,22,22
20 IF (TEMPSI - TWOPI) 22,22,21
21 TEMPSI = TEMPSI - TWOPI
   GO TO 20
22 PSIBPI(IC) = TEMPSI
   ILAST = 4
32 ABIG  =  -1.E20
   DO 35  II = 1,ILAST
   IF( PSIBPI(II) - ABIG) 35,34,34
34 ABIG = PSIBPI(II)
   IBIG = II
35 CONTINUE
   IF(IBIG-ILAST) 36,38,38
36 TEMPSI = PSIBPI(ILAST)
   ITEMP  = JCOR(ILAST)
   PSIBPI(ILAST) = ABIG
   JCOR(ILAST)   = JCOR(IBIG)
   PSIBPI(IBIG)  = TEMPSI
   JCOR(IBIG)    = ITEMP
38 ILAST = ILAST - 1
   IF(ILAST-1) 39,39,32
39 IBB = 1
   IFF = 4
   SPAN = PI
   IF(PSIBPI(4) - PI) 70,70,41
41 IF(PSIBPI(1) - PI) 43,70,70
43 IF(PSIBPI(1) - PIO2) 47,44,44
44 IF(PSIBPI(4) - PI32) 70,70,45
45 SPAN = PI32
   GO TO 70
47 IF(PSIBPI(4) - PI32) 48,48,49
48 SPAN = PI32
   GO TO 70
49 IF(PSIBPI(2) - PI) 53,50,50
50 IBB = 2
   IFF = 1
```

```
      SPAN = PI32
      GO TO 70
   53 IF(PSIBPI(3) - PI) 54,54,57
   54 IBB = 4
      IFF = 3
      SPAN = PI32
      GO TO 70
   57 IF(PSIBPI(3) - PI32) 61,58,58
   58 IBB = 3
      IFF = 2
      SPAN = PI32
      GO TO 70
   61 IF(ABS(PSIBPI(3)-PSIBPI(2)) - ABS(PSIBPI(1)+TWOPI-PSIBPI(4))) 62,
     X  62,65
   62 SPAN = TWOPI
      GO TO 70
   65 IBB = 3
      IFF = 2
      SPAN = TWOPI
   70 PSIBPB = PSIBPI(IBB)
      PSIBPF = PSIBPI(IFF)
      IF(PSIBPF + DELPSI - TWOPI) 71,700,700
  700 PSIBPF = 0.0
   71 NPSIB =  PSIBPB/DELPSI  +0.5
      FNPSIB = FLOAT(NPSIB)
      IF (NPSIB) 73,72,73
   72 PSIBB = FLOAT(NDTAB-1) *DELPSI
      IPSIB0 = NDTAB
      GO TO 74
   73 PSIBB = (FNPSIB - 1.) * DELPSI
      IPSIB0 = NPSIB
   74 PSIB =  PSIBB
      IPSIB = IPSIB0
      RETURN
      END
      SUBROUTINE TORDER(NPT,IBB,IFF,TAB,TB1,TB2,TB3)
C          ACCIDENT RECONSTRUCTION - COMBINES TRAJECTORY AND COLLISION
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
      DIMENSION TAB(1),TB1(1),TB2(1),TB3(1)
C
C        TAB HAS ANGLE VALUES IN RADIANS FROM ZERO TO LESS THAN TWOPI
C          IN NO SPECIAL ORDER.
C        THIS SUBROUTINE ORDERS TAB FOR INCREASING VALUES,AND ARRANGES
C           THE CORRESPONDING VALUES IN TB1,TB2 AND TB3 TO CORRESPOND TO
C           THE NEW ORDER OF TAB..
C        THEN THE INDICES FOR CLOCKWISE SWEEP IN TAB ARE FOUND
C        IBB FOR BEGINNING, IFF FOR FINISH.
C        ALL THE TABLES ARE HANDLED IN PLACE, THAT IS, THE ORDER IS
C           CHANGED IN THE ORIGINAL STORAGE LOCATIONS.
C
C
      ILAST = NPT
   10 ABIG = -1. E20
      DO 14 I=1,ILAST
      IF(TAB(I) - ABIG) 14,12,12
   12 ABIG = TAB(I)
      IBIG = I
   14 CONTINUE
      IF(IBIG - ILAST) 16,18,18
   16 TABTEM = TAB(ILAST)
      TB1TEM = TB1(ILAST)
```

```
      TB2TEM = TB2(ILAST)
      TB3TEM = TB3(ILAST)
      TAB(ILAST) = ABIG
      TB1(ILAST) = TB1(IBIG)
      TB2(ILAST) = TB2(IBIG)
      TB3(ILAST) = TB3(IBIG)
      TAB(IBIG) = TABTEM
      TB1(IBIG) = TB1TEM
      TB2(IBIG) = TB2TEM
      TB3(IBIG) = TB3TEM
   18 ILAST = ILAST - 1
      IF(ILAST-1) 19,19,10
   19 IBB = 1
      IFF = NPT
      SPAN = PI
      IF(TAB(NPT) - PI) 70,70,41
   41 IF(TAB(1) - PI) 43,70,70
   43 IF(TAB(1) - PIO2) 47,44,44
   44 IF(TAB(NPT) - PI32) 70,70,45
   45 SPAN = PI32
      GO TO 70
C
   47 IF(TAB(NPT) - PI32) 48,48,49
   48 SPAN = PI32
      GO TO 70
   49 IPI1 = 0
      IPI2 = 0
      DO 51 I=1,NPT
      IF(TAB(I) - PI) 51,50,50
   50 IPI1 = I-1
      GO TO 52
   51 CONTINUE
   52 IF(IPI1) 70,70,53
   53 IPI2 = IPI1 + 1
      IF ( IPI2-NPT) 54,70,70
   54 IF(TAB(IPI2) - PI32) 59,58,58
   58 IBB = IPI2
      IFF = IPI1
      SPAN = PI32
      GO TO 70
   59 IF(TAB(IPI1) - PIO2) 60,60,61
   60 IBB = IPI2
      IFF = IPI1
      SPAN = PI32
      GO TO 70
   61 IF(ABS(TAB(IPI2)-TAB(IPI1))-ABS(TAB(1)+TWOPI-TAB(NPT)))62,62,65
   62 SPAN = TWOPI
      GO TO 70
   65 IBB = IPI2
      IFF = IPI1
      SPAN = TWOPI
   70 RETURN
      END
      SUBROUTINE SETIND(PS1,PS2,XMX,YMY,PSC1,PSC2,IN1,IN2)
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
C           SET INDICATORS TO OMIT THOSE SURFACES THAT ARE WITHIN
C       + OR - 0.10 RAD     OF BEING PARALLEL TO THE LINE CONNECTING THE
C           CENTERS OF GRAVITY OF THE VEHICLES.
C        SAMPLE CALL
C     CALL SETIND(PSI1,PSI2,XCMXC1,YCMYC1, PSCC1,PSCC2,IND1,IND2)
C
      DATA PI/ 3.141592653/, TWOPI/6.2831853072/
```

```
      IF(XMX) 13,11,13
   11 IF(YMY) 13,12,13
   12 PSC1 = 0.0
      PSC2 = 0.0
      GO TO 16
   13 PSC1 = ATAN2(-YMY,-XMX)
      PSC2 = ATAN2( YMY, XMX)
   16 TEMP1 = ABS(PSC1- PS1)
  161 IF(TEMP1 - PI) 17,17,162
  162 TEMP1 = ABS(TWOPI-TEMP1)
      GO TO 161
   17 IF((TEMP1.LE.0.10) .OR. (TEMP1.GE.3.04)) GO TO 18
      IF( TEMP1.GE.0.88 .AND.TEMP1.LE.2.62) GO TO 20
      IN1 =  1
      GO TO 22
   18 IN1 = - 1
      GO TO 22
   20 IN1 = 0
C      IN1 =-1 END ONLY,  =0 SIDE ONLY,  =1 END AND SIDE
C      IN2 =-1 END ONLY,  =0 SIDE ONLY,  =1 END AND SIDE
   22 TEMP1 = ABS(PSC2- PS2)
  221 IF(TEMP1 - PI) 23,23,222
  222 TEMP1 = TWOPI - TEMP1
      GO TO 221
   23 IF((TEMP1.LE.0.10) .OR. (TEMP1.GE.3.04)) GO TO 30
      IF( TEMP1.GE.0.88 .AND.TEMP1.LE.2.62) GO TO 32
      IN2 = 1
      GO TO 34
   30 IN2 = -1
      GO TO 34
   32 IN2 = 0
   34 RETURN
      END
```

```
      SUBROUTINE CNSTNT
C       SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
      COMMON/INTG/NEQ,T,DT,VAR(12),DER(12)
      EQUIVALENCE (VAR(1),XCP1),(VAR(2),YCP1),(VAR(3),PSI1),
     1            (VAR(4),PSI1D),(VAR(5),U1),(VAR(6),V1),
     2            (VAR(7),XCP2),(VAR(8),YCP2),(VAR(9),PSI2),
     3            (VAR(10),PSI2D),(VAR(11),U2),(VAR(12),V2)
      EQUIVALENCE (DER(1),DXCP1),(DER(2),DYCP1),(DER(3),DPSI1),
     1            (DER(4),DPSI1D),(DER(5),DU1),(DER(6),DV1),
     2            (DER(7),DXCP2),(DER(8),DYCP2),(DER(9),DPSI2),
     3            (DER(10),DPSI2D),(DER(11),DU2),(DER(12),DV2)
      COMMON/INPT/  T0,TF,DTTRA0,DTCOL0,DTCLT0,DTPRN0,UVMIN,PSIDMN,FVEH0
     1             ,XCP10,YCP10,PSI10,PSI1D0,U10,V10,
     2              XCP20,YCP20,PSI20,PSI2D0,U20,V20,
     3              A1,B1,TR1,FIZ1,FM1,PSIR10,XF1,XR1,YS1,
     4              A2,B2,TR2,FIZ2,FM2,PSIR20,XF2,XR2,YS2,
     5              CSTFI(8),TBTQ1,TETQ1,TINCQ1,NTBLQ1,NTQ1,
     6              TBTQ2,TETQ2,TINCQ2,NTBLQ2,NTQ2,TII(8,201),
     7              TBPSF1,TEPSF1,TINCP1,NTBLP1,NPSF1,
     8              TBPSF2,TEPSF2,TINCP2,NTBLP2,NPSF2,PSIFI(4,201),
     9              XBP(2),YBP(2),XMU1,XMU2,CMU
      COMMON/INPT/  DELPS0,DELRO0,ALAMB,ZETAV,AKV(2) ,AMU,C0,C1,C2,
     1              PSLM10,PSLM20,PSLM30,PSLM40,PSLM50,PSLM60,PSLM70,
     2              PSLM80,HED(40),DADE(3),XINPUT(9)
      COMMON/INTGR/JCC(4),JCOR(4),NPSIB,IPSIB0,IPSIB,IBB,IFF,NDTAB,
     1            I9,J9(361,2),  NPSJB,IPSJB,I3,ILAST,IL,IND1,IND2,INDI
     2            ,INDJ,ISTOP,ITZER1,ITZER2,IPZER1,IPZER2,ICOUNT,INDXB,
     3            ICOLLP,ICOLL,IQ,IND,IVEH,LL,ICTVDI,ITCOL,ITFLG1,
     4            ITFLG2,NWRDB1,NWRDE1,NWRDB2,NWRDE2,ICOLAC
C            IN COMMON /COMP/ DO NOT DISTURB THE ORDER OF THE VARIABLES
C                             WHICH ARE EQUIVALENCED TO ARRAYS.
      COMMON/COMP/  DTTRAJ,DTCOLL,DTCOLT,DTPRNT,UVMN2,PSIDMA,TPRINT,DTP,
     1              U1V1SQ,U2V2SQ,DELPSI,DELPST,DELPS2,DELRHO,
     2              PSILM1,PSILM2,PSILM3,PSILM4,PSILM5,PSILM6,PSILM7,
     3              PSILM8,EJJ(4,2),GJJ(4,2),C1OC2,PSCC,PSIR1,PSIR2,
     4            XCMXC1,XCMXC2,YCMYC1,YCMYC2,SPSI1,SPSI2,CPSI1,CPSI2,
     5            XCPI,YCPI,PSII,XCPJ,YCPJ,PSIJ,XCMXC,YCMYC,SPSIJ,
     6            CPSIJ,XFI,XRI,YSI,XFJ,XRJ,YSJ,AKVI,AKVJ,EJ,GJ,
     7            XPJ(4),YPJ(4),ANGYX,SPAN,PSIBPI(4),PSIBPB,PSIBPF,
     8            FNPSIB,PSIBB,PSIB,PSFMPS,SCPSIB,CSPSIB,
     9            RHOBI,RHOBIN(4),PSIBIJ,SPSBIJ,CPSBIJ,XBIJ,YBIJ,PSJTB
      COMMON/COMP/ PSIBPJ,FNPSJB,PSIBJ,RHOBJ,XYJSQR,SPSIB,CPSIB,XBI,YBI,
     1         PRESI,PRESJ,XYSR,DELTA,CRHO,RHOBIC,RHOBMX,RHOBIT,RHOBT,
     2             SPSI21,CPSI21,X2TEM,Y2TEM,XX,YY,PAV,XXYYSR,FN1,SPSF,
     3             CPSF,X1AV,Y1AV,XYAVSR,X2TERM,Y2TERM,SPS21F,CPS21F,
     4             X2AV,Y2AV,VT1AV,VT2AV,VTMVT,FRICT,FNX1,FNY1,FNN1,
     5             FNX2,FNY2,FNN2,SFX1,SFY1,SFN1,SFX2,SFY2,SFN2,
     6              SFX1C,SFX2C,SFY1C,SFY2C,SFN1C,SFN2C,
     7              SFX1T,SFX2T,SFY1T,SFY2T,SFN1T,SFN2T,  TCOL,TEND,
     8              TREST1,TREST2,TCOLS1,TCOLS2,P1TEM,P2TEM,U1V1E,U2V2E,
     9              EXTRA(10)
      COMMON/COMPT/ W(8),XWP(8),YWP(8),FLAGW(8),PSIW(8),TQW(8),
     1            TFX,TFY,TMOM,PSIRR,COEF,COEF1,COEF2,TRD2,TRD21,TRD22,
     2            XBMXB,FNUM,A,B,FIZ,FMASS,XC,YC,U,V,PSIC,PSICD,UD2,
     3            SGNU,SGNI,AOMB,CSPSIC,SNPSIC,TRCPSC,TRSPSC,TRPSCD,
     4            XI,XW,YW,SX,SY,S,XPP,YPP,RHO,RHOP,XMUS,XMU,PSI,XNUM,
     5            XDEN,RATIO,ALFA,F,TIF,FC,FCOS,SMAL,FSMX,FS,BETB,FX,FY
     6            ,CSPSI,SNPSI,XCXC,YCYC,SPJ,CPJ,SPIJ,CPIJ,XFJJ,XRJJ,
     7             YSJJ,EJJI,GJJI,XICORJ,YICORJ,NTQ,TBTQ,TETQ,TINCRQ,
     8            NPSF,TBPSF,TEPSF,TINCRP
```

```
      COMMON/TAB/  TRHOB(361,2),TROBMX(361,2),TPSIB(361,2),
     1             TCPSIB(361,2),TSPSIB(361,2),TXB1(100),TYB1(100),
     2             TPSB1(100),TPRES1(100)
      DIMENSION   XCMXCI(2),YCMYCI(2),SPSII(2),CPSII(2),PSLM0(8),PSLM(8)
      EQUIVALENCE (XCMXCI(1),XCMXC1),(YCMYCI(1),YCMYC1),(SPSII(1),SPSI1)
     1           ,(CPSII(1),CPSI1),(PSLM0(1),PSLM10),(PSLM(1),PSILM1)
C        COMMON VDICON HAS CONSTANTS USED FOR VEHICLE DAMAGE INDEX
      COMMON/VDICON/EXTF1( 9),EXTR1( 9),EXTB1( 9),PSP1(12),
     2              EXTF2( 9),EXTR2( 9),EXTB2( 9),PSP2(12),NCOL51(10),
     2              NCOL52(10),MMM( 9)
      EQUIVALENCE (XINPUT(1),FMOVIE)
      COMMON /MOVIE/ MOVIE,NMSG
      DATA ME/1HE/
      DATA M1/1H1/, M2/1H2/, M3/1H3/, M4/1H4/, M5/1H5/, M6/1H6/
      DATA M7/1H7/, M8/1H8/, M9/1H9/
C
C      TIME INTERVALS AND STOPPING TESTS
      DTTRAJ = DTTRA0
      DTCOLL = DTCOL0
      DTCOLT = DTCLT0
      DTPRNT = DTPRN0
      UVMN2 = UVMIN * ABS(UVMIN)
      PSIDMA = PSIDMN * RAD
C      TERRAIN BOUNDARIES
      INDXB = 1
      XBMXB = XBP(2) - XBP(1)
      IF(XBMXB) 11,10,11
   10 INDXB = 0
      XPP = XBP(1)
      GO TO 12
   11 XI = (YBP(2)-YBP(1))/ XBMXB
      FNUM =YBP(1) - XI*XBP(1)
   12 XCP1 = XCP10
      YCP1 = YCP10
      PSI1 = PSI10 * RAD
      PSI1D = PSI1D0 * RAD
      U1   =U10
      V1   = V10
      PSIR1 = PSIR10*RAD
      TRD21 = TR1/2.0
      COEF1 = FM1*G /(2.0*(A1+B1))
      W(1) = B1* COEF1
      W(3) = A1* COEF1
      W(2) = W(1)
      W(4) = W(3)
C          IVEH = 0 FOR ONE VEHICLE,  =1 FOR TWO VEHICLES
      IVEH = FVEH0 + 0.2
      IVEH = IVEH - 1
      IF(IVEH) 18,15,18
   15 NEQ = 6
      IF(FMOVIE) 16,17,16
   16 STOP
C      CANNOT HAVE COLLISON DAMAGE IN THIS PGM WITH ONLY ONE VEHICLE
   17 RETURN
   18 NEQ = 12
C         NEQ INTEGRATES 6 EQUATIONS FOR ONE VEHICLE, 12 EQ FOR TWO VEH.
      XCP2 = XCP20
      YCP2 = YCP20
      PSI2 = PSI20 * RAD
      PSI2D = PSI2D0 * RAD
      U2   = U20
      V2   = V20
```

```
      PSIR2 = PSIR20*RAD
      TRD22 = TR2/2.0
      COEF2 = FM2*G /(2.0*(A2+B2))
      W(5) = B2*COEF2
      W(7) = A2*COEF2
      W(6) = W(5)
      W(8) = W(7)
C          CORNER IDENTIFICATION
C          1   RF
C          2   LF
C          3   RR
C          4   LR
C          VEHICLE 1
      EJJ(1,1)  = XF1
      EJJ(2,1)  = XF1
      EJJ(3,1)  = XR1
      EJJ(4,1)  = XR1
      GJJ(1,1)  = YS1
      GJJ(2,1)  =-YS1
      GJJ(3,1)  = YS1
      GJJ(4,1)  =-YS1
C          VEHICLE 2
      EJJ(1,2)  = XF2
      EJJ(2,2)  = XF2
      EJJ(3,2)  = XR2
      EJJ(4,2)  = XR2
      GJJ(1,2)  = YS2
      GJJ(2,2)  =-YS2
      GJJ(3,2)  = YS2
      GJJ(4,2)  =-YS2
C               XR1 AND XR2 ARE INPUT AS NEGATIVE QUANTITIES
      DELRHO = DELRO0
      DELPSI = DELPS0 * RAD
      DELPST =  0.01*DELPSI
      DELPS2 = 0.5 * DELPSI
      NDTAB = TWOPI/DELPSI  + 1.01
C          TABLES I AND J  HAVE NDTAB POSSIBLE ENTRIES
      C1OC2 = 0.5 * C1/C2
      PSILM1 = PSLM10 *RAD
      PSILM2 = PSLM20 *RAD
      PSILM3 = PSLM30 *RAD
      PSILM4 = PSLM40 *RAD
      PSILM5 = PSLM50 *RAD
      PSILM6 = PSLM60 *RAD
      PSILM7 = PSLM70 *RAD
      PSILM8 = PSLM80 *RAD
C
   26 IF(FMOVIE) 27,28,27
   27 MOVIE = 1
      NMSG = 2
      GO TO 29
C        MOVIE NE 0 WRITES DAMAGE TABLES ON TAPE AND ON FT02 AT EACH
C        TIME POINT
   28 MOVIE = 0
      NMSG = NOUT
C             EXTF1 AND SO ON USED FOR VEHICLE DAMAGE INDEX
   29 EXTF1(1) = 0.0
      EXTF1(2) = 0.125 * XF1
      EXTF1(3) = 0.250 * XF1
      EXTF1(4) = 0.375 * XF1
      EXTF1(5) = 0.500 * XF1
      EXTF1(6) = 0.625 * XF1
```

```
      EXTF1(7) = 0.846 * XF1
      EXTF1(8) = 0.946 * XF1
      EXTF1(9) = 1.046 * XF1
      EXTF2(1) = 0.0
      EXTF2(2) = 0.125 * XF2
      EXTF2(3) = 0.250 * XF2
      EXTF2(4) = 0.375 * XF2
      EXTF2(5) = 0.500 * XF2
      EXTF2(6) = 0.625 * XF2
      EXTF2(7) = 0.846 * XF2
      EXTF2(8) = 0.946 * XF2
      EXTF2(9) = 1.046 * XF2
      EXTR1(1) = 0.0
      EXTR1(2) = 0.165 * YS1
      EXTR1(3) = 0.253 * YS1
      EXTR1(4) = 0.502 * YS1
      EXTR1(5) = 0.751 * YS1
      EXTR1(6) = 1.000 * YS1
      EXTR1(7) = 1.249 * YS1
      EXTR1(8) = 1.498 * YS1
      EXTR1(9) = 1.747 * YS1
      EXTR2(1) = 0.0
      EXTR2(2) = 0.165 *YS2
      EXTR2(3) = 0.253 *YS2
      EXTR2(4) = 0.502 *YS2
      EXTR2(5) = 0.751 *YS2
      EXTR2(6) = 1.000 *YS2
      EXTR2(7) = 1.249 *YS2
      EXTR2(8) = 1.498 *YS2
      EXTR2(9) = 1.747 *YS2
      EXTB1(1) = 0.0
      EXTB1(2) = -0.084 * XR1
      EXTB1(3) = -0.168 * XR1
      EXTB1(4) = -0.252 * XR1
      EXTB1(5) = -0.336 * XR1
      EXTB1(6) = -0.421 * XR1
      EXTB1(7) = -0.588 * XR1
      EXTB1(8) = -0.769 * XR1
      EXTB1(9) = -0.950 * XR1
      EXTB2(1) = 0.0
      EXTB2(2) = -0.084 * XR2
      EXTB2(3) = -0.168 * XR2
      EXTB2(4) = -0.252 * XR2
      EXTB2(5) = -0.336 * XR2
      EXTB2(6) = -0.421 * XR2
      EXTB2(7) = -0.588 * XR2
      EXTB2(8) = -0.769 * XR2
      EXTB2(9) = -0.950 * XR2
C        ANGLES FOR COMPARISIONS IN VEHICLE DAMAGE INDEX (VDI)
      PSP1(1)= ATAN2(0.500*YS1,XF1)
      PSP1(2)= ATAN2(YS1,XF1)
      PSP1(3)= ATAN2(2.702*YS1,XF1)
      PSP1(4)= ATAN2(1.754*YS1,XR1)
      PSP1(5)= ATAN2(YS1,XR1)
      PSP1(6)= ATAN2(0.500*YS1,XR1)
C      PSP1(1) THRU PSP1(3) ARE IN FIRST QUADRANT
C      PSP1(4) THRU PSP1(6) ARE IN SECOND QUADRANT
C      PSP1(7) THRU PSP1(9) ARE IN THIRD QUADRANT , USES -YS1
C      PSP1(10)THRU PSP1(12)  ARE IN FOURTH QUADRANT,USES -YS1
C                XR1 AND XR2 ARE INPUT AS NEGATIVE QUANTITIES
      PSP1(12)= TWOPI - PSP1(1)
      PSP1(11)= TWOPI - PSP1(2)
```

```
      PSP1(10)= TWOPI - PSP1(3)
      PSP1(9) = TWOPI - PSP1(4)
      PSP1(8) = TWOPI - PSP1(5)
      PSP1(7) = TWOPI - PSP1(6)
      PSP2(1)= ATAN2(0.500*YS2,XF2)
      PSP2(2)= ATAN2(YS2,XF2)
      PSP2(3) = ATAN2(2.702*YS2,XF2)
      PSP2(4)= ATAN2(1.754*YS2,XR2)
      PSP2(5)= ATAN2(YS2,XR2)
      PSP2(6) = ATAN2(0.500*YS2,XR2)
      PSP2(12)=TWOPI - PSP2(1)
      PSP2(11)=TWOPI - PSP2(2)
      PSP2(10)=TWOPI - PSP2(3)
      PSP2(9) =TWOPI - PSP2(4)
      PSP2(8) =TWOPI - PSP2(5)
      PSP2(7) =TWOPI - PSP2(6)
C           VERTICAL DAMAGE LOCATION NOT APPLICABLE (NCOL51 AND NCOL52)
      DO 40 I=1,10
      NCOL51(I) =ME
   40 NCOL52(I) =ME
      MMM(1) = M1
      MMM(2) = M2
      MMM(3) = M3
      MMM(4) = M4
      MMM(5) = M5
      MMM(6) = M6
      MMM(7) = M7
      MMM(8) = M8
      MMM(9) = M9
      RETURN
      END
```

```
      SUBROUTINE RNGDAM(TRHO,TPSI,TCPSI,TSPSI,JJ9,ROU,PSBOU,XOU,YOU,ASTR
     1  ,PSP,PSDA1,PSDA2,PSIM,INPSM,IKD1,IKD2,NARRP,NUM,IKK,IRNG,IWRAP,
     2               IJ,ISTOP,NDTAB,DELPS,XF,XR,YS,NMSG1,IPRTX)
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
      DIMENSION  TRHO(1),TPSI(1),TCPSI(1),TSPSI(1),JJ9(1)
C            ABOVE ARE CALLED FROM PROPER BEGINNING IN DOUBLE-DIMENSION
C            ARRAYS, SUCH AS TRHOB(1,1) OR TRHOB(1,2) FOR TRHOB(361,2)
      DIMENSION  ROU(1),PSBOU(1),XOU(1),YOU(1),ASTR(1),PSIM(1),INPSM(1)
      DIMENSION PSP(1),PSDA1(1),PSDA2(1)
      DIMENSION  IKD1(1),IKD2(1),NARRP(1)
      DATA  TWOPI/6.2831853072/,PIO4/0.7853981634/,PIO6/0.5235987756/
      DATA  PIO12/0.2617993878/
      DATA STAR,BLNK/1H*,1H /
C     SAMPLE CALL
C               CALL  RNGDAM(TRHOB(1,1),TPSIB(1,1),TCPSIB(1,1),
C    1     TSPSIB(1,1),J9(1,1),R1OU,PSB1OU,X1OU,Y1OU,ASTR1,PSP1,PSD11,
C    2     PSD21,PSIM1,INPSM1,IKD11,IKD21,NARRP1,NUM1,IK1,IRNG1,IWRAP1,
C    3               IJ,ISTOP,NDTAB,DELPSI,XF1,XR1,YS1,NMSG, IPRTX)
C
C          SUBROUTINE RNGDAM SAVES THE DAMAGE LOCATIONS, DETERMINES THE
C          RANGE OF DAMAGE AND MIDPOINT OF RANGE (UP TO TEN RANGES)
C          AND MATCHES THE MIDPOINT WITH A DAMAGE INDEX 'N' FROM NARRP
C        NMSG1  DATA SET FOR DAMAGE TABLES (FT02) SET IN SUB CNSTNT,
C        RESET IN MAIN PROG AT END OF RUN TO USE FT06
C        IPRTX EQ 2 AT END OF RUN
C        RNGDAM IS CALLED WITH IPRTX EQ 1 WHEN MOVIE NE 0
C        TO PREPARE DAMAGE TABLES FOR WRITING ON TAPE AT EACH TIME POINT
C
      DO  5 K=1,NDTAB
      ASTR(K) = BLNK
      ROU(K)  = 0.0
      PSBOU(K) =0.0
      XOU(K)   =0.0
    5 YOU(K)   =0.0
      IKK = 0
      IK=0
      IWRAP = 0
      IRNG = 0
      ISAVE= 0
C          IJ =1 OR 2, VEHICLE IDENTIFICATION
C           IKK IS THE NUMBER OF DAMAGE POINTS IN ARRAYS ROU,PSBOU,XOU,
C           YOU, AND ASTR
C           ARRAY IKD1 SAVES THE INDEX OF DAMAGE POINTS WHICH BEGIN
C           RANGES OF DAMAGE.  NUMBER OF ENTRIES IS IRNG.
C           ARRAY IKD2 SAVES THE INDEX OF DAMAGE POINTS WHICH END
C           THESE RANGES OF DAMAGE , NUMBER OF ENTRIES IS IRNG.
C           IWRAP INDICATE THAT ONE RANGE PASSES THROUGH FROM 4TH TO 1ST
C             QUADRANT.
C           ISAVE IS SIGNAL THAT A POSSIBLE RANGE HAS BEGUN.
C           FOR VEHICLE DAMAGE INDEX COL 6 AND 7 USE ORIGINAL ENDPOINTS.
C           FOR VEHICLE DAMAGE INDEX COL 3 AND 4 USE ADJUSTED ENDPOINTS.
C           SUBROUTINE ADJEND  COMPUTES ADJUSTED ENDPOINTS
      DO 22 K= 1,NDTAB
      RHTEST = TRHO(K)
      IF(RHTEST) 15,10,10
   10 IK = IK + 1
      IF(ISAVE) 12,11,12
   11 ISAVE = 1
      IRNG = IRNG + 1
      IKD1(IRNG) = IK
```

```
   12 ROU(IK) = RHTEST
      PSBOU(IK) = TPSI(K)
      XOU(IK)   = RHTEST * TCPSI(K)
      YOU(IK)   = RHTEST * TSPSI(K)
      IF(JJ9(K)) 13, 22,22
   13 ASTR(IK) = STAR
      GO TO 22
   15 IF(ISAVE) 16,22,16
   16 ISAVE = 0
      IKD2(IRNG) = IK
   22 CONTINUE
      IKK = IK
      IF(IKD2(IRNG)) 24,23,24
   23 IKD2(IRNG) = IKK
   24 IF(PSBOU(IKD1(1)) ) 30,25,30
   25 TTTPSE =PSBOU(IKD2(IRNG))
      TTT = TWOPI - TTTPSE
      IF(ABS(TTT-DELPS) -0.25*DELPS) 26,26,30
   26 IKD1 (1)  =  IKD1(IRNG)
      IKD2 (IRNG) = IKD2(1)
      IWRAP = 1
      IRNG = IRNG - 1
      IF(IRNG -1) 27,30,30
   27 IRNG = 1
C
C             TEMPORARY PRINT
   30 WRITE(NMSG1,628)
  628 FORMAT(42H0 FOLLOWING MESSAGE FROM SUBROUTINE RNGDAM)
      WRITE(NMSG1 ,629) IJ,IJ,IJ,IRNG,IWRAP,(I,IKD1(I),IKD2(I),I=1,IRNG)
  629 FORMAT(9H  VEH.NO.,I2,       7X,2H I,11X,4HIKD1,I1,3H(I),5X,4HIKD2
     1      ,I1,3H(I),6H,IRNG=,I3,5X,7H IWRAP=,I3/(16X,I4,12X,I4,9X,I4))
      IF(IPRTX .LT. 2) RETURN
      JJ=1
      CALL ADJEND(PSBOU, XOU, YOU, ASTR, PSP, PSDA1,XF, XR, YS, IKD1,
     1            IRNG,JJ,IJ)
      JJ=2
      CALL ADJEND(PSBOU, XOU, YOU, ASTR, PSP, PSDA2,XF, XR, YS, IKD2,
     1            IRNG,JJ,IJ)
      IB = 1
      IF(IWRAP) 31,35,31
   31 IF(PSDA1(1) - PIO6) 310,310,311
  310 WRITE(6,6310) PSBOU(IKD1(1)),PSDA1(1),IRNG
 6310 FORMAT(66H0 IN SUBROUTINE RNGDAM,WRAP-AROUND,ORIGINAL BEG PT OF FI
     1RST RANGE=,E13.5, 12H,  ADJ. PT =,E13.5/16H NO WRAP,  IRNG=,I3,
     3  16H LOSES ONE RANGE)
      GO TO 35
C
C  31 PSIM(1) = 0.5*((PSBOU(IKD1(1))- TWOPI) + PSBOU(IKD2(1)))
  311 PSIM(1) = 0.5*((PSDA1(1) - TWOPI) + PSDA2(1))
      IF(PSIM(1) -PIO12) 32,32,33
   32 PSIM(1) = PSIM(1) + TWOPI
   33 IF (IRNG-1) 45,45,34
   34 IB=2
   35 DO 40 I = IB,IRNG
   40 PSIM(I) = 0.5*( PSDA1(I) + PSDA2(I))
C          NUM=0 FROM SUBROUTINE SAVMAX WHEN NO ACCEL .GE. 1.0 G,
C          THEN NARRP AND INPSM NOT EVALUATED
   45 IF(NUM) 46,66,46
   46 DO 65 J= 1,IRNG
      CRIT = 0.0
      PSIMT = PSIM(J)
C               PIO12 RAD = 15 DEG
```

```
   48 CRIT = CRIT + PIO12
      IMATCH = 0
      DO 60 I= 1,NUM
      PSTRY = PIO6*FLOAT(NARRP(I))
      TEST = PSIMT - PSTRY
   54 IF( ABS(TEST) - CRIT) 55,55,60
   55 IMATCH = I
      GO TO 64
   60 CONTINUE
      IF(CRIT - PIO4)        48,48,62
C     IF(CRIT -(PIO4-0.0001))48,48,62
   62 ISTOP = 30
      WRITE(6,628)
      WRITE(6,662) ISTOP,IJ,NUM,J,PSIMT,IRNG,PSTRY,TEST,CRIT,
     1             (NARRP(III),III=1,NUM)
  662 FORMAT(8H0 ISTOP=,I3, 26H  NO MATCH FOR VEHICLE NO.,I2,
     1 10H FROM NUM=,I3, 16H VALUES IN NARRP/
     2 10H  AT PSIM(,I3,2H)=, E13.5,11H FROM IRNG=,I3,21H RANGES,  NARRP
     3*PIO6=,E13.5, 7H, TEST=,E13.5, 7H, CRIT=,E13.5 /
     454H USE FIRST VALUE OF NARRP(TESTED AS NARRP*PIO6),NARRP=, 10I4)
      IMATCH = 1
   64 INPSM(J) = IMATCH
C           ARRAY INPSM STORES INDEX OF MATCH FROM NARRP ARRAY
C                      AND IS ITSELF INDEXED TO CORRESPOND TO PSIM ARRAY
   65 CONTINUE
   66 IF(PSIM(1) - TWOPI) 68,67,67
   67 PSIM(1) = PSIM(1) - TWOPI
   68 RETURN
      END
      SUBROUTINE ADJEND(PSBOU,XOU,YOU,ASTR,PSP,PSDD,XF,XR,YS,IKD,IRNG,
     1                  JJ,IJ)
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
C      FOR VEHICLE DAMAGE INDEX, COLUMNS 3 AND 4
C            JJ=1 FOR BEGINNING OF RANGE, JJ=2 FOR LAST POINT OF RANGE
C            IJ= 1 OR 2, VEHICLE IDENTIFICATION
C            CALL ADJEND FROM SUBROUTINE RNGDAM TWICE,ONCE FOR BEGINNING
C             AND ONCE FOR LAST POINTS.
C             SUBROUTINE RNGDAM IS ITSELF CALLED TWICE,ONCE PER VEHICLE.
C          SAMPLE CALL FOR BEGINNING OF ALL DAMAGE RANGES FOR VEHICLE 1
C     JJ=1
C     CALL ADJEND(PSBOU, XOU, YOU, ASTR, PSP, PSDA1,XF, XR, YS, IKD1,
C    1            IRNG,JJ,IJ)
C          SAMPLE CALL FOR LAST POINT OF ALL DAMAGE RANGES FOR VEHICLE 1
C     JJ=2
C     CALL ADJEND(PSBOU, XOU, YOU, ASTR, PSP, PSDA2,XF, XR, YS, IKD2,
C    1            IRNG,JJ,IJ)
      DIMENSION PSBOU(1),XOU(1),YOU(1),ASTR(1),PSP(1),PSDD(1),IKD(1)
      DATA TWOPI/6.2831853072/
      DATA RAD/0.0174532925/
      DATA NBEG/4H BEG/, NLAST/4HLAST/
      DATA STAR/1H*/
      DO 30 I = 1,IRNG
      II = IKD(I)
      PSD = PSBOU(II)
      IF( ASTR(II) - STAR) 11,12,11
   11 PSDD(I) = PSD
      GO TO 30
   12 IF(PSD - PSP(2)) 14,14,13
   13 IF(PSD - PSP(11)) 15,14,14
   14 PSDTEM = ATAN2(YOU(II), XF)
      GO TO 25
   15 IF(PSD - PSP(5)) 18,16,16
```

```
   16 IF(PSD - PSP(8)) 17,17,19
   17 PSDTEM = ATAN2(YOU(II),XR)
      GO TO 25
   18 YSSIDE = YS
      GO TO 20
   19 YSSIDE = - YS
   20 PSDTEM = ATAN2( YSSIDE,XOU(II))
   25 IF(PSDTEM) 26,27,27
   26 PSDTEM = TWOPI + PSDTEM
   27 PSDD(I)  = PSDTEM
   30 CONTINUE
      IF(JJ-1) 35,35,36
   35 NWORD = NBEG
      GO TO 37
   36 NWORD = NLAST
   37 WRITE(6,6038) IJ,NWORD,IRNG
 6038 FORMAT(32H0 MESSAGE FROM SUBROUTINE ADJEND / 2X,8H VEH.NO.,I2,2X,
     1 A4, 11H POINTS FOR , I3, 7H RANGES / 8X,1HI,4X,10H ORIG(DEG),4X,
     2 14H ADJUSTED(DEG) )
      DO 40 I=1,IRNG
      PSDP = PSBOU(IKD(I)) / RAD
      PSDPA = PSDD(I) /RAD
      WRITE(6,6040) I,PSDP,PSDPA
 6040 FORMAT(6X,I3,1X,E13.5,3X,E13.5)
   40 CONTINUE
      RETURN
      END
      SUBROUTINE DAMAGE(PSIM,IRNG ,PSP,PSBOU,XOU,YOU,ASTR,IKD1,IKD2,IKK,
     1                  EXTF,EXTR,EXTB,XF,XR,YS,NCOL3,NCOL4,NCOL6,NCOL7,
     2                  MMM,ISTOP,PSDA1,PSDA2,ARRPSI,INPSM,IJ)
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
      DIMENSION PSIM(1),PSP(1),PSBOU(1),XOU(1),YOU(1),ASTR(1),IKD1(1)
      DIMENSION  IKD2(1),EXTF(1),EXTR(1),EXTB(1),NCOL3(1),NCOL4(1)
      DIMENSION NCOL6(1),NCOL7(1), MMM(1),PSDA1(1),PSDA2(1)
      DIMENSION ARRPSI(1),INPSM(1)
C        ARRAY NCOL5 IS OMITTED FROM ARG. LIST.   ZEROED IN SUBR.CNSTNT
C             IJ IS VEHICLE  IDENTIFICATION NUMBER, FOR PRINTING
C
C     SAMPLE CALL
C     IJ = 1
C               CALL  DAMAGE(PSIM1,IRNG1,PSP1,PSB1OU,X1OU,Y1OU,ASTR1,
C    1          IKD11,IKD21,IK1,EXTF1,EXTR1,EXTB1,XF1,XR1,YS1,NCOL31,
C    2   NCOL41,NCOL61,NCOL71,MMM,ISTOP,PSD11,PSD21,ARRPS1,INPSM1,IJ)
C
      DATA  MF/1HF/,MR/1HR/,MB/1HB/,ML/1HL/
      DATA MY/1HY/,MD/1HD/, MC/1HC/, MZ/1HZ/, MP/1HP/, MS/1HS/,ME/1HE/
      DATA MN/1HN/,MW/1HW/
      DATA  TWOPI/6.2831853072/
      DATA PI/3.1415926536/
C
      DO  505 I= 1,IRNG
      INDAPS = 0
C
      YMNFUL = 0.0
      YMXFUL = 0.0
      XMNFUL = 0.0
      XMXFUL = 0.0
      IB=IKD1(I)
      IL = IKD2(I)
      IF(IB -IL) 230,505,232
C          DO NOT COMPUTE COLS 3,4,  6,7, FOR RANGES WHICH CONSIST OF A
C                SINGLE POINT.       IB=IL
```

```
  230 IBB = IB
      ILB = IL
      IBE = 0
      ILE = 0
      GO TO 233
  232 IBB = 1
      ILB = IL
      IBE = IB
      ILE = IKK
  233 PSIMT = PSIM(I)
      PSD1 = PSDA1(I)
      PSD2 = PSDA2(I)
      IF(PSD1)241,243,240
  240 IF(PSD1-TWOPI)243,243,241
  241 ISTOP = 41
      GO TO 250
  243 IF(PSD2) 245,247,244
  244 IF(PSD2-TWOPI)247,247,245
  245 ISTOP = 42
      GO TO 250
  247 IF(PSIMT)249,310,248
  248 IF(PSIMT-TWOPI) 251,310,249
  249 ISTOP= 43
  250 WRITE(6,6250) ISTOP, I, PSD1, PSD2 ,PSIMT
 6250 FORMAT(8H0 ISTOP=,I3,5X,3H I=,I3, 5X,6H PSD1=,E13.5,5X,6H PSD2=
     1        E13.5,5X,7H PSIMT=,E13.5)
      GO TO 505
  251 IF(PSD1 - PSP(2)) 253,253,252
  252 IF(PSD1 - PSP(11)) 257,253,253
  253 IF(PSD2 - PSP(2))  255,255,254
  254 IF(PSD2 - PSP(11)) 266,255,255
  255 ICORTB = 0
      GO TO 310
  257 IF(PSD1 - PSP(5))  258,260,260
  258 IF(PSD2 - PSP(2))  266,266,2581
 2581 IF(PSD2 - PSP(5))  259,266,266
  259 ICORTB = 0
      GO TO 320
  260 IF(PSD1 - PSP(8))  261,261,263
  261 IF(PSD2 - PSP(5))  266,2611,2611
 2611 IF(PSD2 - PSP(8))  262,262,266
  262 ICORTB = 0
      GO TO 330
  263 IF(PSD2 - PSP(8)) 266,266,264
  264 IF(PSD2 - PSP(11)) 265,266,266
  265 ICORTB = 0
      GO TO 340
  266 ICORTB = 1
      PSTRY = ARRPSI(INPSM(I)) + PI
 2661 IF(PSTRY) 2662,2665,2663
 2662 PSTRY = PSTRY + TWOPI
      GO TO 2661
C
 2663 IF(PSTRY-TWOPI) 2665,2665,2664
 2664 PSTRY = PSTRY - TWOPI
      GO TO 2663
 2665 PSCRIT = PSIMT - PSTRY
 2670 IF(PSCRIT + PI) 2671,2671,2672
 2671 PFLIP =  TWOPI
      GO TO 2674
 2672 IF(PSCRIT - PI) 2675,2673,2673
 2673 PFLIP = - TWOPI
```

```
 2674 PSCRIT = PSCRIT + PFLIP
      GO TO 2670
 2675 PSTEST=PSD2 - PSIMT
 2680 IF(PSTEST + PI) 2681,2681,2682
 2681 PFLIP =  TWOPI
      GO TO 2684
 2682 IF(PSTEST - PI) 2685,2683,2683
 2683 PFLIP = - TWOPI
 2684 PSTEST = PSTEST + PFLIP
      GO TO 2680
 2685 IF(ABS(PSCRIT) - ABS(PSTEST)) 2695,272,272
 2695 IF(ABS(PSCRIT) - 0.242) 272,272,2700
 2700 IF(PSTRY - PSP(2))  2704,2704,2701
 2701 IF(PSTRY - PSP(11)) 2702,2704,2704
 2702 IF(PSTRY - PSP( 5)) 2706,2703,2703
 2703 IF(PSTRY - PSP( 8)) 2708,2708,2710
 2704 IF( PSIMT - PSP( 2)) 310,310,2705
 2705 IF( PSIMT - PSP(11)) 2712,310,310
 2706 IF( PSIMT - PSP( 2)) 2713,2713,2707
 2707 IF( PSIMT - PSP( 5))  320,2713,2713
 2708 IF( PSIMT - PSP( 5)) 2714,2709,2709
 2709 IF( PSIMT - PSP( 8))  330,  330,2714
 2710 IF( PSIMT - PSP(11)) 2711,2715,2715
 2711 IF( PSIMT - PSP( 8)) 2715,2715,340
C       SET INDAPS
 2712 INDAPS = SIGN(1.1,PSCRIT)
      GO TO 310
 2713 INDAPS = SIGN(1.1,PSCRIT)
      GO TO 320
 2714 INDAPS = SIGN(1.1,PSCRIT)
      GO TO 330
 2715 INDAPS = SIGN(1.1,PSCRIT)
      GO TO 340
  272 IF(PSIMT - PSP( 2)) 276,276,273
  273 IF(PSIMT - PSP(11)) 274,277,277
  274 IF(PSIMT - PSP( 5)) 320,275,275
  275 IF(PSIMT - PSP( 8)) 278,278,340
  276 MXMN = -1
      CALL FULSRC( YOU, YMNFUL,IBB,ILB,IBE,ILE,MXMN)
      CALL FULSRC( XOU, XMNFUL,IBB,ILB,IBE,ILE,MXMN)
      IF((YS-YMNFUL) - (XF-XMNFUL)) 2792,310,310
  277 MXMN = + 1
      CALL FULSRC( YOU, YMXFUL,IBB,ILB,IBE,ILE,MXMN)
      MXMN = - 1
      CALL FULSRC( XOU, XMNFUL,IBB,ILB,IBE,ILE,MXMN)
      IF((YS+YMXFUL) - (XF-XMNFUL)) 2793,310,310
  278 IF(PSIMT - PI) 279,330,2791
  279 MXMN = - 1
      CALL FULSRC( YOU, YMNFUL,IBB,ILB,IBE,ILE,MXMN)
      MXMN = + 1
      CALL FULSRC( XOU, XMXFUL,IBB,ILB,IBE,ILE,MXMN)
      IF((YS-YMNFUL) - (XMXFUL-XR)) 2794,330,330
 2791 MXMN = + 1
      CALL FULSRC( YOU, YMXFUL,IBB,ILB,IBE,ILE,MXMN)
      CALL FULSRC( XOU, XMXFUL,IBB,ILB,IBE,ILE,MXMN)
      IF((YS+YMXFUL) - (XMXFUL-XR)) 2795,330,330
 2792 INDAPS= -1
      GO TO 320
 2793 INDAPS= +1
      GO TO 340
 2794 INDAPS= +1
      GO TO 320
```

```
 2795 INDAPS= -1
      GO TO 340
C
  310 NCOL3(I) = MF
      WRITE(6,6310) IJ,I,IRNG,IBB,ILB,IBE,ILE,PSD1,PSD2,PSIM(I),PSIMT,
     1 NCOL3(I),ICORTB,INDAPS
 6310 FORMAT(32H0 MESSAGE FROM SUBROUTINE DAMAGE/
     1 2X,7HVEH.NO.,I2,5X, 9H RANGE I=,I2,3H OF,I3,7H RANGES,
     2 12X,4HIBB=,I4,6H, ILB=,I4,2X,5H IBE=,I4,6H, ILE=,I4/
     3 2X,5HPSD1=,E13.5,7H, PSD2=,E13.5,10H, PSIM(I)=,E13.5,
     4 13H, USED PSIMT=,E13.5,5X,10H NCOL3(I)=,A1 ,2X, 9H, ICORTB=,I2/
     5 2X, 8H INDAPS=,I2)
      IF(INDAPS) 311,3100,350
 3100 IF(PSD1  - PSP(1)) 312,3101,3101
 3101 IF(PSD1  - PSP(12))3102,312,312
 3102 IF(PSD1  - PSP( 9))3103,311,311
 3103 NCOL4(I) = MR
      IF(ICORTB) 412,421,412
  311 IF(PSD2  - PSP(1)) 3112,3110,3110
 3110 IF(PSD2  - PSP(12))3114,3114,3112
 3112 NCOL4(I) = MY
      IF(ICORTB) 411,421,411
 3114 IF(PSD2  - PSP(11))3117,3117,3115
 3115 NCOL4(I) = ML
      IF(ICORTB) 414,421,414
 3117 IF(PSD2  - PSP(4)) 3118,3118,3119
 3118 NCOL4(I) = MD
      IF(ICORTB) 411,421,411
 3119 ISTOP = 44
      GO TO 250
  312 IF(PSD2 - PSP(1))  3122,3122,3120
 3120 IF(PSD2 - PSP(12)) 3124,3124,3122
 3122 NCOL4(I) = MC
      IF(ICORTB) 411,421,411
 3124 IF(PSD2 - PSP(4)) 3125,3125,3126
 3125 NCOL4(I) = MZ
      IF(ICORTB) 411,421,411
 3126 ISTOP = 45
      GO TO 250
  320 NCOL3(I) = MR
      WRITE(6,6310) IJ,I,IRNG,IBB,ILB,IBE,ILE,PSD1,PSD2,PSIM(I),PSIMT,
     1 NCOL3(I),ICORTB,INDAPS
      IF(INDAPS) 321,3200,360
 3200 IF(PSD1 - TWOPI) 3202,3201,3201
 3201 ISTOP = 46
      GO TO 250
 3202 IF(PSD1 - PSP(10)) 3203,321,321
 3203 IF(PSD1 - PSP(3)) 321 ,3204,3204
 3204 IF(PSD1 - PSP(4)) 322 , 3205,3205
 3205 NCOL4(I) = MB
      IF(ICORTB) 413,422,413
  321 IF(PSD2 - PSP(2)) 3210,3210,3212
 3210 ISTOP = 47
      GO TO 250
 3212 IF(PSD2 - PSP(3)) 3213,3213,3215
 3213 NCOL4(I) = MF
      IF(ICORTB) 411,422,411
 3215 IF(PSD2 - PSP(4)) 3216,3216,3217
 3216 NCOL4(I) = MY
      IF(ICORTB) 412,422,412
 3217 NCOL4(I) = MD
      IF(ICORTB) 412,422,412
```

```
 322 IF(PSD2 - PSP(3)) 3220,3220,3221
3220 ISTOP = 49
     GO TO 250
3221 IF(PSD2 - PSP(4)) 3222,3222,3223
3222 NCOL4(I) = MP
     IF(ICORTB) 412,422,412
3223 NCOL4(I) = MZ
     IF(ICORTB) 412,422,412
 330 NCOL3(I) = MB
     WRITE(6,6310) IJ,I,IRNG,IBB,ILB,IBE,ILE,PSD1,PSD2,PSIM(I),PSIMT,
    1 NCOL3(I),ICORTB,INDAPS
     IF(INDAPS) 331,3300,370
3300 IF(PSD1 - PSP(3)) 3301,3302,3302
3301 ISTOP = 51
     GO TO 250
3302 IF(PSD1 - PSP(6)) 331 ,3303,3303
3303 IF(PSD1 - PSP(7)) 332,  3304,3304
3304 NCOL4(I) = ML
     IF(ICORTB) 414,423,414
 331 IF(PSD2 - PSP(5)) 3310,3310,3311
3310 ISTOP = 52
     GO TO 250
3311 IF(PSD2 - PSP(6)) 3312,3312,3313
3312 NCOL4(I) = MR
     IF(ICORTB) 412,423,412
3313 IF(PSD2 - PSP(7)) 3314,3314,3315
3314 NCOL4(I) = MZ
     IF(ICORTB) 413,423,413
3315 NCOL4(I) = MD
     IF(ICORTB) 413,423,413
 332 IF(PSD2 - PSP(6)) 3320,3320,3321
3320 ISTOP = 54
     GO TO 250
3321 IF(PSD2 - PSP(7)) 3322,3322,3323
3322 NCOL4(I) = MC
     IF(ICORTB) 413,423,413
3323 NCOL4(I) = MY
     IF(ICORTB) 413,423,413
 340 NCOL3(I) = ML
     WRITE(6,6310) IJ,I,IRNG,IBB,ILB,IBE,ILE,PSD1,PSD2,PSIM(I),PSIMT,
    1 NCOL3(I),ICORTB,INDAPS
     IF(INDAPS) 341,3400,380
3400 IF(PSD1 - PSP(4)) 3401,3402,3402
3401 ISTOP = 56
     GO TO 250
3402 IF(PSD1 - PSP(9)) 341,3403,3403
3403 IF(PSD1 - PSP(10)) 342,3404,3404
3404 NCOL4(I) = MF
     IF(ICORTB) 411,424,411
 341 IF(PSD2 -PSP(3)) 3410,3411,3411
3410 NCOL4(I) = MD
     IF(ICORTB) 414,424,414
3411 IF(PSD2 - PSP(8)) 3412,3412,3413
3412 ISTOP = 57
     GO TO 250
3413 IF(PSD2 - PSP(9)) 3414,3414,3415
3414 NCOL4(I) = MB
     IF(ICORTB) 413,424,413
3415 IF(PSD2 - PSP(10)) 3416,3416,3417
3416 NCOL4(I) = MZ
     IF(ICORTB) 414,424,414
3417 NCOL4(I) = MD
```

```
      IF(ICORTB) 414,424,414
  342 IF(PSD2 - PSP(3)) 3420,3421,3421
 3420 NCOL4(I) = MY
      IF(ICORTB) 414,424,414
 3421 IF(PSD2 - PSP(9)) 3422,3422,3423
 3422 ISTOP = 59
      GO TO 250
 3423 IF(PSD2 - PSP(10)) 3424,3424, 3425
 3424 NCOL4(I) = MP
      IF(ICORTB) 414,424,414
 3425 NCOL4(I) = MY
      IF(ICORTB) 414,424,414
C
  350 IF(PSD1 - PSP(1)) 352,351,351
  351 IF(PSD1 - PSP(12)) 353,352,352
  352 NCOL4(I) = MZ
      GO TO 421
  353 IF(PSD1 - PSP(11)) 355,354,354
  354 NCOL4(I) = MD
      GO TO 421
  355 NCOL4(I) = MR
      GO TO 412
C
  360 IF(PSD1 - PSP(3)) 362,361,361
  361 IF(PSD1 - PSP(4)) 363,364,364
  362 NCOL4(I) = MD
      GO TO 422
  363 NCOL4(I) = MZ
      GO TO 422
  364 NCOL4(I) = MB
      GO TO 413
C
  370 IF(PSD1 -PSP(6)) 372,371,371
  371 IF(PSD1 - PSP(7)) 373,374,374
  372 NCOL4(I) = MD
      GO TO 423
  373 NCOL4(I) = MY
      GO TO 423
  374 NCOL4(I) = ML
      GO TO 414
C
  380 IF(PSD1 - PSP(9)) 382,381,381
  381 IF(PSD1 - PSP(10)) 383,384,384
  382 NCOL4(I) = MD
      GO TO 424
  383 NCOL4(I) = MY
      GO TO 424
  384 NCOL4(I) = MF
      GO TO 411
C
C
  411 IF(NCOL4(I) - MF) 420,4110,420
 4110 NCOLF = 1
      MXMN = -1
      CALL CORSRC (PSBOU,XOU,PSP(2),PSP(11),XMN,IBB,ILB,IBE,ILE,
     1             NCOLF,MXMN)
      DIS = XF -XMN
      GO TO 415
  412 IF(NCOL4(I) - MR) 420,4120,420
 4120 NCOLF = 0
      MXMN = -1
      CALL CORSRC (PSBOU,YOU,PSP(2),PSP( 5),YMN,IBB,ILB,IBE,ILE,
```

```
     1            NCOLF,MXMN)
      DIS = YS - YMN
      GO TO 415
  413 IF(NCOL4(I) - MB) 420,4130,420
 4130 NCOLF = 0
      MXMN =  1
      CALL CORSRC (PSBOU,XOU,PSP(5),PSP( 8),XMX,IBB,ILB,IBE,ILE,
     1            NCOLF,MXMN)
      DIS = XMX - XR
      GO TO 415
  414 IF(NCOL4(I) - ML) 420,4140,420
 4140 NCOLF = 0
      MXMN =  1
      CALL CORSRC (PSBOU,YOU,PSP(8),PSP(11),YMX,IBB,ILB,IBE,ILE,
     1            NCOLF,MXMN)
      DIS = YS + YMX
  415 IF(DIS) 4150,4152,4152
 4150 ISTOP = 61
 4151 WRITE(6,6415) ISTOP,DIS
 6415 FORMAT(8H0 ISTOP=,I3,5X,5H DIS=,E13.5,5X,14H SHOULD BE POS  )
      GO TO 505
 4152 IF(DIS -4.5) 4153,4153,4154
 4153 NCOL6(I) = MS
      GO TO 500
 4154 IF(DIS - 16.5) 4155,4155,4156
 4155 NCOL6(I) = ME
      GO TO 500
 4156 NCOL6(I) = MW
      GO TO 500
C
  420 NCOL = NCOL3(I)
      IF(NCOL - MF) 4201,421,4201
 4201 IF(NCOL - MR) 4202,422,4202
 4202 IF(NCOL - MB) 4203,423,4203
 4203 IF(NCOL - ML) 4204,424,4204
 4204 ISTOP = 67
      WRITE(6,6506) ISTOP,I,NCOL
      GO TO 505
  421 DIS = YOU(IL) - YOU(IB)
      GO TO 425
  422 DIS = XOU(IB) - XOU(IL)
      GO TO 425
  423 DIS = YOU(IB) - YOU(IL)
      GO TO 425
  424 DIS = XOU(IL) - XOU(IB)
  425 IF (DIS) 4250,4252,4252
 4250 ISTOP = 62
      GO TO 4151
 4252 IF(DIS - 16.0) 4253,4253,4254
 4253 NCOL6(I) = MN
      GO TO 500
 4254 NCOL6(I) = MW
C
C
  500 NCOL = NCOL3(I)
      IF(NCOL - MF) 5002,5010,5002
 5002 IF(NCOL - MR) 5003,5020,5003
 5003 IF(NCOL - MB) 5004,5030,5004
 5004 IF(NCOL - ML) 5005,5040,5005
 5005 ISTOP = 63
 5006 WRITE(6,6506) ISTOP, I,NCOL
 6506 FORMAT(8H0 ISTOP=,I3,5X,6HNCOL3(,I3,2H)=,A1)
```

```
      GO TO 505
 5010 MXMN = -1
      NCOLF = 1
      CALL DEFMJ(PSBOU,XOU,ASTR,PSP(2),PSP(11),XMIN,IKK,IBB,ILB,IBE,ILE,
     1         INDAPS,NCOLF,MXMN)
      EXT = XF - XMIN
      IF(EXT - 0.0) 5011,5011,5013
 5011 ISTOP = 64
 5012 WRITE(6,6501) ISTOP,EXT
 6501 FORMAT(8H0 ISTOP=,I3,5X,4HEXT=,E13.5)
      GO TO 505
 5013 DO 5015 II = 2,9
      IF(EXT -EXTF(II)) 5014,5014,5015
 5014 NCOL7(I) = MMM(II-1)
      GO TO 505
 5015 CONTINUE
      NCOL7(I) = MMM(9)
      GO TO 505
 5020 MXMN = -1
      NCOLF = 0
      CALL DEFMJ(PSBOU,YOU,ASTR,PSP(2),PSP( 5),YMIN,IKK,IBB,ILB,IBE,ILE,
     1         INDAPS,NCOLF,MXMN)
      EXT = YS - YMIN
 5021 IF(EXT) 5022,5022,5023
 5022 ISTOP = 65
      GO TO 5012
 5023 DO 5025 II = 2,9
      IF(EXT - EXTR(II)) 5024,5024,5025
 5024 NCOL7(I) = MMM(II-1)
      GO TO 505
 5025 CONTINUE
      NCOL7(I) = MMM(9)
      GO TO 505
 5030 MXMN = 1
      NCOLF = 0
      CALL DEFMJ(PSBOU,XOU,ASTR,PSP(5),PSP( 8),XMAX,IKK,IBB,ILB,IBE,ILE,
     1         INDAPS,NCOLF,MXMN)
      EXT = XMAX - XR
      IF(EXT) 5031,5031,5032
 5031 ISTOP = 66
      GO TO 5012
 5032 DO 5035 II = 2,9
      IF(EXT - EXTB(II)) 5034,5034,5035
 5034 NCOL7(I) = MMM(II-1)
      GO TO 505
 5035 CONTINUE
      NCOL7(I) = MMM(9)
      GO TO 505
 5040 MXMN = 1
      NCOLF = 0
      CALL DEFMJ(PSBOU,YOU,ASTR,PSP(8),PSP(11),YMAX,IKK,IBB,ILB,IBE,ILE,
     1         INDAPS,NCOLF,MXMN)
      EXT = YS + YMAX
      GO TO 5021
  505 CONTINUE
      RETURN
      END
      SUBROUTINE FULSRC( XXOU,XM,IBB,ILB,IBE,ILE,MXMN)
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
      DIMENSION XXOU(1)
C
C        SEARCHES FOR MAX  OR MIN OF X OR Y WITHIN THE APPROPRIATE RANGE
```

```
C                OF VEHICLE DAMAGE, RANGE NOT LIMITED NOR RESTRICTED
C
C          CHOOSES MAX IF MXMN = +1
C          CHOOSES MIN IF MXMN = -1
C           SAMPLE CALLS
C     MXMN = -1
C     CALL FULSRC( XOU, XMNFUL,IBB,ILB,IBE,ILE,MXMN)
C       XMNFUL IS THE MINIMUM X IN THIS RANGE
C     MXMN = +1
C     CALL FULSRC( YOU, YMXFUL,IBB,ILB,IBE,ILE,MXMN)
C       YMXFUL IS THE MAXIMUM Y IN THIS RANGE
C
      IB = IBB
      IL = ILB
      XMN = -1.E20
      IF(MXMN) 2,3,3
    2 XMN = 1.E20
      GO TO 15
    3 DO 8 II=IB,IL
      IF(XXOU(II) - XMN) 8,5,5
    5 XMN = XXOU(II)
    8 CONTINUE
      IF(IBE) 9,28,9
    9 IF(IB-IBE) 10,28,10
   10 IB = IBE
      IL = ILE
      GO TO 3
C
   15 DO 20 II= IB,IL
      IF(XXOU(II) - XMN) 17,17,20
   17 XMN = XXOU(II)
   20 CONTINUE
      IF(IBE) 21,28,21
   21 IF(IB-IBE) 22,28,22
   22 IB = IBE
      IL = ILE
      GO TO 15
   28 XM = XMN
      RETURN
      END
      SUBROUTINE CORSRC (PPSBOU,XXOU,PSPP1,PSPP2,XM,IBB,ILB,IBE,ILE,
     1                   NCOLF,MXMN)
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
      DIMENSION PPSBOU(1),XXOU(1)
C
C          CHOOSES MAX IF MXMN = +1
C          CHOOSES MIN IF MXMN = -1
C          SPECIAL LIMIT LOGIC FOR FRONT OF CAR, NCOLF =1
C          OTHER LIMIT LOGIC FOR ALL OTHER CASES, NCOLF=0
C          CALLING PROGRAM SETS NCOLF
C           SAMPLE CALLS
C     NCOLF = 1
C     MXMN = -1
C     CALL CORSRC (PSBOU,XOU,PSP(2),PSP(11),XMN,IBB,ILB,IBE,ILE,
C    1             NCOLF,MXMN)
C     DIS = XF - XMN
C     NCOLF = 0
C     MXMN =  1
C     CALL CORSRC (PSBOU,YOU,PSP(8),PSP(11),YMX,IBB,ILB,IBE,ILE,
C    1             NCOLF,MXMN)
C     DIS = YS + YMX
C
```

```
      IB = IBB
      IL = ILB
      XMN = -1.E20
      IF(MXMN) 2,3,3
    2 XMN = 1.E20
    3 DO 25 II = IB,IL
      PSBTT = PPSBOU(II)
      IF(NCOLF) 4,7,4
    4 IF(PSBTT-PSPP1) 10,10,5
    5 IF(PSBTT-PSPP2) 25,10,10
    7 IF(PSBTT - PSPP1) 25,8,8
    8 IF(PSBTT - PSPP2) 10,10,25
   10 IF(MXMN) 11,12,12
   11 IF(XXOU(II) - XMN) 20,20,25
   12 IF(XXOU(II) - XMN) 25,20,20
   20 XMN = XXOU(II)
   25 CONTINUE
      IF(IBE) 26,28,26
   26 IF (IB-IBE) 27,28,27
   27 IB=IBE
      IL = ILE
      GO TO 3
   28 XM = XMN
      RETURN
      END
      SUBROUTINE DEFMJ( PPSBOU,XXOU,ASTRO,PSPP1,PSPP2,XM,IKK,IBB,ILB,
     1         IBE,ILE,INDAPS,NCOLF,MXMN)
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
      DIMENSION PPSBOU(1),XXOU(1),ASTRO(1)
      DATA STAR/1H*/
C
C          CHOOSES MAX IF MXMN = +1
C          CHOOSES MIN IF MXMN = -1
C           CALLING PROGRAM SETS MXMN
C          SPECIAL LIMIT LOGIC FOR FRONT OF CAR, NCOLF =1
C          OTHER LIMIT LOGIC FOR ALL OTHER CASES, NCOLF=0
C          CALLING PROGRAM SETS NCOLF
C       SAMPLE CALL
C       NCOLF = 1
C     MXMN = -1
C     CALL DEFMJ(PSBOU,XOU,ASTR,PSP(2),PSP(11),XMIN,IKK,IBB,ILB,IBE,ILE,
C    1         INDAPS,NCOLF,MXMN)
C     NCOLF = 0
C     MXMN = -1
C     CALL DEFMJ(PSBOU,YOU,ASTR,PSP(2),PSP( 5),YMIN,IKK,IBB,ILB,IBE,ILE,
C    1         INDAPS,NCOLF,MXMN)
C
      IB = IBB
      IL = ILB
      XMN = -1.E20
      IF(MXMN) 2,3,3
    2 XMN = 1.E20
    3 DO 25 II = IB,IL
      IF(INDAPS) 4,10,4
    4 PSBTT = PPSBOU(II)
      IF(NCOLF) 5,7,5
    5 IF(PSBTT-PSPP1) 10,10,6
    6 IF(PSBTT-PSPP2) 25,10,10
    7 IF(PSBTT - PSPP1) 25,8,8
    8 IF(PSBTT - PSPP2) 10,10,25
   10 IF(ASTRO(II) - STAR) 17,11,17
   11 IIB = II - 1
```

```
      IF(IIB) 12,12,13
   12 IIB = 1
   13 IF(ASTRO(IIB) - STAR) 14,17,14
   14 IIL = II + 1
      IF(IIL - IKK) 16,16,15
   15 IIL = IKK
   16 IF(ASTRO(IIL) - STAR) 25,17,25
   17 IF(MXMN) 18,19,19
   18 IF(XXOU(II) - XMN) 20,20,25
   19 IF(XXOU(II) - XMN) 25,20,20
   20 XMN = XXOU(II)
   25 CONTINUE
      IF(IBE) 26,28,26
   26 IF (IB-IBE) 27,28,27
   27 IB=IBE
      IL = ILE
      GO TO 3
   28 XM = XMN
      RETURN
      END
      SUBROUTINE NCOLDV(NARRPS,ARRDV ,MMM,NUM ,JN,ISTOPP,NCOL1P,NCOL2P,
     1                  DVSUMM)
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS - SMAC
C      COMPUTES NCOL1,NCOL2 - THE FIRST TWO COLUMNS OF VDI
C      SUMS DELTA V IF SEVERAL CLOCK DIRECTIONS(NARRPS) SEEM TO OCCUR
C      WITHIN ONE RANGE OF DAMAGE.
C
C      SAMPLE CALL FROM SUBROUTINE OUT2
C     JNCOL = INPSM1(I)
C     CALL NCOLDV(NARRP1,ARRDV1,MMM,NUM1,JNCOL,ISTOPP,NCOL1,NCOL2,DVSUM)
C
      DIMENSION NARRPS(1),ARRDV(1),MMM(1)
      DATA MZERO/1H0/,M1/1H1/
      ISTOPP = 0
      DVSUMM = ARRDV(JN)
      NCOL = NARRPS(JN)
      NCOL1P = MZERO
      IF(NCOL-9) 66,66,65
   65 NCOL1P = M1
      NCOL = NCOL - 10
   66 IF(NCOL) 67,69,70
   67 ISTOPP = 1
      GO TO 95
   69 NCOL2P = MZERO
      GO TO 80
   70 DO 72 J = 1,9
      IF(NCOL - J) 72,71,72
   71 NCOL2P = MMM(J)
      GO TO 80
   72 CONTINUE
   80 IF(NUM-1) 95,95,81
   81 NCOL = NARRPS(JN)
      NCOLP = NCOL+1
      IF(NCOLP - 12) 83,83,82
   82 NCOLP = 1
   83 NCOLM = NCOL-1
      IF(NCOLM) 84,84,85
   84 NCOLM = 12
   85 DO 90 J = 1,NUM
      IF( J-JN) 86,90,86
   86 NTRY = NARRPS(J)
      IF (NTRY - NCOL) 87,89,87
```

```
87 IF (NTRY - NCOLP) 88,89,88
88 IF(NTRY - NCOLM) 90,89,90
89 DVSUMM = DVSUMM + ARRDV(J)
90 CONTINUE
95 RETURN
   END
```

```
      SUBROUTINE INPUT(*)
C     Version 04-13-98
C     SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
C
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
      COMMON/INTG/NEQ,T,DT,VAR(12),DER(12)
      EQUIVALENCE (VAR(1),XCP1),(VAR(2),YCP1),(VAR(3),PSI1),
     1            (VAR(4),PSI1D),(VAR(5),U1),(VAR(6),V1),
     2            (VAR(7),XCP2),(VAR(8),YCP2),(VAR(9),PSI2),
     3            (VAR(10),PSI2D),(VAR(11),U2),(VAR(12),V2)
      EQUIVALENCE (DER(1),DXCP1),(DER(2),DYCP1),(DER(3),DPSI1),
     1            (DER(4),DPSI1D),(DER(5),DU1),(DER(6),DV1),
     2            (DER(7),DXCP2),(DER(8),DYCP2),(DER(9),DPSI2),
     3            (DER(10),DPSI2D),(DER(11),DU2),(DER(12),DV2)
      COMMON/INPT/  T0,TF,DTTRA0,DTCOL0,DTCLT0,DTPRN0,UVMIN,PSIDMN,FVEH0
     1             ,XCP10,YCP10,PSI10,PSI1D0,U10,V10,
     2              XCP20,YCP20,PSI20,PSI2D0,U20,V20,
     3              A1,B1,TR1,FIZ1,FM1,PSIR10,XF1,XR1,YS1,
     4              A2,B2,TR2,FIZ2,FM2,PSIR20,XF2,XR2,YS2,
     5              CSTFI(8),TBTQ1,TETQ1,TINCQ1,NTBLQ1,NTQ1,
     6              TBTQ2,TETQ2,TINCQ2,NTBLQ2,NTQ2,TII(8,201),
     7              TBPSF1,TEPSF1,TINCP1,NTBLP1,NPSF1,
     8              TBPSF2,TEPSF2,TINCP2,NTBLP2,NPSF2,PSIFI(4,201),
     9              XBP(2),YBP(2),XMU1,XMU2,CMU
      COMMON/INPT/  DELPS0,DELRO0,ALAMB,ZETAV,AKV(2) ,AMU,C0,C1,C2,
     1              PSLM10,PSLM20,PSLM30,PSLM40,PSLM50,PSLM60,PSLM70,
     2              PSLM80,HED(40),DADE(3),XINPUT(9)
C
      COMMON/OPTIONS/ IOPTION, DTINT, STEPPSI, STEPDT, SINPSI, SINFREQ,
     1                GAIN, DISLKAHD, XYTB(2,201), NOUT1, NOUT2
C
      COMMON/DELAY/ TDGT, DTDGT, DY1LST, YAW1LST, DY2LST, YAW2LST
C
      COMMON/INTGR/JCC(4),JCOR(4),NPSIB,IPSIB0,IPSIB,IBB,IFF,NDTAB,
     1             I9,J9(361,2),  NPSJB,IPSJB,I3,ILAST,IL,IND1,IND2,INDI
     2             ,INDJ,ISTOP,ITZER1,ITZER2,IPZER1,IPZER2,ICOUNT,INDXB,
     3             ICOLLP,ICOLL,IQ,IND,IVEH,LL,ICTVDI,ITCOL,ITFLG1,
     4             ITFLG2,NWRDB1,NWRDE1,NWRDB2,NWRDE2,ICOLAC
C            IN COMMON /COMP/ DO NOT DISTURB THE ORDER OF THE VARIABLES
C                             WHICH ARE EQUIVALENCED TO ARRAYS.
      COMMON/COMP/  DTTRAJ,DTCOLL,DTCOLT,DTPRNT,UVMN2,PSIDMA,TPRINT,DTP,
     1              U1V1SQ,U2V2SQ,DELPSI,DELPST,DELPS2,DELRHO,
     2              PSILM1,PSILM2,PSILM3,PSILM4,PSILM5,PSILM6,PSILM7,
     3              PSILM8,EJJ(4,2),GJJ(4,2),C1OC2,PSCC,PSIR1,PSIR2,
     4             XCMXC1,XCMXC2,YCMYC1,YCMYC2,SPSI1,SPSI2,CPSI1,CPSI2,
     5             XCPI,YCPI,PSII,XCPJ,YCPJ,PSIJ,XCMXC,YCMYC,SPSIJ,
     6             CPSIJ,XFI,XRI,YSI,XFJ,XRJ,YSJ,AKVI,AKVJ,EJ,GJ,
     7             XPJ(4),YPJ(4),ANGYX,SPAN,PSIBPI(4),PSIBPB,PSIBPF,
     8             FNPSIB,PSIBB,PSIB,PSFMPS,SCPSIB,CSPSIB,
     9             RHOBI,RHOBIN(4),PSIBIJ,SPSBIJ,CPSBIJ,XBIJ,YBIJ,PSJTB
      COMMON/COMP/ PSIBPJ,FNPSJB,PSIBJ,RHOBJ,XYJSQR,SPSIB,CPSIB,XBI,YBI,
     1         PRESI,PRESJ,XYSR,DELTA,CRHO,RHOBIC,RHOBMX,RHOBIT,RHOBT,
     2             SPSI21,CPSI21,X2TEM,Y2TEM,XX,YY,PAV,XXYYSR,FN1,SPSF,
     3             CPSF,X1AV,Y1AV,XYAVSR,X2TERM,Y2TERM,SPS21F,CPS21F,
     4             X2AV,Y2AV,VT1AV,VT2AV,VTMVT,FRICT,FNX1,FNY1,FNN1,
     5             FNX2,FNY2,FNN2,SFX1,SFY1,SFN1,SFX2,SFY2,SFN2,
     6              SFX1C,SFX2C,SFY1C,SFY2C,SFN1C,SFN2C,
     7              SFX1T,SFX2T,SFY1T,SFY2T,SFN1T,SFN2T,  TCOL,TEND,
     8              TREST1,TREST2,TCOLS1,TCOLS2,P1TEM,P2TEM,U1V1E,U2V2E,
     9              EXTRA(10)
      COMMON/COMPT/ W(8),XWP(8),YWP(8),FLAGW(8),PSIW(8),TQW(8),
```

```
     1             TFX,TFY,TMOM,PSIRR,COEF,COEF1,COEF2,TRD2,TRD21,TRD22,
     2             XBMXB,FNUM,A,B,FIZ,FMASS,XC,YC,U,V,PSIC,PSICD,UD2,
     3             SGNU,SGNI,AOMB,CSPSIC,SNPSIC,TRCPSC,TRSPSC,TRPSCD,
     4             XI,XW,YW,SX,SY,S,XPP,YPP,RHO,RHOP,XMUS,XMU,PSI,XNUM,
     5             XDEN,RATIO,ALFA,F,TIF,FC,FCOS,SMAL,FSMX,FS,BETB,FX,FY
     6             ,CSPSI,SNPSI,XCXC,YCYC,SPJ,CPJ,SPIJ,CPIJ,XFJJ,XRJJ,
     7              YSJJ,EJJI,GJJI,XICORJ,YICORJ,NTQ,TBTQ,TETQ,TINCRQ,
     8             NPSF,TBPSF,TEPSF,TINCRP
      COMMON/TAB/  TRHOB(361,2),TROBMX(361,2),TPSIB(361,2),
     1             TCPSIB(361,2),TSPSIB(361,2),TXB1(100),TYB1(100),
     2             TPSB1(100),TPRES1(100)
C
      DIMENSION   XCMXCI(2),YCMYCI(2),SPSII(2),CPSII(2),PSLM0(8),PSLM(8)
      EQUIVALENCE (XCMXCI(1),XCMXC1),(YCMYCI(1),YCMYC1),(SPSII(1),SPSI1)
     1           ,(CPSII(1),CPSI1),(PSLM0(1),PSLM10),(PSLM(1),PSILM1)
      EQUIVALENCE (XINPUT(1),FMOVIE)
      DIMENSION DUM(9),CARDIM(20)
      DIMENSION TTARG(201)
      EQUIVALENCE (TXB1(1),TTARG(1))
C
      DATA BLANK/4H    /
      DATA  NAMEQ/4H  TI/,NAMEP/4HPSIF/,NO1/1/,NO2/2/
C
    1 READ(NIN,1000,END=500)(HED(I),I=1,40)
 1000 FORMAT(20A4)
      GO TO 2
C
  500 WRITE(NOUT,1001)
 1001 FORMAT('1End of input - SMAC completed')
      RETURN 1
C
    2 WRITE(NOUT,1002) HED
 1002 FORMAT(1H1,20A4/1X,20A4)
          DO 3 I=1,3
          II = 37 + I
          DADE(I) = HED(II)
    3     HED(II) = BLANK
C
C     PUNCH DATE IN COL 69 THRU 80 ON SECOND HEADING CARD
C
    4 READ(NIN,1003) CARDIM, DUM,ICARD
c    4 READ(NIN,1003) DUM,ICARD
      write(6,*) ' just read icard = ', icard
 1003 FORMAT( 20A4,T1,9F8.0,I8)
c 1003 FORMAT( 9F8.0,I8)
      WRITE(NOUT,1004) CARDIM
 1004 FORMAT(1X,20A4)
      IF(ICARD.GE.9999) GO TO 120
      GO TO (5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,21) ,ICARD
      WRITE(NOUT,1005) CARDIM
 1005 FORMAT(19HOILLEGAL INPUT CARD /1X,20A4)
      STOP
C
    5 T0 = DUM(1)
      TF = DUM(2)
C             THE COMPUTING AND PRINTING INTERVALS CANNOT BE ZERO.
C         IF THE INTERVALS ARE BLANK ON INPUT, THEY ARE SET TO STANDARD.
      IF(DUM(3)) 51,52,51
   51 DTTRA0 = DUM(3)
      GO TO 53
   52 DTTRA0 = 0.025
   53 IF(DUM(4)) 54,55,54
```

```
   54 DTCOL0 = DUM(4)
      GO TO 56
   55 DTCOL0 = 0.001
   56 IF(DUM(5)) 57,58,57
   57 DTCLT0 = DUM(5)
      GO TO 59
   58 DTCLT0 = 0.005
   59 IF(DUM(6)) 60,61,60
   60 DTPRN0 = DUM(6)
      GO TO 62
   61 DTPRN0 = DTCOL0
C            ASSUME DTCOL0 IS THE SMALLEST INTERVAL
C             WILL PRINT  EVERY INTERVAL
   62 IF(DUM(9)) 65,65,63
   63 IF(DUM(9) -2.0) 64,64,65
   64 FVEH0 = DUM(9)
      GO TO 67
   65 WRITE(NOUT,1065)
 1065 FORMAT(42H0 STOP.  NUMBER OF VEHICLES NOT ONE OR TWO)
      STOP
   67 UVMIN = DUM(7)
      PSIDMN = DUM(8)
      GO TO 4
    6 XCP10 = DUM(1)
      YCP10 = DUM(2)
      PSI10 = DUM(3)
      PSI1D0= DUM(4)
      U10   = DUM(5)
      V10   = DUM(6)
      GO TO 4
    7 XCP20 = DUM(1)
      YCP20 = DUM(2)
      PSI20 = DUM(3)
      PSI2D0= DUM(4)
      U20   = DUM(5)
      V20   = DUM(6)
      GO TO 4
    8 A1 = DUM(1)
      B1 = DUM(2)
      TR1 = DUM(3)
      FIZ1 = DUM(4)
      FM1 = DUM(5)
      PSIR10 = DUM(6)
      XF1 = DUM(7)
      XR1 = DUM(8)
      YS1 = DUM(9)
      GO TO 4
C              XR1 AND XR2 ARE INPUT AS NEGATIVE QUANTITIES
    9 A2 = DUM(1)
      B2 = DUM(2)
      TR2 = DUM(3)
      FIZ2 = DUM(4)
      FM2 = DUM(5)
      PSIR20 = DUM(6)
      XF2 = DUM(7)
      XR2 = DUM(8)
      YS2 = DUM(9)
      GO TO 4
C            CARD 6   CORNERING STIFFNESS FOR VEH. NO. 1
   10 DO 91  IJ = 1,4
   91 CSTFI(IJ) = DUM(IJ)
      GO TO 4
```

```
C            CARD 7   CORNERING STIFFNESS FOR VEH. NO. 2
   11 DO 92 IJ = 5,8
   92 CSTFI(IJ) = DUM(IJ-4)
      GO TO 4
C
C           CARD 8 TORQUE FOR EACH WHEEL FOR VEHICLE 1, TII(8,201)
C     FOLLOWED BY AT MOST 7 VALUES TO A CARD FOR WHEEL1,THEN WHEEL2,ETC.
C           AFTER FINISHING ENTRIES FOR  A WHEEL, START ENTRIES FOR NEXT
C           WHEEL ON A NEW CARD
   12 TBTQ1= DUM(1)
      TETQ1= DUM(2)
      TINCQ1 = DUM(3)
      NTQ1 = (TETQ1 -TBTQ1)/TINCQ1 + 1.2
      NTBLQ1 = DUM(4) + 0.1
C     NTBLQ1- NON-ZERO MEANS DO NOT READ TABLE FOR TORQUE
      IF(NTBLQ1 .NE. 0) GO TO 4
      NTQ = NTQ1
      IB = 1
      IL = 4
      NUM = NO1
   99 DO 101 I = IB,IL
      DO 100 IJ = 1,NTQ,7
      IJL = MIN0(IJ+6,NTQ )
      READ(NIN,1006) CARDIM,(TII(I,K),K=IJ,IJL)
 1006 FORMAT(20A4,T1, 7F10.0)
      WRITE(NOUT,1007) CARDIM,NUM,NAMEQ,I,IJ,IJL
 1007 FORMAT(1X,20A4,5X,7HVEH.NO.,I2,A4,1H(,I2,1H,2HJ=,I4,3H TO,I4,1H))
  100 CONTINUE
  101 CONTINUE
      GO TO 4
C
C           CARD 9 TORQUE FOR EACH WHEEL FOR VEHICLE 2, TII(8,201)
   13 TBTQ2= DUM(1)
      TETQ2= DUM(2)
      TINCQ2 = DUM(3)
      NTQ2 = (TETQ2 -TBTQ2)/TINCQ2 + 1.2
      NTBLQ2 = DUM(4) + 0.1
C     NTBLQ2- NON-ZERO MEANS DO NOT READ TABLE FOR TORQUE
      IF(NTBLQ2 .NE. 0) GO TO 4
      NTQ = NTQ2
      IB = 5
      IL = 8
      NUM = NO2
      GO TO 99
C
C
C     CARD 10 STEER FOR FRONT WHEELS FOR VEHICLE 1, PSIFI(4,201)
C     FOLLOWED BY AT MOST 7 VALUES TO A CARD FOR WHEEL1,THEN WHEEL2,ETC.
C           AFTER FINISHING ENTRIES FOR  A WHEEL, START ENTRIES FOR NEXT
C           WHEEL ON A NEW CARD
   14 TBPSF1 = DUM(1)
      TEPSF1 = DUM(2)
      TINCP1 = DUM(3)
      NPSF1 = (TEPSF1-TBPSF1)/TINCP1 + 1.2
      NTBLP1 = DUM(4) + 0.1
C     NTBLP1- NON-ZERO MEANS DO NOT READ TABLE FOR STEER
      IF(NTBLP1 .NE. 0) GO TO 4
      NPSF = NPSF1
      IB = 1
      IL = 2
      NUM = NO1
  109 DO 111 I = IB,IL
```

```
      DO 110 IJ = 1,NPSF,7
      IJL = MIN0(IJ+6,NPSF)
      READ( NIN,1006) CARDIM, (PSIFI(I,K),K=IJ,IJL)
      WRITE(NOUT,1007) CARDIM,NUM,NAMEP,I,IJ,IJL
  110 CONTINUE
  111 CONTINUE
      GO TO 4
C
C     CARD 11 STEER FOR FRONT WHEELS FOR VEHICLE 2, PSIFI(4,201)
   15 TBPSF2 = DUM(1)
      TEPSF2 = DUM(2)
      TINCP2 = DUM(3)
      NPSF2 = (TEPSF2-TBPSF2)/TINCP2 + 1.2
      NTBLP2 = DUM(4) + 0.1
C     NTBLP2- NON-ZERO MEANS DO NOT READ TABLE FOR STEER
      IF(NTBLP2 .NE. 0) GO TO 4
      NPSF = NPSF2
      IB = 3
      IL = 4
      NUM = NO2
      GO TO 109
C
C           CARD 12  TERRAIN ZONES
   16 XBP(1) = DUM(1)
      YBP(1) = DUM(2)
      XBP(2) = DUM(3)
      YBP(2) = DUM(4)
      XMU1   = DUM(5)
      XMU2   = DUM(6)
      CMU    = DUM(7)
      GO TO 4
C         CARD 13    COLLISION PARAMETERS
   17 DELPS0= DUM(1)
      DELRO0= DUM(2)
      ALAMB = DUM(3)
      ZETAV = DUM(4)
      AKV(1)= DUM(5)
      AKV(2)= DUM(6)
      AMU   = DUM(7)
      FMOVIE = DUM(8)
C        IF FMOVIE (EIGHTH FIELD ON CARD 13) NE 0.0, WRITE TAPE FOR
C        MOVIE,WITH TIRE TRACKS AND DAMAGE TABLES AT EACH TIME POINT,BUT
C        COMPUTE VDI ONLY AT END OF RUN.  PRINT INTERMEDIATE DAMAGE
C        TABLES ON FT02.
C        IF FMOVIE EQ 0.0, WRITE TAPE WITH TIRE TRACKS AT EACH TIME
C        POINT, BUT WITH DAMAGE TABLE AND VDI ONLY AT END OF RUN.
C        IF FMOVIE EQ 0.0 DO NOT WRITE FT02.
      GO TO 4
C         CARD 14    COLLISION PARAMETERS
   18 C0 =  DUM(1)
      C1 =  DUM(2)
      C2 =  DUM(3)
      GO TO 4
C           CARD 15, CRITERIA FOR COLLISION LIMITS
C            SET TO STANDARDS IN MAIN PROG.
C          THIS CARD NEED NOT BE FURNISHED. IF USED, GIVE ALL FIELDS.
   19 DO 20 I=1,8
   20 PSLM0(I) = DUM(I)
      GO TO 4
C         CARD 16    DESIRED TRAJECTORY
C       IF IOPTION .NE. 0, THEN
C       IOPT = 1, STEP INPUT
```

```
C       IOPT = 2, SINUSOIDAL INPUT
C       IOPT = 3, FEEDBACK CONTROL
C       IOPT = 4, FEEDBACK CONTROL with time delay
C       IOPT = 5, INPUT DESIRED TRAJECTORY
C       FOR OPTIONS 1,2 RUN WITH ONLY 1 VEHICLE
C
   21 IOPTION =  DUM(1)
      IF (IOPTION .EQ. 1) THEN
          FVEH0 = 1.0
          DTINT = 0.25
          WRITE(6,*) ' ENTER STEERING STEP INPUT PARAMETERS'
          WRITE(6,*) ' AMPLITUDE (DEG) AND TIME TO GO UP TO AMP (SEC):'
          READ(5,*) STEPPSI, STEPDT
      ELSEIF ( IOPTION .EQ. 2 ) THEN
          FVEH0 = 1.0
          DTINT = 0.25
          WRITE(6,*) ' ENTER STEERING SINUSOIDAL INPUT PARAMETERS'
          WRITE(6,*) ' AMPLITDUE (DEG) AND FREQUENCY (HZ):'
          READ(5,*) SINPSI, SINFREQ
      ELSEIF ( IOPTION .EQ. 3 ) THEN
          WRITE(6,*) ' ENTER FEEDBACK CONTROL PARAMETERS'
          WRITE(6,*) ' GAIN (0.01-0.20) AND LOOK-AHEAD DISTANCE (M):'
          READ(5,*) GAIN, DISLKAHD
      ELSEIF ( IOPTION .EQ. 4 ) THEN
          WRITE(6,*) ' ENTER FEEDBACK CONTROL PARAMETERS'
          WRITE(6,*) ' GAIN (0.01-0.20) AND LOOK-AHEAD DISTANCE (M):'
          READ(5,*) GAIN, DISLKAHD
          WRITE(6,*) ' ENTER TIME DELAY (DIGITIZED CYCLE IN ACTUATION)'
          WRITE(6,*) ' DTDGT'
          READ(5,*) DTDGT
      ELSEIF ( IOPTION .EQ. 5 ) THEN
          OPEN(UNIT=21, FILE='DESTRAJ.DAT')
          DO 22 I = 1,201
          READ(21,1101, END=24) XYTB(1,I), XYTB(2,I)
 1101 FORMAT(2F10.3)
   22     CONTINUE
   24 CLOSE(21)
      ENDIF
      GO TO 4
  120 IF(FMOVIE) 121,200,121
  121 IF(FVEH0 - 2.0) 122,200,122
  122 WRITE(6,1010) FMOVIE,FVEH0
 1010 FORMAT(10H0 FMOVIE =,F3.0,5X,8H FVEH0 =,F3.0 /
     1113H  NON-EXISTENT DAMAGE TABLES (ONE CAR CASE) SHOULD NOT BE WRIT
     2TEN ON TAPE AT EACH TIME POINT (FMOVIE.NE.0.). STOP )
      STOP
C           * * * * * * * * * *
  200 WRITE(NOUT,1008) HED,DADE
 1008 FORMAT(1H1,25X,20A4/26X,20A4/58X,3A4/)
      WRITE(NOUT,1012)  XCP10,   XCP20,   DELPS0,
     1               YCP10,   YCP20,   DELRO0,
     2               PSI10,   PSI20,   ALAMB,
     3               PSI1D0,  PSI2D0,  ZETAV,
     4               U10,    U20,      V10,V20
 1012 FORMAT(1H0,30X,18HINITIAL CONDITIONS,40X,21HCALCULATION CONSTANTS/
     X 13X,13HVEHICLE NO. 1,27X,13HVEHICLE NO. 2 /
     1/ 9X,10HXC10'    =,F8.3,7H INCHES,15X,10HXC20'    =,F8.3,7H INCHES
     2 ,16X,10HDELPSI   =,F8.3,8H DEGREES /
     3 9X,10HYC10'    =,F8.3,7H INCHES,15X,10HYC20'    =,F8.3,7H INCHES,
     416X,10HDELRHO   =,F8.3,7H INCHES /
     5 9X,10HPSI10    =,F8.3,8H DEGREES,14X,10HPSI20    =,F8.3,8H DEGREE
     6S,15X,10HLAMBDA   =,F8.3,21H LB/IN,PRESSURE ERROR/
```

```
    7 9X,10HPSI1D0   =,F8.3,8H DEG/SEC,14X,10HPSI2D0   =,F8.3,8H DEG/SE
    8C,15X,10HZETAV    =,F8.3,21H IN/SEC,MIN.FOR FRICT/
    9 9X,10HU10      =,F8.3,7H IN/SEC,15X,10HU20      =,F8.3,7H IN/SEC/
    X 9X,10HV10      =,F8.3,7H IN/SEC,15X,10HV20      =,F8.3,7H IN/SEC)
     WRITE(NOUT,1016) A1,    A2,      AKV(1),
    1                 B1,    B2,      AKV(2),
    2                 TR1,   TR2,     AMU,
    3                 FIZ1,  FIZ2,    C0,
    4                 FM1,   FM2,     C1
1016 FORMAT(1H0,22X,34HDIMENSIONS AND INERTIAL PROPERTIES,34X,16HDEFORM
    1ABLE LAYER/ 9X,10HA1       =,F8.3,7H INCHES,15X,10HA2       =,F8.3
    2,7H INCHES,16X,10HKV1      =,F8.3,11H LB/(IN**2)/ 9X,10HB1       =
    3,F8.3,7H INCHES,15X,10HB2       =,F8.3,7H INCHES,16X,10HKV2      =
    4,F8.3,11H LB/(IN**2) /  9X,10HTR1      =,F8.3,7H INCHES,15X,10HTR2
    5      =,F8.3,7H INCHES,16X,10HMU,FRICT =,F8.3/ 9X,2HI1,7X,1H=,F8.0
    6,13H LB-SEC**2-IN, 9X,2HI2,7X,1H=,F8.0,13H LB-SEC**2-IN,10X,2HC0,
    77X,1H=,F8.3,12H RESTITUTION/ 9X,2HM1,7X,1H=,F8.3,13H LB-SEC**2/IN,
    8 9X,2HM2,7X,1H=,F8.3,13H LB-SEC**2/IN,10X,2HC1,2X,1H=,E13.5,2X,
    9 7H VERSUS )
     WRITE(NOUT,1018) PSIR10,  PSIR20,  C2,
    1                 XF1,     XF2,
    2                 XR1,     XR2,
    3                 YS1,      YS2
1018 FORMAT( 9X,10HPSIR10   =,F8.3,8H DEGREES,14X,10HPSIR20   =,F8.3,
    1 8H DEGREES,15X,2HC2,2X,1H=,E13.5, 11H DEFLECTION /
    2 9X,10HXF1      =,F8.3,7H INCHES,15X,10HXF2      =,F8.3,7H INCHES/
    3 9X,10HXR1      =,F8.3,7H INCHES,15X,10HXR2      =,F8.3,7H INCHES/
    4 9X,10HYS1      =,F8.3,7H INCHES,15X,10HYS2      =,F8.3,7H INCHES)
     WRITE(NOUT,1020)  CSTFI(1),     CSTFI(5),   XBP(1),     YBP(1),
    1                  CSTFI(2),     CSTFI(6),   XBP(2),     YBP(2),
    2                  CSTFI(3),     CSTFI(7),   XMU1,
    3                  CSTFI(4),     CSTFI(8),   XMU2,   CMU
1020 FORMAT(1H0,55X,15HTIRE PROPERTIES /30X,20H CORNERING STIFFNESS ,
    1 41X,37H TIRE-TERRAIN COEF AND TERRAIN ZONES /9X,10HC(1)     =,
    2 F8.0,7H LB/RAD, 15X,10HC(5)     =,F8.0,7H LB/RAD,16X,6HXB1' =,
    3 F8.3,4H IN.,   5X, 6HYB1' =,    F8.3,4H IN.   / 9X,10HC(2)     =,
    4 F8.0,7H  ''   , 15X,10HC(6)     =,F8.0,7H  ''   ,16X,6HXB2' =,
    5 F8.3,4H IN.,   5X, 6HYB2' =,    F8.3,4H IN.   / 9X,10HC(3)     =,
    6 F8.0,7H  ''   , 15X,10HC(7)     =,F8.0,7H  ''   ,16X,6HXMU1 =,
    7 F8.3 / 9X,10HC(4)     =,F8.0,7H  ''   ,15X,10HC(8)     =,F8.0,
    8  7H  ''   ,16X,6HXMU2 =,F8.3 /90X,6HCMU  =,E13.5 )
     WRITE(NOUT,1022) PSLM10,     PSLM50,        T0,
    1                 PSLM20,     PSLM60,        TF,
    2                 PSLM30,     PSLM70,        DTTRA0,
    3                 PSLM40,     PSLM80,        DTCOL0
1022 FORMAT(1H0,12X,17H PSIB RANGE TESTS,20X,22H PSIBI FOR RHOBI TESTS,
    119X,20HPROGRAM CONTROL DATA /
    2 13X,19H COLLISION CRITERIA, 18X,19H COLLISION CRITERIA /
    3                              9X,10HPSILIM1  =,F8.3,8H DEGREES,14X
    4,10HPSILIM5  =,F8.3,8H DEGREES,15X,10HT0       =,F8.3,11H SEC.,BEG
    5IN/ 9X,10HPSILIM2  =,F8.3,5H   '',17X,10HPSILIM6  =,F8.3,5H   '',
    618X,10HTF       =,F8.3,9H  ''  END/ 9X,10HPSILIM3  =,F8.3,5H   '',
    717X,10HPSILIM7  =,F8.3,5H   '',18X,10HDTTRAJ   =,F8.3,23H  ''  INT
    8EG. INTVL,TRAJ/ 9X,10HPSILIM4  =,F8.3,5H   '',17X,10HPSILIM8  =,
    9F8.3,5H   '',18X,10HDTCOLL   =,F8.3,23H  ''  INTEG.INTVL,COLL )
     WRITE(NOUT,1024)  DTCLT0,DTPRN0,UVMIN,PSIDMN,FVEH0,FMOVIE
1024 FORMAT( 90X,10HDTCOLT   =,F8.3, 23H  ''  INTEG.INTVL,CPOS /
    1        90X,10HDTPRNT   =,F8.3,21H  ''  PRINT INTERVAL    /
    2        90X,10HUVMIN    =,F8.3,21H IN/SEC STOPPING TEST   /
    3        90X,10HPSIDOT   =,F8.3,21HDEG/SEC STOPPING TEST   /
    4        90X,16HNO.OF VEHICLES =,F3.0 /
    5        90X,10H FMOVIE = ,F3.0,29H(ZERO,FINAL DAMAGE TABLE TAPE/
```

```
    6       103X,29H(NON-ZERO,DAMAGE HISTORY TAPE/
    7       103X,29H(ALSO WRITTEN ON FORTRAN 2.  /
    8       103X,29H   (TAPE IS ALWAYS FORTRAN 1) )
     LINES = 49
     ITZER1 = 1
     IB = 1
     ILL = 4
     NTQ = NTQ1
     NUM = NO1
 220 DO 221 I = IB,ILL
     DO 221 J = 1,NTQ
     IF(TII(I,J)) 225,221,225
 221 CONTINUE
     WRITE(NOUT,1030) IB,ILL,NTQ,NUM
1030 FORMAT(14H0((TII(I,K),I=,I2,1H,,I2,6H),K=1,,I4,40H) TORQUE TABLES
    XALL ZERO FOR VEHICLE NO.,I2)
     LINES = LINES + 2
     IF(IB-1) 222,222,224
 222 ITZER1 = 0
 223 ITZER2 = 1
     IB = 5
     ILL = 8
     NTQ = NTQ2
     NUM = NO2
     GO TO 220
 224 ITZER2 = 0
     GO TO 229
 225 IF(IB-1) 223,223,229
 229 IPZER1 = 1
     IB = 1
     ILL = 2
     NPSF = NPSF1
     NUM = NO1
 230 DO 231 I = IB,ILL
     DO 231 J = 1,NPSF
     IF( PSIFI(I,J)) 235,231,235
 231 CONTINUE
     WRITE(NOUT,1032) IB,ILL,NPSF,NUM
1032 FORMAT(16H0((PSIFI(I,K),I=,I2,1H,,I2,6H),K=1,,I4,39H) STEER TABLES
    X ALL ZERO FOR VEHICLE NO., I2)
     LINES = LINES + 2
     IF(IB-1) 232,232,234
 232 IPZER1 = 0
 233 IPZER2 = 1
     IB = 3
     ILL = 4
     NPSF = NPSF2
     NUM = NO2
     GO TO 230
 234 IPZER2 = 0
     GO TO 239
 235 IF(IB-1) 233,233,239
 239 IF(ITZER1 + ITZER2 + IPZER1 + IPZER2) 240,290,240
 240 NTPR = 0
     TTARG(1) = 0.0
     IF(ITZER1) 242,257,242
 242 IF(TINCQ1) 243,257,243
 243 NTQ = NTQ1
     IB = 1
     ILL = 4
     NUM = NO1
     Y = TBTQ1
```

```
      TINCRQ = TINCQ1
  245 WRITE(NOUT,1009)
 1009 FORMAT(1H0)
      LINES = LINES + 2
      DO 246 I = 1,NTQ
      TTARG(I) = Y
      Y = Y + TINCRQ
  246 CONTINUE
      NNADD = 0
      IF(MOD(NTQ,3) .NE. 0) NNADD = 1
      NTPR = NTQ/3 + NNADD
      NTPR2 =  2*NTPR
      LINES = LINES + NTPR + 4
      IF(LINES.LE.60) GO TO 250
      WRITE(NOUT,1008) HED,DADE
      LINES = NTPR + 8
  250 WRITE(NOUT,1050) NUM
 1050 FORMAT(15H0   VEHICLE NO.,I2 /
     1        1X,   3( 14X, 29H TRACTIVE OR BRAKING FORCE LB )/ 1X,
     2 3(7X,4H SEC, 6X,2HRF,5X,3H LF, 5X,3H RR, 5X,3H LR))
      DO 255 J = 1,NTPR
      IJ = J
      IJL = MIN0(NTQ,IJ+NTPR2 )
      WRITE(NOUT,1052)(TTARG(II),(TII(K,II),K=IB,ILL),II=IJ,IJL,NTPR)
 1052 FORMAT(1X,3(F11.3,4F8.2))
  255 CONTINUE
      IF(IB-5) 257,270,270
  257 IF(ITZER2)258,270,258
  258 IF(TINCQ2) 259,270,259
  259 NTQ = NTQ2
      IB = 5
      ILL = 8
      NUM = NO2
      Y = TBTQ2
      TINCRQ = TINCQ2
      GO TO 245
C
  270 IF(IPZER1) 271,280,271
  271 IF(TINCP1) 272,280,272
  272 NPSF = NPSF1
      IB = 1
      ILL = 2
      NUM = NO1
      Y = TBPSF1
      TINCRP = TINCP1
  275 WRITE(NOUT,1009)
      LINES = LINES + 2
      DO 276 I = 1,NPSF
      TTARG(I) = Y
      Y = Y + TINCRP
  276 CONTINUE
      NNADD = 0
      IF(MOD(NPSF ,5) .NE. 0) NNADD = 1
      NTPR = NPSF/5  + NNADD
      NTPR5 = 4*NTPR
      LINES = LINES + NTPR + 4
      IF(LINES.LE.60) GO TO 278
      WRITE(NOUT,1008) HED,DADE
      LINES = NTPR + 8
  278 WRITE(NOUT,1054) NUM
 1054 FORMAT(15H0   VEHICLE NO.,I2 /
     1        1X,  5( 9X,17H STEER ANGLES,DEG)  /1X,5(5X,4H SEC,1X,16H
```

```
    1 PSIF1   PSIF2))
      DO 279 J = 1,NTPR
      IJ = J
      IJL = MIN0(NPSF , IJ+NTPR5 )
      WRITE(NOUT,1056)(TTARG(II),(PSIFI(K,II),K=IB,ILL),II=IJ,IJL,NTPR)
1056 FORMAT(1X,5(F10.3,2F8.3))
 279 CONTINUE
      IF(IB-3) 280,290,290
 280 IF(IPZER2) 282,290,282
 282 IF(TINCP2) 283,290,283
 283 NPSF = NPSF2
      IB = 3
      ILL = 4
      NUM = NO2
      Y = TBPSF2
      TINCRP = TINCP2
      GO TO 275
 290 RETURN
      END
```

```
      SUBROUTINE SMAC
C     Version 03-02-83
C     SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
C     PROGRAM VERSION 10-31-78
C
C     THE FOLLOWING MODIFICATIONS HAVE BEEN MADE TO THE ORIGINAL SOURCE
C     RECEIVED FROM CALSPAN
C
C         (A)  STATEMENT NUMBER 55 IN SUBROUTINE TRAJ WHICH READ
C                                  55 RATIO = XNUM/XDEN
C              NOW READS
C                                  55 RATIO = ATAN2(XNUM,XDEN)
C
C         (B)  AN HSRI VERSION DATE WRITE STATEMENT HAS BEEN ADDED AS
C              THE FIRST EXECUTABLE STATEMENT IN MAIN
C
C         (C)  THE UNREFERENCED VARIABLE "BLNK" WAS DELETED FROM
C              SUBROUTINES OUT2, ADJEND, DAMAGE, DEFMJ, AND PLOTAP
C
C         (D)  THE UNREFERENCED VARIABLE "STAR" WAS DELETED FROM
C              SUBROUTINES OUT2 AND DAMAGE
C
C         (E)  THE UNREFERENCED VARIABLES "PIO4", "PIO6", AND "PIO9"
C              WERE DELETED FROM SUBROUTINE DAMAGE
C
C         (F)  THE MAIN PROGRAM HAS BEEN CONVERTED TO A SUBROUTINE
C
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
C
      COMMON/INTG/NEQ,T,DT,VAR(12),DER(12)
C
      EQUIVALENCE (VAR(1),XCP1),(VAR(2),YCP1),(VAR(3),PSI1),
     1            (VAR(4),PSI1D),(VAR(5),U1),(VAR(6),V1),
     2            (VAR(7),XCP2),(VAR(8),YCP2),(VAR(9),PSI2),
     3            (VAR(10),PSI2D),(VAR(11),U2),(VAR(12),V2)
C
      EQUIVALENCE (DER(1),DXCP1),(DER(2),DYCP1),(DER(3),DPSI1),
     1            (DER(4),DPSI1D),(DER(5),DU1),(DER(6),DV1),
     2            (DER(7),DXCP2),(DER(8),DYCP2),(DER(9),DPSI2),
     3            (DER(10),DPSI2D),(DER(11),DU2),(DER(12),DV2)
C
      COMMON/INPT/  T0,TF,DTTRA0,DTCOL0,DTCLT0,DTPRN0,UVMIN,PSIDMN,FVEH0
     1             ,XCP10,YCP10,PSI10,PSI1D0,U10,V10,
     2              XCP20,YCP20,PSI20,PSI2D0,U20,V20,
     3              A1,B1,TR1,FIZ1,FM1,PSIR10,XF1,XR1,YS1,
     4              A2,B2,TR2,FIZ2,FM2,PSIR20,XF2,XR2,YS2,
     5              CSTFI(8),TBTQ1,TETQ1,TINCQ1,NTBLQ1,NTQ1,
     6              TBTQ2,TETQ2,TINCQ2,NTBLQ2,NTQ2,TII(8,201),
     7              TBPSF1,TEPSF1,TINCP1,NTBLP1,NPSF1,
     8              TBPSF2,TEPSF2,TINCP2,NTBLP2,NPSF2,PSIFI(4,201),
     9              XBP(2),YBP(2),XMU1,XMU2,CMU
C
      COMMON/INPT/  DELPS0,DELRO0,ALAMB,ZETAV,AKV(2), AMU,C0,C1,C2,
     1              PSLM10,PSLM20,PSLM30,PSLM40,PSLM50,PSLM60,PSLM70,
     2              PSLM80,HED(40),DADE(3),XINPUT(9)
C
      COMMON /OPTIONS/ IOPTION, DTINT, STEPPSI, STEPDT, SINPSI, SINFREQ,
     1                 GAIN, DISLKAHD, XYTB(2,201), NOUT1, NOUT2
      COMMON/DELAY/ TDGT, DTDGT, DY1LST, YAW1LST, DY2LST, YAW2LST
C
```

```
      COMMON/INTGR/JCC(4),JCOR(4),NPSIB,IPSIB0,IPSIB,IBB,IFF,NDTAB,
     1             I9,J9(361,2),  NPSJB,IPSJB,I3,ILAST,IL,IND1,IND2,INDI
     2             ,INDJ,ISTOP,ITZER1,ITZER2,IPZER1,IPZER2,ICOUNT,INDXB,
     3             ICOLLP,ICOLL,IQ,IND,IVEH,LL,ICTVDI,ITCOL,ITFLG1,
     4             ITFLG2,NWRDB1,NWRDE1,NWRDB2,NWRDE2,ICOLAC
C
C     IN COMMON /COMP/ DO NOT DISTURB THE ORDER OF THE VARIABLES
C     WHICH ARE EQUIVALENCED TO ARRAYS.
C
      COMMON/COMP/  DTTRAJ,DTCOLL,DTCOLT,DTPRNT,UVMN2,PSIDMA,TPRINT,DTP,
     1              U1V1SQ,U2V2SQ,DELPSI,DELPST,DELPS2,DELRHO,
     2              PSILM1,PSILM2,PSILM3,PSILM4,PSILM5,PSILM6,PSILM7,
     3              PSILM8,EJJ(4,2),GJJ(4,2),C1OC2,PSCC,PSIR1,PSIR2,
     4             XCMXC1,XCMXC2,YCMYC1,YCMYC2,SPSI1,SPSI2,CPSI1,CPSI2,
     5             XCPI,YCPI,PSII,XCPJ,YCPJ,PSIJ,XCMXC,YCMYC,SPSIJ,
     6             CPSIJ,XFI,XRI,YSI,XFJ,XRJ,YSJ,AKVI,AKVJ,EJ,GJ,
     7             XPJ(4),YPJ(4),ANGYX,SPAN,PSIBPI(4),PSIBPB,PSIBPF,
     8             FNPSIB,PSIBB,PSIB,PSFMPS,SCPSIB,CSPSIB,
     9             RHOBI,RHOBIN(4),PSIBIJ,SPSBIJ,CPSBIJ,XBIJ,YBIJ,PSJTB
C
      COMMON/COMP/ PSIBPJ,FNPSJB,PSIBJ,RHOBJ,XYJSQR,SPSIB,CPSIB,XBI,YBI,
     1         PRESI,PRESJ,XYSR,DELTA,CRHO,RHOBIC,RHOBMX,RHOBIT,RHOBT,
     2             SPSI21,CPSI21,X2TEM,Y2TEM,XX,YY,PAV,XXYYSR,FN1,SPSF,
     3             CPSF,X1AV,Y1AV,XYAVSR,X2TERM,Y2TERM,SPS21F,CPS21F,
     4             X2AV,Y2AV,VT1AV,VT2AV,VTMVT,FRICT,FNX1,FNY1,FNN1,
     5             FNX2,FNY2,FNN2,SFX1,SFY1,SFN1,SFX2,SFY2,SFN2,
     6              SFX1C,SFX2C,SFY1C,SFY2C,SFN1C,SFN2C,
     7              SFX1T,SFX2T,SFY1T,SFY2T,SFN1T,SFN2T,  TCOL,TEND,
     8              TREST1,TREST2,TCOLS1,TCOLS2,P1TEM,P2TEM,U1V1E,U2V2E,
     9              EXTRA(10)
C
      COMMON/COMPT/ W(8),XWP(8),YWP(8),FLAGW(8),PSIW(8),TQW(8),
     1             TFX,TFY,TMOM,PSIRR,COEF,COEF1,COEF2,TRD2,TRD21,TRD22,
     2             XBMXB,FNUM,A,B,FIZ,FMASS,XC,YC,U,V,PSIC,PSICD,UD2,
     3             SGNU,SGNI,AOMB,CSPSIC,SNPSIC,TRCPSC,TRSPSC,TRPSCD,
     4             XI,XW,YW,SX,SY,S,XPP,YPP,RHO,RHOP,XMUS,XMU,PSI,XNUM,
     5             XDEN,RATIO,ALFA,F,TIF,FC,FCOS,SMAL,FSMX,FS,BETB,FX,FY
     6             ,CSPSI,SNPSI,XCXC,YCYC,SPJ,CPJ,SPIJ,CPIJ,XFJJ,XRJJ,
     7              YSJJ,EJJI,GJJI,XICORJ,YICORJ,NTQ,TBTQ,TETQ,TINCRQ,
     8             NPSF,TBPSF,TEPSF,TINCRP
C
      COMMON/TAB/  TRHOB(361,2),TROBMX(361,2),TPSIB(361,2),
     1             TCPSIB(361,2),TSPSIB(361,2),TXB1(100),TYB1(100),
     2             TPSB1(100),TPRES1(100)
C
      DIMENSION   XCMXCI(2),YCMYCI(2),SPSII(2),CPSII(2),PSLM0(8),PSLM(8)
C
      EQUIVALENCE (XCMXCI(1),XCMXC1),(YCMYCI(1),YCMYC1),(SPSII(1),SPSI1)
     1           ,(CPSII(1),CPSI1),(PSLM0(1),PSLM10),(PSLM(1),PSILM1)
C
C     COMMON VDICON HAS CONSTANTS USED FOR VEHICLE DAMAGE INDEX
C
      COMMON/VDICON/EXTF1( 9),EXTR1( 9),EXTB1( 9),PSP1(12),
     2              EXTF2( 9),EXTR2( 9),EXTB2( 9),PSP2(12),NCOL51(10),
     2              NCOL52(10),MMM( 9)
C
      EQUIVALENCE (XINPUT(1),FMOVIE)
C
      COMMON/COLLPT/ XCP1C,YCP1C,PSI1C,U1C,V1C,PSI1DC,
     1               XCP2C,YCP2C,PSI2C,U2C,V2C,PSI2DC
C
      COMMON/OUTPT/TMOU(55),XCP1OU(55),YCP1OU(55),PSI1OU(55),U1OU(55),
```

```
     1            V1OU(55),PS1DOU(55),AX1OU(55),AY1OU(55),XCP2OU(55),
     2             YCP2OU(55),PSI2OU(55),U2OU(55),V2OU(55),PS2DOU(55),
     3          AX2OU(55),AY2OU(55),ALF1OU(55),ALF2OU(55),XWP1OU(55),
     4             XWP2OU( 55),XWP3OU( 55),XWP4OU( 55), XWP5OU( 55),
     5             XWP6OU( 55),XWP7OU( 55),XWP8OU( 55), YWP1OU( 55),
     6             YWP2OU( 55),YWP3OU( 55),YWP4OU( 55), YWP5OU( 55),
     7             YWP6OU( 55),YWP7OU( 55),YWP8OU( 55), FLAG1( 55),
     8             FLAG2( 55), FLAG3( 55),FLAG4( 55),FLAG5( 55),
     9         FLAG6( 55),FLAG7( 55),FLAG8( 55),ACC1OU(55),ACC2OU(55)
C
      COMMON/OUTPT /ARRXY1(10),ARRX1(10),ARRY1(10),ARRT1(10),ARRPS1(10),
     1   ARRDV1(10),ARRXY2(10),ARRX2(10),ARRY2(10),ARRT2(10),ARRPS2(10),
     2   ARRDV2(10),ABIG1,AXBIG1,AYBIG1,TBIG1,DELVS1,ACC1,ACCP1,AXX1,
     3  AYY1,ACTST1,ABIG2,AXBIG2,AYBIG2,TBIG2,DELVS2,ACC2,ACCP2,AXX2,
     4  AYY2,ACTST2,TM,TIMEP,NARRP1(10),NARRP2(10),NUM1,NUM2,
     5         ISAV1,ISAV2,IACDV1,IACDV2,IPAGE
C
      COMMON /MOVIE/ MOVIE,NMSG
C
      DIMENSION XNTG0(26),XNPT0(2556),INTGR0(768),TAB0(4010),COMP0(189)
      DIMENSION COMPT0(129)
      EQUIVALENCE (T,XNTG0(1)),(T0,XNPT0(1)),(JCC(1),INTGR0(1))
      EQUIVALENCE(DTTRAJ,COMP0(1)),(W(1),COMPT0(1)),(TRHOB(1,1),TAB0(1))
      DIMENSION VDICN(107),COLLP1(12)
      EQUIVALENCE(VDICN (1),EXTF1(1)) ,(COLLP1(1),XCP1C)
      DIMENSION OUTPT0(2644)
      EQUIVALENCE (TMOU(1),OUTPT0(1))
      DATA NO1/1/, NO2/2/
C
C      G     = 386.4
C      RAD   = .0174532925
C      PI    = 3.1415926536
C      TWOPI = 2.0 * PI
C      PIO2  = 0.5 * PI
C      PI32  = 1.5 * PI
C      ND2   = 2
C      ND4   = 4
C      ND8   = 8
C      NIN   = 5
C      NOUT  = 6
      NMSG  = 6
      MOVIE = 0
C
C     PRINT HSRI VERSION DATE
C
C      WRITE(NOUT,100)
C  100 FORMAT('0SMAC - HSRI Version 10-31-78')
C
C       WRITE(NOUT, 100)
C  100 FORMAT('0SMAC - PATH VERSION 1-1-97')
C
         DO 8   I = 1,2556
    8     XNPT0(I) = 0.0
C
C     INPUT AREA CLEARED ONLY ONCE PER JOB
C
      PSLM10 = 70.
      PSLM20 = 110.
      PSLM30 = 250.
      PSLM40 = 290.
      PSLM50 = 10.
      PSLM60 = 170.
```

```
      PSLM70 = 190.
      PSLM80 = 350.
C
   10     DO 11 I = 1,26
   11     XNTG0(I) = 0.0
C
          DO 12 I = 1,189
   12     COMP0(I) = 0.0
C
          DO 13 I = 1,129
   13     COMPT0(I) = 0.0
C
          DO 14 I = 1,768
   14     INTGR0(I) =0
C
          DO 15 I = 1,1444
   15     TAB0(I) = -1.0
C
          DO 16 I = 1445,4010
   16     TAB0(I) = 0.0
C
          DO 160 I = 1,107
  160     VDICN(I) = 0.0
C
          DO 161 I = 1,12
  161     COLLP1(I) = 0.0
C
          DO 17 I = 1,2644
   17     OUTPT0(I) = 0.0
C
      IPRTY = 1
      ABIG1 = -1.E10
      ABIG2 = -1.E10
C
C     SUBROUTINE INPUT READS CARDS FURNISHED FOR THIS CASE, ALLOWS
C     SUCCESSIVE CASES WITH ONLY THOSE CARDS WHICH ARE CHANGED,
C     AND PRINTS ALL INPUT FOR EACH CASE.
C
C      CALL INPUT(&9999)
      CALL INPUT(*9999)
C
C     SUBROUTINE CNSTNT RESETS NMSG=2,MOVIE=1 IF FMOVIE.NE.0,INPUT
C
      CALL CNSTNT
      TPRINT = T0
C
      TDGT = T0
C
      T = T0
      TCOL = T0
      TIMEP = T0
      TM = T0
      TTADD = 0.1 * DTCOLL
      ICOLTR = 1
      DT = DTCOLT
      DTP = DT
C      CALL PLOTAP(1)
      CALL RNGKT1(1,  NEQ,T,DT,VAR,DER)
      CALL ACCEL(G,DU1,DV1,DPSI1,U1,V1,AXX1,AYY1,ACC1)
      IF(IVEH.EQ.0) GO TO 19
      CALL ACCEL(G,DU2,DV2,DPSI2,U2,V2,AXX2,AYY2,ACC2)
C
```

```
C     SUBR SAVMAX CALLS SUBR DELTAV, INITIALIZES.
C
      CALL SAVMAX (ARRXY1,ARRX1,ARRY1,ARRT1,ARRDV1,ARRPS1,NARRP1,
     1            TM,TIMEP,AXX1,AYY1,ACC1,ACCP1,DELVS1,ACTST1,
     2            ABIG1,AXBIG1,AYBIG1,TBIG1,NUM1,ISAV1 ,IACDV1,
     3            IPRTY,ISTOP)
      CALL SAVMAX (ARRXY2,ARRX2,ARRY2,ARRT2,ARRDV2,ARRPS2,NARRP2,
     1            TM,TIMEP,AXX2,AYY2,ACC2,ACCP2,DELVS2,ACTST2,
     2            ABIG2,AXBIG2,AYBIG2,TBIG2,NUM2,ISAV2 ,IACDV2,
     3            IPRTY,ISTOP)
      GO TO 31
C
C     T IS T0,  IF TWO VEHICLES, CHECK FOR POSSIBLE CONTACT AT ST 31
C
   18 WRITE(NOUT,6018) T,DT,DTP
 6018 FORMAT('0 At time =',F8.4,' sec, New interval =',F8.4,' sec, Previ
     1ous interval =',F8.4,' sec')
      CALL RNGKT1(1,  NEQ,T,DT,VAR,DER)
      CALL ACCEL(G,DU1,DV1,DPSI1,U1,V1,AXX1,AYY1,ACC1)
      IF(IVEH.EQ.0) GO TO 19
      CALL ACCEL(G,DU2,DV2,DPSI2,U2,V2,AXX2,AYY2,ACC2)
   19 IF(ISTOP.NE.0) GO TO 80
C
C     OMIT CALL TO OUTDIS
C     IF(IVEH.NE.0) CALL OUTDIS
C     ENTRY OUTDIS IN SUBROUTINE OUTPUT PRINTS RHOB TABLE EVERY STEP
C
   20 IF(TPRINT.GT.(T+0.1*DT)) GO TO 21
      CALL OUTPUT(1)
C      CALL PLOTAP(2)
      TPRINT = TPRINT + DTPRNT
   21 ICOLLP = 0
C
      IF (IOPTION .NE. 4) GO TO 23
      IF(TDGT.GT.(T+0.1*DT)) GO TO 23
C      WRITE(14,*) T, TDGT
      DY1LST = YCP1
      YAW1LST = PSI1
      DY2LST = YCP2
      YAW2LST = PSI2
      TDGT = TDGT + DTDGT
   23 CONTINUE
C
      CALL RNGKT1(2,  NEQ,T,DT,VAR,DER)
      IF(ISTOP.NE.0) GO TO 80
      IF(T.GE.TF) GO TO 80
      TIMEP = TM
      TM = T
      ACCP1 = ACC1
      CALL ACCEL(G,DU1,DV1,DPSI1,U1,V1,AXX1,AYY1,ACC1)
      IF(IVEH.EQ.0) GO TO 220
      ACCP2 = ACC2
      CALL ACCEL(G,DU2,DV2,DPSI2,U2,V2,AXX2,AYY2,ACC2)
      CALL SAVMAX (ARRXY1,ARRX1,ARRY1,ARRT1,ARRDV1,ARRPS1,NARRP1,
     1            TM,TIMEP,AXX1,AYY1,ACC1,ACCP1,DELVS1,ACTST1,
     2            ABIG1,AXBIG1,AYBIG1,TBIG1,NUM1,ISAV1 ,IACDV1,
     3            IPRTY,ISTOP)
      CALL SAVMAX (ARRXY2,ARRX2,ARRY2,ARRT2,ARRDV2,ARRPS2,NARRP2,
     1            TM,TIMEP,AXX2,AYY2,ACC2,ACCP2,DELVS2,ACTST2,
     2            ABIG2,AXBIG2,AYBIG2,TBIG2,NUM2,ISAV2 ,IACDV2,
     3            IPRTY,ISTOP)
C
```

```
  220 ITFG1P  = ITFLG1
      ITFG2P = ITFLG2
      ITFLG1 = 0
      ITFLG2 = 0
      U1V1SQ = U1**2 + V1**2
      IF(U1V1SQ.GT.UVMN2) GO TO 25
      IF(ABS(PSI1D).GT.PSIDMA) GO TO 25
      ITFLG1 = 1
      IF(ITFLG1.EQ.ITFG1P) GO TO 241
      TREST1 = T
      TCOLS1 = TREST1 - TCOL
      U1V1E = (SQRT(U1V1SQ))/ 12.0
      P1TEM = PSI1D/RAD
  241 IF(IVEH) 26,80,26
C
   25 IF(IVEH.EQ.0) GO TO 20
   26 U2V2SQ = U2**2 + V2**2
      IF(U2V2SQ.GT.UVMN2) GO TO 31
      IF(ABS(PSI2D).GT.PSIDMA) GO TO 31
      ITFLG2 = 1
      IF(ITFLG2.EQ.ITFG2P) GO TO 281
      TREST2 = T
      TCOLS2 = TREST2 - TCOL
      U2V2E  = (SQRT(U2V2SQ))/ 12.0
      P2TEM  = PSI2D/RAD
  281 IF(ITFLG1) 80,31,80
C
C     OMIT CALL TO OUTDIS
C
C  31 CALL OUTDIS
C
C     ST 31 IS REACHED ONLY WHEN IVEH.NE.0
C     TEST FOR CONTACT
C     COSINE(PSI1-PSI2) SAME AS COSINE(PSI2-PSI1)
C
   31 CPIJ = CPSI1 * CPSI2  +  SPSI1 * SPSI2
      I = 1
      J = 2
   32 XCXC = XCMXCI(I)
      YCYC = YCMYCI(I)
      SPJ  = SPSII(J)
      CPJ  = CPSII(J)
      SPIJ = SPSII(I) * CPJ  -  CPSII(I)*SPJ
      XFJJ = EJJ(1,J)
      XRJJ = EJJ(3,J)
      YSJJ = GJJ(1,J)
      XICRTM = YCYC*SPJ + XCXC*CPJ
      YICRTM = YCYC*CPJ - XCXC*SPJ
C
C        VEHICLE I CORNER POINTS IN VEHICLE J COORDINATES
C        RF = 1,  LF = 2,  RR = 3,  LR = 4
C
         DO 35 IC = 1,4
         EJJI = EJJ(IC,I)
         GJJI = GJJ(IC,I)
         XICORJ = XICRTM             + EJJI*CPIJ - GJJI*SPIJ
         YICORJ = YICRTM             + EJJI*SPIJ + GJJI*CPIJ
         IF((XRJJ.LT.XICORJ).AND.(XICORJ.LT.XFJJ).AND.
     1     ((-YSJJ).LT.YICORJ).AND.(YICORJ.LT.YSJJ)) GO TO 37
   35    CONTINUE
      IF(I.GT.1) GO TO 45
      I = 2
```

```
      J = 1
      GO TO 32
C
   37 IF(ITCOL.NE.0) GO TO 39
      TCOL   = T
      ITCOL  = 1
      XCP1C  = XCP1
      YCP1C  = YCP1
      PSI1C  = PSI1
      U1C    = U1
      V1C    = V1
      PSI1DC = PSI1D
      XCP2C  = XCP2
      YCP2C  = YCP2
      PSI2C  = PSI2
      U2C    = U2
      V2C    = V2
      PSI2DC = PSI2D
      GO TO 41
C
   39 IF(ICOLTR.EQ.0) GO TO 42
      IF((ACC1.LE.1.0).AND.(ACC2.LE.1.0)) GO TO 20
   41 ICOLL  = 1
      ICOLTR = 0
      DTP    = DT
      DT     = DTCOLL
      DTPRNT = DTCOLL
      TPRINT = T
C
C      TDGT = T
C
      IQ     = 0
      CALL SETIND(PSI1,PSI2,XCMXC1,YCMYC1, PSCC1,PSCC2,IND1,IND2)
      GO TO 18
C
   42 IF((ACC1.GT.1.0).OR.(ACC2.GT.1.0)) GO TO 20
      IF(ICOLAC.GT.5) GO TO 44
      ICOLAC= ICOLAC+1
      GO TO 20
C
   44 TTEST = T + TTADD
      IF( AMOD(TTEST,DTCOLT)  - DTCOLL) 47,20,20
   45 IF(ICOLTR.NE.0) GO TO 50
      TTEST = T + TTADD
      IF( AMOD(TTEST,DTCOLT)  - DTCOLL) 47,20,20
   47 ICOLTR = 1
      ICOLLP = 1
      ICOLAC = 0
      DTP = DT
      DT = DTCOLT
C
C     DTPRNT=DTPRN0     RESET NOT DONE
C
      TPRINT = T
C
C      TDGT = T
C
      IND = 1
      GO TO 53
C
   50 IF (IQ.NE.100) GO TO 53
      DTP = DT
```

```
      DT  = DTTRAJ
      TPRINT = T
C
C      TDGT = T
C
      IND = 1
   53 IQ = IQ + 1
   55 IF(IND.EQ.0) GO TO 20
      IND = 0
      GO TO 18
C
   80 NMSG  = NOUT
      IPRTY = 2
      ACCP1 = ACC1
      CALL ACCEL(G,DU1,DV1,DPSI1,U1,V1,AXX1,AYY1,ACC1)
      IF(IVEH.EQ.0) GO TO 802
      ACCP2 = ACC2
      CALL ACCEL(G,DU2,DV2,DPSI2,U2,V2,AXX2,AYY2,ACC2)
      CALL SAVMAX (ARRXY1,ARRX1,ARRY1,ARRT1,ARRDV1,ARRPS1,NARRP1,
     1            TM,TIMEP,AXX1,AYY1,ACC1,ACCP1,DELVS1,ACTST1,
     2            ABIG1,AXBIG1,AYBIG1,TBIG1,NUM1,ISAV1 ,IACDV1,
     3            IPRTY,ISTOP)
      CALL SAVMAX (ARRXY2,ARRX2,ARRY2,ARRT2,ARRDV2,ARRPS2,NARRP2,
     1            TM,TIMEP,AXX2,AYY2,ACC2,ACCP2,DELVS2,ACTST2,
     2            ABIG2,AXBIG2,AYBIG2,TBIG2,NUM2,ISAV2 ,IACDV2,
     3            IPRTY,ISTOP)
  802 CALL OUTPUT(2)
C      CALL PLOTAP(3)
      IF(ITFLG1.EQ.0) GO TO 84
      WRITE(NOUT,6025) TREST1,NO1,U1V1E,NO1,P1TEM
 6025 FORMAT('0  At T =',F8.4,' sec, Vehicle ',I2,' moving slowly'/22X,
     1       ' Speed =',E13.5,' ft/sec, PSI ',I1,' D =',E13.5,
     2' deg/sec')
      IF(IVEH.NE.0) GO TO 85
      WRITE(NOUT,6026)
 6026 FORMAT('0  STOP, this case has one vehicle')
      GO TO 90
C
   84 IF(IVEH.EQ.0) GO TO 90
   85 IF(ITFLG2.EQ.0) GO TO 90
      WRITE(NOUT,6025) TREST2,NO2,U2V2E,NO2,P2TEM
      IF(ITFLG1.EQ.0) GO TO 90
      WRITE(NOUT,6028) T
 6028 FORMAT('0 STOP AT T=',F8.4,' sec, both vehicles moving slowly')
   90 IF(ISTOP.EQ.0) GO TO 10
C
C     ST 10 REPEATS WITH NEW INPUT, RETAINING DESIRED OLD INPUT
C
      WRITE(NOUT,6080)  T,ISTOP

 6080 FORMAT('0 STOP AT ',F9.4,' sec, ISTOP =',I3)
 9999 RETURN
      END
```

```
      PROGRAM SMACPATH
C     Version 11-12-97
C     SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
C     PROGRAM VERSION 10-31-78
C
C     THE FOLLOWING MODIFICATIONS HAVE BEEN MADE TO THE ORIGINAL SOURCE
C     RECEIVED FROM CALSPAN
C
C     MOVE COMMON BLOCK/CONST/ TO MAIN PROGRAM
C     USE FILE UNITS 15, 16 FOR INPUT, OUTPUT
C
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
      COMMON /OPTIONS/ IOPTION,DTINT,STEPPSI,STEPDT,SINPSI,SINFREQ,
     1                 GAIN,DISLKAHD,XYTB(2,201),NOUT1,NOUT2
C
C
      G     = 386.4
      RAD   = .0174532925
      PI    = 3.1415926536
      TWOPI = 2.0 * PI
      PIO2  = 0.5 * PI
      PI32  = 1.5 * PI
      ND2   = 2
      ND4   = 4
      ND8   = 8
      NIN   = 15
      NOUT  = 16
C       OPEN ADDITIONAL FILES FOR OPTIONS
      NOUT1 = 17
      NOUT2 = 18
C
C     OPEN FILES FOR INPUT AND OUTPUT
C
      OPEN (UNIT=NIN, FILE='INPUT.DAT')
      OPEN (UNIT=NOUT, FILE='OUTPUT.DAT', STATUS='UNKNOWN')
      OPEN (UNIT=NOUT1, FILE='VEH1.DAT', STATUS='UNKNOWN')
      OPEN (UNIT=NOUT2, FILE='VEH2.DAT', STATUS='UNKNOWN')
C
C     CALL SMAC
C
      CALL SMAC
C
C     CLOSE FILES
C
      CLOSE(NIN)
      CLOSE(NOUT)
      CLOSE(NOUT1)
      CLOSE(NOUT2)

      STOP
      END
```

```
      SUBROUTINE OUTPUT(IPRT)
C     Version 03-03-83
C     SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
C
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
C
      COMMON/INTG/NEQ,T,DT,VAR(12),DER(12)
C
      EQUIVALENCE (VAR(1),XCP1),(VAR(2),YCP1),(VAR(3),PSI1),
     1            (VAR(4),PSI1D),(VAR(5),U1),(VAR(6),V1),
     2            (VAR(7),XCP2),(VAR(8),YCP2),(VAR(9),PSI2),
     3            (VAR(10),PSI2D),(VAR(11),U2),(VAR(12),V2)
C
      EQUIVALENCE (DER(1),DXCP1),(DER(2),DYCP1),(DER(3),DPSI1),
     1            (DER(4),DPSI1D),(DER(5),DU1),(DER(6),DV1),
     2            (DER(7),DXCP2),(DER(8),DYCP2),(DER(9),DPSI2),
     3            (DER(10),DPSI2D),(DER(11),DU2),(DER(12),DV2)
C
      COMMON/INPT/  T0,TF,DTTRA0,DTCOL0,DTCLT0,DTPRN0,UVMIN,PSIDMN,FVEH0
     1             ,XCP10,YCP10,PSI10,PSI1D0,U10,V10,
     2              XCP20,YCP20,PSI20,PSI2D0,U20,V20,
     3              A1,B1,TR1,FIZ1,FM1,PSIR10,XF1,XR1,YS1,
     4              A2,B2,TR2,FIZ2,FM2,PSIR20,XF2,XR2,YS2,
     5              CSTFI(8),TBTQ1,TETQ1,TINCQ1,NTBLQ1,NTQ1,
     6              TBTQ2,TETQ2,TINCQ2,NTBLQ2,NTQ2,TII(8,201),
     7              TBPSF1,TEPSF1,TINCP1,NTBLP1,NPSF1,
     8              TBPSF2,TEPSF2,TINCP2,NTBLP2,NPSF2,PSIFI(4,201),
     9              XBP(2),YBP(2),XMU1,XMU2,CMU
C
      COMMON/INPT/  DELPS0,DELRO0,ALAMB,ZETAV,AKV(2) ,AMU,C0,C1,C2,
     1              PSLM10,PSLM20,PSLM30,PSLM40,PSLM50,PSLM60,PSLM70,
     2              PSLM80,HED(40),DADE(3),XINPUT(9)
C
      COMMON/OPTIONS/ IOPTION, DTINT, STEPPSI, STEPDT, SINPSI, SINFREQ,
     1                GAIN, DISLKAHD, XYTB(2,201), NOUT1, NOUT2
C
      COMMON/INTGR/JCC(4),JCOR(4),NPSIB,IPSIB0,IPSIB,IBB,IFF,NDTAB,
     1             I9,J9(361,2),  NPSJB,IPSJB,I3,ILAST,IL,IND1,IND2,INDI
     2             ,INDJ,ISTOP,ITZER1,ITZER2,IPZER1,IPZER2,ICOUNT,INDXB,
     3             ICOLLP,ICOLL,IQ,IND,IVEH,LL,ICTVDI,ITCOL,ITFLG1,
     4             ITFLG2,NWRDB1,NWRDE1,NWRDB2,NWRDE2,ICOLAC
C
C     IN COMMON /COMP/ DO NOT DISTURB THE ORDER OF THE VARIABLES
C     WHICH ARE EQUIVALENCED TO ARRAYS.
C
      COMMON/COMP/  DTTRAJ,DTCOLL,DTCOLT,DTPRNT,UVMN2,PSIDMA,TPRINT,DTP,
     1              U1V1SQ,U2V2SQ,DELPSI,DELPST,DELPS2,DELRHO,
     2              PSILM1,PSILM2,PSILM3,PSILM4,PSILM5,PSILM6,PSILM7,
     3              PSILM8,EJJ(4,2),GJJ(4,2),C1OC2,PSCC,PSIR1,PSIR2,
     4             XCMXC1,XCMXC2,YCMYC1,YCMYC2,SPSI1,SPSI2,CPSI1,CPSI2,
     5             XCPI,YCPI,PSII,XCPJ,YCPJ,PSIJ,XCMXC,YCMYC,SPSIJ,
     6             CPSIJ,XFI,XRI,YSI,XFJ,XRJ,YSJ,AKVI,AKVJ,EJ,GJ,
     7             XPJ(4),YPJ(4),ANGYX,SPAN,PSIBPI(4),PSIBPB,PSIBPF,
     8             FNPSIB,PSIBB,PSIB,PSFMPS,SCPSIB,CSPSIB,
     9             RHOBI,RHOBIN(4),PSIBIJ,SPSBIJ,CPSBIJ,XBIJ,YBIJ,PSJTB
C
      COMMON/COMP/ PSIBPJ,FNPSJB,PSIBJ,RHOBJ,XYJSQR,SPSIB,CPSIB,XBI,YBI,
     1         PRESI,PRESJ,XYSR,DELTA,CRHO,RHOBIC,RHOBMX,RHOBIT,RHOBT,
     2             SPSI21,CPSI21,X2TEM,Y2TEM,XX,YY,PAV,XXYYSR,FN1,SPSF,
     3             CPSF,X1AV,Y1AV,XYAVSR,X2TERM,Y2TERM,SPS21F,CPS21F,
     4             X2AV,Y2AV,VT1AV,VT2AV,VTMVT,FRICT,FNX1,FNY1,FNN1,
```

```
     5             FNX2,FNY2,FNN2,SFX1,SFY1,SFN1,SFX2,SFY2,SFN2,
     6              SFX1C,SFX2C,SFY1C,SFY2C,SFN1C,SFN2C,
     7              SFX1T,SFX2T,SFY1T,SFY2T,SFN1T,SFN2T, TCOL,TEND,
     8              TREST1,TREST2,TCOLS1,TCOLS2,P1TEM,P2TEM,U1V1E,U2V2E,
     9              EXTRA(10)
C
      COMMON/COMPT/ W(8),XWP(8),YWP(8),FLAGW(8),PSIW(8),TQW(8),
     1             TFX,TFY,TMOM,PSIRR,COEF,COEF1,COEF2,TRD2,TRD21,TRD22,
     2             XBMXB,FNUM,A,B,FIZ,FMASS,XC,YC,U,V,PSIC,PSICD,UD2,
     3             SGNU,SGNI,AOMB,CSPSIC,SNPSIC,TRCPSC,TRSPSC,TRPSCD,
     4             XI,XW,YW,SX,SY,S,XPP,YPP,RHO,RHOP,XMUS,XMU,PSI,XNUM,
     5             XDEN,RATIO,ALFA,F,TIF,FC,FCOS,SMAL,FSMX,FS,BETB,FX,FY
     6             ,CSPSI,SNPSI,XCXC,YCYC,SPJ,CPJ,SPIJ,CPIJ,XFJJ,XRJJ,
     7              YSJJ,EJJI,GJJI,XICORJ,YICORJ,NTQ,TBTQ,TETQ,TINCRQ,
     8             NPSF,TBPSF,TEPSF,TINCRPC
      COMMON/TAB/  TRHOB(361,2),TROBMX(361,2),TPSIB(361,2),
     1             TCPSIB(361,2),TSPSIB(361,2),TXB1(100),TYB1(100),
     2             TPSB1(100),TPRES1(100)
C
      DIMENSION   XCMXCI(2),YCMYCI(2),SPSII(2),CPSII(2),PSLM0(8),PSLM(8)
      EQUIVALENCE (XCMXCI(1),XCMXC1),(YCMYCI(1),YCMYC1),(SPSII(1),SPSI1)
     1           ,(CPSII(1),CPSI1),(PSLM0(1),PSLM10),(PSLM(1),PSILM1)
C
C     COMMON VDICON HAS CONSTANTS USED FOR VEHICLE DAMAGE INDEX
C
      COMMON/VDICON/EXTF1( 9),EXTR1( 9),EXTB1( 9),PSP1(12),
     2              EXTF2( 9),EXTR2( 9),EXTB2( 9),PSP2(12),NCOL51(10),
     2              NCOL52(10),MMM( 9)
C
      COMMON/OUTPT/TMOU(55),XCP1OU(55),YCP1OU(55),PSI1OU(55),U1OU(55),
     1            V1OU(55),PS1DOU(55),AX1OU(55),AY1OU(55),XCP2OU(55),
     2             YCP2OU(55),PSI2OU(55),U2OU(55),V2OU(55),PS2DOU(55),
     3          AX2OU(55),AY2OU(55),ALF1OU(55),ALF2OU(55),XWP1OU(55),
     4             XWP2OU( 55),XWP3OU( 55),XWP4OU( 55), XWP5OU( 55),
     5             XWP6OU( 55),XWP7OU( 55),XWP8OU( 55), YWP1OU( 55),
     6             YWP2OU( 55),YWP3OU( 55),YWP4OU( 55), YWP5OU( 55),
     7             YWP6OU( 55),YWP7OU( 55),YWP8OU( 55), FLAG1( 55),
     8             FLAG2( 55), FLAG3( 55),FLAG4( 55),FLAG5( 55),
     9         FLAG6( 55),FLAG7( 55),FLAG8( 55),ACC1OU(55),ACC2OU(55)
C
      COMMON/OUTPT /ARRXY1(10),ARRX1(10),ARRY1(10),ARRT1(10),ARRPS1(10),
     1   ARRDV1(10),ARRXY2(10),ARRX2(10),ARRY2(10),ARRT2(10),ARRPS2(10),
     2   ARRDV2(10),ABIG1,AXBIG1,AYBIG1,TBIG1,DELVS1,ACC1,ACCP1,AXX1,
     3  AYY1,ACTST1,ABIG2,AXBIG2,AYBIG2,TBIG2,DELVS2,ACC2,ACCP2,AXX2,
     4  AYY2,ACTST2,TM,TIMEP,NARRP1(10),NARRP2(10),NUM1,NUM2,
     5         ISAV1,ISAV2,IACDV1,IACDV2,IPAGE
C
      DIMENSION PSIW1RF(55), PSIW2RF(55)
      COMMON /MOVIE/ MOVIE,NMSG
C
      DATA NO1/1/,NO2/2/
      DATA NENDT/4HT = /,NSEC/4H SEC/,NPOR/4HPOR /,NBLNK/4H    /
C
      LL  = LL + 1
      TMOU(LL) = T
      XCP1OU(LL) = XCP1/12.
      YCP1OU(LL) = YCP1/12.
      PSI1OU(LL) = PSI1/RAD
      U1OU(LL)   = U1/12.
      V1OU(LL)   = V1/12.
      PS1DOU(LL) = PSI1D/RAD
      AX1OU(LL)  = AXX1
```

```
      AY1OU(LL)  = AYY1
      ACC1OU(LL) = ACC1
      ALF1OU(LL) = ANGLE(U1,V1)/RAD
      PSIW1RF(LL) = PSIW(1)/RAD
      XWP1OU(LL) = XWP(1)/ 12.
      XWP2OU(LL) = XWP(2)/ 12.
      XWP3OU(LL) = XWP(3)/ 12.
      XWP4OU(LL) = XWP(4)/ 12.
      YWP1OU(LL) = YWP(1)/ 12.
      YWP2OU(LL) = YWP(2)/ 12.
      YWP3OU(LL) = YWP(3)/ 12.
      YWP4OU(LL) = YWP(4)/ 12.
      FLAG1(LL)  = FLAGW(1)
      FLAG2(LL)  = FLAGW(2)
      FLAG3(LL)  = FLAGW(3)
      FLAG4(LL)  = FLAGW(4)
C
C     RF = 1,  LF = 2,  RR = 3,  LR = 4
C
      IF(IVEH.EQ.0) GO TO 9
      XCP2OU(LL) = XCP2/12.
      YCP2OU(LL) = YCP2/12.
      PSI2OU(LL) = PSI2/RAD
      U2OU(LL)   = U2/12.
      V2OU(LL)   = V2/12.
      PS2DOU(LL) = PSI2D/RAD
      AX2OU(LL)  = AXX2
      AY2OU(LL)  = AYY2
      ACC2OU(LL) = ACC2
      ALF2OU(LL) = ANGLE(U2,V2)/RAD
      PSIW2RF(LL) = PSIW(5)/RAD
      XWP5OU(LL) = XWP(5)/12.
      XWP6OU(LL) = XWP(6)/12.
      XWP7OU(LL) = XWP(7)/12.
      XWP8OU(LL) = XWP(8)/12.
      YWP5OU(LL) = YWP(5)/12.
      YWP6OU(LL) = YWP(6)/12.
      YWP7OU(LL) = YWP(7)/12.
      YWP8OU(LL) = YWP(8)/12.
      FLAG5(LL)  = FLAGW(5)
      FLAG6(LL)  = FLAGW(6)
      FLAG7(LL)  = FLAGW(7)
      FLAG8(LL)  = FLAGW(8)
C
    9 IF(LL .LT. 51 .AND. IPRT.LT.2) GO TO 16
C
C     PRINT EITHER AT FULL PAGE OR AT A PROGRAMMED STOP
C
      IPAGE = 1
   10 WRITE(NOUT,1010) IPAGE,HED,DADE
 1010 FORMAT(1H1,124X,4HPAGE,I3/26X,20A4/26X,20A4/58X,3A4)
      IF (IPAGE.LT.1 .OR.IPAGE .GT. 5) GO TO 50
      GO TO (11,12,13,14,15) , IPAGE
C
   11 WRITE(NOUT,1011) (NO1,IJ=1,8),
     1             (TMOU(I),XCP1OU(I),YCP1OU(I),PSI1OU(I), U1OU(I),
     2            V1OU(I),PS1DOU(I),AX1OU(I),AY1OU(I),ACC1OU(I),I=1,LL)
 1011 FORMAT(58X,11HVEHICLE NO.,I2 / 20X,13HC.G. POSITION,4X,
     1 13HHEADING ANGLE, 11X,10HVELOCITIES,11X,16HANGULAR VELOCITY, 9X,
     2 12HACCELERATION / 4X,4HTIME,10X,2HXC,I1,1H',8X, 2HYC,I1,1H',7X,
     3 3HPSI,I1,14X, 3HFWD, 9X,3HLAT,13X,3HPSI,I1,3HDOT,11X,2HAX,I1,
     49X,2HAY,I1,7X,3HACC,I1/4X,4HSEC.,11X,2HFT,10X,2HFT, 8X,3HDEG,13X,
```

```
     5 6HFT/SEC,7X, 6HFT/SEC,11X, 7HDEG/SEC, 12X,1HG,11X,1HG,10X,1HG /
     6 (F8.4,2X,2F12.2,F12.2,5X,2F12.2,5X,F12.2,5X,2F12.2,F11.2))
C
C       ADD ADDITIONAL OUTPUT FOR VEH1.DAT
C
      WRITE(NOUT1,9011)
     1              (TMOU(I),
     2               YCP1OU(I)*0.3048,PSI1OU(I),PSIW1RF(I),I=1,LL)
 9011 FORMAT(
     6 F8.4,2X,F12.2,2X,F12.2,2X,F12.2)
C
      IPAGE = IPAGE + 1
      GO TO 10
C
   12 WRITE(NOUT,1030) (NO1,IJ=1,3)
 1030 FORMAT('0',58X,'VEHICLE NO',I2,'    (*ASTERISK INDICATES SKIDDING
     1TIRE)'/5X,'TIME',7X,'VELOCITY VECTOR',34X,'TIRE TRACKS(FT)'/
     26X,'SEC',7X,'ATAN(V',I1,'/U',I2,') DEG',8X,'RF',20X,'LF',20X,'RR',
     320X,'LR'/38X,3HX1',8X,3HY1',8X,3HX2',8X,3HY2',8X,3HX3',
     48X,3HY3',8X,3HX4',8X,3HY4')
      WRITE(NOUT,1034)(TMOU(I),ALF1OU(I),XWP1OU(I),FLAG1(I),
     1             YWP1OU(I),FLAG1(I),XWP2OU(I),FLAG2(I),YWP2OU(I),
     2             FLAG2(I),XWP3OU(I),FLAG3(I),YWP3OU (I),FLAG3(I),
     3             XWP4OU(I),FLAG4(I),YWP4OU(I),FLAG4(I),I=1,LL)
 1034 FORMAT(1X,F9.3,4X,F12.3,8X,F8.1,A2,1X,F8.1,A2,1X,F8.1,A2,1X,
     1       F8.1,A2,1X, F8.1,A2,1X, F8.1,A2,1X, F8.1,A2,1X,F8.1,A2)
      IPAGE = IPAGE + 1
      IF(IVEH.NE.0) GO TO 10
      IF(IPRT.GE.2) GO TO 45
      LL = 0
      RETURN
C
   13 WRITE(NOUT,1011) (NO2,IJ=1,8),
     1              (TMOU(I),XCP2OU(I),YCP2OU(I),PSI2OU(I), U2OU(I),
     2            V2OU(I),PS2DOU(I),AX2OU(I),AY2OU(I),ACC2OU(I),I=1,LL)
C
C       ADD ADDITIONAL OUTPUT FOR VEH2.DAT
C
      WRITE(NOUT2,9011)
     1              (TMOU(I),
     2               YCP2OU(I)*0.3048,PSI2OU(I),PSIW2RF(I),I=1,LL)
C
      IPAGE = IPAGE + 1
      GO TO 10
C
   14 WRITE(NOUT,1030) (NO2,IJ=1,3)
      WRITE(NOUT,1034)(TMOU(I),ALF2OU(I),XWP5OU(I),FLAG5(I),
     1             YWP5OU(I),FLAG5(I),XWP6OU(I),FLAG6(I),YWP6OU(I),
     2             FLAG6(I),XWP7OU(I),FLAG7(I),YWP7OU (I),FLAG7(I),
     3             XWP8OU(I),FLAG8(I),YWP8OU(I),FLAG8(I),I=1,LL)
      IPAGE = IPAGE + 1
   15 LL = 0
   16 IF( IPRT.GE.2 .OR. MOVIE.NE.0 ) GO TO 19
      RETURN
C
C
      ENTRY OUTDIS
      PSI1TM = PSI1/RAD
      PS1DTM = PSI1D/RAD
      PSI2TM = PSI2/RAD
      PS2DTM = PSI2D/RAD
      IF(IPAGE - 5) 141,140,50
```

```
  140 WRITE(6,1010) IPAGE,HED,DADE
      IPAGE = 0
  141 WRITE(6,1012) T, XCP1,YCP1,PSI1TM,PS1DTM,XCP2,YCP2,PSI2TM,PS2DTM
 1012 FORMAT(//,6H TIME=,F8.4,4H SEC, 5X,7H  XCP1=,E14.6,3H IN,
     1   7H, YCP1=,E14.6,3H IN,7H, PSI1=,E14.6,4H DEG,8H, PSI1D=,E14.6,
     2 4H DEG/                          23X,7H  XCP2=,E14.6,3H IN,
     3   7H, YCP2=,E14.6,3H IN,7H, PSI2=,E14.6,4H DEG,8H, PSI2D=,E14.6,
     4  4H DEG )
C
      WRITE(6,1021) DXCP1,DYCP1,DU1,DV1,DXCP2,DYCP2,DU2,DV2
 1021 FORMAT(23H VALUES AFTER RK STEP 4,
     1            7H DXCP1=,E14.6,3X,7H DYCP1=,E14.6,3X,5H DU1=,E14.6,
     2 3X,5H DV1=,E14.6  /
     3       23X, 7H DXCP2=,E14.6,3X,7H DYCP2=,E14.6,3X,5H DU2=,E14.6,
     4 3X,5H DV2=,E14.6  )
      WRITE(6,1019)
 1019 FORMAT(120H USING ENTRY OUTDIS. FIRST TWO COLUMNS OF VDI ARE SET
     1 FROM INCOMPLETE INFORMATION(NARRP  ZER0 AND ARRPS  NOT ORDERED) )
C
   19 CALL OUT2(IPRT)
      IF(IPRT .LT. 2) RETURN
C
C        WHEN MOVIE.NE.0, OUT2 CALLS RNGDAM TO PREPARE DAMAGE TABLES
C        FOR WRITING.   VDI IS COMPUTED ONLY AT END OF RUN.
C
   45 TEND = T - TCOL
  453 IF(ITFLG1) 454,455,454
  454 NWRDB1 = NPOR
      NWRDE1 = NBLNK
      WRITE(6,1036) NO1,NENDT,TEND,NSEC,NWRDB1
 1036 FORMAT(13H0 VEHICLE NO.,I2, 10H, ELAPSED ,A4,F8.4,A4,5X,A4)
      GO TO 456
  455 NWRDB1 = NENDT
      NWRDE1 = NSEC
      WRITE(6,1036) NO1,NWRDB1,TEND,NWRDE1
  456 IF(IVEH) 460,465,460
  460 IF(ITFLG2) 461,462,461
  461 NWRDB2 = NPOR
      NWRDE2 = NBLNK
      WRITE(6,1036) NO2,NENDT,TEND,NSEC,NWRDB2
      GO TO 465
  462 NWRDB2 = NENDT
      NWRDE2 = NSEC
      WRITE(6,1036) NO2,NWRDB2,TEND,NWRDE2
  465 RETURN
   50 ISTOP = 50
      WRITE(6,1020)T, ISTOP
 1020 FORMAT(  3H0AT,F8.3,12HSEC., ISTOP=, I4)
      RETURN
      END
      FUNCTION ANGLE(X,Y)
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
      IF(X) 60,12,20
   12 IF(Y) 14,16,18
   14 ANGLE = PI32
      RETURN
   16 WRITE(NOUT,17)
   17 FORMAT(43H0 X=Y=0.0, THUS ANGLE IS UNDEFINED. SET=0.0)
      ANGLE = 0.0
      RETURN
   18 ANGLE = PIO2
```

```
      RETURN
   20 ANGLE = ATAN(Y/X)
      IF(Y) 26,80,80
   26 ANGLE = ANGLE + TWOPI
      RETURN
   60 IF(Y) 66,62,66
   62 ANGLE = PI
      RETURN
   66 ANGLE = ATAN(Y/X)   + PI
   80 RETURN
      END
C
      SUBROUTINE OUT2(IPRTX)
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
      COMMON/INTG/NEQ,T,DT,VAR(12),DER(12)
      EQUIVALENCE (VAR(1),XCP1),(VAR(2),YCP1),(VAR(3),PSI1),
     1            (VAR(4),PSI1D),(VAR(5),U1),(VAR(6),V1),
     2            (VAR(7),XCP2),(VAR(8),YCP2),(VAR(9),PSI2),
     3            (VAR(10),PSI2D),(VAR(11),U2),(VAR(12),V2)
      EQUIVALENCE (DER(1),DXCP1),(DER(2),DYCP1),(DER(3),DPSI1),
     1            (DER(4),DPSI1D),(DER(5),DU1),(DER(6),DV1),
     2            (DER(7),DXCP2),(DER(8),DYCP2),(DER(9),DPSI2),
     3            (DER(10),DPSI2D),(DER(11),DU2),(DER(12),DV2)
      COMMON/INPT/  T0,TF,DTTRA0,DTCOL0,DTCLT0,DTPRN0,UVMIN,PSIDMN,FVEH0
     1             ,XCP10,YCP10,PSI10,PSI1D0,U10,V10,
     2              XCP20,YCP20,PSI20,PSI2D0,U20,V20,
     3              A1,B1,TR1,FIZ1,FM1,PSIR10,XF1,XR1,YS1,
     4              A2,B2,TR2,FIZ2,FM2,PSIR20,XF2,XR2,YS2,
     5              CSTFI(8),TBTQ1,TETQ1,TINCQ1,NTBLQ1,NTQ1,
     6              TBTQ2,TETQ2,TINCQ2,NTBLQ2,NTQ2,TII(8,201),
     7              TBPSF1,TEPSF1,TINCP1,NTBLP1,NPSF1,
     8              TBPSF2,TEPSF2,TINCP2,NTBLP2,NPSF2,PSIFI(4,201),
     9              XBP(2),YBP(2),XMU1,XMU2,CMU
      COMMON/INPT/  DELPS0,DELRO0,ALAMB,ZETAV,AKV(2) ,AMU,C0,C1,C2,
     1              PSLM10,PSLM20,PSLM30,PSLM40,PSLM50,PSLM60,PSLM70,
     2              PSLM80,HED(40),DADE(3),XINPUT(9)
      COMMON/INTGR/JCC(4),JCOR(4),NPSIB,IPSIB0,IPSIB,IBB,IFF,NDTAB,
     1            I9,J9(361,2),  NPSJB,IPSJB,I3,ILAST,IL,IND1,IND2,INDI
     2            ,INDJ,ISTOP,ITZER1,ITZER2,IPZER1,IPZER2,ICOUNT,INDXB,
     3            ICOLLP,ICOLL,IQ,IND,IVEH,LL,ICTVDI,ITCOL,ITFLG1,
     4            ITFLG2,NWRDB1,NWRDE1,NWRDB2,NWRDE2,ICOLAC
C            IN COMMON /COMP/ DO NOT DISTURB THE ORDER OF THE VARIABLES
C                             WHICH ARE EQUIVALENCED TO ARRAYS.
      COMMON/COMP/  DTTRAJ,DTCOLL,DTCOLT,DTPRNT,UVMN2,PSIDMA,TPRINT,DTP,
     1              U1V1SQ,U2V2SQ,DELPSI,DELPST,DELPS2,DELRHO,
     2              PSILM1,PSILM2,PSILM3,PSILM4,PSILM5,PSILM6,PSILM7,
     3              PSILM8,EJJ(4,2),GJJ(4,2),C1OC2,PSCC,PSIR1,PSIR2,
     4             XCMXC1,XCMXC2,YCMYC1,YCMYC2,SPSI1,SPSI2,CPSI1,CPSI2,
     5             XCPI,YCPI,PSII,XCPJ,YCPJ,PSIJ,XCMXC,YCMYC,SPSIJ,
     6             CPSIJ,XFI,XRI,YSI,XFJ,XRJ,YSJ,AKVI,AKVJ,EJ,GJ,
     7             XPJ(4),YPJ(4),ANGYX,SPAN,PSIBPI(4),PSIBPB,PSIBPF,
     8             FNPSIB,PSIBB,PSIB,PSFMPS,SCPSIB,CSPSIB,
     9             RHOBI,RHOBIN(4),PSIBIJ,SPSBIJ,CPSBIJ,XBIJ,YBIJ,PSJTB
      COMMON/COMP/ PSIBPJ,FNPSJB,PSIBJ,RHOBJ,XYJSQR,SPSIB,CPSIB,XBI,YBI,
     1         PRESI,PRESJ,XYSR,DELTA,CRHO,RHOBIC,RHOBMX,RHOBIT,RHOBT,
     2             SPSI21,CPSI21,X2TEM,Y2TEM,XX,YY,PAV,XXYYSR,FN1,SPSF,
     3             CPSF,X1AV,Y1AV,XYAVSR,X2TERM,Y2TERM,SPS21F,CPS21F,
     4             X2AV,Y2AV,VT1AV,VT2AV,VTMVT,FRICT,FNX1,FNY1,FNN1,
     5             FNX2,FNY2,FNN2,SFX1,SFY1,SFN1,SFX2,SFY2,SFN2,
     6              SFX1C,SFX2C,SFY1C,SFY2C,SFN1C,SFN2C,
     7              SFX1T,SFX2T,SFY1T,SFY2T,SFN1T,SFN2T, TCOL,TEND,
```

```
     8              TREST1,TREST2,TCOLS1,TCOLS2,P1TEM,P2TEM,U1V1E,U2V2E,
     9              EXTRA(10)
      DIMENSION   XCMXCI(2),YCMYCI(2),SPSII(2),CPSII(2),PSLM0(8),PSLM(8)
      EQUIVALENCE (XCMXCI(1),XCMXC1),(YCMYCI(1),YCMYC1),(SPSII(1),SPSI1)
     1           ,(CPSII(1),CPSI1),(PSLM0(1),PSLM10),(PSLM(1),PSILM1)
      COMMON/TAB/  TRHOB(361,2),TROBMX(361,2),TPSIB(361,2),
     1             TCPSIB(361,2),TSPSIB(361,2),TXB1(100),TYB1(100),
     2             TPSB1(100),TPRES1(100)
C         COMMON VDICON HAS CONSTANTS USED FOR VEHICLE DAMAGE INDEX
      COMMON/VDICON/EXTF1( 9),EXTR1( 9),EXTB1( 9),PSP1(12),
     2              EXTF2( 9),EXTR2( 9),EXTB2( 9),PSP2(12),NCOL51(10),
     2              NCOL52(10),MMM( 9)
      COMMON/OUTPT/TMOU(55),XCP1OU(55),YCP1OU(55),PSI1OU(55),U1OU(55),
     1            V1OU(55),PS1DOU(55),AX1OU(55),AY1OU(55),XCP2OU(55),
     2             YCP2OU(55),PSI2OU(55),U2OU(55),V2OU(55),PS2DOU(55),
     3          AX2OU(55),AY2OU(55),ALF1OU(55),ALF2OU(55),XWP1OU(55),
     4             XWP2OU( 55),XWP3OU( 55),XWP4OU( 55), XWP5OU( 55),
     5             XWP6OU( 55),XWP7OU( 55),XWP8OU( 55), YWP1OU( 55),
     6             YWP2OU( 55),YWP3OU( 55),YWP4OU( 55), YWP5OU( 55),
     7             YWP6OU( 55),YWP7OU( 55),YWP8OU( 55), FLAG1( 55),
     8             FLAG2( 55), FLAG3( 55),FLAG4( 55),FLAG5( 55),
     9         FLAG6( 55),FLAG7( 55),FLAG8( 55),ACC1OU(55),ACC2OU(55)
      COMMON/OUTPT /ARRXY1(10),ARRX1(10),ARRY1(10),ARRT1(10),ARRPS1(10),
     1   ARRDV1(10),ARRXY2(10),ARRX2(10),ARRY2(10),ARRT2(10),ARRPS2(10),
     2   ARRDV2(10),ABIG1,AXBIG1,AYBIG1,TBIG1,DELVS1,ACC1,ACCP1,AXX1,
     3  AYY1,ACTST1,ABIG2,AXBIG2,AYBIG2,TBIG2,DELVS2,ACC2,ACCP2,AXX2,
     4  AYY2,ACTST2,TM,TIMEP,NARRP1(10),NARRP2(10),NUM1,NUM2,
     5         ISAV1,ISAV2,IACDV1,IACDV2,IPAGE
      COMMON /MOVIE/ MOVIE,NMSG
      COMMON/OUTPT1/X1OU(361),X2OU(361),Y1OU(361),Y2OU(361),R1OU(361),
     1           R2OU(361),PSB1OU(361),PSB2OU(361),ASTR1(361),ASTR2(361)
     2              ,PSIM1(10),INPSM1(10),IKD11(10),IKD21(10),IRNG1,IK1,
     3        IWRAP1,PSIM2(10),INPSM2(10),IKD12(10),IKD22(10),IRNG2,IK2,
     4        IWRAP2,NCOL11(10),NCOL12(10),NCOL21(10),NCOL22(10),
     5              NCOL31(10),NCOL32(10),NCOL41(10),NCOL42(10),
     6              NCOL61(10),NCOL62(10),NCOL71(10),NCOL72(10)
     7              ,PSD11(10),PSD21(10),PSD12(10),PSD22(10),DVVDI1(10),
     8              DVVDI2(10)
      DIMENSION OUTP0(3876)
      EQUIVALENCE(OUTP0 (1),X1OU(1))
C
C
   17 IF(ICTVDI) 23,18,23
   18 DO 20 I=1,3876
   20 OUTP0(I) = 0.0
      N6 = NOUT
      IF(MOVIE.EQ.0) GO TO 23
C        PAGE HEADING FOR BEGINNING OF EXTRA DATA SET  FT02
      WRITE(NMSG,1010) IPAGE,HED,DADE
      ICTVDI = 1
C
   23 IJ = 1
      IF(IPRTX.LT.2) GO TO 232
C
C        IPRTX EQ 2 SIGNIFIES END OF RUN. PASSED FROM  SUBROUTINE OUTPUT
C        CALL DAMAGE ONLY AT END OF RUN
C        SAVMAX DOES SOME SPECIAL PROCESSING AT END OF RUN ONLY.
C       IPRTX EQ 1 AT THIS POINT WHEN MOVIE NE 0, OR OUTDIS CALLED.
C        MOVIE NE 0 PUTS DAMAGE TABLES ON TAPE AT EACH TIME POINT, AND
C        WRITES ON EXTRA DATA SET FT02 (NMSG=2,THEN NMSG=6 WHEN IPRTX=2)
C
C
```

```
C           TEMPORARY MESSAGE
      WRITE(NOUT,620)
  620 FORMAT(80H0 MESSAGE FROM SUBROUTINE OUT2,REFERRING TO VALUES OBTAI
     XNED IN SUBROUTINE SAVMAX)
      IF(NUM1) 230,230,231
  230 WRITE(NOUT,623) IJ,NUM1
  623 FORMAT(9H  VEH.NO., I2, 35H,  NUMBER OF MAX ACCEL .GE. 1.0 G =,I3,
     1 37H, DO NOT COMPUTE VEHICLE DAMAGE INDEX)
C             NUM1 =0 IMPLIES RIGID BARRIER, WITH NO ACCEL .GE.1.0 G
C            AVOID VDI COMPUTATION FOR CASES WHERE THERE IS NO
C           ACCELERATION .GE. 1.0 G                 SIMILARLY FOR VEH 2
      GO TO 232
  231 WRITE(NOUT ,621)(ARRT1(I),ARRXY1(I),ARRX1(I),ARRY1(I),ARRPS1(I),
     1     NARRP1(I),ARRDV1(I), I=1,NUM1)
  621 FORMAT(120H0    ARRT1             ARRXY1               ARRX1
     1         ARRY1         ARRPS1(RAD)         NARRP1     ARRDV1(MPH)/
     2 (F9.4,4E20.6, I8,E20.6))
C
  232 CALL            RNGDAM(TRHOB(1,1),TPSIB(1,1),TCPSIB(1,1),
     1     TSPSIB(1,1),J9(1,1),R1OU,PSB1OU,X1OU,Y1OU,ASTR1,PSP1,PSD11,
     2     PSD21,PSIM1,INPSM1,IKD11,IKD21,NARRP1,NUM1,IK1,IRNG1,IWRAP1,
     3                 IJ,ISTOP,NDTAB,DELPSI,XF1,XR1,YS1,NMSG,IPRTX)
C            RADIAN VALUES FOR PSB1OU,PSIM1,PSD11,PSD21,PSB2OU,PSIM2,
C                             PSD12,PSD22 RETURNED FROM SUBROUTINE RNGDAM
C           IK1=NO.OF ENTRIES IN R1OU,PSB1OU,X1OU,Y1OU,ASTR1
C           NUM1=NO.OF ENTRIES IN NARRP1
C           IRNG1=NO.OF ENTRIES IN PSIM1,INPSM1,IKD11,IKD21,PSD11,PSD21
      IF(IPRTX.LT.2) GO TO 240
      IF(NUM1) 239,240,239
C             SUBROUTINE DAMAGE WILL SET STANDARD CODE
  239 CALL DAMAGE(PSIM1,IRNG1,PSP1,PSB1OU,X1OU,Y1OU,ASTR1,
     1            IKD11,IKD21,IK1,EXTF1,EXTR1,EXTB1,XF1,XR1,YS1,
     2 NCOL31,NCOL41,NCOL61,NCOL71,MMM,ISTOP,PSD11,PSD21,ARRPS1,INPSM1,
     3  IJ)
C
  240 IJ = 2
      IF(IPRTX.LT.2) GO TO 243
C
C           TEMPORARY MESSAGE
      WRITE(NOUT,620)
      IF(NUM2) 241,241,242
  241 WRITE(NOUT,623) IJ,NUM2
      GO TO 243
  242 WRITE(NOUT ,622)(ARRT2(I),ARRXY2(I),ARRX2(I),ARRY2(I),ARRPS2(I),
     1     NARRP2(I),ARRDV2(I), I=1,NUM2)
  622 FORMAT(120H0    ARRT2             ARRXY2               ARRX2
     1         ARRY2         ARRPS2(RAD)         NARRP2     ARRDV2(MPH)/
     2 (F9.4,4E20.6, I8,E20.6))
C
  243 CALL            RNGDAM(TRHOB(1,2),TPSIB(1,2),TCPSIB(1,2),
     1     TSPSIB(1,2),J9(1,2),R2OU,PSB2OU,X2OU,Y2OU,ASTR2,PSP2,PSD12,
     2     PSD22,PSIM2,INPSM2,IKD12,IKD22,NARRP2,NUM2,IK2,IRNG2,IWRAP2,
     3                 IJ,ISTOP,NDTAB,DELPSI,XF2,XR2,YS2,NMSG,IPRTX)
      IF(IPRTX.LT.2) GO TO 248
      IF(NUM2) 247,248,247
  247 CALL DAMAGE(PSIM2,IRNG2,PSP2,PSB2OU,X2OU,Y2OU,ASTR2,
     1            IKD12,IKD22,IK2,EXTF2,EXTR2,EXTB2,XF2,XR2,YS2,
     2 NCOL32,NCOL42,NCOL62,NCOL72,MMM,ISTOP,PSD12,PSD22,ARRPS2,INPSM2,
     3  IJ)
C
  248 IF( IPAGE - 5) 27,26,50
   26 WRITE(NMSG,1010) IPAGE,HED,DADE
```

```
 1010 FORMAT(1H1,124X,4HPAGE,I3/26X,20A4/26X,20A4/58X,3A4)
C        IPAGE EQ 5 HERE AT END OF RUN, OR AFTER FULL PAGES PRINTED,
C        SO PRINT HEADING.
C        IPAGE IS LESS THAN 5 AT THIS POINT WHEN ENTRY OUTDIS IS USED.
C        IPAGE ALSO .LT.5 HERE WHEN MOVIE NE 0
   27 IF(IPRTX.LT.2)  WRITE(NMSG,1011) T
 1011 FORMAT(12H0    TIME = , F8.4)
      WRITE(NMSG,1013) IK1,IK2
 1013 FORMAT( 51X,14HDAMAGE SUMMARY, 2X, 29H(DISPLACED POINTS, * J POINT
     1) /26X,13HVEHICLE NO. 1,52X,13HVEHICLE NO. 2/26X,I5,7H POINTS,53X,
     1 I5,7H POINTS/  10X,5HRHOB1,7X,5HPSIB1,
     2  9X,2HX1,10X,2HY1,25X, 5HRHOB2,7X,5HPSIB2,9X,2HX2,10X,2HY2 /
     3 10X,6HINCHES,7X,3HDEG, 8X,6HINCHES,6X,6HINCHES, 23X, 6HINCHES,
     4  7X,3HDEG, 8X,6HINCHES, 6X,6HINCHES )
      IF(IK1 - IK2)  28,30,32
   28 LK  = IK1
      IKK = LK + 1
      LKK = IK2
      LCO = LKK
      GO TO 34
   30 LK = IK1
      IKK = 0
      LKK = 0
      LCO = LK
      GO TO 34
   32 LK = IK2
      IKK = LK + 1
      LKK = IK1
      LCO = LKK
   34 DO 340 I =1,IK1
  340 PSB1OU(I) = PSB1OU(I) /RAD
      DO 341 I=1,IK2
  341 PSB2OU(I) = PSB2OU(I) /RAD
      IF (LK) 35,350,35
   35 WRITE(NMSG,1014)(ASTR1(IJ),R1OU(IJ),PSB1OU(IJ),X1OU(IJ),Y1OU(IJ),
     X                ASTR2(IJ),R2OU(IJ),PSB2OU(IJ),X2OU(IJ),Y2OU(IJ),
     X                IJ=1,LK)
 1014 FORMAT(2X,A1,1X,2F12.3,2F12.4, 15X,A1,1X,2F12.3,2F12.4)
  350 IF(LKK) 36,40,36
   36 IF(LKK-IK1)39,37,39
   37 WRITE(NMSG,1016)( ASTR1(IJ),R1OU(IJ),PSB1OU(IJ),X1OU(IJ),Y1OU(IJ),
     X                IJ=IKK,LKK)
 1016 FORMAT(2X,A1,1X,2F12.3,2F12.4)
      GO TO 40
   39 WRITE(NMSG,1018)(ASTR2(IJ),R2OU(IJ),PSB2OU(IJ),X2OU(IJ),Y2OU(IJ),
     X                IJ=IKK,LKK)
 1018 FORMAT(67X,A1,1X,2F12.3,2F12.4)
C
   40 IF(IRNG1 - IRNG2) 41,42,43
   41 LR = IRNG1
      IRR = LR + 1
      LRR = IRNG2
      LCO = LCO + LRR
      GO TO 44
   42 LR = IRNG1
      IRR  = 0
      LRR  = 0
      LCO = LCO + LR
      GO TO 44
   43 LR = IRNG2
      IRR = LR + 1
      LRR = IRNG1
```

```
      LCO = LCO + LRR
C        LCO COUNTS POSSIBLE LINES FOR DAMAGE ARRAYS
C            COMPARE TO 45, ALLOWING 15 LINES FOR HEADINGS.
   44 IF(LCO - 45) 442,440,440
C             IPAGE IS LESS THAN 5 IF THE ENTRY OUTDIS WAS USED,
C                 OR MOVIE NE  0
  440 IF(IPAGE-5) 442,441,441
  441 IPAGE = 6
      WRITE(NMSG,1010) IPAGE,HED,DADE
      IPAGE = 0
  442 IF(IPRTX-2) 4421,4422,4422
 4421 WRITE(NMSG,1030) IRNG1,IRNG2
 1030 FORMAT(5X,6HIRNG1=,I3,5X,6HIRNG2=,I3)
      RETURN
C              VDI NOT COMPUTED WHEN IPRTX LT 2
C
 4422 WRITE(NOUT,1022) IRNG1,IRNG2
 1022 FORMAT( 1H0,25X,13HVEHICLE NO. 1,52X,13HVEHICLE NO. 2/28X,I3,7H RA
     1NGES,55X,I3,7H RANGES / 8X,5HBEGIN,13X,3HEND, 7X,8HMIDPOINT,2X,
     210HVEH.DAMAGE,3X,3HVEL,12X,5HBEGIN,13X,3HEND, 7X,8HMIDPOINT,2X,
     310HVEH.DAMAGE,3X,3HVEL    / 4X,5HRHOB1,3X,5HPSIB1,4X,5HRHOB1,3X,
     45HPSIB1,5X,5HPSIM1,4X,5HINDEX,3X,7HDELTA V,  7X,5HRHOB2,3X,5HPSIB2
     5,4X,5HRHOB2,3X,5HPSIB2,5X,5HPSIM2,4X,5HINDEX,3X,7HDELTA V   /
     6 4X,6HINCHES,3X,3HDEG,5X,6HINCHES,3X,3HDEG,7X,3HDEG,15X,3HMPH,
     7 9X,6HINCHES,3X,3HDEG,5X,6HINCHES,3X,3HDEG,7X,3HDEG,15X,3HMPH    )
C      CALLS TO ENTRY OUTDIS HAVE NOT COMPLETED RESULTS FOR FIRST TWO
C       COLUMNS OF VDI
      IF(NUM1) 62,61,62
   61 NCOL51(1) = 0
C            INTEGER SHOULD NOT BE PRINTABLE WITH FORMAT A
      DVVDI1(1) = ARRDV1(1)
C      ASSUME ONLY THE FIRST DELTA V FROM SUBROUTINE SAVMAX HAS VALUE.
C      ASSUME IRNG1 = 1 WHEN NUM1 = 0
C      NARRP AND INPSM NOT EVALUATED WHEN NUM=0 (SUBROUTINE RNGDAM)
      GO  TO 80
   62 DO 75 I=1,IRNG1
      JNCOL = INPSM1(I)
      CALL NCOLDV(NARRP1,ARRDV1,MMM,NUM1,JNCOL,ISTOPP,NCOL1,NCOL2,DVSUM)
      IF(ISTOPP) 67,69,67
   67 ISTOP = 70
      NCOL = NARRP1(JNCOL)
   68 WRITE(NOUT,680) ISTOP,NCOL
  680 FORMAT(8H0 ISTOP=,I3,5X,6H NCOL=,I3)
      GO TO 75
   69 NCOL11(I) = NCOL1
      NCOL21(I) = NCOL2
      DVVDI1(I) = DVSUM
   75 CONTINUE
   80 IF(NUM2) 82,81,82
   81 NCOL52(1) = 0
      DVVDI2(1) = ARRDV2(1)
      GO TO 96
   82 DO 95 I = 1,IRNG2
      JNCOL = INPSM2(I)
      CALL NCOLDV(NARRP2,ARRDV2,MMM,NUM2,JNCOL,ISTOPP,NCOL1,NCOL2,DVSUM)
      IF(ISTOPP) 87,89,87
   87 ISTOP = 71
      NCOL = NARRP2(JNCOL)
      WRITE(NOUT,680) ISTOP,NCOL
      GO TO 95
   89 NCOL12(I) = NCOL1
      NCOL22(I) = NCOL2
```

```
      DVVDI2(I) = DVSUM
   95 CONTINUE
C                IF NUM1 OR NUM2 =0, THE VDI FOR THAT VEHICLE IS NOT
C                   COMPUTED,AND SHOULD CONTAIN INTEGER 0, NOT PRINTABLE
C                  IRNG1 IS PROBABLY =1 WHEN NUM1=0.STORE ONLY DVVDI1(1)
C                  SIMILARLY FOR VEHICLE 2 IF NUM2=0.
   96 DO 97 I=1,IRNG1
   97 PSIM1(I) = PSIM1(I)/RAD
      DO 99 I=1,IRNG2
   99 PSIM2(I) = PSIM2(I)/RAD
      IF(LR) 443,445,443
  443 WRITE(N6,1024)(ASTR1(IKD11(IJ)),R1OU(IKD11(IJ)),PSB1OU(IKD11(IJ)),
     1               ASTR1(IKD21(IJ)),R1OU(IKD21(IJ)),PSB1OU(IKD21(IJ)),
     2               PSIM1(IJ),NCOL11(IJ),NCOL21(IJ),NCOL31(IJ),
     3           NCOL41(IJ),NCOL51(IJ),NCOL61(IJ),NCOL71(IJ),DVVDI1(IJ),
     4               ASTR2(IKD12(IJ)),R2OU(IKD12(IJ)),PSB2OU(IKD12(IJ)),
     5               ASTR2(IKD22(IJ)),R2OU(IKD22(IJ)),PSB2OU(IKD22(IJ)),
     6               PSIM2(IJ),NCOL12(IJ),NCOL22(IJ),NCOL32(IJ),
     7              NCOL42(IJ),NCOL52(IJ),NCOL62(IJ),NCOL72(IJ),
     8               DVVDI2(IJ), IJ=1,LR)
 1024 FORMAT(1X,A1,F7.2,F8.2,2X,A1,F7.2,F8.2,1X,F8.2,3X,7A1,2X,F7.2,
     1       4X,A1,F7.2,F8.2,2X,A1,F7.2,F8.2,1X,F8.2,3X,7A1,2X,F7.2)
  445 IF(LRR) 446,45 ,446
  446 IF(LRR-IRNG1) 448,447,448
  447 WRITE(N6,1026)(ASTR1(IKD11(IJ)),R1OU(IKD11(IJ)),PSB1OU(IKD11(IJ)),
     1               ASTR1(IKD21(IJ)),R1OU(IKD21(IJ)),PSB1OU(IKD21(IJ)),
     2               PSIM1(IJ),NCOL11(IJ),NCOL21(IJ),NCOL31(IJ),
     3              NCOL41(IJ),NCOL51(IJ),NCOL61(IJ),NCOL71(IJ),
     4               DVVDI1(IJ), IJ=IRR,LRR)
 1026 FORMAT(1X,A1,F7.2,F8.2,2X,A1,F7.2,F8.2,1X,F8.2,3X,7A1,2X,F7.2)
      GO TO 45
  448 WRITE(N6,1028)(ASTR2(IKD12(IJ)),R2OU(IKD12(IJ)),PSB2OU(IKD12(IJ)),
     1               ASTR2(IKD22(IJ)),R2OU(IKD22(IJ)),PSB2OU(IKD22(IJ)),
     2               PSIM2(IJ),NCOL12(IJ),NCOL22(IJ),NCOL32(IJ),
     3              NCOL42(IJ),NCOL52(IJ),NCOL62(IJ),NCOL72(IJ),
     4               DVVDI2(IJ), IJ=IRR,LRR)
 1028 FORMAT(67X,A1,F7.2,F8.2,2X,A1,F7.2,F8.2,1X,F8.2,3X,7A1,2X,F7.2)
   45 RETURN
   50 ISTOP = 51
      WRITE(NOUT,1020) T, ISTOP
 1020 FORMAT(  3H0AT,F8.3,12HSEC., ISTOP=, I4)
      RETURN
      END
```

```
      SUBROUTINE ACCEL (G,DU ,DV ,DPSI ,U ,V ,AXX ,AYY ,ACC )
      AXX = (DU -V*DPSI)/G
      AYY = (DV + U*DPSI)/G
      ACC = SQRT(AXX**2 + AYY**2)
      RETURN
      END
      SUBROUTINE SAVMAX (ARRXY ,ARRX ,ARRY ,ARRT ,ARRDV ,ARRPSI,NARRPS,
     1            TM,TIMEP,AXX ,AYY ,ACC ,ACCP ,DELVS ,ACTST ,
     2            ABIG ,AXBIG ,AYBIG ,TBIG ,NUM ,ISAVE ,IACDV ,
     3            IPRTX,ISTOP)
C       SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
C
C          SAVES THE MAXIMA OF ACCELERATION CORRESPONDING TO IMPACT.
C       SAVES THOSE MAXIMA WHICH ARE FOLLOWED BY RESULTANT ACCELERATION
C       ACC = SQRT(AX1OU**2 + AY1OU**2) LESS THAN 1.G AND THEN FOLLOWED
C       BY RESULTANT ACCELERATION GREATER THAN 1.G.
C
C       AT END OF RUN ARRANGE IN ORDER OF DECREASING MAXIMA,
C       AND COMPUTE ANGLES AND INDICES FOR DIRECTION OF FORCE AT
C       PRINCIPAL IMPACT.
C       ALSO CONVERT ARRDV TO MPH FROM G*SEC
C
C       CALLS SUBROUTINE DELTAV,WHICH COMPUTES DELVS,ACTST,IACDV
C
      DIMENSION ARRXY(1),ARRX(1),ARRY(1),ARRT(1),ARRPSI(1),NARRPS(1)
      DIMENSION ARRDV(1)
      DATA TWOPI/6.2831853072/, PIO6/0.5235987756/
C         SAMPLE CALL
C     CALL SAVMAX (ARRXY1,ARRX1,ARRY1,ARRT1,ARRDV1,ARRPS1,NARRP1,
C    1            TM,TIMEP,AXX1,AYY1,ACC1,ACCP1,DELVS1,ACTST1,
C    2            ABIG1,AXBIG1,AYBIG1,TBIG1,NUM1,ISAV1 ,IACDV1,
C    3            IPRTY,ISTOP)
C
C       USES AXX,AYY,ACC,TM,IPRTX. SUBROUTINE DELTAV ALSO USES ACCP,TIMEP
C       RETURNS ABIG,AXBIG,AYBIG,TBIG,NUM,ISAVE
C       AND ARRAYS ARRXY,ARRX,ARRY,ARRT,ARRDV,ARRPSI,NARRPS.
C       RETURNS DELVS,ACTST,IACDV WHICH ARE COMPUTED IN SUBROUTINE DELTAV
C        *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C       CALLING PROGRAM MUST ZERO ISAVE, NUM, DELVS
C       SET ABIG.LE.-1.E10 AND SET IPRTX.LE.1 BEFORE FIRST CALL TO SAVMAX
C       ALSO SET TM=T0 AND TIMEP=T0 BEFORE FIRST CALL TO SAVMAX
C        *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
      DATA CONVER/21.9546/
C       CONVER CONVERTS ACCELERATION*SEC IN G*SEC TO MPH
C     DELVS IS VELOCITY CHANGE OF PASSENGER COMPARTMENT DURING COLLISION
C       DELVS IS COMPUTED IN SUBROUTINE DELTAV AND IS STORED IN ARRAY
C       ARRDV IN SUBROUTINE SAVMAX TO CORRESPOND TO MAX ACCEL AND TO VDI
C       AT END OF RUN,ARRDV IS CONVERTED TO MPH (DELVS IS G*SEC)
C       IACDV =  1,2,3 FOR COMPARISON OF CONSISTENT DOMINANT COMPONENTS
C       IN ACCELERATION FOR ACCUMULATING DELVS IN SUBROUTINE DELTAV
C       IACDV=0 FOR ACC.LE.1.0,   IACDV=-10 FOR SUDDEN FLIP
C       ACTST HAS VALUE PLUS OR MINUS ONE, OR ZERO,USED TO COMPARE
C       DIRECTION OF ACCELERATION
C
C
      CALL DELTAV (TM,TIMEP,AXX,AYY,ACC,ACCP,DELVS,ACTST,IACDV,ABIG)
      IF(IACDV + 10)  9,26, 9
    9 IF(ISAVE) 15,10,15
   10 IF(ACC  - ABIG) 15,15,11
   11 IF(ACC - 1.0) 30,30,13
   13 ABIG = ACC
```

```
      AXBIG = AXX
      AYBIG = AYY
      TBIG = TM
      GO TO 30
   15 IF(ACC - 1.0) 18,18,25
C       ISAVE = 1 FOR FIRST OCCURRENCE OF RESULTANT .LE. 1.G AFTER MAX.
C       ISAVE = 0 INDICATES RESULTANT .LE. 1.G NOT YET DETECTED AFTER MAX
   18 IF(ISAVE) 30,19,30
   19 ISAVE = 1
      GO TO 30
   25 IF(ISAVE) 26,30,26
   26 NUM = NUM + 1
C
      IF(NUM - 10) 28,28,27
   27 WRITE(6,6127) TM,TBIG,NUM
 6127 FORMAT(10H0 AT TIME=,E14.6,23H EXAMINE MAX FOR TIME =,E14.6 /
     1 5X,12HBECAUSE NUM=,I3,16H RESET NUM TO 10)
      NUM = 10
      GO TO 37
C
   28 ARRXY(NUM) = ABIG
      ARRX(NUM) = AXBIG
      ARRY(NUM) = AYBIG
      ARRT(NUM) = TBIG
      ARRDV(NUM) = DELVS
   29 ABIG = ACC
      AXBIG = AXX
      AYBIG = AYY
      TBIG = TM
      DELVS = 0.0
      CALL DELTAV (TM,TIMEP,AXX,AYY,ACC,ACCP,DELVS,ACTST,IACDV,ABIG)
      ISAVE = 0
C
   30 IF(IPRTX -2) 31,33,33
   31 RETURN
C
   33 IF(NUM) 26,26,34
   34 IF(ISAVE) 26,35,26
   35 IF(NUM - 1) 48,48,37
   37 IB = 1
   38 ABIGA= -1.E10
      IBIG = 0
      DO 40 I=IB,NUM
      AA = ARRXY(I)
      IF(AA -ABIGA) 40,40,39
   39 ABIGA = AA
      IBIG = I
   40 CONTINUE
      IF(IB - IBIG) 42,45,41
   41 ISTOP = 5
      WRITE(6,6141) ISTOP
 6141 FORMAT( 9H0  ISTOP=,I3,21H IN SUBROUTINE SAVMAX)
      RETURN
   42 ASAV = ARRXY(IB)
      ARRXY(IB) = ABIGA
      ARRXY(IBIG) = ASAV
      XSAV = ARRX(IB)
      YSAV = ARRY(IB)
      TSAV = ARRT(IB)
      DLVSAV = ARRDV(IB)
      ARRX(IB) = ARRX(IBIG)
      ARRY(IB) = ARRY(IBIG)
```

```
      ARRT(IB) = ARRT(IBIG)
      ARRDV(IB) = ARRDV(IBIG)
      ARRX(IBIG) = XSAV
      ARRY(IBIG) = YSAV
      ARRT(IBIG) = TSAV
      ARRDV(IBIG) = DLVSAV
   45 IB = IB + 1
      IF(IB - NUM) 38,46,46
   46 IF(IPRTX - 2) 47,48,48
C
   47 IF(ARRXY(NUM) - ABIG) 470,29,29
  470 WRITE(6,6127) TM,TBIG,NUM
      WRITE(6,6471) ABIG,ARRXY(NUM)
 6471 FORMAT(16H0 NEW MAX, ABIG=,E14.6, 50H, IS GREATER THAN SMALLEST PR
     1EVIOUS MAX,ARRXY(10)=,E14.6/
     2 5X,78H REPLACE ARRXY(10) WITH ABIG AND REPLACE TENTH PLACE IN
     3 OTHER ARRAYS ALSO.)
      GO TO 28
C
   48 DO 481 I=1,NUM
  481 ARRDV(I) = CONVER*ARRDV(I)
C        ****
C       AT DO 50
C       AFTER ST 48 EXAMINE FOR LACK OF SIGNIFICANT ACCEL. IMPLIES
C      STATIONARY OBJECT. RETURN NUM=0.  IN SUBR OUT2,NUM=0 DELETES VDI
      DO 50 I=1,NUM
      IF(ARRXY(I) - 1.0) 50,50,52
   50 CONTINUE
      NUM = 0
      GO TO 71
C        ****
   52 DO 70 I=1,NUM
      IF(ARRY(I)) 63,61,64
   61 IF(ARRX(I)) 64,62,64
   62 ARRPSI(I) = 0.0
      NPSITM= 6
      GO TO 70
   63 PQUAD = TWOPI
      GO TO 65
   64 PQUAD = 0.0
   65 PSITEM = ATAN2(ARRY(I),ARRX(I))
      ARRPSI(I) = PSITEM + PQUAD
      NPSITM = ARRPSI(I)/PIO6 + 6.5
      IF(NPSITM - 12) 70,70,69
   69 NPSITM = NPSITM - 12
   70 NARRPS(I) = NPSITM
   71 RETURN
      END
      SUBROUTINE DELTAV (TMM,TMMP,AX,AY,AC,ACP,DVS,ACTS,IAC,ABIG)
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS - SMAC
C      CALLED EVERY INTERVAL AT BEGINNING OF SAVMAX, SOMETIMES CALLED
C      AGAIN IN THAT INTERVAL AFTER 'STORE' IN SAVMAX.
C      IAC = 3,2,1,0,-1,-2,-3,-7,-8,-9,-10
C      ABIG IS NEG ONLY AT BEGINNING UNTIL AC IS .GT.1.0 ONCE.
      IF(IAC) 3,7,7
    3 IF(IAC + 6) 7,7,4
    4 IF(AC - 1.0) 7,7,5
    5 IF(ABIG) 7,6,6
    6 IAC = IAC-6
      RETURN
C
    7 ACX = AC
```

```
      ACPX =ACP
      IACX = IAC
      ACTSX = ACTS
      IPLUS =0
   15 IF(IACX) 20,20,36
   20 IF(AX) 22,24,22
   22 ABSAC = ABS(AY/AX)
      IF(ABSAC - 1.23) 26,26,24
   24 IACT = 3
      ACTST = SIGN(1.0,AY)
      GO TO 31
   26 IF(ABSAC - 0.84) 30,28,28
   28 IACT = 1
      ACTST = SIGN(1.0,AX*AY)
      GO TO 31
   30 IACT = 2
      ACTST = SIGN(1.0,AX)
   31 IAC = IACT
      ACTS = ACTST
      IF(IACX + 10) 321,32,321
C      AC MUST BE .GT.1.0 HERE     FLIP (-10) WAS SET IN  CALL IN THIS I
   32 ACPX = 0.0
      GO TO 48
  321 IF(IACX + 6) 322,323,323
  322 IACX =  IACX + 6
      GO TO 324
  323 IF(IACX) 324,325,325
  324 IACX = - IACX
      IPLUS = 1
C
  325 IF(AC - 1.0) 326,326, 36
  326 IF(ACP - 1.0) 327,327, 36
  327 IF(IAC) 329,329,328
  328 IAC = -IAC
  329 RETURN
C
C
C      USE PREVIOUSLY SET INDICATORS
C      IACX = 1,2,3
   36 IF(IACX - 1) 40,38,40
   38 IF(ACTSX - SIGN(1.0,AX*AY)) 46,48,46
   40 IF(IACX - 2) 44,42,44
   42 IF(ACTSX - SIGN(1.0,AX)) 46,48,46
   44 IF(ACTSX - SIGN(1.0,AY)) 46,48,46
C      AT ST 46 NO MATCH
   46 IF(IPLUS) 47,47,475
   47 IF(ABIG) 475,470,470
  470 IAC = -10
      ACX = 0.0
      GO TO 48
  475 ACPX = 0.0
      IPLUS = 0
   48 DVS = DVS + 0.5*(ACX + ACPX)*(TMM-TMMP)
      IF(IAC) 50,49,49
   49 IF(AC -1.0) 327,327,50
   50 RETURN
      END
C      SUBROUTINE RNGKT1(INIT,N,/X/,/H/,YSP,YP)
      SUBROUTINE RNGKT1(INIT,N,X,H,YSP,YP)
C        AUXILIARY SUBROUTINE  FROM CAL PROGRAM LIBRARY, MARCH 1971
C        SUBROUTINE RNGKT1 (CU0068)
C*****************************************************************
```

```
C*                                                                     *
C*                        SUBROUTINE RNGKT1                            *
C*                             CU 0068                                 *
C*                                                                     *
C*    PURPOSE                                                          *
C*       HYBRID PRECISION (SINGLE PRECISION LINKAGES, BUT DOUBLE       *
C*       PRECISION INTERNAL COMPUTATION)  RUNGE KUTTA INTEGRATION      *
C*       (BLUM'S MODIFICATION), 4TH-ORDER, FOR UP TO AND INCLUDING     *
C*       30 FIRST-ORDER DIFFERENTIAL EQUATIONS.                        *
C*                                                                     *
C*    USAGE                                                            *
C*       CALL RNGKT1 (INIT,N,X,H,Y,YP)                                 *
C*                                                                     *
C*    DESCRIPTION OF PARAMETERS                                        *
C*       INIT  =1  FOR INITIALIZATION,                                 *
C*             =2 FOR INTEGRATE ONE STEP                               *
C*       N     NUMBER OF FIRST-ORDER DIFFERENTIAL EQUATIONS            *
C*       X     INDEPENDENT VARIABLE (SINGLE PRECISION)                 *
C*       H     STEP SIZE (SINGLE PRECISION)                            *
C*       Y     ARRAY OF DEPENDENT VARIABLES (SINGLE PRECISION)         *
C*       YP    ARRAY OF DERIVATIVES OF Y WRT X (SINGLE PRECISION)      *
C*                                                                     *
C*    REMARKS                                                          *
C*       USER MUST SUPPLY A SUBROUTINE DAUX, SUCH THAT 'CALL DAUX'     *
C*       (WITHOUT EXPLICIT ARGUMENTS) CAUSES THE YP VALUES TO BE       *
C*       CALCULATED   USING X AND VALUES FROM THE Y ARRAY.             *
C*                                                                     *
C*       INITIALIZATION (INIT = 1) IS REQUIRED AFTER USER SETS         *
C*       INITIAL VALUES OF X AND THE Y-ARRAY.  RE-INITIALIZATION       *
C*       IS NOT NECESSARY FOR CHANGES IN STEP SIZE, H.                 *
C*                                                                     *
C*    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                    *
C*       DAUX     (SEE PREVIOUS REMARKS)                               *
C*                                                                     *
C*    METHOD                                                           *
C*       THE E.K. BLUM MODIFICATION OF THE RUNGE-KUTTA FOURTH-         *
C*       ORDER METHOD IS USED. SEE                                     *
C*           'MATHEMATICS OF COMPUTATION', APRIL, 1962  PP 176-187     *
C*                                                                     *
C*    AUTHOR DATE                                                      *
C*       W. FRYER                                                      *
C*       CORNELL AERO. LAB                                             *
C*       OCTOBER, 1966                                                 *
C*                                                                     *
C*       NOTE. CODING IS ADAPTED FROM SQUARE PARTEE                    *
C*             INTEGRATOR PROGRAMS PINT1 AND DINT.                     *
C*             DECKS CU0053, CU0054.                                   *
C*                                                                     *
C***********************************************************************
C
C
      REAL YSP(1), YP(1)
      DOUBLE PRECISION P(30),Q(30),X0,Y(30),HD
C
      GO TO (4,8), INIT
    4 CALL DAUX
      DO 5  I=1,N
    5 Y(I) = YSP(I)
      X0 = X
      RETURN
C
    8 HD = H
```

```
      DO 10 I=1,N
      P(I) = HD*YP(I)
      Y(I) = Y(I) + 0.5D0*P(I)
      YSP(I) = Y(I)
   10 Q(I) = P(I)
C
      X = X0 + 0.5D0*HD
      CALL DAUX
C
      DO 20 I=1,N
      P(I) = HD*YP(I)
      Y(I) = Y(I) + 0.5D0*(P(I)-Q(I))
      YSP(I) = Y(I)
   20 Q(I) = Q(I)/6.0D0
C
      CALL DAUX
C
      DO 30 I = 1,N
      P(I) = HD*YP(I) - 0.5D0*P(I)
      Y(I) = Y(I) + P(I)
      YSP(I) = Y(I)
   30 Q(I) = Q(I) - P(I)
C
      X0 = X0 + HD
      X = X0
      CALL DAUX
C
      DO 40 I = 1,N
      P(I) = HD*YP(I) + (P(I)+P(I))
      Y(I) = Y(I) + Q(I) + P(I)/6.0D0
   40 YSP(I) = Y(I)
C
      CALL DAUX
C
      RETURN
      END
      SUBROUTINE DAUX
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
      COMMON/INTG/NEQ,T,DT,VAR(12),DER(12)
      EQUIVALENCE (VAR(1),XCP1),(VAR(2),YCP1),(VAR(3),PSI1),
     1            (VAR(4),PSI1D),(VAR(5),U1),(VAR(6),V1),
     2            (VAR(7),XCP2),(VAR(8),YCP2),(VAR(9),PSI2),
     3            (VAR(10),PSI2D),(VAR(11),U2),(VAR(12),V2)
      EQUIVALENCE (DER(1),DXCP1),(DER(2),DYCP1),(DER(3),DPSI1),
     1            (DER(4),DPSI1D),(DER(5),DU1),(DER(6),DV1),
     2            (DER(7),DXCP2),(DER(8),DYCP2),(DER(9),DPSI2),
     3            (DER(10),DPSI2D),(DER(11),DU2),(DER(12),DV2)
      COMMON/INPT/  T0,TF,DTTRA0,DTCOL0,DTCLT0,DTPRN0,UVMIN,PSIDMN,FVEH0
     1             ,XCP10,YCP10,PSI10,PSI1D0,U10,V10,
     2              XCP20,YCP20,PSI20,PSI2D0,U20,V20,
     3              A1,B1,TR1,FIZ1,FM1,PSIR10,XF1,XR1,YS1,
     4              A2,B2,TR2,FIZ2,FM2,PSIR20,XF2,XR2,YS2,
     5              CSTFI(8),TBTQ1,TETQ1,TINCQ1,NTBLQ1,NTQ1,
     6              TBTQ2,TETQ2,TINCQ2,NTBLQ2,NTQ2,TII(8,201),
     7              TBPSF1,TEPSF1,TINCP1,NTBLP1,NPSF1,
     8              TBPSF2,TEPSF2,TINCP2,NTBLP2,NPSF2,PSIFI(4,201),
     9              XBP(2),YBP(2),XMU1,XMU2,CMU
      COMMON/INPT/  DELPS0,DELRO0,ALAMB,ZETAV,AKV(2) ,AMU,C0,C1,C2,
     1              PSLM10,PSLM20,PSLM30,PSLM40,PSLM50,PSLM60,PSLM70,
     2              PSLM80,HED(40),DADE(3),XINPUT(9)
      COMMON/INTGR/JCC(4),JCOR(4),NPSIB,IPSIB0,IPSIB,IBB,IFF,NDTAB,
```

```
     1             I9,J9(361,2),  NPSJB,IPSJB,I3,ILAST,IL,IND1,IND2,INDI
     2             ,INDJ,ISTOP,ITZER1,ITZER2,IPZER1,IPZER2,ICOUNT,INDXB,
     3             ICOLLP,ICOLL,IQ,IND,IVEH,LL,ICTVDI,ITCOL,ITFLG1,
     4             ITFLG2,NWRDB1,NWRDE1,NWRDB2,NWRDE2,ICOLAC
C             IN COMMON /COMP/ DO NOT DISTURB THE ORDER OF THE VARIABLES
C                              WHICH ARE EQUIVALENCED TO ARRAYS.
      COMMON/COMP/  DTTRAJ,DTCOLL,DTCOLT,DTPRNT,UVMN2,PSIDMA,TPRINT,DTP,
     1              U1V1SQ,U2V2SQ,DELPSI,DELPST,DELPS2,DELRHO,
     2              PSILM1,PSILM2,PSILM3,PSILM4,PSILM5,PSILM6,PSILM7,
     3              PSILM8,EJJ(4,2),GJJ(4,2),C1OC2,PSCC,PSIR1,PSIR2,
     4             XCMXC1,XCMXC2,YCMYC1,YCMYC2,SPSI1,SPSI2,CPSI1,CPSI2,
     5             XCPI,YCPI,PSII,XCPJ,YCPJ,PSIJ,XCMXC,YCMYC,SPSIJ,
     6             CPSIJ,XFI,XRI,YSI,XFJ,XRJ,YSJ,AKVI,AKVJ,EJ,GJ,
     7             XPJ(4),YPJ(4),ANGYX,SPAN,PSIBPI(4),PSIBPB,PSIBPF,
     8             FNPSIB,PSIBB,PSIB,PSFMPS,SCPSIB,CSPSIB,
     9             RHOBI,RHOBIN(4),PSIBIJ,SPSBIJ,CPSBIJ,XBIJ,YBIJ,PSJTB
      COMMON/COMP/ PSIBPJ,FNPSJB,PSIBJ,RHOBJ,XYJSQR,SPSIB,CPSIB,XBI,YBI,
     1         PRESI,PRESJ,XYSR,DELTA,CRHO,RHOBIC,RHOBMX,RHOBIT,RHOBT,
     2             SPSI21,CPSI21,X2TEM,Y2TEM,XX,YY,PAV,XXYYSR,FN1,SPSF,
     3             CPSF,X1AV,Y1AV,XYAVSR,X2TERM,Y2TERM,SPS21F,CPS21F,
     4             X2AV,Y2AV,VT1AV,VT2AV,VTMVT,FRICT,FNX1,FNY1,FNN1,
     5             FNX2,FNY2,FNN2,SFX1,SFY1,SFN1,SFX2,SFY2,SFN2,
     6              SFX1C,SFX2C,SFY1C,SFY2C,SFN1C,SFN2C,
     7              SFX1T,SFX2T,SFY1T,SFY2T,SFN1T,SFN2T,  TCOL,TEND,
     8              TREST1,TREST2,TCOLS1,TCOLS2,P1TEM,P2TEM,U1V1E,U2V2E,
     9              EXTRA(10)
      COMMON/COMPT/ W(8),XWP(8),YWP(8),FLAGW(8),PSIW(8),TQW(8),
     1             TFX,TFY,TMOM,PSIRR,COEF,COEF1,COEF2,TRD2,TRD21,TRD22,
     2             XBMXB,FNUM,A,B,FIZ,FMASS,XC,YC,U,V,PSIC,PSICD,UD2,
     3             SGNU,SGNI,AOMB,CSPSIC,SNPSIC,TRCPSC,TRSPSC,TRPSCD,
     4             XI,XW,YW,SX,SY,S,XPP,YPP,RHO,RHOP,XMUS,XMU,PSI,XNUM,
     5             XDEN,RATIO,ALFA,F,TIF,FC,FCOS,SMAL,FSMX,FS,BETB,FX,FY
     6             ,CSPSI,SNPSI,XCXC,YCYC,SPJ,CPJ,SPIJ,CPIJ,XFJJ,XRJJ,
     7              YSJJ,EJJI,GJJI,XICORJ,YICORJ,NTQ,TBTQ,TETQ,TINCRQ,
     8             NPSF,TBPSF,TEPSF,TINCRP
      COMMON/TAB/  TRHOB(361,2),TROBMX(361,2),TPSIB(361,2),
     1             TCPSIB(361,2),TSPSIB(361,2),TXB1(100),TYB1(100),
     2             TPSB1(100),TPRES1(100)
      DIMENSION   XCMXCI(2),YCMYCI(2),SPSII(2),CPSII(2),PSLM0(8),PSLM(8)
      EQUIVALENCE (XCMXCI(1),XCMXC1),(YCMYCI(1),YCMYC1),(SPSII(1),SPSI1)
     1           ,(CPSII(1),CPSI1),(PSLM0(1),PSLM10),(PSLM(1),PSILM1)
C
      IF(ISTOP.NE. 0) RETURN
      SFX1 = 0.0
      SFY1 = 0.0
      SFN1 = 0.0
      SFX1C = 0.0
      SFY1C = 0.0
      SFN1C = 0.0
      SFX1T = 0.0
      SFY1T = 0.0
      SFN1T = 0.0
      SPSI1     = SIN(PSI1)
      CPSI1   = COS(PSI1)
      IF(IVEH) 5,7,5
    5 SFX2 = 0.0
      SFY2 = 0.0
      SFN2 = 0.0
      SFX2C = 0.0
      SFY2C = 0.0
      SFN2C = 0.0
      SFX2T = 0.0
```

```
      SFY2T = 0.0
      SFN2T = 0.0
      SPSI2     = SIN(PSI2)
      CPSI2     = COS(PSI2)
      XCMXC1    = XCP1 - XCP2
      XCMXCI(2) = - XCMXC1
      YCMYC1    = YCP1 - YCP2
      YCMYCI(2) = - YCMYC1
    7 CALL TRAJ
      IF(IVEH)  9,20,9
    9 IF (ICOLL) 10,15,10
   10 CALL COLL
      SFX1 = SFX1T + SFX1C
      SFY1 = SFY1T + SFY1C
      SFN1 = SFN1T + SFN1C
      SFX2 = SFX2T + SFX2C
      SFY2 = SFY2T + SFY2C
      SFN2 = SFN2T + SFN2C
      GO TO 25
   15 SFX2 = SFX2T
      SFY2 = SFY2T
      SFN2 = SFN2T
   20 SFX1 = SFX1T
      SFY1 = SFY1T
      SFN1 = SFN1T
C             SET UP DERIVATIVES FOR RUNGE-KUTTA INTEGRATION
      IF(IVEH) 25,27,25
   25 DPSI2D = SFN2/FIZ2
      DPSI2  = PSI2D
      DU2  =  SFX2/FM2 +  V2*PSI2D
      DV2  =  SFY2/FM2 -  U2*PSI2D
      DXCP2  =  U2*CPSI2 - V2*SPSI2
      DYCP2  =  U2*SPSI2 + V2*CPSI2
   27 DPSI1D = SFN1/FIZ1
      DPSI1  = PSI1D
      DU1  =  SFX1/FM1 +  V1*PSI1D
      DV1  =  SFY1/FM1 -  U1*PSI1D
      DXCP1 =   U1*CPSI1 - V1*SPSI1
      DYCP1 =   U1*SPSI1 + V1*CPSI1
      RETURN
      END
```

```
      SUBROUTINE TRAJ
C       SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
      COMMON/INTG/NEQ,T,DT,VAR(12),DER(12)
      EQUIVALENCE (VAR(1),XCP1),(VAR(2),YCP1),(VAR(3),PSI1),
     1            (VAR(4),PSI1D),(VAR(5),U1),(VAR(6),V1),
     2            (VAR(7),XCP2),(VAR(8),YCP2),(VAR(9),PSI2),
     3            (VAR(10),PSI2D),(VAR(11),U2),(VAR(12),V2)
      EQUIVALENCE (DER(1),DXCP1),(DER(2),DYCP1),(DER(3),DPSI1),
     1            (DER(4),DPSI1D),(DER(5),DU1),(DER(6),DV1),
     2            (DER(7),DXCP2),(DER(8),DYCP2),(DER(9),DPSI2),
     3            (DER(10),DPSI2D),(DER(11),DU2),(DER(12),DV2)
      COMMON/INPT/  T0,TF,DTTRA0,DTCOL0,DTCLT0,DTPRN0,UVMIN,PSIDMN,FVEH0
     1             ,XCP10,YCP10,PSI10,PSI1D0,U10,V10,
     2              XCP20,YCP20,PSI20,PSI2D0,U20,V20,
     3              A1,B1,TR1,FIZ1,FM1,PSIR10,XF1,XR1,YS1,
     4              A2,B2,TR2,FIZ2,FM2,PSIR20,XF2,XR2,YS2,
     5              CSTFI(8),TBTQ1,TETQ1,TINCQ1,NTBLQ1,NTQ1,
     6              TBTQ2,TETQ2,TINCQ2,NTBLQ2,NTQ2,TII(8,201),
     7              TBPSF1,TEPSF1,TINCP1,NTBLP1,NPSF1,
     8              TBPSF2,TEPSF2,TINCP2,NTBLP2,NPSF2,PSIFI(4,201),
     9              XBP(2),YBP(2),XMU1,XMU2,CMU
      COMMON/INPT/  DELPS0,DELRO0,ALAMB,ZETAV,AKV(2) ,AMU,C0,C1,C2,
     1              PSLM10,PSLM20,PSLM30,PSLM40,PSLM50,PSLM60,PSLM70,
     2              PSLM80,HED(40),DADE(3),XINPUT(9)
      COMMON /OPTIONS/ IOPTION,DTINT,STEPPSI,STEPDT,SINPSI,SINFREQ,
     1                 GAIN,DISLKAHD,XYTB(2,201),NOUT1,NOUT2
      COMMON/INTGR/JCC(4),JCOR(4),NPSIB,IPSIB0,IPSIB,IBB,IFF,NDTAB,
     1             I9,J9(361,2),  NPSJB,IPSJB,I3,ILAST,IL,IND1,IND2,INDI
     2             ,INDJ,ISTOP,ITZER1,ITZER2,IPZER1,IPZER2,ICOUNT,INDXB,
     3             ICOLLP,ICOLL,IQ,IND,IVEH,LL,ICTVDI,ITCOL,ITFLG1,
     4             ITFLG2,NWRDB1,NWRDE1,NWRDB2,NWRDE2,ICOLAC
C             IN COMMON /COMP/ DO NOT DISTURB THE ORDER OF THE VARIABLES
C                              WHICH ARE EQUIVALENCED TO ARRAYS.
      COMMON/COMP/  DTTRAJ,DTCOLL,DTCOLT,DTPRNT,UVMN2,PSIDMA,TPRINT,DTP,
     1              U1V1SQ,U2V2SQ,DELPSI,DELPST,DELPS2,DELRHO,
     2              PSILM1,PSILM2,PSILM3,PSILM4,PSILM5,PSILM6,PSILM7,
     3              PSILM8,EJJ(4,2),GJJ(4,2),C1OC2,PSCC,PSIR1,PSIR2,
     4             XCMXC1,XCMXC2,YCMYC1,YCMYC2,SPSI1,SPSI2,CPSI1,CPSI2,
     5             XCPI,YCPI,PSII,XCPJ,YCPJ,PSIJ,XCMXC,YCMYC,SPSIJ,
     6             CPSIJ,XFI,XRI,YSI,XFJ,XRJ,YSJ,AKVI,AKVJ,EJ,GJ,
     7             XPJ(4),YPJ(4),ANGYX,SPAN,PSIBPI(4),PSIBPB,PSIBPF,
     8             FNPSIB,PSIBB,PSIB,PSFMPS,SCPSIB,CSPSIB,
     9             RHOBI,RHOBIN(4),PSIBIJ,SPSBIJ,CPSBIJ,XBIJ,YBIJ,PSJTB
      COMMON/COMP/ PSIBPJ,FNPSJB,PSIBJ,RHOBJ,XYJSQR,SPSIB,CPSIB,XBI,YBI,
     1         PRESI,PRESJ,XYSR,DELTA,CRHO,RHOBIC,RHOBMX,RHOBIT,RHOBT,
     2             SPSI21,CPSI21,X2TEM,Y2TEM,XX,YY,PAV,XXYYSR,FN1,SPSF,
     3             CPSF,X1AV,Y1AV,XYAVSR,X2TERM,Y2TERM,SPS21F,CPS21F,
     4             X2AV,Y2AV,VT1AV,VT2AV,VTMVT,FRICT,FNX1,FNY1,FNN1,
     5             FNX2,FNY2,FNN2,SFX1,SFY1,SFN1,SFX2,SFY2,SFN2,
     6              SFX1C,SFX2C,SFY1C,SFY2C,SFN1C,SFN2C,
     7              SFX1T,SFX2T,SFY1T,SFY2T,SFN1T,SFN2T,  TCOL,TEND,
     8              TREST1,TREST2,TCOLS1,TCOLS2,P1TEM,P2TEM,U1V1E,U2V2E,
     9              EXTRA(10)
      COMMON/COMPT/ W(8),XWP(8),YWP(8),FLAGW(8),PSIW(8),TQW(8),
     1             TFX,TFY,TMOM,PSIRR,COEF,COEF1,COEF2,TRD2,TRD21,TRD22,
     2             XBMXB,FNUM,A,B,FIZ,FMASS,XC,YC,U,V,PSIC,PSICD,UD2,
     3             SGNU,SGNI,AOMB,CSPSIC,SNPSIC,TRCPSC,TRSPSC,TRPSCD,
     4             XI,XW,YW,SX,SY,S,XPP,YPP,RHO,RHOP,XMUS,XMU,PSI,XNUM,
     5             XDEN,RATIO,ALFA,F,TIF,FC,FCOS,SMAL,FSMX,FS,BETB,FX,FY
     6             ,CSPSI,SNPSI,XCXC,YCYC,SPJ,CPJ,SPIJ,CPIJ,XFJJ,XRJJ,
```

```
     7              YSJJ,EJJI,GJJI,XICORJ,YICORJ,NTQ,TBTQ,TETQ,TINCRQ,
     8             NPSF,TBPSF,TEPSF,TINCRP
      COMMON/TAB/  TRHOB(361,2),TROBMX(361,2),TPSIB(361,2),
     1             TCPSIB(361,2),TSPSIB(361,2),TXB1(100),TYB1(100),
     2             TPSB1(100),TPRES1(100)
      DIMENSION   XCMXCI(2),YCMYCI(2),SPSII(2),CPSII(2),PSLM0(8),PSLM(8)
      EQUIVALENCE (XCMXCI(1),XCMXC1),(YCMYCI(1),YCMYC1),(SPSII(1),SPSI1)
     1           ,(CPSII(1),CPSI1),(PSLM0(1),PSLM10),(PSLM(1),PSILM1)
      DATA BLNK/1H /,STAR/1H*/
C
      IF(ISTOP.NE. 0) RETURN
      XC = XCP1
      YC = YCP1
      U = U1
      V = V1
      PSIC = PSI1
      CSPSIC = CPSI1
      SNPSIC = SPSI1
      PSICD = PSI1D
      A = A1
      B = B1
      COEF = COEF1
      TRD2 = TRD21
      PSIRR = PSIR1
      FIZ = FIZ1
      FMASS = FM1
      TBPSF = TBPSF1
      TEPSF = TEPSF1
      TINCRP = TINCP1
      NPSF = NPSF1
      TBTQ = TBTQ1
      TETQ = TETQ1
      TINCRQ = TINCQ1
      NTQ = NTQ1
      KK = 0
      II = 0
   30 SGNU = SIGN(1.0,U)
      UD2 = ABS(0.5*U)
      TRCPSC  = CSPSIC * TRD2
      TRSPSC  = SNPSIC * TRD2
      TRPSCD = PSICD * TRD2
      TFX = 0.0
      TFY = 0.0
      TMOM = 0.0
      AOMB = A
      SGNI = 1.0
C          LOOP FOR WHEELS
C        RF = 1,  LF = 2,  RR = 3,  LR = 4
      DO 105 I=1,4
      III = I + II
      SGNI = -1.0*SGNI
      IF(I - 2) 32,32,31
   31 AOMB = -B
   32 XW = XC + AOMB*CSPSIC + SGNI * TRSPSC
      YW = YC + AOMB*SNPSIC - SGNI * TRCPSC
      XWP(III) = XW
      YWP(III) = YW
      IF(INDXB) 35,37,36
   35 ISTOP = 15
      WRITE(NOUT,6035) ISTOP
 6035 FORMAT( 8H0 ISTOP=,I4,17H, INDXB  NEGATIVE)
      RETURN
```

```
   36 XPP= FNUM *XW/(YW - XI*XW)
   37 YPP=YW*XPP/XW
      SX = U + SGNI*TRPSCD
      SY = V + AOMB*PSICD
      RHO = SQRT(XW*XW + YW*YW)
      RHOP=SQRT(XPP*XPP + YPP*YPP)
      IF(RHO - RHOP) 38,38,39
   38 XMUS = XMU1
      GO TO 40
   39 XMUS = XMU2
   40 S   = SQRT( SX*SX + SY*SY)
      XMU = XMUS*(1.0-CMU*S)
      IF(I-2) 42,42,41
   41 PSI = PSIRR
      GO TO 45
   42 IF(NPSF-1) 43,43,44
   43 PSI = 0.0
C       NO STEER TABLES
      GO TO 45
   44 K = I + KK
C       Original program uses a routine of
C       interpolation to get steering angle from given table
C
C       In the new routine, the steering anlge is determined
C       by the selected option, thus a new call needs
C       to be inserted here.  The new subroutine should return a
C       a steering angle
C
C       If KK = 0 (I = 1,2) the routine is calculating for Vehicle 1
C       If KK = 2 (I = 1,2) the routine is calculating for Vehicle 2
C
      IF (IOPTION .EQ. 0) THEN
         CALL INTRPV(ND4,K,PSIFI,TBPSF,TEPSF,TINCRP,T,PSI)
      ELSEIF (IOPTION .EQ. 1) THEN
         CALL STEPSTR(T, PSI)
      ELSEIF (IOPTION .EQ. 2) THEN
         CALL SINSTR(T, PSI)
      ELSEIF (IOPTION .GT. 2 .AND. IOPTION .LT. 5) THEN
         CALL STRCTRL(K,XC,YC,PSIC,U,V,PSICD,PSI)
      ENDIF
      PSI = PSI * RAD
   45 PSIW(III) = PSI
      XNUM = SY
      XDEN= ABS(SX)
      IF(XDEN)55,50,55
   50 IF(XNUM)54,52,54
   52 RATIO = 0.0
      GO TO 56
   54 RATIO = 1.57
      GO TO 56
   55 RATIO = ATAN2(XNUM,XDEN)
   56 ALFA = RATIO - PSI*SGNU
   57 F = XMU * W(III)
      IF(NTQ-1) 58,58,59
   58 TIF = 0.0
C       NO TORQUE TABLES
      GO TO 60
   59 CALL INTRPV (ND8,III,TII,TBTQ,TETQ,TINCRQ,T,TIF)
   60 TQW(III) = TIF
      IF(TIF) 70,61,62
   61 FC = 0.0
      GO TO 80
```

```
   62 IF(TIF-F) 64,64,66
   64 FC = TIF
      GO TO 80
   66 FC = F
      GO TO 80
   70 FCOS = F*COS(ALFA)
      IF(UD2-1.0) 72,73,73
   72 SMAL = UD2
      GO TO 74
   73 SMAL = 1.0
   74 IF( ABS(TIF) - FCOS) 76,76,78
   76 FC = TIF*SGNU*SMAL
      GO TO 80
   78 FC = -FCOS*SGNU*SMAL
   80 IF( ABS(FC) - (ABS(F) -1.0)) 86,84,84
   84 FS = 0.0
      FLAGW(III) = STAR
      GO TO 100
   86 IF(ABS(U) - 0.5) 88,90,90
   88 IF(ABS(V) - 0.5) 84,90,90
   90 FSMX = SQRT(F*F - FC*FC)
      BETB = CSTFI(III) * ALFA /FSMX
      IF(ABS(BETB)-3.0)94,92,92
   92 FS = FSMX * SIGN(1.0,BETB)
      FLAGW(III) = STAR
      GO TO 100
   94 FS = FSMX*BETB*(BETB*BETB/27.0 - ABS(BETB)/3.0 +1.0)
      FLAGW(III) = BLNK
  100 CSPSI = COS(PSI)
      SNPSI = SIN(PSI)
      FY = FS*CSPSI + FC*SNPSI
      FX = -FS*SNPSI + FC*CSPSI
      TFX = TFX + FX
      TFY = TFY + FY
      TMOM = TMOM  + SGNI * TRD2 * FX   + AOMB * FY
  105 CONTINUE
C           END OF LOOP FOR WHEELS
      IF(II) 115,110,115
  110 SFX1T = TFX
      SFY1T = TFY
      SFN1T = TMOM
      IF (IVEH) 112,117,112
  112 XC = XCP2
      YC = YCP2
      U = U2
      V = V2
      PSIC = PSI2
      CSPSIC = CPSI2
      SNPSIC = SPSI2
      PSICD = PSI2D
      A = A2
      B = B2
      COEF = COEF2
      TRD2 = TRD22
      PSIRR = PSIR2
      FIZ = FIZ2
      FMASS = FM2
      TBPSF = TBPSF2
      TEPSF = TEPSF2
      TINCRP = TINCP2
      NPSF = NPSF2
      TBTQ = TBTQ2
```

```
      TETQ = TETQ2
      TINCRQ = TINCQ2
      NTQ = NTQ2
      KK = 2
      II = 4
      GO TO 30
  115 SFX2T = TFX
      SFY2T = TFY
      SFN2T = TMOM
  117 RETURN
      END
      SUBROUTINE INTRPV(ND,ID,TABLE,XMIN,XMAX,DX,X,Y)
C      SIMULATION MODEL OF AUTOMOBILE COLLISIONS -SMAC
C       QUADRATIC
      DIMENSION TABLE(ND,1)
    1 XLK = AMIN1(X,XMAX)
      XLK = AMAX1(XLK,XMIN)
      N1 = (XLK-XMIN)/DX+1.2
      N2 = N1+1
      NT = (XMAX-XMIN)/DX+1.2
      N0 = N1-1
    2 IF(N0.GT.0) GO TO 3
      N0 = N1
      N1 = N2
      N2 = N1+1
    3 IF(N2.LE.NT) GO TO 4
      N2 = N1
      N1 = N0
      N0 = N1-1
    4 XXX = FLOAT(N0)*DX+XMIN
      DX2 = DX**2
      TT1 = TABLE(ID,N1)
      TT0 = TABLE(ID,N0)
      A = ( TABLE(ID,N2) -2.0*TT1 + TT0) / (2.0*DX2)
      B = (TT1 -TT0)/DX -A*(2.0*XXX - DX)
      C = TT1 -(A*XXX**2 + B*XXX)
      Y = (A*XLK+B)*XLK+C
      RETURN
      END
C
C       New subroutine for creating steering input
C
      SUBROUTINE STEPSTR(T,PSI)
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
      COMMON /OPTIONS/ IOPTION, DTINT, STEPPSI, STEPDT, SINPSI, SINFREQ,
     1                 GAIN, DISLKAHD, XYTB(2,201), NOUT1, NOUT2
C       STEP FUNCTION STARTS AFTER 0.25 SECOND OF INITIATION
      IF (T .LE. DTINT ) THEN
         PSI = 0.0
      ELSE
         IF (T .LT. (DTINT+STEPDT) ) THEN
            PSI = (T-DTINT)/STEPDT*STEPPSI
         ELSE
            PSI = STEPPSI
         ENDIF
      ENDIF
      RETURN
      END
      SUBROUTINE SINSTR(T,PSI)
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
      COMMON /OPTIONS/ IOPTION, DTINT, STEPPSI, STEPDT, SINPSI, SINFREQ,
     1                 GAIN, DISLKAHD, XYTB(2,201), NOUT1, NOUT2
```

```
C       SIN FUNCTION STARTS AFTER 0.25 SECOND
      IF (T .LE. DTINT ) THEN
         PSI = 0.0
      ELSE
         PSI = SINPSI*SIN(TWOPI*SINFREQ*(T-DTINT))
      ENDIF
      RETURN
      END
      SUBROUTINE STRCTRL(K,X,Y,YAW,U,V,YAWD,PSI)
C
C     Cal subroutine to find latereal position and yaw deviation
C
      CALL DEVIATE(X,Y,YAW,DEVLAT)
C
C       CALL SUBROUTINE TO GET DESIRED YAW ANGLE
C
      CALL YAWCAL(X,Y,YAW,DESYAW)
C
C     Implement control algorithms to calculate steering input
C     given deviation and desired yaw, and other variables in common
C     blocks to determine desired steeering angle
C
      CALL CNTRL(K,X,Y,YAW,U,V,YAWD,DEVLAT,DESYAW,PSI)
      RETURN
      END
C
C       CALL subroutine to find latereal position deviation
C
      SUBROUTINE DEVIATE(X,Y,YAW,DEVY)
      COMMON /DESTRAJ/ IXYDES, XYTB(2,201)
      DUMMY = X+Y+YAW
      DEVY = Y
      RETURN
      END
C
C       CALL SUBROUTINE TO GET DESIRED YAW ANGLE
C
      SUBROUTINE YAWCAL(X,Y,YAW,DYAW)
      DUMMY = X+Y+YAW
      DYAW = 0.0
      RETURN
      END
C
C       SUBROUTINE TO DETERMINE THE STEERING ANGLE
C
      SUBROUTINE CNTRL(K,X,Y,YAW,U,V,YAWD,DY,DYAW,CNTLPSI)
      COMMON/CONST/ G,RAD,PI,TWOPI,PIO2,PI32,ND2,ND4,ND8,NIN,NOUT
      COMMON/INTG/NEQ,T,DT,VAR(12),DER(12)
      COMMON/INTGR/JCC(4),JCOR(4),NPSIB,IPSIB0,IPSIB,IBB,IFF,NDTAB,
     1             I9,J9(361,2),  NPSJB,IPSJB,I3,ILAST,IL,IND1,IND2,INDI
     2             ,INDJ,ISTOP,ITZER1,ITZER2,IPZER1,IPZER2,ICOUNT,INDXB,
     3             ICOLLP,ICOLL,IQ,IND,IVEH,LL,ICTVDI,ITCOL,ITFLG1,
     4             ITFLG2,NWRDB1,NWRDE1,NWRDB2,NWRDE2,ICOLAC
C
      COMMON /OPTIONS/ IOPTION, DTINT, STEPPSI, STEPDT, SINPSI, SINFREQ,
     1                 GAIN, DISLKAHD, XYTB(2,201), NOUT1, NOUT2
      COMMON/DELAY/ TDGT, DTDGT, DY1LST, YAW1LST, DY2LST, YAW2LST
C
C     K = 1, 2 VEHICLE 1
C     K = 3, 4 VEHICLE 2
C
C       FIRST CONTROL LAW
```

```
C       PSI = - GAIN * ( DY + DISLKAHD*YAW )
C               PSI IN RADIAN
C               DY AND LOOK-AHEAD DISTANCE IN METER
C
      DUMMY = X+Y+YAW+U+V+YAWD+DY+DYAW
      IF ( K .LT. 3 ) THEN
         ICAR = 1
         IF (IOPTION .EQ. 4) THEN
            DYX = DY1LST
            YAWX = YAW1LST
         ELSE
            DYX = Y
            YAWX = YAW
         ENDIF
      ELSEIF ( ICOLL .NE. 0 ) THEN
C
C     K.GT. 2 CAR NO2 AND ICOLL INDICATING COLLISION HAS OCCURRED
C
         ICAR = 2
         IF (IOPTION .EQ. 4) THEN
            DYX = DY2LST
            YAWX = YAW2LST
         ELSE
            DYX = Y
            YAWX = YAW
         ENDIF
      ENDIF
      CNTLPSI = - GAIN*(DYX*0.0254 + YAWX*DISLKAHD)/RAD
C      WRITE(6,*) K, T, DYM, CNTLPSI
      RETURN
      END
```