MATHEMATICAL RECONSTRUCTION OF HIGHWAY ACCIDENTS - FURTHER EXTENSIONS AND REFINEMENTS OF THE CRASH COMPUTER PROGRAM

EDC Library Ref. No. 1010

Received May 27, 77
from NHTSA

DOT HS-802 287

ATHEMATICAL RECONSTRUCTION HIGHWAY ACCIDENTS - FURTHER TENSIONS AND REFINEMENTS OF THE RASH COMPUTER PROGRAM

Contract No. DOT-HS-5-01124
March 1977
Final Report

PREPARED FOR:

U.S. DEPARTMENT OF TRANSPORTATION
NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION
WASHINGTON, D.C. 20590

NOTICE

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. DOT HS 802 287	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Mathematical Reconstruction of Highway Accidents - Further Extensions and Refinements of the Crash Computer Program		5. Report Date March 1977 6. Performing Organization Code
7. Author(s) Raymond R. McHenry and James P. Lynch		8. Performing Organization Report No. ZQ-5708-V-5
9. Performing Organization Name and Address Calspan Corporation 4455 Genesee Street Buffalo, New York 14221		10. Work Unit No. 11. Contract or Grant No. DOT-HS-5-01124
12. Sponsoring Agency Name and Address U.S. Department of Transportation National Highway Traffic Safety Administration Washington, D. C. 20590		13. Type of Report and Period Covered Final Report February-November 1976 14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The Calspan Reconstruction of Accident Speeds on the Highway (CRASH) computer program was modified to (1) incorporate an optional trajectory simulation routine, based on the corresponding portion of the Simulation Model of Automobile Collisions (SMAC) computer program, to permit automatic testing and refinement of the CRASH estimates of separation velocities, (2) provide an optional abbreviated format for the time-sharing version of CRASH, (3) provide an optional batch mode of operation, and (4) revise and extend the output format. This report documents the analytical bases for the modifications that have been incorporated in the CRASH2 computer program.

17. Key Words Collisions Accident Investigation Crash Responses Computer Program Reconstruction	18. Distribution Statement [illegible]		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 38	22. Price

Form DOT F 1700.7 (8-69)

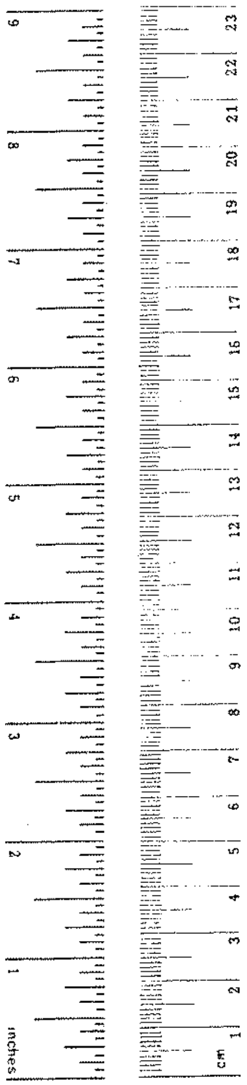
METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

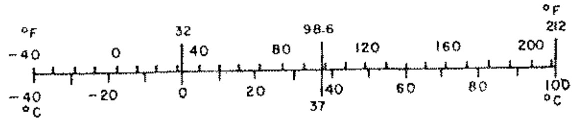
---

*1 in ≡ 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



FOREWORD

This report summarizes part of the research results achieved within the continuation, covered by Modifications No. 3 and 4, of Contract No. DOT-HS-5-01124 with the National Highway Traffic Safety Administration, U. S. Department of Transportation. A separate report, in the form of a User's Manual for the CRASH Program (Calspan Report No. ZQ-5708-V-4, October 1976) and the CRASH2 computer program have also been prepared and submitted under the cited contract continuation.

The opinions, findings and conclusions expressed in this report are those of the authors and not necessarily those of the National Highway Traffic Safety Administration.

This report has been reviewed and is approved by:

Edwin A. Kidd

Edwin A. Kidd, Head
Transportation Research Department

ACKNOWLEDGEMENT

The NHTSA Contract Technical Managers for this program of analytical research have been Dr. Kenneth J. Tharp and Mr. Thomas Noga.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The CRASH* computer program has been developed with the objective of achieving uniformity and improved accuracy in the interpretation of physical evidence from automobile accidents. In particular, it serves to generate approximations of the directions and magnitudes of the speed changes, ΔV_i, experienced by the two vehicles and of the collision speeds, on the basis of measured physical evidence.

The program is designed to accommodate a range of accident evidence, from VDIs only at one extreme to complete definitions of rest and impact positions as well as damage dimensions at the other. In full-capability runs, multiple outputs of speed changes are provided with identification of the basis for each approximation. In this manner, it is possible for the user to select the approximation result based on the most reliable items of evidence while assuring that the various different items of evidence are compatible.

In preliminary applications to evidence generated by staged collisions, a range of ± 12% accuracy of reconstructed speeds and speed-changes has been indicated (Reference 1). However, the present level of accuracy, with the trajectory-testing option and other refinements is believed to be significantly better than the earlier findings. The potential accuracy, with planned refinements in the stored vehicle parameter data and empirical coefficients, is expected to approach the range of ± 5%.

---

*Calspan Reconstruction of Accident Speeds on the Highway.

The computer costs for time-sharing operation of the CRASH program range from approximately $1.50 to $10.00 per case, depending on the extent of options used. The upper end of the indicated cost range corresponds to a run in which the option for testing and refining the trajectory analysis portion of the calculation has been exercised. In the batch processing mode of program operation, the computer costs have been found to range from approximately $0.50 to $0.90. Note that the trajectory simulation option is not included in the batch processing mode.

This report documents the analytical bases and the computer aspects of modifications incorporated in the CRASH2 computer program. It should be noted that a number of options and features beyond those specified in the Work Statement have been included within this program modification. For example, a number of IBM assembly language routines were eliminated for compatibility with non-IBM equipment. The MENU feature which controls the various options, the backspace and question mark (with expanded explanations of questions) features, and the RERUN option all fall within this category. Multiple revisions were made of the output format, as definitions of the desired format were changed. Calculations were incorporated to shift the user-entered midpoint dimension of the damage, D, to the centroid of the damage area prior to the calculation of effective masses in the DAMAGE routine. Also, a considerable amount of effort was applied to troubleshooting the inputs of application runs in which user difficulties had been encountered. As a result of the cited diversions of effort, the effort on some detailed task items (e.g., lower limits on errors and vehicle damage data) had to be curtailed.

The need for incorporating lower limits on errors was diminished by the fact that the trajectory errors were found to either converge to acceptable limits within five iterations or to diverge and, thereby, to clearly indicate a need for input revisions. The lower limit feature would not, therefore, enhance results in any way. Rather, it would serve only to end the iterative process at less than five iterations in some cases.

Emphasis was further shifted away from this program feature by a decision to omit the trajectory option from batch-processed runs.

Effort on refining the damage tables was delayed by the complexity of developing a proper and accurate procedure for extracting crush properties from the results of staged two-car collisions. Such a procedure is being developed under Contract No. DOT-HS-6-01372. However, it was not available for use within the performance period of the presently reported research.

Conclusions and recommendations based on results of this research are presented in Section 2. The results of the research are summarized in Section 3. References are listed in Section 4.

2. CONCLUSIONS AND RECOMMENDATIONS

2.1 Conclusions

2.1.1 The trajectory simulation option constitutes a highly desirable diagnostic tool for cases in which results based on damage differ significantly from those based on the trajectory analysis.

2.1.2 User convenience has been substantially improved through introduction of the ABBREVIATED and RERUN options.

2.1.3 The batch processing mode of CRASH operation provides significant reductions of operating costs from those of the time-sharing mode.

2.1.4 The extensive modifications that have been made to the CRASH program coding improve clarity and make the program more suitable for non-IBM computer systems.

2.2 Recommendations

2.2.1 In the course of future development of the CRASH computer program, the distinction between the needs of a "production" type program and those of a developmental prototype must be recognized. The introduction of an excessive number of user options and adjustments can work against the original objective of uniformity of evidence interpretations. Therefore, analytical refinements and extensions must be "built-in" and made automatic to the greatest extent possible.

2.2.2 Provisions for inclusion of the effects of underride/override on the interpretation of vehicle crush dimensions should be incorporated.

2.2.3 Simple approximation techniques and corresponding user questions related to common complicating factors (e.g., secondary collisions, terrain features, rollovers) should be prepared for future incorporation in the CRASH program. As application experience is gained, it is anticipated that a need to extend generality will become apparent.

2.2.4 The RERUN option should be extended to include automatic presentation of questions affected by a change in a YES or NO response.

2.2.5 Additional logic should be incorporated in the error calculations within the trajectory simulation option to permit user entry of either positive or negative heading angles. The remainder of the CRASH2 program currently contains such logic.

2.2.6 An additional internal check of user inputs should be incorporated in CRASH2 to verify that the specified spinout dimensions and conditions are compatible with a common velocity at the damage centroids.

3. DISCUSSION OF RESULTS

3.1 Trajectory Simulation Routine

The objective of this program extension has been to test the match of rest position and orientation and of other items of trajectory evidence that is achieved in a time-history simulation of the spinout using the SPIN2 values for the separation velocities. A further objective has been to automatically make adjustments, as required, in the separation conditions to improve the evidence match. In particular, the independent variables from SPIN2 consist of the magnitude and direction of the linear velocity at separation, $\dot{S}_s$ (inches/second) and γ_s (radians), and the angular velocity, $\dot{\psi}_s$ (degrees/second). The trajectory evidence consists of up to five separate items. On the basis of discrepancies between the predicted (i.e., time-history simulation) and measured items of trajectory evidence, adjustments of the independent variables are made in up to five iterative steps. Weighting factors are applied to the individual errors, reflecting their relative importance in the overall evidence match.

Symbols

X'_s, Y'_s, ψ'_s = Position and heading at separation, ft. and deg.

X'_R, Y'_R, ψ'_R = Measured position and heading at rest, feet and degrees

X'_1, Y'_1, ψ'_1 = Measured position and heading at end of rotation, feet and degrees

X'_2, Y'_2 = Measured intermediate position on curved trajectory, feet

X''_R, Y''_R, ψ''_R = Predicted position and heading at rest, feet and degrees

X''_1, Y''_1, ψ''_1 = Predicted position and heading at end of rotation, feet and degrees

X''_2, Y''_2 = Predicted intermediate position on curved trajectory, feet. Determined, for JCURV=1, in the following manner: At each point in time the distance between the vehicle center of gravity and X'_2, Y'_2 is calculated and compared with the value from the previous time increment. The smaller value is retained.

ε_1 = Rest position error

ε_2 = End of rotation position error

ε_3 = Rest orientation error

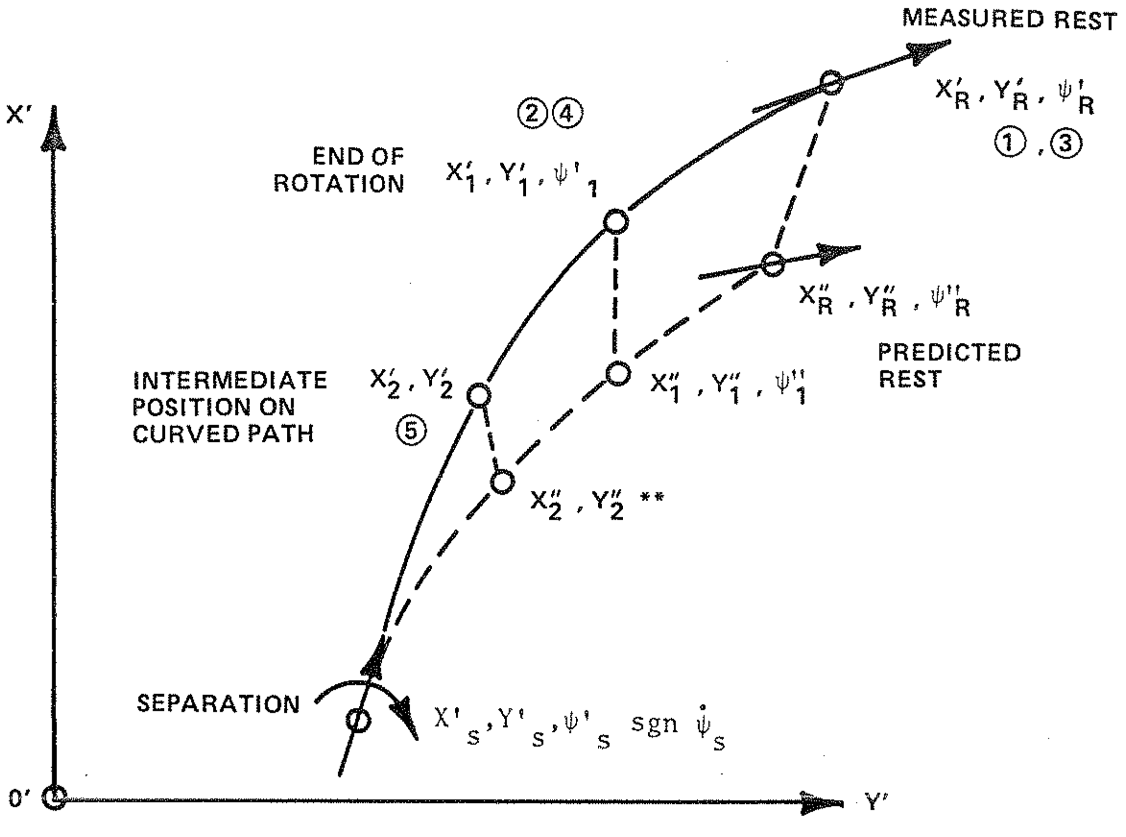
ε_4 = End of rotation orientation error

ε_5 = Intermediate position on curved path error

C_1 through C_5 = Weighting factors applied to the calculated errors

Error in positions along the spinout trajectories can occur in the form of range and/or azimuth discrepancies (i.e., the magnitude of a position error alone does not distinguish between points on a circle around the measured position). In the following, the range and azimuth components of position errors are separately calculated and are identified by the symbols γ (azimuth) or ℓ (range) combined with the error number. Orientation errors do not have components.

The calculation of errors is depicted schematically in Figure 1. The analytical relationships are as follows.



**PREDICTED PATH NOT ANALYTICALLY DEFINED.
LOGIC REQUIRED TO DEFINE X_2'', Y_2''.

Figure 1 SCHEMATIC SKETCH OF ERROR CALCULATIONS

3.1.1 Position Errors

(a) Rest Position Azimuth Error

$$\gamma_R = \arctan\left(\frac{Y'_R - Y'_s}{X'_R - X'_s}\right)$$

$$\gamma'_R = \arctan\left(\frac{Y''_R - Y'_s}{X''_R - X'_s}\right)$$

$$\varepsilon_{1\gamma} = \gamma_R - \gamma'_R$$

Form of correction:

$$\gamma_s = \gamma_s + \varepsilon_{1\gamma}$$

(b) Rest Position Range Error

$$\ell_R = \sqrt{(X'_R - X'_s)^2 + (Y'_R - Y'_s)^2}$$

$$\ell'_R = \sqrt{(X''_R - X'_s)^2 + (Y''_R - Y'_s)^2}$$

$$\varepsilon_{1\ell} = \frac{\ell_R - \ell'_R}{\ell'_R}$$

Form of correction:

$$\dot{S}_s = \dot{S}_s \sqrt{(1 + \varepsilon_{1\ell})}$$

(c) End-of-Rotation Azimuth Error

$$\gamma_1 = \arctan\left(\frac{Y'_1 - Y'_s}{X'_1 - X'_s}\right)$$

$$\gamma'_1 = \arctan\left(\frac{Y''_1 - Y'_s}{X''_1 - X'_s}\right)$$

$$\varepsilon_{2\gamma} = \gamma_1 - \gamma'_1$$

<u>Form of correction:</u>

$$\gamma_s = \gamma_s + \varepsilon_{2\gamma}$$

(d) End-of-Rotation Range Error

$$\ell_1 = \sqrt{(X'_1 - X'_s)^2 + (Y'_1 - Y'_s)^2}$$

$$\ell'_1 = \sqrt{(X''_1 - X'_s)^2 + (Y''_1 - Y'_s)^2}$$

$$\varepsilon_{2\ell} = \frac{\ell_1 - \ell'_1}{\ell'_1}$$

<u>Form of correction:</u>

$$\dot{S}_s = \dot{S}_s \sqrt{(1 + \varepsilon_{2\ell})}$$

(e) Curved Path Azimuth Error

$$\gamma_2 = \arctan\left(\frac{Y'_2-Y'_s}{X'_2-X'_s}\right)$$

$$\gamma'_2 = \arctan\left(\frac{Y''_2-Y'_s}{X''_2-X'_s}\right)$$

$$\varepsilon_{5\gamma} = \gamma_2-\gamma'_2$$

<u>Form of correction:</u>

$$\gamma_s = \gamma_s + \varepsilon_{5\gamma}$$

(f) Curved Path Range Error

$$\ell_2 = \sqrt{(X'_2-X'_s)^2 + (Y'_2-Y'_s)^2}$$

$$\ell'_2 = \sqrt{(X''_2-X'_s)^2 + (Y''_2-Y'_s)^2}$$

$$\varepsilon_{5\ell} = \frac{\ell_2-\ell'_2}{\ell'_2}$$

<u>Form of correction:</u>

$$\dot{S}_s = \dot{S}_s \sqrt{(1 + \varepsilon_{5\ell})}$$

3.1.2 Orientation Errors

(a) Rest Position Orientation Error*

$$\varepsilon_3 = \left(\frac{\psi'_R - \psi''_R}{\psi'_R - \psi'_s}\right)$$

Form of correction:

$$\dot{\psi}_s = \dot{\psi}_s \sqrt{(1 + \varepsilon_3)}$$

If $\varepsilon_3 \leq -0.99$, set $\varepsilon_3 = -0.99$

(b) End-of-Rotation Orientation Error*

$$\varepsilon_4 = \left(\frac{\psi'_1 - \psi''_1}{\psi'_1 - \psi'_s}\right)$$

Form of correction:

$$\dot{\psi}_s = \dot{\psi}_s \sqrt{(1 + \varepsilon_4)}$$

If $\varepsilon_4 \leq -0.99$, set $\varepsilon_4 = -0.99$

---

* It was discovered late in the research program that this error calculation, as programmed, requires positive algebraic signs for ψ'_s and ψ'_R (see Recommendation 2.2.5).

3.1.3 Iterative Adjustments of Independent Variables

Let A_{ji} = Adjustment coefficient

C_i = Weighting factor

If $(\varepsilon_i)_{max} < |\varepsilon_i|$, for any value of i,

$$Q_n = \sum_{i=1}^{i=5} C_i |\varepsilon_i| \quad \text{(i.e., weighted sum of errors)}$$

The independent variables are adjusted in the following manner:

$$A_1 = \frac{C_1 A_{11} + C_2 A_{12} + C_5 A_{15}}{(C_1 + C_2 + C_5)}$$

$$A_2 = \frac{C_3 A_{23} + C_4 A_{24}}{(C_3 + C_4)}$$

$$A_3 = \frac{C_1 A_{31} + C_2 A_{32} + C_5 A_{35}}{(C_1 + C_2 + C_5)}$$

$$\dot{S}_s = A_1 \dot{S}_s$$

$$\dot{\psi}_s = A_2 \dot{\psi}_s$$

$$\gamma_s = A_3 + \gamma_s$$

where A_{ij} are defined as follows:

A_{ji}

i	j=1	j=2	j=3
1	$\sqrt{(1 + \varepsilon_{1\ell})}$	1.00	$\varepsilon_{1\gamma}$
2	$\sqrt{(1 + \varepsilon_{2\ell})}$	1.00	$\varepsilon_{2\gamma}$
3	1.00	$\sqrt{(1 + \varepsilon_{3})}$	1.00
4	1.00	$\sqrt{(1 + \varepsilon_{4})}$	1.00
5	$\sqrt{(1 + \varepsilon_{5\ell})}$	1.00	$\varepsilon_{5\gamma}$

On the basis of a limited number of trial runs, the following values for maximum errors and weighting factors were adopted.

$\varepsilon_1 = 0.10$ $\quad C_1 = 1.0$

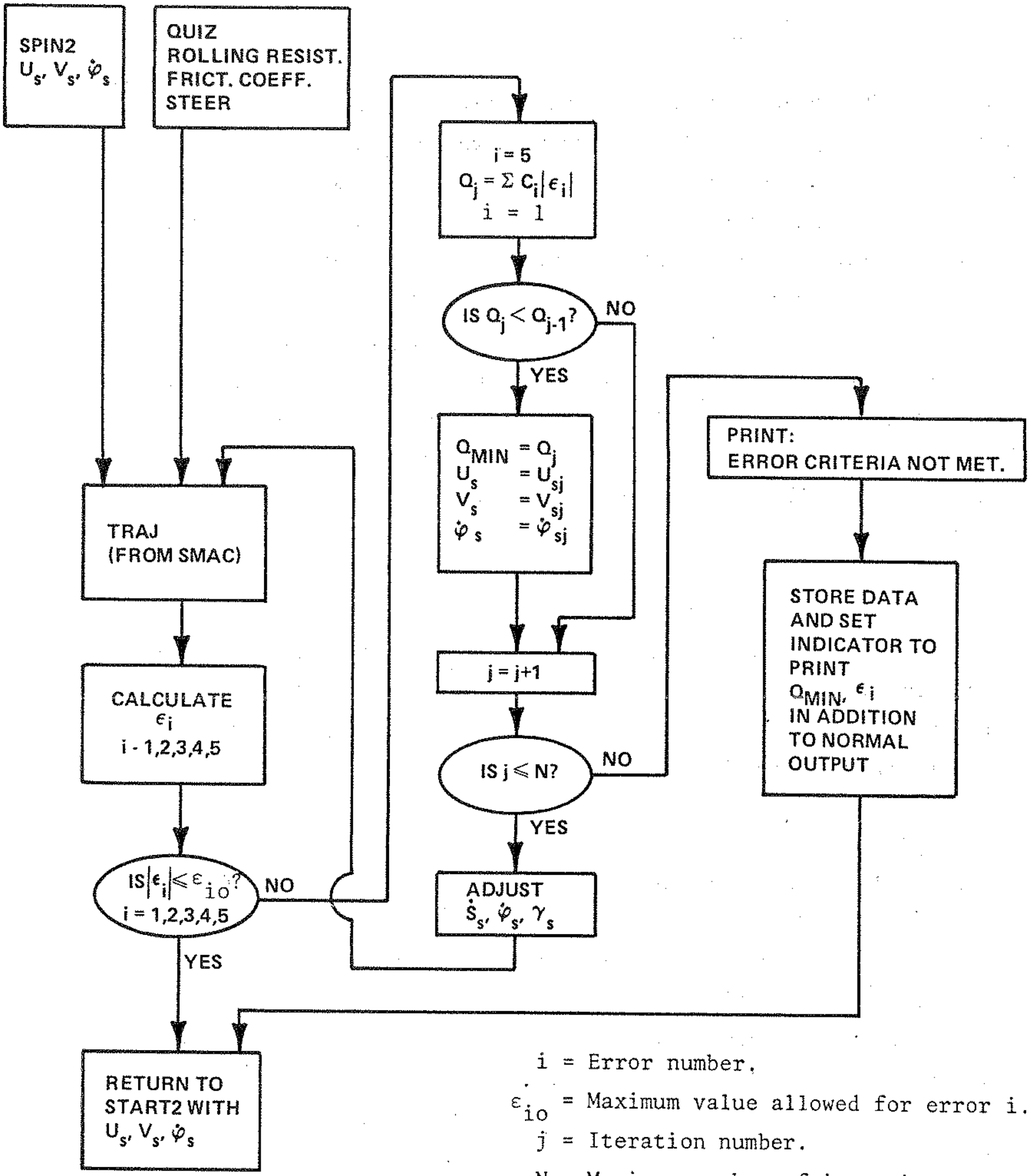
$\varepsilon_2 = 0.15$ $\quad C_2 = 0.5$

$\varepsilon_3 = 0.10$ $\quad C_3 = 1.0$

$\varepsilon_4 = 0.15$ $\quad C_4 = 0.5$

$\varepsilon_5 = 0.15$ $\quad C_5 = 0.25$

A schematic flowchart of the iterative loop is shown in Figure 2.



i = Error number.

ε_{io} = Maximum value allowed for error i.

j = Iteration number.

N = Maximum number of iterations.

Figure 2 SCHEMATIC FLOW CHART OF ITERATIVE LOOP

3.2 Batch Processor

The CRASH program is an iteractive investigation aid designed to provide investigations with a uniform, accurate, and convenient interpretation of accident evidence. There are certain situations where the interactive nature of the CRASH program is too costly and/or too time-consuming for the purpose envisioned. For this reason, a batch processing feature was built into the CRASH program. This batch processor allows users to construct a file of CRASH input data representing one or more accident cases. This input file of multiple cases can be submitted to the CRASH program using the batch processing procedure, and the results are placed into a multiple case output file. Users who wish to process large data banks of accident cases with the CRASH program will find the batch processing procedure much more convenient to use.

A complete description of the batch processing input formats is given in Section 5 of the CRASH2 User's Manual. Since the addition of the batch processor complicates program installation somewhat, the following information provides some of the computer installation details needed to install and run the CRASH2 program.

3.2.1 Modes of Operation

The program may be run in either interactive or batch processing mode. In the interactive mode, a set of questions is posed to the user at a computer terminal, answers are keyed in and analyzed for correctness, and the analysis results are displayed at the user's terminal. In the batch processing mode, the user may keypunch or edit a set of strictly formatted data cards which may be submitted to the CRASH2 program as a background batch job. The program produces the results on the high speed printer.

3.2.2 Source Language

The CRASH2 program was developed in IBM FORTRAN IV, G-Level compiler language. CRASH2 is quite close to ANS FORTRAN, but certain features such as enclosing literals within apostrophes, dimensioning within a type specification depart from the traditional ANS FORTRAN construct. This is no hindrance since most versions of FORTRAN IV allow usage of these techniques. CRASH2 can also be compiled on the IBM FORTRAN IV H-Level optimizing compiler which will reduce run costs at the penalty of increased storage requirements.

3.2.3 Input-Output Units

Three input/output units must be defined for the CRASH2 program. They are summarized in Table 1.

3.2.4 Storage Requirements

CRASH2 is a very large program, almost five boxes of computer cards. While the analytical techniques are simple, the "idiot-proofing" accounts for most of the code. Also, the timesharing and batch processing versions are merged to give the advantage of one single program to install and maintain.

As a result, the CRASH2 program requires 171 K bytes of storage on an IBM System/370 computer. This is not an unreasonable size for a batch processing application of the CRASH2 program; but on the Calspan TSO timesharing service, the 171 K byte load module will not fit in a 256 K byte timesharing region. The only recourse is to overlay the program, which reduces it's effective size to 99 K bytes. Figure 3 is a layout of the CRASH program in overlay and non-overlay setups. The overlay technique is ideal for this program since CRASH2 has no loops or extensive iterations that could contribute to disc swapping overhead expense.

Table 1 CRASH2 INPUT/OUTPUT REQUIREMENTS

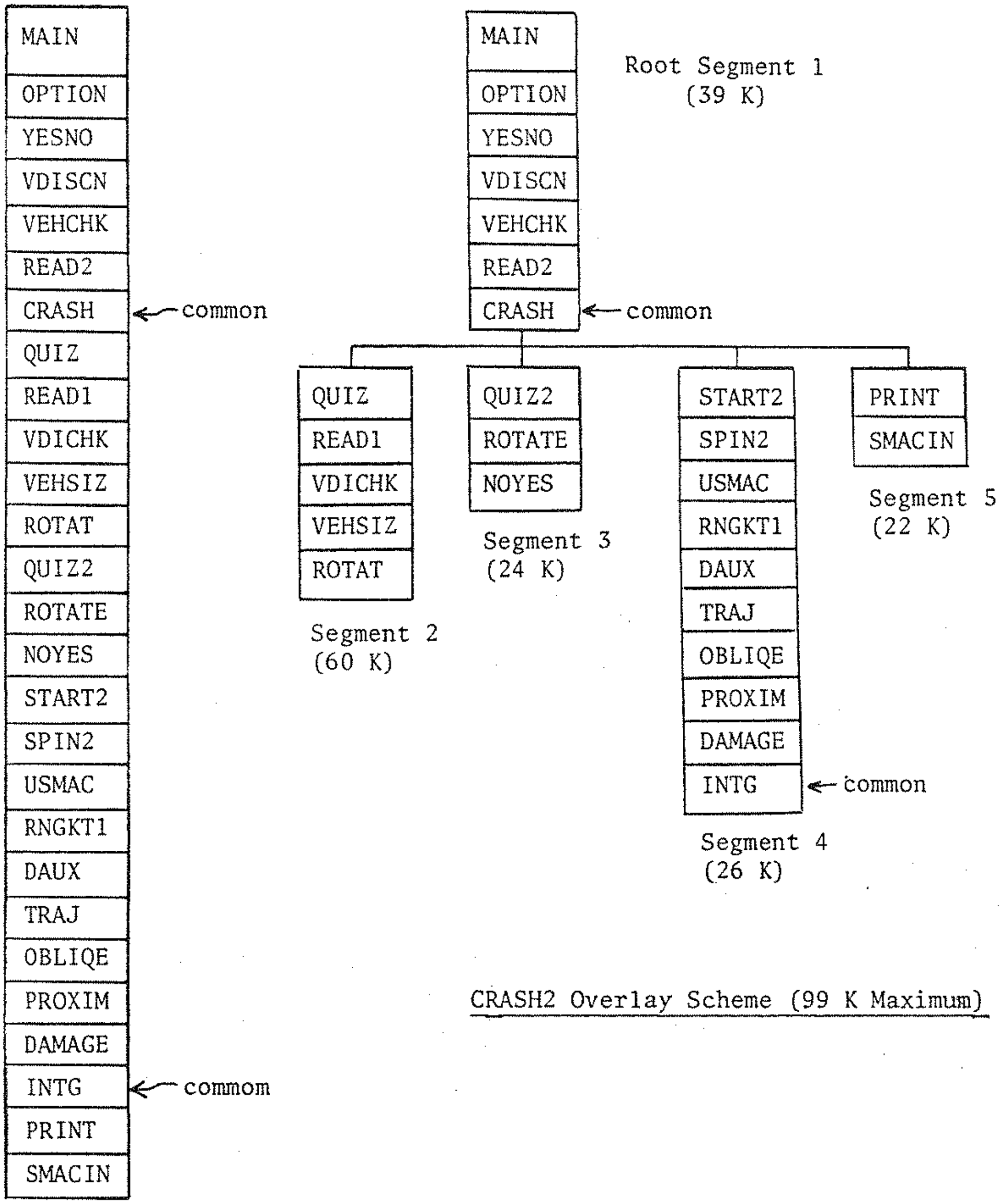
FORTRAN Input/Output Unit	Purpose	Record Length	Suggested Block Size	Equipment for Timesharing Usage	Equipment for Batch Processing
5	data input	80	3120	terminal	card reader file terminal
6	program output and messages	80	3120	terminal	printer file terminal
7	program output (SMAC data cards)	80	1600	terminal file card punch	card punch file printer terminal

Users trying to install the program on a computer with restricted region size, such as a minicomputer, might consider tailoring the program to their particular needs. For example, in the overlay setup segment 2 is for timesharing while segment 3 is for batch processing. By modifying the main program, either one of the segments can be dropped. The trajectory simulation capability can be removed by dropping the questions concerning it in Subroutine QUIZ, removing the appropriate subroutine calls to it in Subroutine SPIN2, and then removing the trajectory simulation routines USMAC, RNGKT1, DAUX, and TRAJ. Lastly, while it would require some work, Subroutine QUIZ which asks the idiot-proofed questions is virtually straight line programming and is thus a candidate for overlay management.

As an aid to those users attempting to create an IBM System/370 overlay load module, Figure 4 shows the job control language required. In the example of Figure 4, LINK is a cataloged procedure that executes the link editor program, the 'OVLY' term in the parameter list requests the overlay feature. The "INCLUDE LOADLIB (CRASH2)" statement identifies the module that has all the component parts while "MEMBER=CRASH2O" identifies the name of the final overlay module.

3.2.5 Compilation

Users are advised to compile the CRASH2 program in pieces - few computer systems offer a large enough region size to compile five boxes of FORTRAN cards in one pass. Calspan experience is that a box and one-half at a time is best, which is about the size of the QUIZ subroutine. This will require a region size of 800 K bytes on an IBM computer system.



CRASH2 Overlay Scheme (99 K Maximum)

CRASH2 NON-OVERLAY

Figure 3 LAYOUT OF CRASH2 PROGRAM (NON-OVERLAY AND OVERLAY METHODS)

```
//  EXEC  LINK,PARM.LKED='LIST,MAP,LET,ONLY',MEMBER=CRASH2O
//LKED.SYSIN  DD  *
  ENTRY MAIN
  INSERT MAIN,OPTION,YESNO,VDISCN,VEHCHK,READ2,CRASH
  OVERLAY ONE
  INSERT QUIZ,READ1,VDICHK,VEHSIZ,ROTAT
  OVERLAY ONE
  INSERT QUIZ2,ROTATE,NOYES
  OVERLAY ONE
  INSERT START2,SPIN2,USMAC,RNGKT1,DAUX,TRAJ,OBLIQE,PROXIM,DAMAGE,INTG
  OVERLAY ONE
  INSERT PRINT,SMACIN
  INCLUDE LOADLIB(CRASH2)
/*
```

Figure 4   JOB CONTROL LANGUAGE TO OVERLAY CRASH2 PROGRAM

3.2.6 Subroutines Required

CRASH2 consists of a main program, 24 subroutines, and 2 named commons. Excepting for the usual FORTRAN-supplied trignometric and mathematical function subroutines, no other special subroutines are required.

3.2.7 Executing The Program

At Calspan, we have an IBM System 370 TSO timesharing service. To run the interactive version of the CRASH2 program, a file called a "CLIST" must be set up. Figure 5 lists the "CLIST" used on Calspan's TSO system. The programmer is referred to the appropriate IBM TSO manuals. CRASH20 is the overlaid load module. While TSO automatically handles FORTRAN I/O units 5 and 6, the I/O unit #7 must be erased, deallocated, and redefined each time the program is run.

With a functioning command procedure stored in the command procedure library, all the CRASH2 user has to do to run the program is type:

```
EXEC  (CRASH2) (CR)
```

To run CRASH2 in the batch processing environment, a simple "LOADGO" procedure will suffice. Figure 6 shows the makeup of a CRASH2 batch processing deck.

```
00010 TERMINAL NOLINES LINESIZE(80)
00020 DELETE 'LCJY.SMACDATA.DATA'
00030 FREI DA('LCJY.SMACDATA.DATA') ATTR(DCB1)
00040 ATTR DCB1 LRECL(80) BLKSIZE(1600) RECFM(F B)
00050 ALLOC FILE(FT07F001) DA('LCJY.SMACDATA.DATA')
      NEW SPACE(1 1) TRACKS USING(DCB1)
00060 CALL 'LOADLIB(CRASH20)'
00070 END
```

NOTE: This file should be loaded into the command procedure library as member "CRASH2".

Figure 5 COMMAND PROCEDURE FOR INTERACTIVE CRASH2

```
//LCJYZIPF JOB (X,W01001,5386,52,JY),'JIM LYNCH'
// EXEC LOADGO,GCORE=350K
//GO.SYSLIN DD DSN=LOADLIB(JYCRASH2),DISP=SHR
//GO.SYSIN DD *
               ↑

CRASH2 BATCH CARDS GO HERE

               ↓
/*
```

Figure 6 JOB CONTROL LANGUAGE FOR CRASH2 BATCH PROCESSING

Load library module JYCRASH2 is the non-overlaid copy of the CRASH2 program. Calspan's LOADGO procedure includes automatically the proper definition of FORTRAN input/output units 5, 6 and 7.

3.3 Extension of Vehicle Categories

The total number of vehicle categories was extended from the original four (Reference 2) to seven, where one category is an SAE moving barrier (Category 7). The stored data table is presented in Table 2. Note that the values for crush properties (i.e., A, B, G for front, side and rear) are gross approximations which will be refined in the near future under Contract No. DOT-HS-6-01372.

Table 2
VEHICLE CATEGORIES

		1	2	3	4	5	6	7	8	UNITS
		MINICAR	SUBCOMPACT	COMPACT	INTERMEDIATE	FULL SIZE	LARGE	RIGID	BARRIER	
a		45.1	46.3	51.3	54.7	58.1	60.1	54.0	50.0	INCHES
b		48.1	50.1	55.5	59.2	63.0	65.1	66.0	50.0	INCHES
M		5.70*	7.90*	9.18*	10.99*	12.59*	13.74*	10.35	10^6	LB SEC^2/IN.
RSQ		2006.	2951.	3324.	3741.	4040.	4229.	4029.	10^6	IN^2
X_F		76.0	83.3	89.8	98.8	101.8	104.2	84.0	50.0	INCHES
X_R		-83.8	-91.6	-106.4	-114.0	-121.9	-125.2	-96.0	-50.0	INCHES
Y_s		30.4	33.6	36.3	38.5	39.9	39.9	39.0	50.0	INCHES
T		51.1	54.6	58.9	61.8	63.7	63.6	60.0	50.0	INCHES
F	A	130.5	130.5	154.6	281.8	307.5	307.5	0.0	0.0	LB/IN.
	B	58.72	58.72	69.57	33.82	36.89	36.89	0.0	0.0	LB/IN^2
	G	144.94	144.94	171.78	1174.3	1281.1	1281.1	0.0	0.0	LB
R, L	A	82.21	82.21	111.8	43.72	49.19	49.19	0.0	0.0	LB/IN.
	B	42.76	42.76	58.16	47.23	53.13	53.13	0.0	0.0	LB/IN^2
	G	79.04	79.04	107.5	20.24	22.77	22.77	0.0	0.0	LB
B	A	65.98	65.98	78.18	85.51	93.28	93.28	0.0	0.0	LB/IN.
	B	13.20	13.20	15.64	17.11	18.66	18.66	0.0	0.0	LB/IN^2
	G	164.97	164.97	195.45	213.78	233.21	233.21	0.0	0.0	LB
WB		93.2	96.4	106.8	113.9	121.1	125.2	120.0	–	INCHES
	CURB WGT	1902.	2753.	3247.	3947.	4565.	5009.	4000.	–	LBS
TRACK		51.1	54.6	58.9	61.8	63.7	63.6	60.0	–	INCHES
LENGTH		159.8	174.9	196.2	212.8	223.7	229.4	180.0	–	INCHES
WIDTH		60.8	67.2	72.6	77.0	79.8	79.8	78.0	–	INCHES

*NOTE THAT M VALUES INCLUDE ALLOWANCES FOR 2 PASSENGER LOADING.

$g = 356.4$ in/sec^2

3.4 Centroid of Damage Region

(1) The values of C_i must be entered proceeding in the positive direction of the vehicle coordinate system (i.e., from left to right across an end and from rear to front along a side).

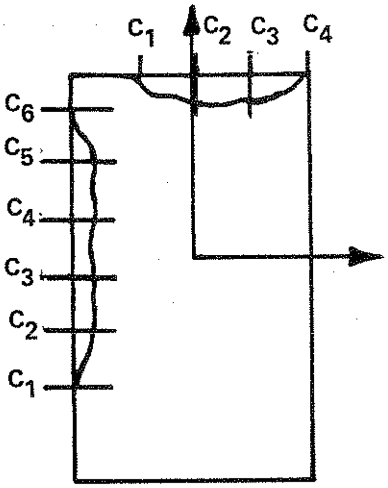


Figure 7 ORDER OF EXTENT ENTRIES

(2) Extent Adjustment (2 pt., 4 pt., 6 pt.)

Taking integration strips parallel to the direction of the C_i dimensions as elements of area and the following form of functional relationship between extent (c) and width (ℓ),

$$c = C_1 + (C_2 - C_1)\frac{\ell}{L} \quad \text{(2 point definition of damage profile),} \qquad (1)$$

the distance from the original side or end surface to the centroid of the damaged area is obtained from:

$$\text{TEMP} = \frac{\int_0^L \left(\frac{1}{2} c\right)(c\,d\ell)}{\int_0^L c\,d\ell} \tag{2}$$

Application of equation (2) to two, four and six point definition of the damage profile yields:

$$\text{TEMP} = \frac{(C_1^2 + C_1C_2 + C_2^2)}{3\ (C_1 + C_2)}$$

$$\text{TEMP} = \frac{(C_1^2 + 2C_2^2 + 2C_3^2 + C_4^2 + C_1C_2 + C_2C_3 + C_3C_4)}{3\ (C_1 + 2C_2 + 2C_3 + C_4)}$$

$$\text{TEMP} = \frac{(C_1^2 + 2C_2^2 + 2C_3^2 + 2C_4^2 + 2C_5^2 + C_6^2 + C_1C_2 + C_2C_3 + C_3C_4 + C_4C_5 + C_5C_6)}{3\ (C_1 + 2C_2 + 2C_3 + 2C_4 + 2C_5 + C_6)}$$

SIDE CONTACT

TEMP1 = YS-TEMP

END CONTACT

TEMP1 = XF-TEMP

OR

TEMP1 = -XR-TEMP

(3) Midpoint Adjustment (2 pt., 4 pt., 6 pt.)

Again taking integration strips parallel to the direction of the C_i dimensions as elements of area and the same form of functional relationship between extent (c) and width (ℓ) (i.e., equation 1) the distance from the midpoint of the damaged area, taken parallel to the original side or end surface, is obtained from:

$$\text{TEMP 2} = \frac{\int_0^L c\ (\ell - \frac{L}{2})\ d\ell}{\int_0^L c d\ell} \tag{3}$$

Application of equation (3) to two, four and six point definitions of the damage profile yields:

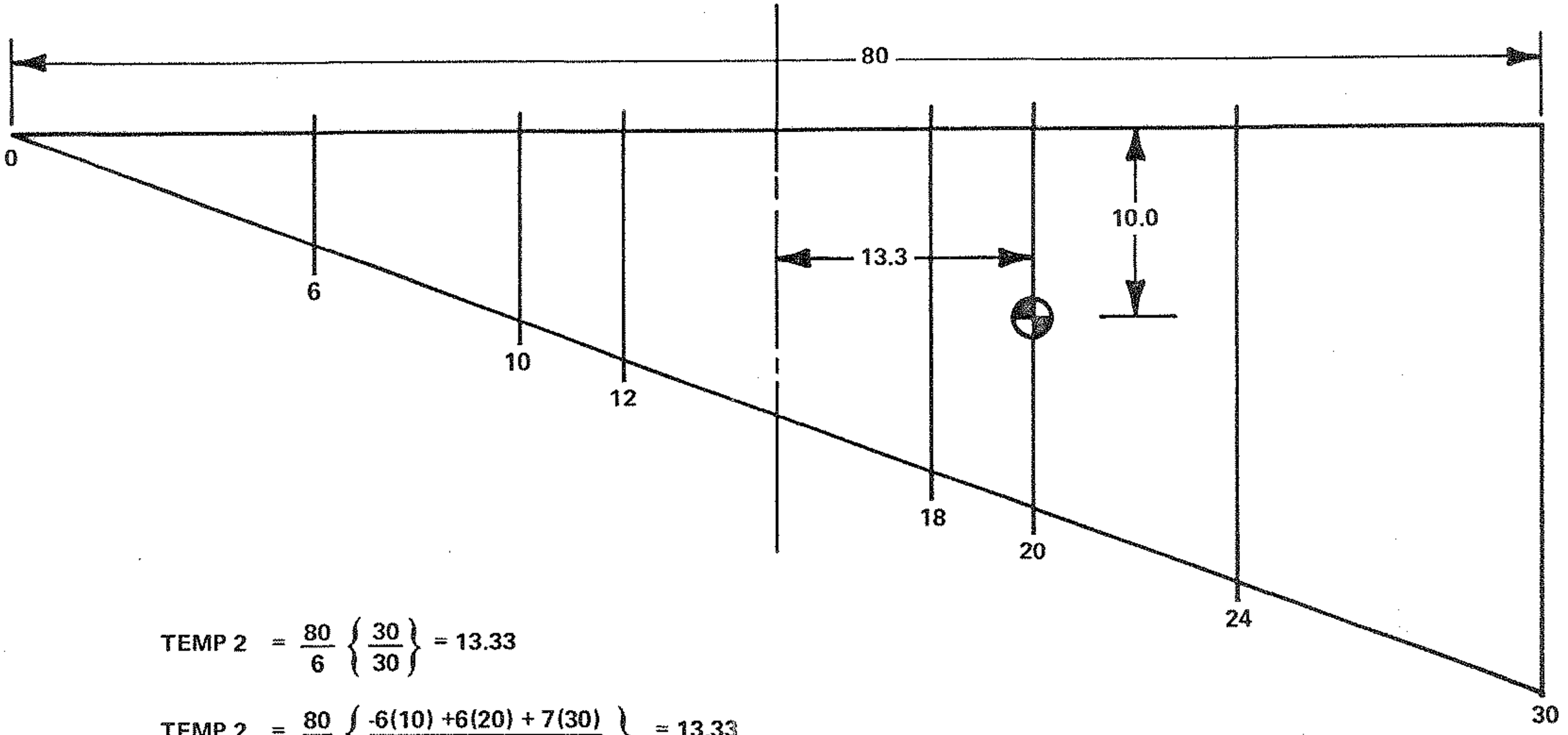
$$\text{TEMP2} = \frac{L}{6} \left\{ \frac{C_2 - C_1}{C_1 + C_2} \right\}$$

$$\text{TEMP2} = \frac{L}{18} \left\{ \frac{-7C_1 - 6C_2 + 6C_3 + 7C_4}{C_1 + 2C_2 + 2C_3 + C_4} \right\}$$

$$\text{TEMP2} = \frac{L}{30} \left\{ \frac{-13C_1 - 18C_2 - 6C_3 + 6C_4 + 18C_5 + 13C_6}{C_1 + 2C_2 + 2C_3 + 2C_4 + 2C_5 + C_6} \right\}$$

$$D = D + \text{TEMP2}$$

A sample application is depicted in Figure 8.



$$\text{TEMP 2} = \frac{80}{6}\left\{\frac{30}{30}\right\} = 13.33$$

$$\text{TEMP 2} = \frac{80}{18}\left\{\frac{-6(10)+6(20)+7(30)}{2(10)+2(20)+30}\right\} = 13.33$$

$$\text{TEMP 2} = \frac{80}{30}\left\{\frac{-18(6)-6(12)+6(18)+18(24)+13(30)}{2(6)+2(12)+2(18)+2(24)+30}\right\} = 13.33$$

$$\text{TEMP} = \frac{30^2}{90} = 10.0$$

$$\text{TEMP} = \frac{2700}{270} = 10.0$$

$$\text{TEMP} = \frac{4500}{450} = 10.0$$

Figure 8 SAMPLE CENTROID CALCULATION

4. REFERENCES

1. McHenry, Raymond R., "The CRASH Program - A Simplified Collision Reconstruction Program," Proceedings of the Motor Vehicle Collision Investigation Symposium, Calspan Corporation, October 6-10, 1975.

2. McHenry, Raymond R., "User's Manual for the CRASH Computer Program," Calspan Report No. ZQ-5708-V-3, January 1976.