Spring 1981

Digital Acoustic Tracking Analysis Program

George H. Ford
University of Central Florida

DIGITAL ACOUSTIC TRACKING ANALYSIS PROGRAM

BY

GEORGE H. FORD, JR.
B.S.E.E., University of Florida, 1976

RESEARCH REPORT

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering
in the Graduate Studies Program
of the College of Engineering
at the University of Central Florida;
Orlando, Florida

Winter Quarter
1981

DIGITAL ACOUSTIC TRACKING ANALYSIS PROGRAM

BY

George H. Ford, Jr.

ABSTRACT

The purpose of this report is to investigate the processing of tracking data for acoustic targets. The programs developed for two- and three-dimensional space calculate the target's position via "hyperbolic-fix" navigation (geometric) considerations using the Newton-Raphson algorithm.

The computer programs and the tracking solution approach contained herein is based on knowledge of only the sensors' locations and the relative time-difference at which a target's referenced, singular, acoustic pressure wavefronts are received at the sensors.

Omnidirectional sensors are found to be sufficient for the two-dimensional space tracking problem. However, it is found that the three-space problem requires usage of directional frequency and ranging (DIFAR) sensors.

Line printer plots are provided for the target position solutions; also, tabular track position solutions are provided.

Dr. Fred O. Simons, Jr.
Director of Research Report

ACKNOWLEDGEMENTS

Appreciation and gratitude are expressed to my wife for her support and encouragement; to my chairman Dr. Fred Simons for his support and encouragement; to the Naval Training Equipment Center for part-time use of its Harris Datacraft 6024/4 computer; and, to Mr. George T. Kirby for his encouragement and allowance for pursuit of this project while under his supervision at the Naval Training Equipment Center, Code N-4133, Orlando, Florida.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii

LIST OF FIGURES vi

PROBLEM STATEMENT vii

I. BACKGROUND INVOLVEMENT: ASW 1

II. PROBLEM SCENARIO 2

III. REAL WORLD TRACKING PROBLEM 13

IV. TWO SPACE PROBLEM 16

V. THREE SPACE PROBLEM: METHOD ONE 26

VI. THREE SPACE PROBLEM: METHOD TWO 27

VII. SIMULATION LIMITATIONS 37

VIII. SUMMARY AND CONCLUSIONS 40

APPENDIX A. TWO SPACE DATA SIMULATION PROGRAM . 44

APPENDIX B. TWO SPACE PROGRAM 53

APPENDIX C. TWO SPACE SIMULATION DATA 78

APPENDIX D. TWO SPACE COMPUTER DATA SOLUTION . 81

APPENDIX E. THREE SPACE: METHOD ONE: PROGRAM . 92

APPENDIX F. THREE SPACE SIMULATION DATA . . . 118

APPENDIX G. THREE SPACE: METHOD TWO: SIMULATION DATA PROGRAM 123

APPENDIX H. THREE SPACE: METHOD TWO: PLANAR PROGRAM 133

APPENDIX I. THREE SPACE: METHOD TWO: PLANAR SIMULATION DATA 162

APPENDIX J. THREE SPACE: METHOD TWO: TRACKING SOLUTION 167

APPENDIX K. COMPUTER OBJECT-CODE RUN-TIME PROGRAMS 180

SELECTED BIBLIOGRAPHY 182

LIST OF FIGURES

1. SCENARIO EXAMPLE: WATER 3

2. WAVEFRONT SPHERICAL SPREADING 5

3. ANALYSIS EQUATIONS 7

4. HYPERBOLIC CURVE SOLUTIONS 11

5. SUBROUTINE FLOWCHART: DATA SIMULATION 17

6. EXAMPLE OF TWO SPACE DATA GENERATION INPUTS 18

7. EXAMPLE OF TWO SPACE DATA STORAGE IN COMPUTER FILE . 19

8. SUBROUTINE FLOWCHART: TRACKING PROBLEM 20

9. EXAMPLE OF TWO SPACE COMPUTER PRINTOUT SOLUTION . . . 23

10. EXAMPLE OF TWO SPACE COMPUTER PLOT SOLUTION 24

11. EXAMPLE OF TWO SPACE COMPUTER PLOT SOLUTION 25

12. EXAMPLE OF THREE SPACE DATA GENERATION INPUTS 28

13. EXAMPLE OF THREE SPACE DATA STORAGE IN COMPUTER FILE 29

14. EXAMPLE OF THREE SPACE COMPUTER PRINTOUT SOLUTION . . 30

15. EXAMPLE OF THREE SPACE COMPUTER PRINTOUT SOLUTION . . 31

16. EXAMPLE OF THREE SPACE COMPUTER PRINTOUT SOLUTION . . 32

17. EXAMPLE OF THREE SPACE COMPUTER PRINTOUT SOLUTION . . 33

18. EXAMPLE OF THREE SPACE COMPUTER PRINTOUT SOLUTION . . 34

19. EXAMPLE OF THREE SPACE COMPUTER PRINTOUT SOLUTION . . 35

20. EXAMPLE OF THREE SPACE COMPUTER PRINTOUT SOLUTION . . 36

21. SCHEME FOR WAVEFRONT PHASE DETECTION FOR TRACKING . . 41

PROBLEM STATEMENT: Design an acoustic tracking program for a digital computer solution with one target (source) and three sensors in two and three dimensional space. The solution is to be implemented given only time (phase) delays of target to sensor wavefront propagation, known sensor locations, and known speed of sound for the medium. The Newton-Raphson algorithm is to be implemented in the computer solution.

SECTION I

BACKGROUND INVOLVEMENT: ASW

The author's interest in this research topic stems from work experience as a civilian engineer in the Department of the Navy in Anti-Submarine Warfare (ASW) acoustics studies. Utilizing knowledge from formal education and work experience, the author's interest is to pursue the problem statement topic and interject insight into the research report for use by other interested persons.

The computer programs designed and implemented are original; however, the concepts for the tracking solution are the same as commonly used in navigation today. The foundation for this historic problem is discussed in Section III of this report. All references, examples, illustrations, and scenario examples discussed in this report are purely academic in their presentation and analysis. No material of a sensitive nature has been drawn upon for this report (all references are available to the public); and, to the best of the author's knowledge no sensitive methodology, processing, or other techniques are used in this report. Any similarity to processing techniques in actual use which might be used in the systems noted in Section III, or any other systems, is coincidental.

SECTION II

PROBLEM SCENARIO

An acoustic tracking program has many practical applications: expansion to radar telemetry problems; home and business security monitoring; proximity detection systems; sonar data processing; anti-submarine warfare; weapons test range tracking; etc. In these and other similar instances it is desirable to gather as little information as possible about the target yet retaining the capability of tracking the object. It is also desirable to present such tracking data in a way that is readily discernible by a human observer. For example, along with various acoustic data, it would be useful to present mapped information of the target's trajectory, sensor locations, and locations of any other environmental landmarks.

Figure 1 shows an example scenario. The medium is water, the sensors are sonobuoys, and the target is an active noise emission torpedo. What is required to track the target are the (X,Y,Z) coordinates of the sensors (as referenced in Figure 1), the velocity of sound in water (assumed constant), and the time differences or phase delays with which the active emission reaches the sensors. It is assumed that the sensors provide coherent detection giving rise to the time difference with which a wavefront arrives from the target source to any

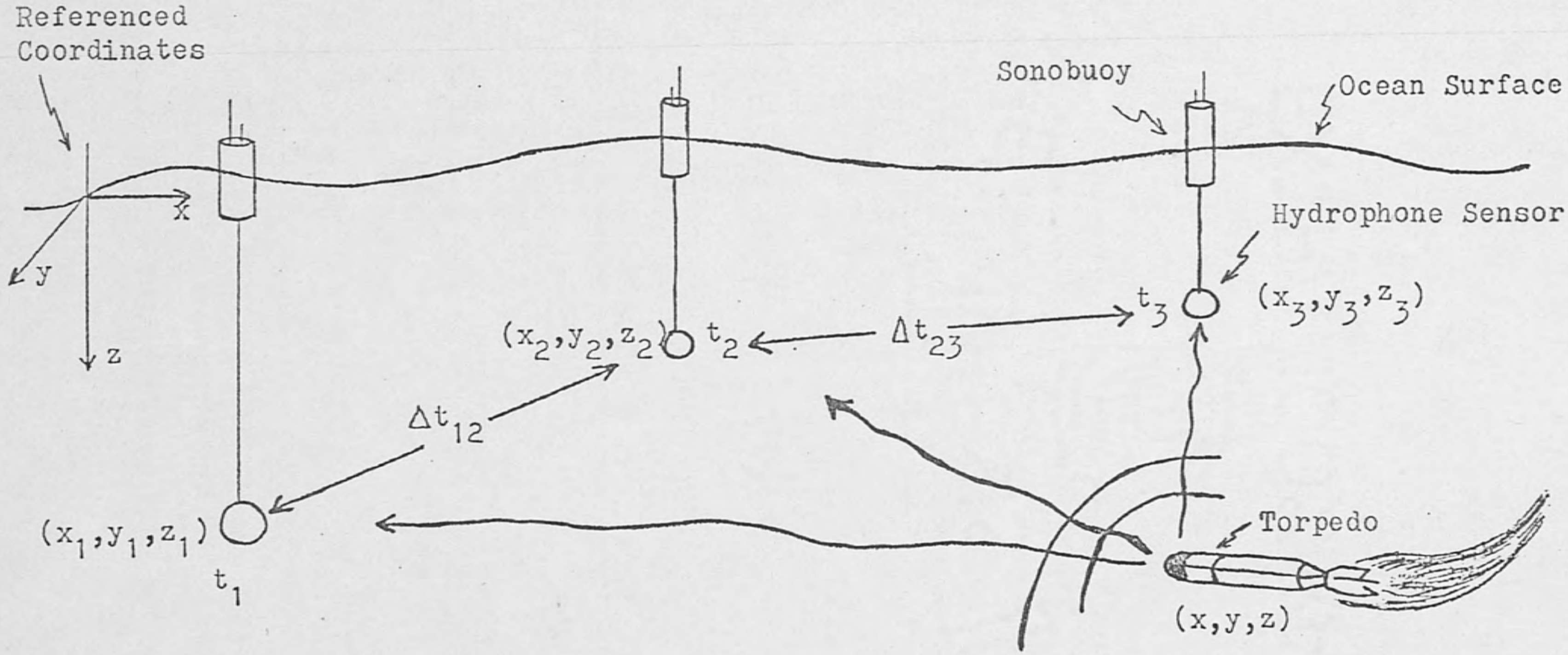


FIGURE 1. SCENARIO EXAMPLE: WATER

pair of sensor receivers; or, the sensor data has been preprocessed to yield time-difference data. The variable Δt_{12} ($=t_1-t_2$=DTT12) is used for the time difference between sensors one and two with which the wavefront is received. The sensors will initially be assumed to be omni-directional, that is, the direction with which the source's acoustic signal wavefront approaches the sensors will initially be assumed to be negligible with respect to requirements of known data necessary for the tracking solution.

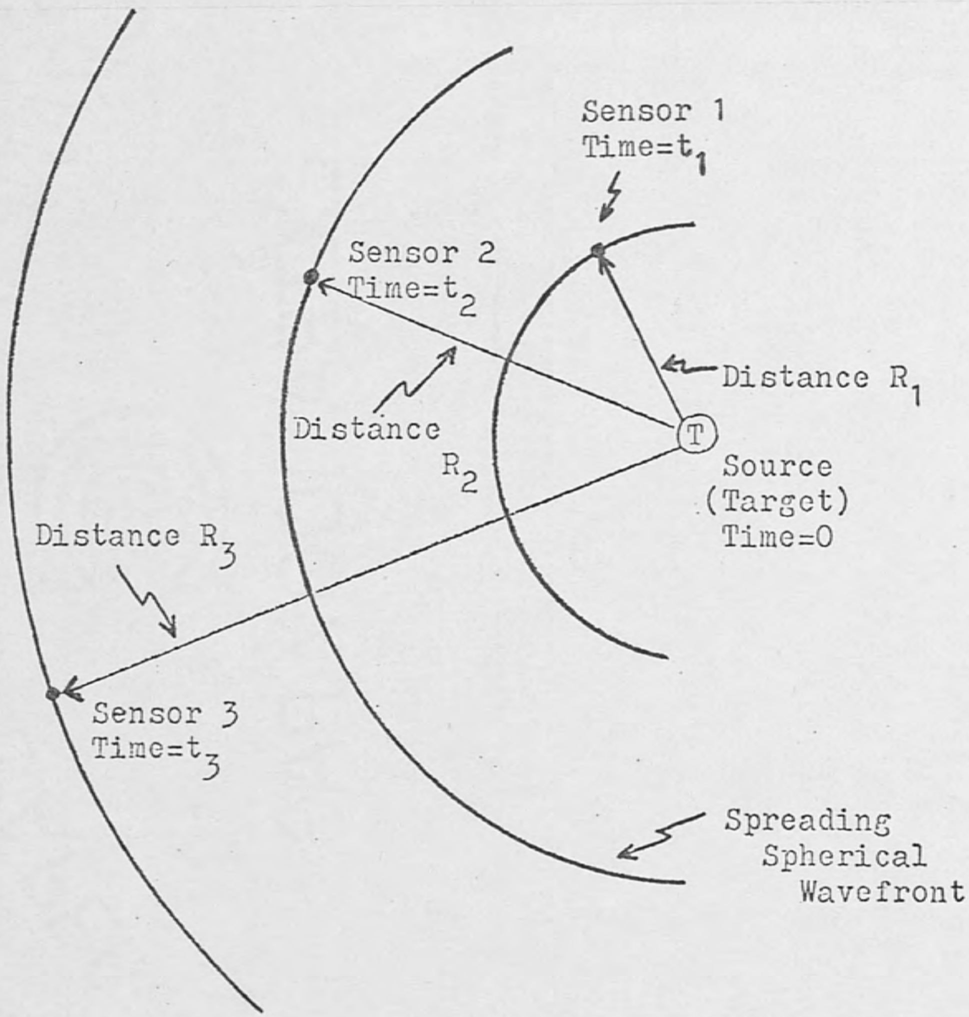
In order to see how it is possible to solve for the unknown (X,Y,Z) coordinates of the target see Figure 2. The wavefront for a given frequency which is emitting from the target is assumed to have the properties of spherical spreading. This is consistent with sonar acoustic theory.[1] In addition, the following assumptions are made: no propagation loss with range of the emission; constant acoustic velocity characteristics for the medium; and, no multi-path,[2] convergence zone,[3] nor Lloyd's Mirror (Bathtub) effect. As seen in Figure 2, the spherical spreading of the wavefront arrives at the three

---

[1] Robert J. Urick, Principles of Underwater Sound for Engineers (New York: McGraw-Hill Book Co., 1967), pp. 83, 92-93.

[2] Ivan Tolstoy and C.S. Clay, Ocean Acoustics: Theory and Experiment in Underwater Sound (New York: McGraw-Hill Book Co., 1966), pp. 85, 104-105.

[3] Ibid., pp. 145,149.



(1) $R = Vt$
(2) $\Delta R = V \Delta t$

(3) $\Delta R_{12} = V \Delta t_{12}$

where, $\Delta R_{12} = R_1 - R_2$

and $\Delta t_{12} = t_1 - t_2$

V=Velocity of Sound (Medium)

FIGURE 2. WAVEFRONT SPHERICAL SPREADING

sensors at the respective times of t_1, t_2, and t_3. The distances of travel for the wavefront from the source to the sensors are proportional to these times and are denoted as R_1, R_2, and R_3. The basic equation is that $R=Vt$, where distance (R) is the range from a sensor to the target, the velocity (V) is the speed of sound for the medium, and the time (t) is the time at which the wavefront is received at the sensor (Figure 2, Equation (1)). As shown in Figure 2 (Equation (2)), implementing the necessary difference equation (holding velocity constant) yields the difference in times for which two sensors receive the wavefront. This time difference is thus directly proportional to the difference in the wavefront's distances to the target. The distance equations for three-space are provided in Figure 3 (Equations (B)). The unsubscripted coordinates are those of the source target. The application of these distance equations to the difference equation previously mentioned gives rise to the time lag equations of Figure 3 (Equations (C)). The time lag or time difference equation is of the form of the Equation C-5. These equations, in turn, can be solved simultaneously for the unknown coordinates of the source target by using the Newton-Raphson algorithm.[4]

---

[4]Francis Scheid, Numerical Analysis, Schaum's Outline Series in Mathematics (New York: McGraw-Hill Book Co., 1968), pp. 312, 327-328.

Location

(A-1) Sensor 1 (X_1, Y_1, Z_1)

(A-2) Sensor 2 (X_2, Y_2, Z_2)

(A-3) Sensor 3 (X_3, Y_3, Z_3)

Distance Equations

(B-1) $R_1=((X-X_1)^2+(Y-Y_1)^2+(Z-Z_1)^2)^{\frac{1}{2}}$

(B-2) $R_2=((X-X_2)^2+(Y-Y_2)^2+(Z-Z_2)^2)^{\frac{1}{2}}$

(B-3) $R_3=((X-X_3)^2+(Y-Y_3)^2+(Z-Z_3)^2)^{\frac{1}{2}}$

Time Lag (Delay) Equations

(C-1) $\Delta R=V\Delta t$

(C-2) $R_{12}= V \cdot DTT_{12}$

(C-3) $R_{13}= V \cdot DTT_{13}$

(C-4) $R_{23}= V \cdot DTT_{23}$

(C-5) $DTT_{12}=\frac{(R_1-R_2)}{V}$

Equations for Newton-Raphson Algorithm

(D-1) $R_{12}-V \cdot DTT_{12}=E(X,Y,Z)$

(D-2) $R_{13}-V \cdot DTT_{13}=F(X,Y,Z)$

(D-3) $R_{23}-V \cdot DTT_{23}=G(X,Y,Z)$

(D-4) $-E=\frac{\partial E}{\partial x} \cdot h+\frac{\partial E}{\partial y} \cdot k+\frac{\partial E}{\partial z} \cdot p$

(D-5) $-F=\frac{\partial F}{\partial x} \cdot h+\frac{\partial F}{\partial y} \cdot k+\frac{\partial F}{\partial z} \cdot p$

(D-6) $-G=\frac{\partial G}{\partial x} \cdot h+\frac{\partial G}{\partial y} \cdot k+\frac{\partial G}{\partial z} \cdot p$

FIGURE 3. ANALYSIS EQUATIONS

The set of Equations (D) of Figure 3 form the basis set of simultaneous difference distance/time equations (D-1, D-2, and D-3) to which the Newton-Raphson algorithm must be applied. The resultant equations which must be solved, using Cramer's Rule for simultaneous equations, are found in equations D-4 through D-6. The variables h, k, and p are increments related to the cordinates X, Y, and Z. These increments (which are the unknowns for these equations) cause the equations D-4 through D-6 to diverge or converge to an (X,Y,Z) solution point for the simultaneous equations D-1 through D-3. Programming value checks to the iterative variables h, k, and p determines whether or not a solution has been found within some predetermined tolerance.

The Newton-Raphson algorithm is an iterative technique for solving systems with unknowns governed by a set of simultaneous equations. The method is good for N unknowns in N equations (for N positive, integer) either linear or non-linear.[5] However, divergence in lieu of convergence to a solution is a setback of the technique, since it exhibits instability for convergence in some applications and requires a good initial guess for the unknown coordinate variables.

---

[5]Francis Scheid, Numerical Analysis, Schaum's Outline Series in Mathematics (New York: McGraw-Hill Book Co., 1968), p. 328.

That is to say, the initial coordinates guess must be within close proximity of the solution in order for the method to converge or yield a correct solution.[6] The partial derivatives linearize the non-linear hyperbolic-fix distance equations (B) (Fig. 3). The increments (h,k,p) are for the coordinates added to the old guess or solution for the next guess or solution. If convergence occurs, the increments will get smaller than some pre-set acceptable tolerance value; and, the unknown target coordinate solution will be found. The hyperbolic fix distance difference equations are non-linear since the time-lag equations (Figure 3, Equations (C)) squared twice (to get rid of the distance-difference square-root terms (Figure 3, Equations (B))) will demonstrate cross-multiplication of X, Y, and Z variables of orders greater than one.

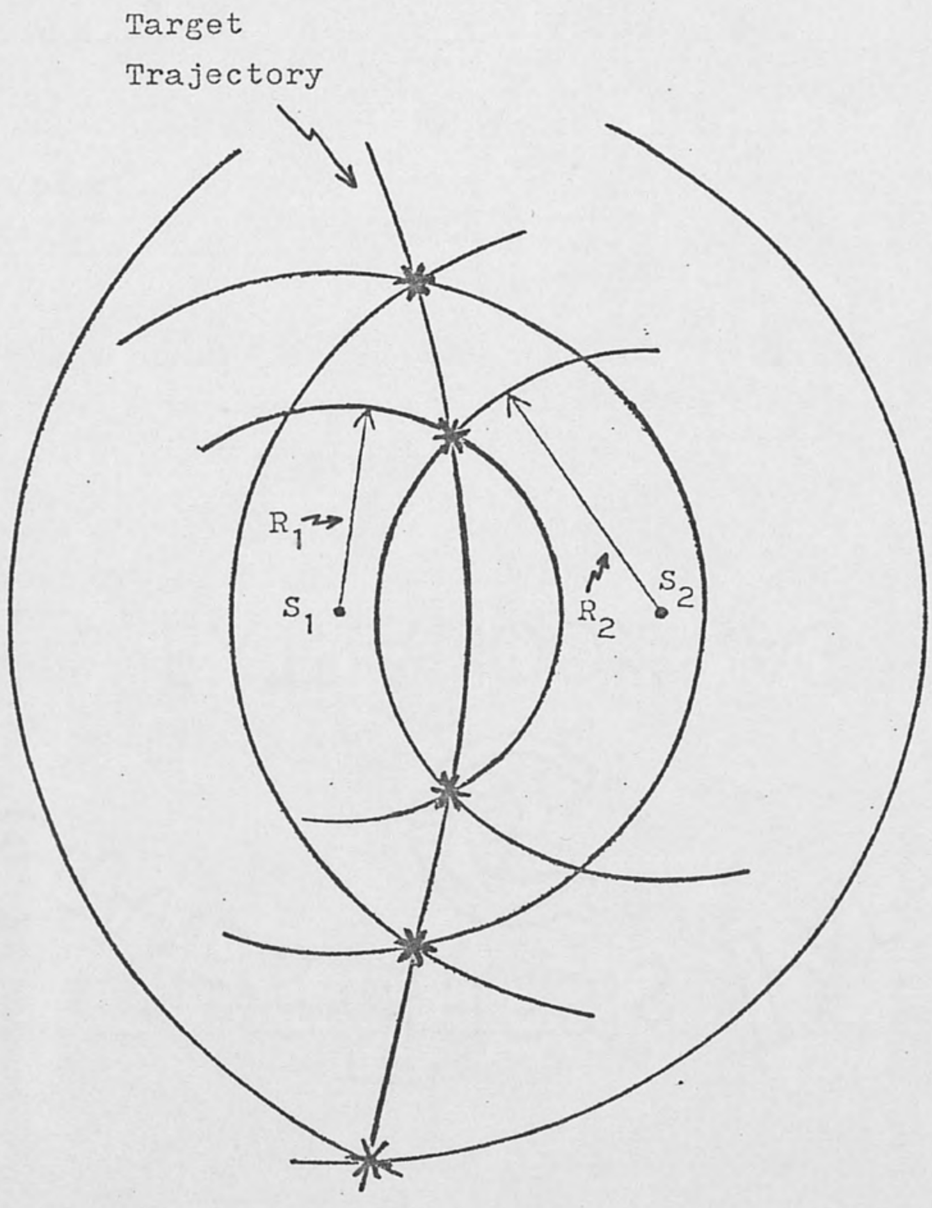
The two-space problem has the same set of equations and scenario, except that the Z variable coordinate is not used. The two-space problem still requires three sensors, since two distinct pairs of sensors are required for difference-in-distance and difference-in-time considerations to yield two simultaneous distance equations for the Newton-Raphson algorithm solution. The Newton-Raphson system

---

[6]Kaj L. Nielson, Methods in Numerical Analysis, 2nd ed. (New York: The MacMillan Co., 1967), pp. 209-210, 213-215 passim.

equations (Figure 3, Equations D-4 through D-6) reduce to two equations in two unknown variables.

In order to graphically visualize the solution (in the simpler two-space problem) see Figure 4. Here a single pair of sensors is analyzed. This analysis is in accordance with the time-difference equations of Figure 3 (Equations C-5 and D-1 through D-6). For the moment, consider the sensors in Figure 4 to be simultaneously emitting some tonal frequency. The wavefronts spread spherically (as mentioned previously) from the sensors in concentric circles. Note: only arcs of circles are shown for clarity. If the target is initially randomly placed between the two sensors, then the wavefronts will travel distances R_1 and R_2 in the times t_1 and t_2, respectively, to reach the target. If the relative difference between distances and between times is held constant, then the solution for the location of the target is found to be positioned on a curve of a hyperbola. This is shown in Figure 4, where the asterisks are the solution positions to the target for the various pairs of arcs emanating from sensors S_1 and S_2. Note that the hyperbolic solution curve is a trajectory of possible target positions for a given distance and time difference as given in Equations D-4 through D-6 (Figure 3). This navigation technique for locating a vessel or target, is referred to as "hyperbolic-fix" navigation. The sensors



$$\Delta R_{12} = R_1 - R_2$$
$$= \text{constant}$$

FIGURE 4. HYPERBOLIC CURVE SOLUTIONS

can be thought as emitting the wavefronts since the target/sensor relationship exhibits the duality principle of signal transmission and reception.[7]

The solution set for increasing radii is also a hyperbolic curve. This is seen by the form of the distance-difference equations of Figure 3 and plot of Figure 4. The two solution points for any given pair of radii (Figure 4) is resolved to one solution by the introduction of a third sensor to yield a second unique sensor pair, and thus a second unique set of distance- and time-differences. The two-space method of tracking a target has historic foundation in maritime hyperbolic-fix navigation techniques that will be discussed further in the next section.

The three-space problem has similar graphical analysis, yielding pairs of hyperboloids which intersect for the navigational fix solution. This is more difficult to show graphically and will be omitted in this report..

The above scenario (three sonobuoys and one target in a water medium), equations, assumptions, and techniques have been used to design the two- and three- space tracking programs for this research report.

---

[7]Robert J. Urick, Principles of Underwater Sound for Engineers (New York: McGraw-Hill Book Co., 1967), pp. 103-109 passim.

SECTION III

REAL WORLD TRACKING PROBLEM

This section will establish precedents for using hyperbolic-fix navigation systems in tracking targets. These systems have the property mentioned in the previous section that a constant distance-difference from two fixed sensors to the target is used to provide tracking navigation. The systems usually consist of two or more synchronized transmitters/receivers, at known locations. Airborne equipment usually consists of passive, receiving equipment.[8] The following is a brief description of all vehicle tracking systems known to use a variation of this type of navigation.

1. Loran A: This system operates by measurement of two range-differences. Coverage is to a range of about 1000 miles. The system requires at least three transmitters. Passive operation.
   Ambiguities - occur close to transmitting sites; geometric degeneration of accuracy along baseline extensions.
   This system is primarily for air navigation.[9]

2. Loran C: This system is a newer, improved version of Loran A (standard loran). Ships

---

[8] U.S., Department of the Navy, Navy Hydrographic Office, Air Navigation, Hydrographic Office Publication No. 216 (Washington, D.C.: Government Printing Office, 1967), p. 345.

[9] International Telephone and Telegraph Corp., Reference Data for Radio Engineers (Indianapolis: Howard W. Sams and Co., Inc., 1975), p. 32-7.

and aircraft utilize this system.[10] This system operates by measurement of two-range differences. Coverage is to a range of about 2000 miles. Passive operation.

Ambiguities - occur close to transmitting sites; geometric degeneration along base line extensions.[11]

3. Gee: This a British system similar to loran. It operates mainly in the VHF band, and hence reception is limited to line-of-sight. Four ground stations are used, 70 to 80 miles apart, transmitting accurately-timed, pulsed signals.[12]

4. Decca: This is a low-frequency, continuous wave, hyperbolic system. Navigation is accomplished by comparing the phase of signals from two or more stations, which transmit phase-synchronized signals.[13]

5. Lorac: This system is the United States equivalent of the Decca system.[14]

6. Sofar: This system depends upon transmission of sound in water. The initials stand for sound fixing and ranging. Time measuring equipment is used at the various synchronized stations, with time-differences being determined for navigation target

---

[10] U.S., Department of the Navy, Navy Hydrographic Office, Air Navigation, Hydrographic Office Publication No. 216 (Washington, D.C.: Government Printing Office, 1967), p. 361.

[11] International Telephone and Telegraph Corp., Reference Data for Radio Engineers (Indianapolis: Howard W. Sams and Co., Inc., 1975), p. 32-7.

[12] U.S., Department of the Navy, Navy Hydrographic Office, Air Navigation, Hydrographic Office Publication No. 216 (Washington, D.C.: Government Printing Office, 1967), p. 361.

[13] Ibid., p. 362.

[14] Ibid., p. 363.

tracking. The practical range is about 2000 to 3000 miles.[15]

7. Consol: This system operates by determining bearing angles from ground sites (three in-line antennas). Coverage is 1000 to 1500 miles. Two stations are required for a fix. Passive operation.
   Accuracy - a few nautical miles (depends upon range, diurnal effect, sea-land, and propagation).
   Ambiguities - radial bearing line is ambiguous from one angular sector to the next; geometric degeneration along the line of the three antennas (base line extensions).[16]

8. Consolan: This system is the United States version of consol. Greatest range is over sea water paths from 1000 to 1400 miles. The system is not usable within 50 miles of the stations. The greatest accuracy is obtained on a line normal to the line of stations and the useful portion of the pattern is the sector of 140 degrees broadside to the antennas. The pattern off the ends of the antennas, plus or minus 20 degrees, is considered useless for bearing information for tracking purposes.[17]

---

[15] U.S., Department of the Navy, Navy Hydrographic Office, Air Navigation, Hydrographic Office Publication No. 216 (Washington, D.C.: Government Printing Office, 1967), p. 363.

[16] International Telephone and Telegraph Corp., Reference Data for Radio Engineers (Indianapolis: Howard W. Sams and Co., Inc., 1975), p. 32-8.

[17] U.S., Department of the Navy, Navy Hydrographic Office, Air Navigation, Hydrographic Office Publication No. 216 (Washington, D.C.: Government Printing Office, 1967), p. 365.

SECTION IV

TWO SPACE PROBLEM

The two space problem is constrained to the XY-plane as previously viewed in Figure 1. The Z coordinate is not used in determining the tracking of the target, that is, the depth of the target is assumed constant or negligible.

The flowchart of Figure 5 is for the subroutines used in the Two/Three Space Data Simulation Program of Appendix A/G for the time-lag-difference inputs to the programs discussed in Section IV and Section V of this report. SIMDAT is the main routine, prompting target track, sound velocity, and sensor locations from the operator. PTCHK generates the time-difference (lag) data to be used in the tracking programs. DATXYZ stores the time-difference data and number of points with sensor locations on a disk file in the computer system for use by the tracking program. The Two Space data prompted by the routine SIMDAT can be seen in Figure 6. The resulting data stored by the routine DATXYZ is viewed in Figure 7.

The flowchart of Figure 8 is for the subroutines used in the Two/Three Space Tracking Program of Appendix B/E and H. This program uses the data in the disk file to locate and track the target trajectory. TRACK is the main program which

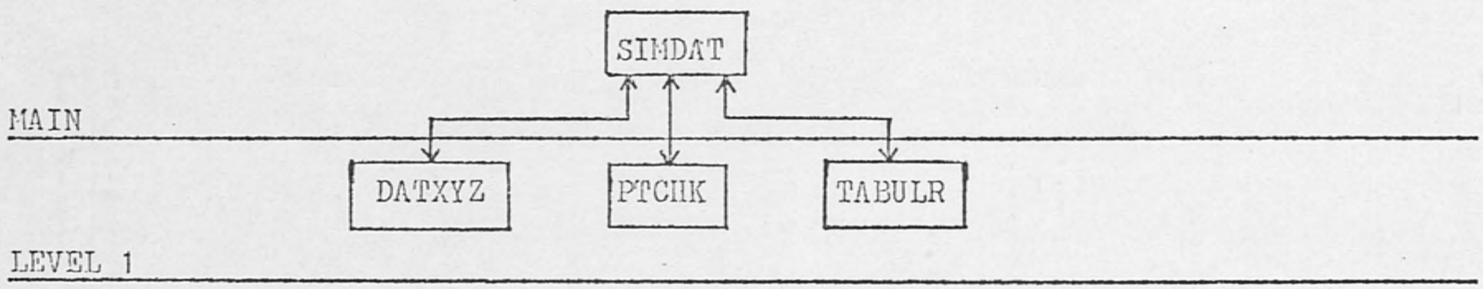


FIGURE 5. SUBROUTINE FLOWCHART: DATA SIMULATION

VELOCITY OF MEDIUM IS 1450.000 (M/SEC)

SENSOR POSITIONS

	X	Y	Z
1	75.000	100.000	0.000
2	175.000	75.000	0.000
3	200.000	200.000	0.000

VEHICLE (SOURCE) PATH

	X	Y	Z
1	80.000	10.000	0.000
2	100.000	40.000	0.000
3	110.000	70.000	0.000
4	110.000	100.000	0.000
5	100.000	120.000	0.000
6	80.000	140.000	0.000
7	90.000	165.000	0.000
8	110.000	175.000	0.000
9	130.000	180.000	0.000
10	155.000	170.000	0.000
11	175.000	150.000	0.000
12	200.000	135.000	0.000
13	225.000	150.000	0.000
14	239.000	175.000	0.000
15	250.000	210.000	0.000
16	239.000	239.000	0.000
17	250.000	245.000	0.000
18	265.000	255.000	0.000
19	280.000	245.000	0.000
20	310.000	235.000	0.000
21	340.000	225.000	0.000
22	365.000	220.000	0.000
23	375.000	210.000	0.000

FIGURE 6. EXAMPLE OF TWO SPACE DATA GENERATION INPUTS

25

-0.017220594790	-0.092816047347	-0.075595452558
-0.012251535714	-0.085296291476	-0.073044755762
-0.013168483976	-0.077252524636	-0.064084040660
-0.023890994059	-0.068645683083	-0.044754689024
-0.038240453266	-0.066239216249	-0.027998762983
-0.051584383166	-0.064726062005	-0.013141678839
-0.039369561179	-0.033603892059	0.005765669121
-0.025175088022	-0.007339987362	0.017835100660
-0.011830453375	0.016745751932	0.028576205307
0.006357944092	0.036012544135	0.029654600043
0.025381654396	0.038552896164	0.013171241767
0.044694861970	0.044694861970	0.000000000000
0.046879380084	0.070491161084	0.023611780999
0.042489118105	0.092421281794	0.049932163689
0.036045388945	0.107386255820	0.071340866882
0.026852376460	0.110225562490	0.083373186031
0.028591256405	0.110343616070	0.081752359672
0.030315778434	0.110384182670	0.080068404239
0.035369088664	0.109868821950	0.074499733292
0.042532683105	0.107298373000	0.064765689896
0.048283224324	0.103991089610	0.055707865294
0.051612916190	0.101820272510	0.050207356323
0.053953335126	0.099479603271	0.045526268145
0.056410658679	0.096891074063	0.040480415385
0.060020033422	0.092047979923	0.032027946502
75.0000	100.0000	0.0000
175.0000	75.0000	0.0000
200.0000	200.0000	0.0000

FIGURE 7. EXAMPLE OF TWO SPACE DATA STORAGE IN COMPUTER FILE

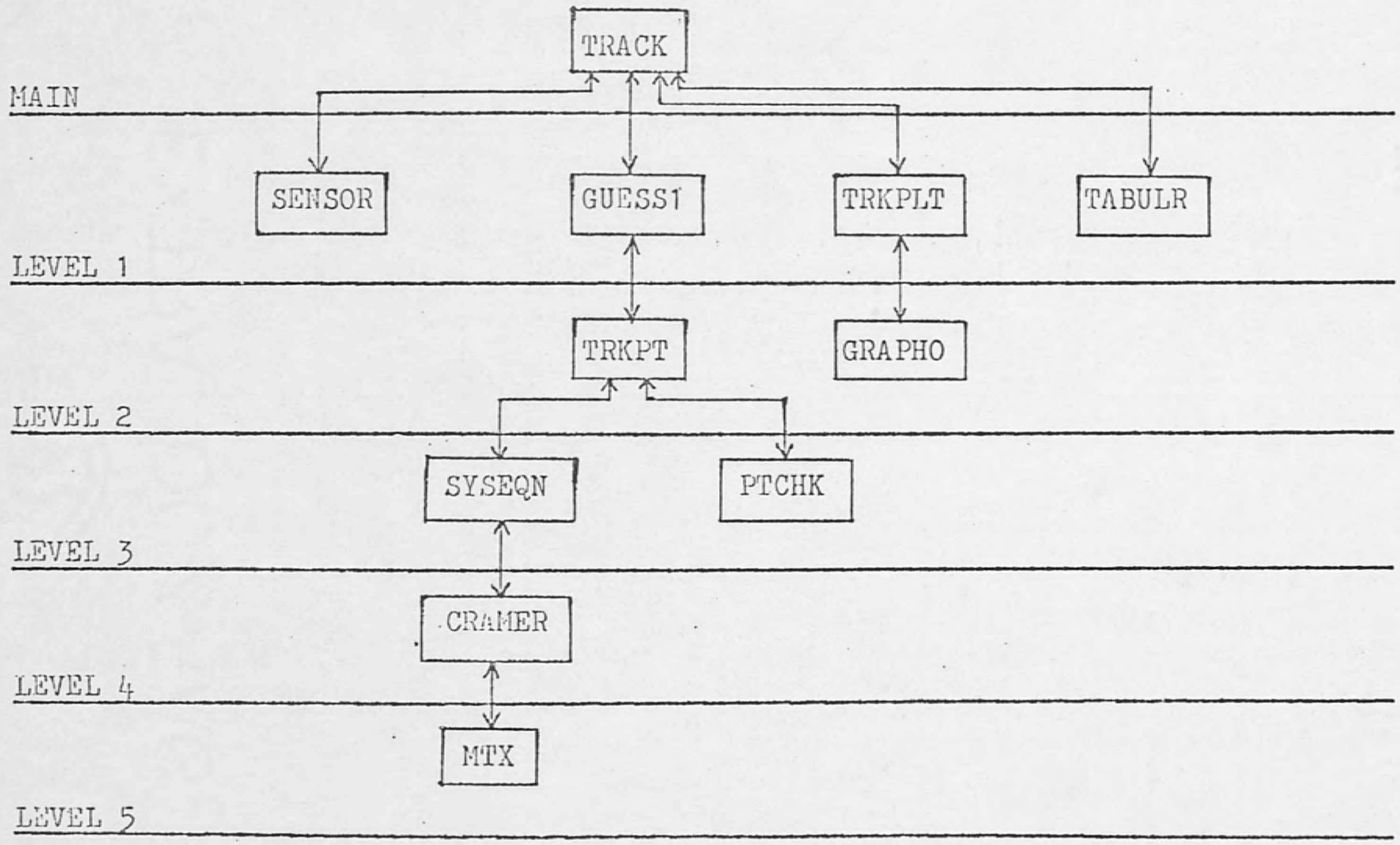


FIGURE 8. SUBROUTINE FLOWCHART: TRACKING PROBLEM

gathers the disk data and controls the solution algorithm, tabular display, and trajectory plot subroutines. SENSOR is a routine provided for future applications which would be designed to utilize real-time processing of actual data. GUESS1 provides decision processes to try to locate a good first target track point for the Newton-Raphson algorithm; and, as a last resort asks the operator for an initial guess for the first track point approximation. TRKPT is the subroutine which continues searching the next trajectory locations for a specified number of track points utilizing the last and next-to-last solutions as initial guesses for the present unknown track point location. SYSEQN is called by the subroutine TRKPT to solve the Newton-Raphson algorithm for the distance-difference system equations to within a specified solution tolerance. SYSEQN calls subroutine CRAMER to solve for the X, Y, and Z increments using Cramer's Rule. Subroutine MTX solves the determinants of the matrices required by the Cramer's Rule procedure. The subroutine PTCHK verifies that the solution found is indeed the solution to the unknown track point in question by use of the distance-difference equations (Figure 3, Equations (C)) satisfaction by the assumed solution point.

After all points of the trajectory have been solved, TABULR gives a tabular listing of the velocity of the medium, sensor locations, and the trajectory point locations which have been found via the data processing described above (see Figure 9). TRKPLT generates track plot data in arrays to be plotted out to the computer line printer via subroutine GRAPHO. A sample computer solution to a two-space data trajectory can be seen in Figure 10 and Figure 11. The plots show sensors numbered one, two, and three; and, the target locations as "T". Additional exhibits can be viewed in Appendix D.

VELOCITY OF MEDIUM IS 1450.000 (M/SEC)

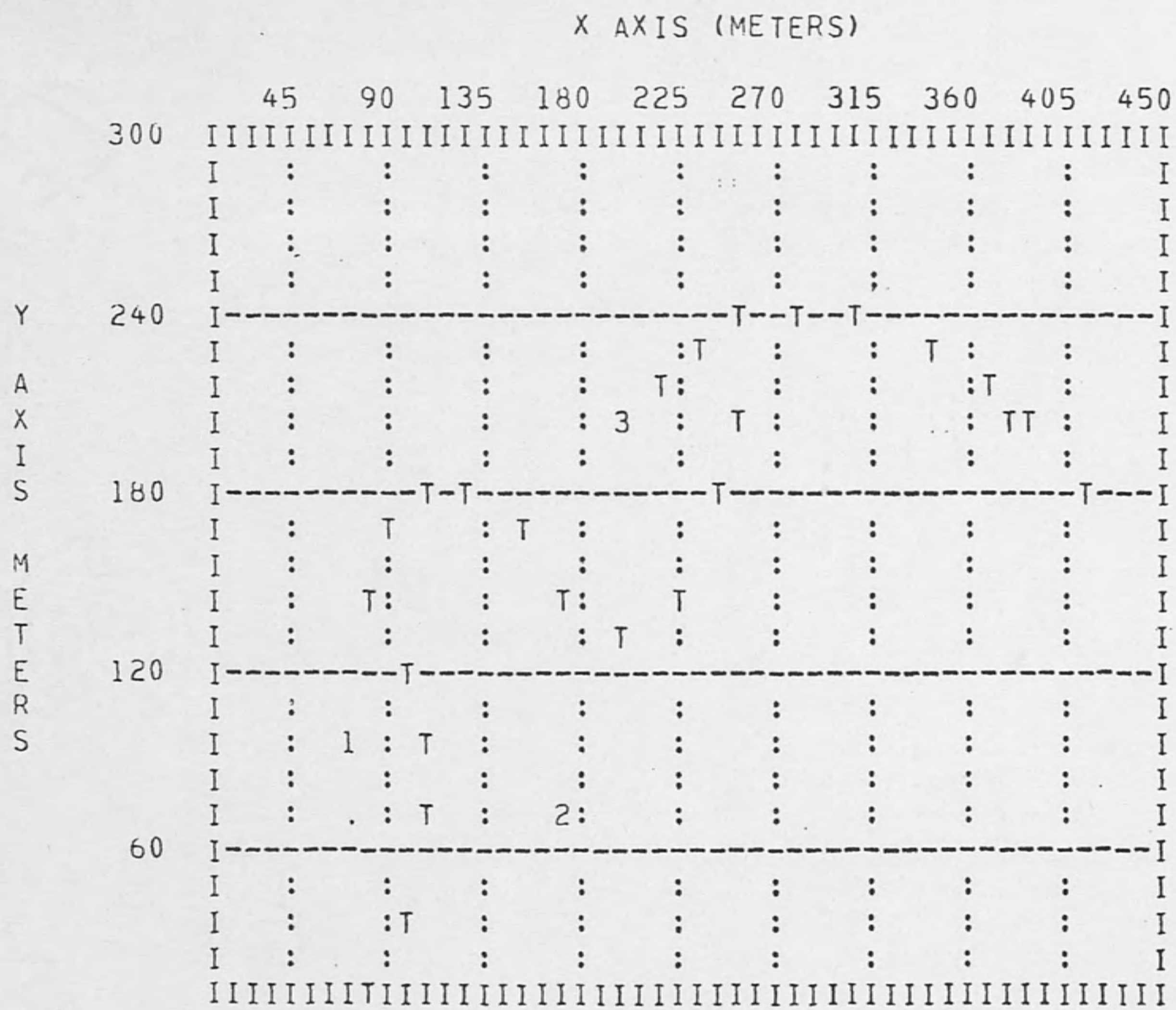
SENSOR POSITIONS

	X	Y	Z
1	75.000	100.000	0.000
2	175.000	75.000	0.000
3	200.000	200.000	0.000

VEHICLE (SOURCE) PATH

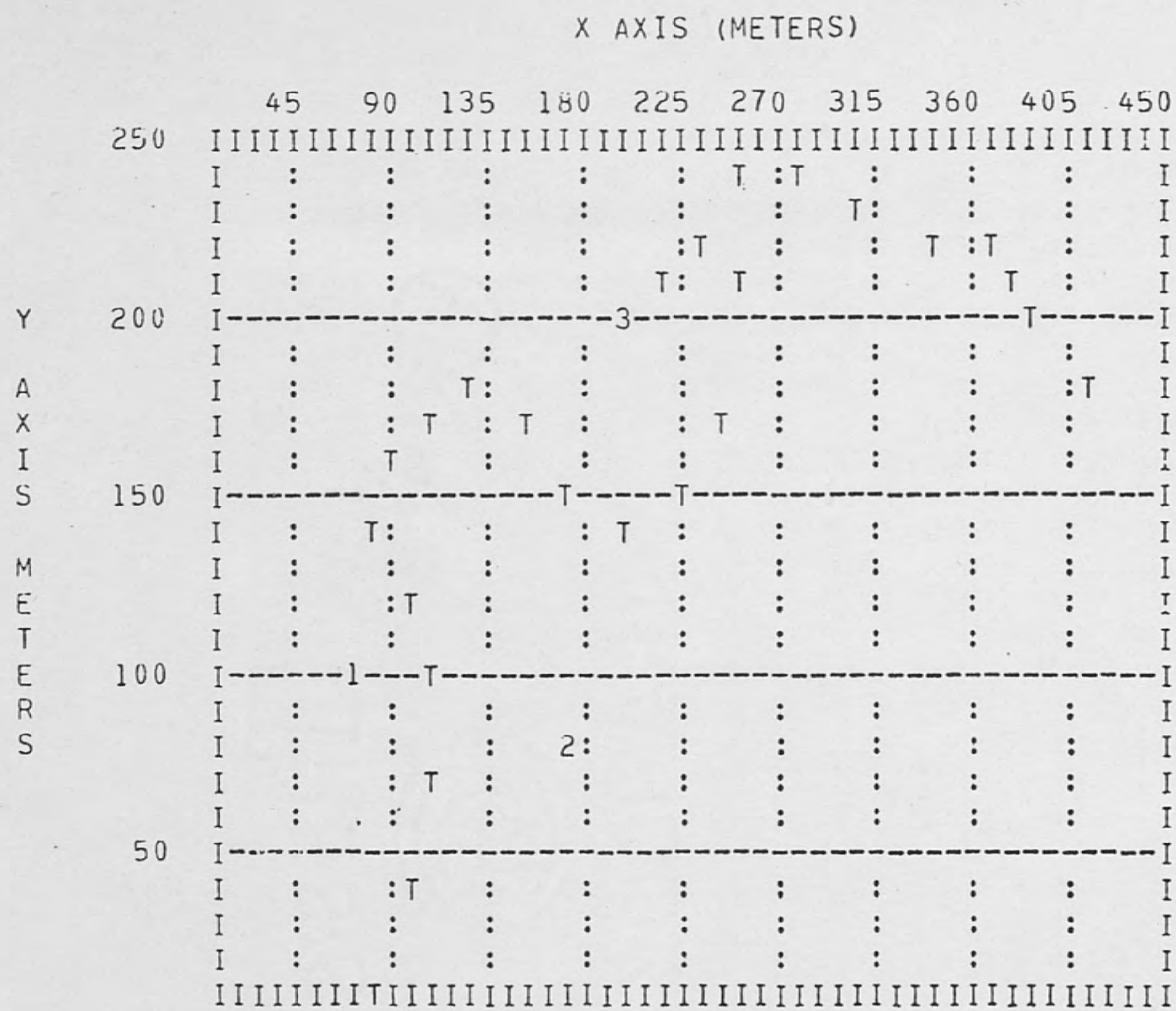
	X	Y	Z
1	80.000	10.000	0.000
2	100.000	40.000	0.000
3	110.000	70.000	0.000
4	110.000	100.000	0.000
5	100.000	120.000	0.000
6	80.000	140.000	0.000
7	90.000	165.000	0.000
8	110.000	175.000	0.000
9	130.000	180.000	0.000
10	155.000	170.000	0.000
11	175.000	150.000	0.000
12	200.000	135.000	0.000
13	225.000	150.000	0.000
14	239.000	175.000	0.000
15	250.000	210.000	0.000
16	218.838	210.593	0.000
17	233.596	223.215	0.000
18	252.745	239.660	0.000
19	280.000	245.000	0.000
20	310.000	235.000	0.000
21	340.000	225.000	0.000
22	365.000	220.000	0.000
23	375.000	210.000	0.000

FIGURE 9. EXAMPLE OF TWO SPACE COMPUTER PRINTOUT SOLUTION



SF= 1.00 XMAX= 500.00 YMAX= 300.00 ZMAX= 300.00

FIGURE 10. EXAMPLE OF TWO SPACE COMPUTER PLOT SOLUTION



SF= 1.00 XMAX= 500.00 YMAX= 300.00 ZMAX= 300.00

FIGURE 11. EXAMPLE OF TWO SPACE COMPUTER PLOT SOLUTION

SECTION V

THREE SPACE PROBLEM: METHOD ONE

The Three Space Program of Appendix E utilizes the distance equations of three coordinate variables and three simultaneous equations with three unknowns in its Newton-Raphson algorithm. The time-difference data for this program has been generated by the program of Appendix A and is provided in Appendix F.

The method failed to provide convergence for any solution points in the trajectory of the target. Other sensor/target paths were tried with no success. This is further discussed in Section VII.

SECTION VI

THREE SPACE PROBLEM: METHOD TWO

The Three Space Program of Appendix H solves three dimensional trajectory paths by utilizing the Two Space Method of analysis once each for the XY, YZ, and XZ planes. Whereas programs of Sections IV and V assumed omni-directional sensors, here the assumption is that directional sensors are utilized. The planes used imply directional sensitivity with respect to each of the cartesian coordinate axes. A separate data simulation program providing information for three planar, directional time-differences has been developed for this three-space tracking program (see Appendix G). The generated time-differince data can be seen in tabular form in Figure 12 and as it appears in the disk data file in Figure 13.

The computer solution is presented in Figures 14 through 20. Note that the trajectory as graphed represents the three planes of view.

Appendix K contains the various computer object-code run-time programs necessary to run the main data simulation and tracking programs listed in this report.

VELOCITY OF MEDIUM IS 1450.000 (M/SEC)

SENSOR POSITIONS

	X	Y	Z
1	350.000	150.000	100.000
2	200.000	300.000	100.000
3	200.000	150.000	250.000

VEHICLE (SOURCE) PATH

	X	Y	Z
1	20.000	10.000	10.000
2	30.000	20.000	15.000
3	40.000	25.000	20.000
4	50.000	30.000	25.000
5	50.000	35.000	30.000
6	60.000	40.000	35.000
7	60.000	40.000	39.000
8	65.000	45.000	40.000
9	75.000	50.000	45.000
10	85.000	70.000	50.000
11	95.000	90.000	60.000
12	115.000	110.000	65.000
13	135.000	117.000	70.000
14	155.000	135.000	80.000
15	175.000	145.000	100.000

FIGURE 12. EXAMPLE OF THREE SPACE DATA GENERATION INPUTS

15

0.011826209146	0.089954401074	0.078128191929
0.012297609448	0.090613086341	0.078315476892
0.011099401978	0.090492064292	0.079392662314
0.009821368412	0.090355973314	0.080534604902
0.011571501350	0.091224755044	0.079653253695
0.010251577365	0.091114696714	0.080863119349
0.010251577365	0.091114696714	0.080863119349
0.010480062101	0.091517611866	0.081037549765
0.009040690055	0.091406477125	0.082365787070
0.013561650552	0.094291692035	0.080730041483
0.018742482973	0.097261980499	0.078519497525
0.020850597142	0.099612743227	0.078762146086
0.016080612748	0.099738363784	0.083657751036
0.016930867856	0.102166832580	0.085235964732
0.012460847806	0.103156080290	0.090695232491
-0.094628523616	-0.076838420703	0.017790102914
-0.094686380756	-0.078095678876	0.016590701881
-0.095166792437	-0.078182469970	0.016984322468
-0.095664419587	-0.078269160703	0.017395258885
-0.096179504341	-0.078354973495	0.017824530847
-0.096712098119	-0.078438875947	0.018273222173
-0.097433352892	-0.077358702821	0.020074650072
-0.097261980499	-0.078519497525	0.018742482973
-0.097828546632	-0.078595015485	0.019233531148
-0.097263610067	-0.083494331699	0.013769278368
-0.097699695003	-0.087681077892	0.010018617112
-0.096583490897	-0.093878760858	0.002704730038
-0.097134142954	-0.095449512937	0.001684630018
-0.097384621645	-0.100455504590	-0.003070882947
-0.103448275860	-0.100057455300	0.003390820554
0.097107937623	0.029001812088	-0.068106125535
0.097262678990	0.028312801639	-0.068949877351
0.097428119773	0.027570825757	-0.069857294016
0.097605395592	0.026770052594	-0.070835342998
0.098295849960	0.028819239242	-0.069476610717
0.098511539055	0.028021648152	-0.070489890903
0.099057885480	0.029741265057	-0.069316620424
0.098975483141	0.028687974433	-0.070287508710
0.099228324420	0.027822020122	-0.071406304298
0.099500944250	0.026876062967	-0.072624881283
0.100522182550	0.028300176911	-0.072222005648
0.100460831620	0.023447832203	-0.077012999421
0.100340571130	0.017728511290	-0.082612059844
0.101226668150	0.013908899234	-0.087317768924
0.103448275860	0.015814439133	-0.087633836729

350.0000	150.0000	100.0000
200.0000	300.0000	100.0000
200.0000	150.0000	250.0000

FIGURE 13. EXAMPLE OF THREE SPACE DATA STORAGE IN COMPUTER FILE

VELOCITY OF MEDIUM IS 1450.000 (M/SEC)

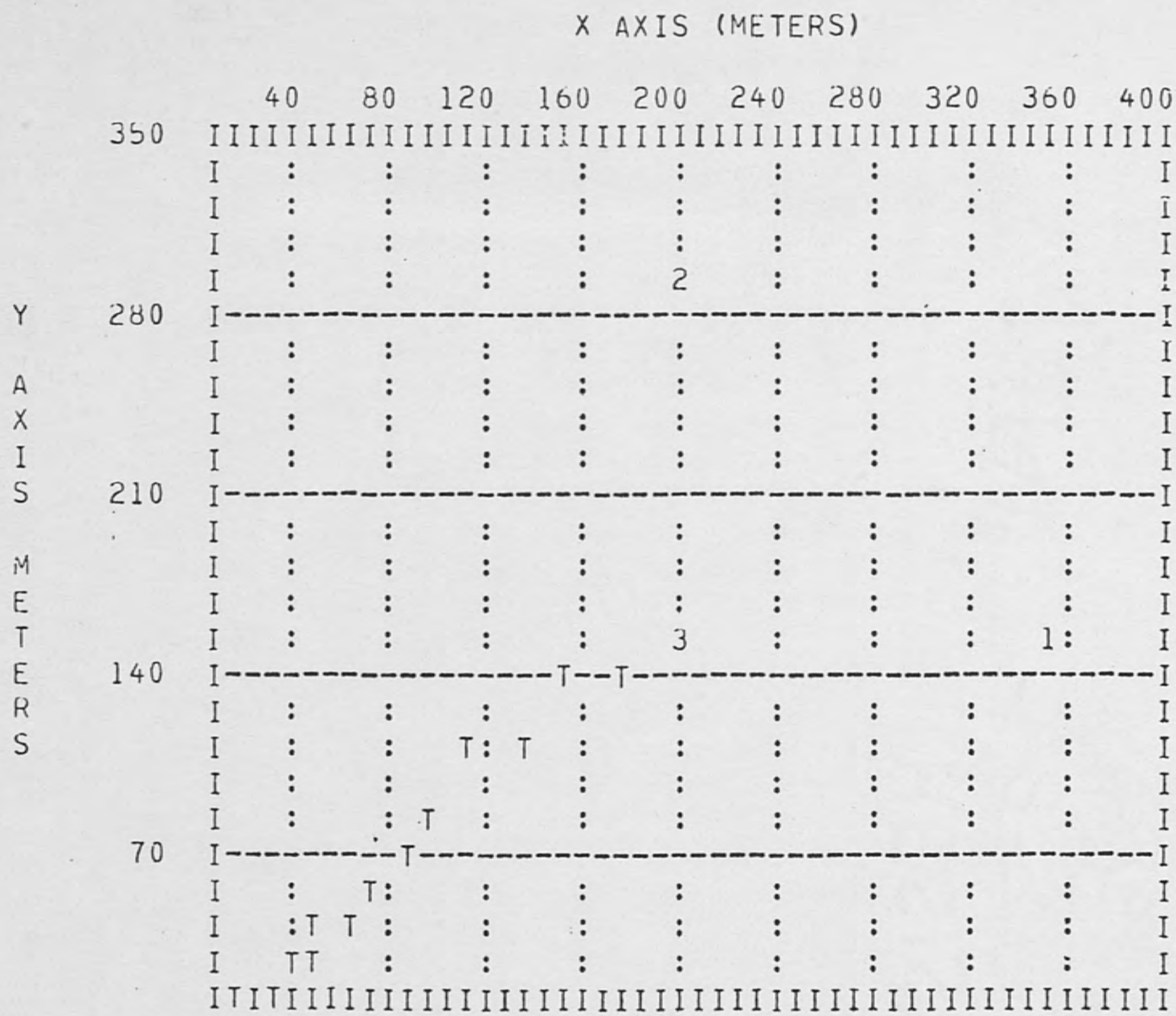
SENSOR POSITIONS

	X	Y	Z
1	350.000	150.000	100.000
2	200.000	300.000	100.000
3	200.000	150.000	250.000

VEHICLE (SOURCE) PATH

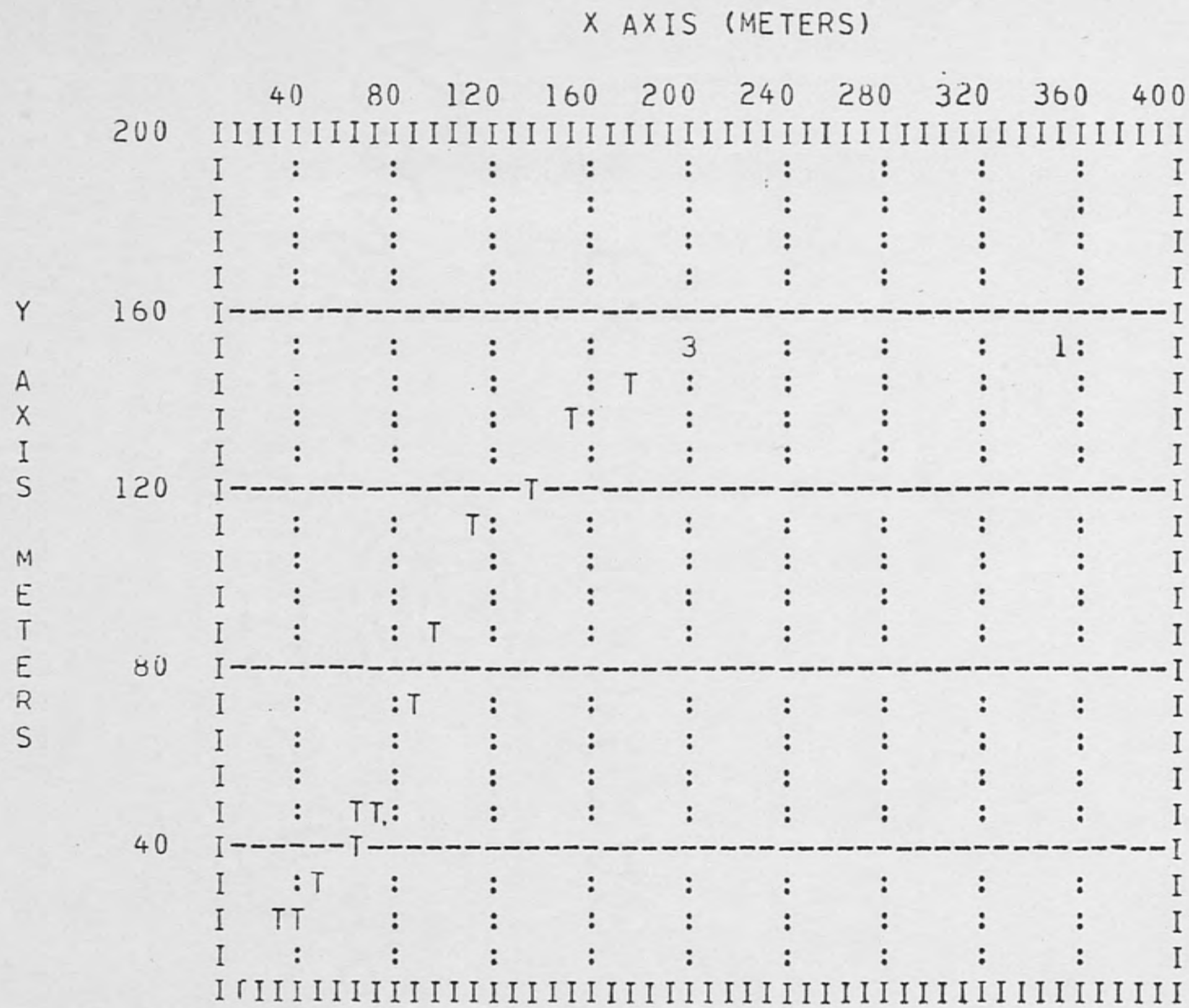
	X	Y	Z
1	20.000	10.000	10.000
2	30.000	20.000	15.000
3	40.000	25.000	20.000
4	50.001	30.000	25.000
5	50.000	35.000	30.000
6	60.001	40.000	35.000
7	60.000	40.000	39.000
8	65.000	45.000	40.000
9	75.001	50.000	45.000
10	85.001	70.000	50.000
11	95.000	90.000	60.000
12	115.000	110.000	65.000
13	135.000	117.000	70.000
14	155.000	135.000	80.000
15	174.671	144.745	99.721

FIGURE 14. EXAMPLE OF THREE SPACE COMPUTER PRINTOUT SOLUTION



SF= 1.00 XMAX= 400.00 YMAX= 350.00 ZMAX= 300.00

FIGURE 15. EXAMPLE OF THREE SPACE COMPUTER PLOT SOLUTION



SF= 1.00 XMAX= 400.00 YMAX= 350.00 ZMAX= 300.00

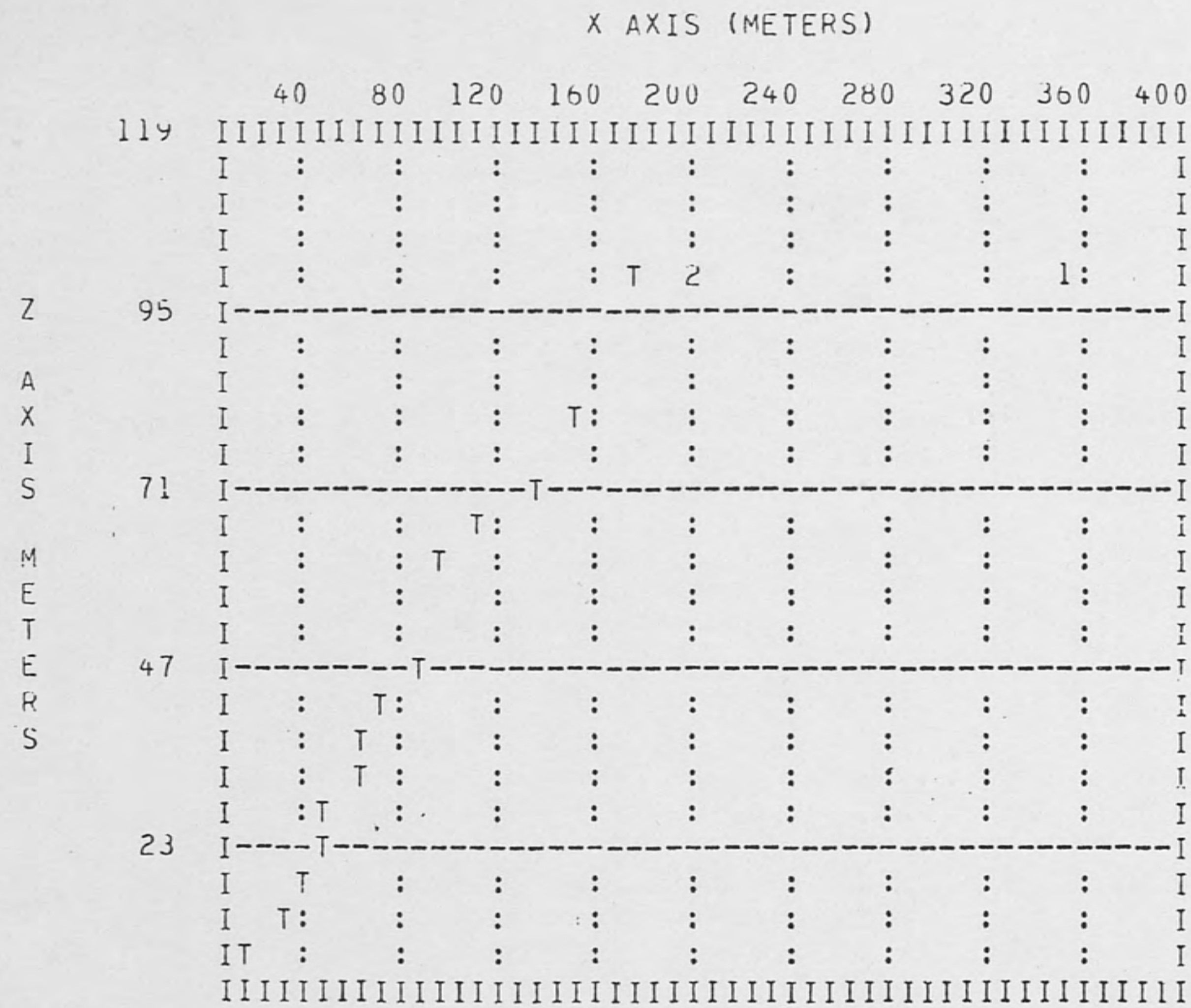
FIGURE 16. EXAMPLE OF THREE SPACE COMPUTER PLOT SOLUTION

```
                     X AXIS (METERS)

             40  80 120 160 200 240 280 320 360 400
      300 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
          I   :   :   :   :   :   :   :   :   :   I
          I   :   :   :   :   :   :   :   :   :   I
          I   :   :   :   :   :   :   :   :   :   I
          I   :   :   :   :   3   :   :   :   :   I
Z     240 I---------------------------------------I
          I   :   :   :   :   :   :   :   :   :   I
A         I   :   :   :   :   :   :   :   :   :   I
X         I   :   :   :   :   :   :   :   :   :   I
I         I   :   :   :   :   :   :   :   :   :   I
S     180 I---------------------------------------I
          I   :   :   :   :   :   :   :   :   :   I
M         I   :   :   :   :   :   :   :   :   :   I
E         I   :   :   :   :   :   :   :   :   :   I
T         I   :   :   :   :   :   :   :   :   :   I
E     120 I---------------------------------------I
R         I   :   :   :   :   :   :   :   :   :   I
S         I   :   :   :   : T 2   :   :   :  1:   I
          I   :   :   :  T:   :   :   :   :   :   I
          I   :   :   : T :   :   :   :   :   :   I
       60 I--------T-T----------------------------I
          I   :  T:T  :   :   :   :   :   :   :   I
          I   : T :   :   :   :   :   :   :   :   I
          I   TT  :   :   :   :   :   :   :   :   I
          ITITIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII

SF=   1.00      XMAX= 400.00      YMAX= 350.00      ZMAX= 300.00
```

FIGURE 17. EXAMPLE OF THREE SPACE COMPUTER PLOT SOLUTION



SF= 1.00 XMAX= 400.00 YMAX= 350.00 ZMAX= 300.00

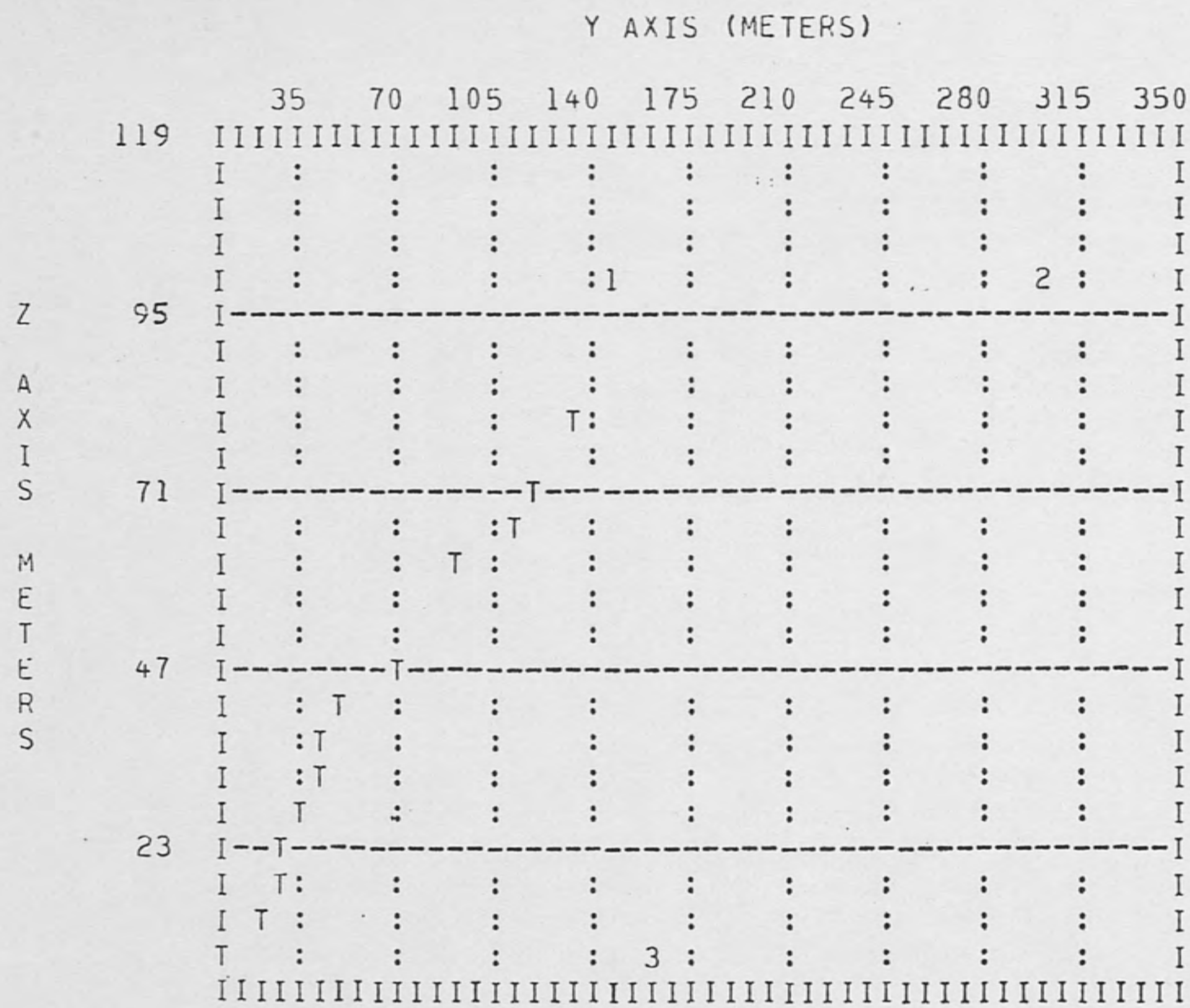
FIGURE 18. EXAMPLE OF THREE SPACE COMPUTER PLOT SOLUTION

```
                         Y AXIS (METERS)

               35   70  105  140  175  210  245  280  315  350
        300  IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
             I   :    :    :    :    :    :    :    :    :    I
             I   :    :    :    :    :    :    :    :    :    I
             I   :    :    :    :    :    :    :    :    :    I
             I   :    :    :    :3   :    :    :    :    :    I
Z       240  I----------------------------------------------------I
             I   :    :    :    :    :    :    :    :    :    I
A            I   :    :    :    :    :    :    :    :    :    I
X            I   :    :    :    :    :    :    :    :    :    I
I            I   :    :    :    :    :    :    :    :    :    I
S       180  I----------------------------------------------------I
             I   :    :    :    :    :    :    :    :    :    I
M            I   :    :    :    :    :    :    :    :    :    I
E            I   :    :    :    :    :    :    :    :    :    I
T            I   :    :    :    :    :    :    :    :    :    I
E       120  I----------------------------------------------------I
R            I   :    :    :    :    :    :    :    :    :    I
S            I   :    :    :    :1   :    :    :    :  2 :    I
             I   :    :    :   T:    :    :    :    :    :    I
             I   :    :    : T  :    :    :    :    :    :    I
         60  I-------------T--T-----------------------------------I
             I   : T  T    :    :    :    :    :    :    :    I
             I   :T   :    :    :    :    :    :    :    :    I
             I  TT    :    :    :    :    :    :    :    :    I
             TITIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII

     SF=   1.00     XMAX= 400.00     YMAX= 350.00     ZMAX= 300.00
```

FIGURE 19. EXAMPLE OF THREE SPACE COMPUTER PLOT SOLUTION



SF= 1.00 XMAX= 400.00 YMAX= 350.00 ZMAX= 300.00

FIGURE 20. EXAMPLE OF THREE SPACE COMPUTER PLOT SOLUTION

SECTION VII

SIMULATION LIMITATIONS

In addition to limitations imposed by the assumptions in Section II (p. 4), the following are noted. When target paths approach the proximity of a sensor, the track point solution is distorted and tends toward the condition of divergence of the solution; or, for less critical nearness, tends to simple distortion in the solution track point coordinate values. The distortion can be observed by comparing the tabular listings of Appendix C to that for Appendix D for the track points 16 through 18. This effect is found in the operational systems such as Loran and Consol (Section III) with occurrence of ambiquities due to sensor nearfield geometric effects. This effect can be explained mathematically as follows. The Newton-Raphson algorithm utilizes partial derivatives of the distance-difference equations (Figure 3) with respect to the coordinate variables. These same partials are found in the gradient of the distance-difference equations. The cross-product of the gradient of two functions with the cross-product of the derivative of the unit-coordinate-vector defines a tangent line at the intersection of the two functions:[18]

---

[18] Wilfred Kaplan, Advanced Calculus (Reading, Mass.: Addison-Wesley Publishing Co., 1973), pp. 158-159.

$$d\hat{r} \times (\nabla\hat{E} \times \nabla\hat{F})=0 \qquad (\hat{r}= x\hat{i} + y\hat{j}).$$

Specially inserted computer calculation printouts were implemented in the software design during program development. (NOTE: for real-time, fast computer applications using the supplied programs, all unnecessary printouts should be deleted). These special data printouts were used to experimentally determine that the partial derivatives become large as the target approaches a sensor; also, the partials of one distance equation approach the condition of being multiples of the other distance equations' partial derivatives in the limit. If the gradients of the two functions are multiples of each other, that is, parallel, then

$$\nabla\hat{E} \times \nabla\hat{F} \rightarrow 0.$$

Thus the tangent line (and intersection solution) disappears, as noted by divergence in the solution iterations. This presents a major concern for the Three Space - Method One - approach, since the solution depends on the intersection of three three-space, simultaneous, hyperboloid functions. The intersection for a solution would have to be **well** defined and unique to exist for convergence.

Another limitation of the simulation is large distances for the target with respect to a tight cluster grouping of the sensors which does not provide sufficient base-line resolution for the sensors to gather unique time-difference data. This is similar to parallax problems in measuring astronomical distances. The radii from the sensors to the target approach some equal magnitude and parallel vector, that is, the sensors appear as one sensor to a distant target.

Since $\nabla\hat{E} \times \nabla\hat{F} = |\hat{E}||\hat{F}|\sin\theta$, the best solution is for the hyperbolic distance-difference equations (Figure 3) to be at an angle θ for $\sin\theta = 1$, or $\theta = 90$ degrees. This condition is best accomplished for widespread sensors placed at vertices of an equilateral triangle. In three-space, this may mean a tetrahedral spacing of sensors is optimal, that is, selecting optimal detection-pairs of sensors for tracking data gathering. However, a tetrahedral spacing of sensors in three-space was not experimentally tested in the tracking program.

The ideas behind the three-space tetrahedral grouping of sensors are as follows. The two-space program sensor tracking approach is optimal for the planar equilateral triangle, that is, with sensors at the vertices of the triangle with the target normally transgressing the plane formed by the sensors. The target optimally must normally transgress the sensor plane keeping sufficiently far from the sensors, yet keeping close enough so that the sensors obtain unique data in order to avoid the ambiquities as previously noted. The obvious extension to three space is a four point placement of sensors at the vertices of a regular tetrahedron. This would provide four equilateral triangle faces (and planes) determined by subsets of three sensors.

SECTION VIII

SUMMARY AND CONCLUSIONS

The simulation limitations of Section VII imitate range limitation and sensor proximity ambiquities that are theoretically similar to those denoted for the systems described in Section III. Regarding sonobuoy sensors, the older SOFAR system is similar to the Two Space Program problem of Section IV with omni-directional sensors. The newer directional frequency and ranging sonobuoy systems are similar to the Section VI three-space trajectory algorithm, where directional sensors are assumed which exhibit planar directional sensitivity.

Future experimental applications of the two- and three-space programs herein would be to access real-time data and process it according to Figure 21. The scenario might be (for two-space) omni-directional microphones picking up a certain prominent frequency of a person's speech as the person walks about a room. The desired solution would be to use the two-space program to track and plot the person's location and trajectory based upon the time-difference data due to the reception of the prominent frequency's wavefronts at the microphones.

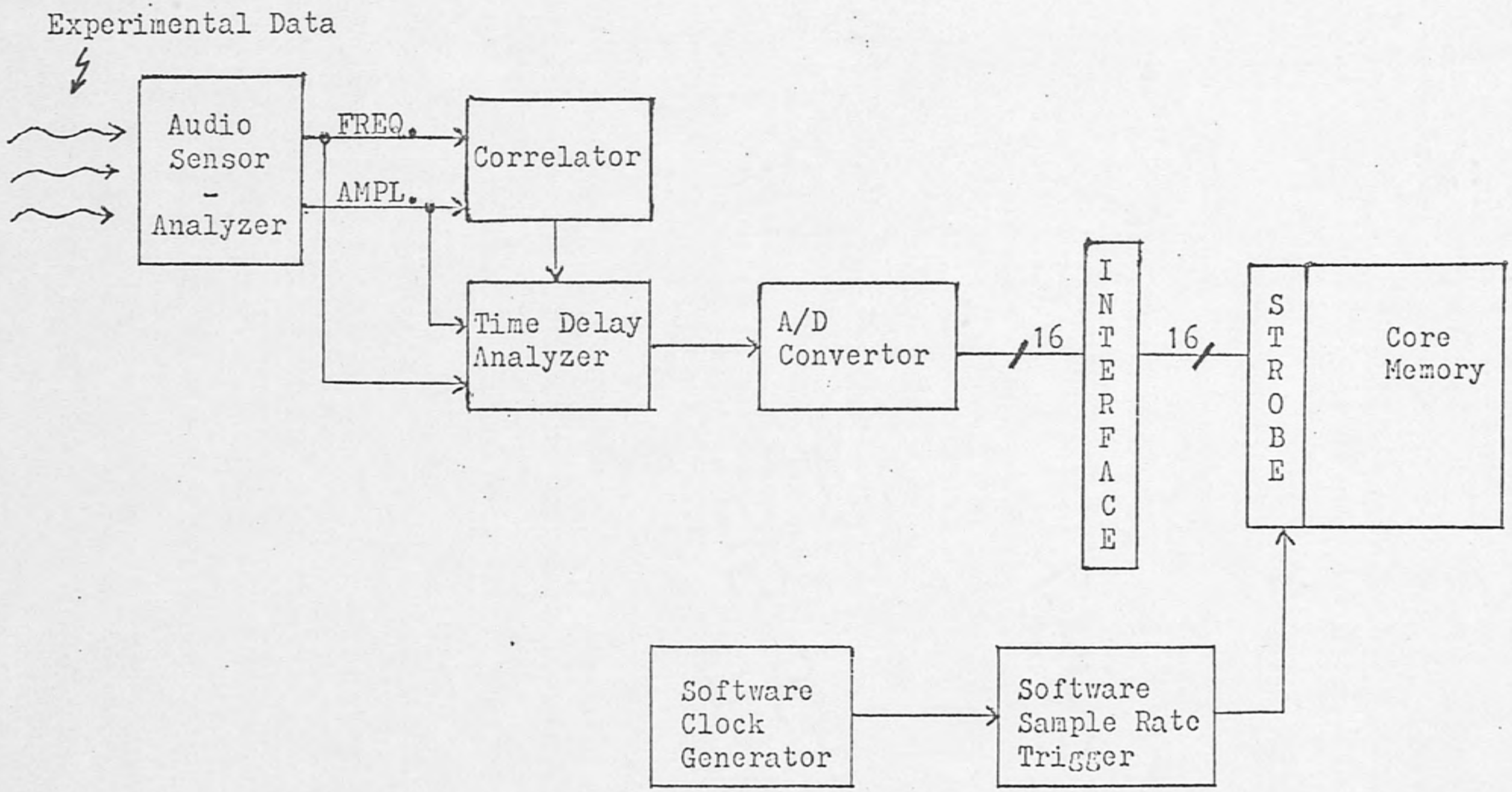


FIGURE 21. SCHEME FOR WAVEFRONT PHASE DETECTION FOR TRACKING

An application for the Three Space (Method Two) Program would be sonar data processing. Here North-South, East-West, and depth directional coordinate sensitivities might be used to make acoustic sensors sensitive to reception of target acoustic wavefronts. This is done for gathering the necessary time-difference data to provide a target tracking capability via the computer program contained herein.

An acoustic tracking program has many other practical considerations: expansion to radar telemetry problems; home and business security monitoring via audio detection; acoustic proximity detection systems for land and sea mines or torpedoes; anti-submarine warfare signal processing; weapons test range tracking; etc. The number of applications is limited only by the engineer's imagination. While only the sensors have been plotted with the experimental target paths, it would be useful to extend (for real applications) the inclusion of such graphic data such as environmental landmarks for operator references. Also, a graphic cathode-ray tube display would be more appealing than the current limitation to graphic line printer plots.

Future use with expansion of the capabilities of the programs herein might include: 1.) devising some means of compensation for near-field target/sensor anomalies in order to avoid divergence of solution track points, 2.) implementation of a more comprehensive first solution track point guess search method in lieu of the present GUESS1 subroutine in order to automate this processing with a high probability of success under total computer program control, and 3.) expanding the programs to correlate tracking data from two or more targets with their respective trajectory.

This method of using time-difference data of acoustic wavefronts is useful particularly where it is desirable to gather as little information as possible about some object with the capability of locating and tracking the object's trajectory.

The emphasis for consideration of this report is that the smallest set of data has been demonstrated, by experimental computer analysis, to be sufficient for tracking a target. This data includes: speed of sound in the medium, location of the sensors, and time-difference data for arrival of a prominent frequency's wavefront at the sensors which is emitted from the target. With three sensors it has been shown that a trajectory for a target can be found in both two- and three-space; and, the geometric considerations have foundation in hyperbolic-fix navigation techniques.

APPENDIX A

TWO SPACE DATA SIMULATION PROGRAM, pp. 45-52.

DESCRIPTION:

This program generates time-difference data for a known vehicle source track given the velocity of sound in the medium and the locations of the sensors. The program prints out all tracking parameters for operator verification of the data; then, stores the time-difference data and sensor locations in a disk file. See Appendix C for relevant data generated by this program.

```
JOB SIMDAT DLIST4                                                    00:48:43
OPTIONS 21,8,9
FORTRAN
                              MODULE:  *MAIN*   OPTIONS: SE LO MA
 61516-02 ENHANCED SAU FORTRAN COMPILER       REVISION LEVEL 33.071978


      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS MAIN ROUTINE CREATES SIMULATED TRACKING DATA (VIA TIME LAG
C     DIFFERENCES BETWEEN SENSORS): BASED UPON PROBLEM SCENARIO OF THE
C     TARGET/SENSOR LOCATIONS.
C
    5 WRITE(17,100)
      WRITE(17,110)
      READ(17,) NPTS
      WRITE(17,120)
      READ(17,) V
      DO 10 I=1,3
      WRITE(17,130)
      READ(17,) XSENS(I),YSENS(I),ZSENS(I)
   10 CONTINUE
      DO 20 J=1,'NPTS
      WRITE(17,140)
      READ(17,) XTRK(J),YTRK(J),ZTRK(J)
      CALL PTCHK(J,2.0,DX)
   20 CONTINUE
      CALL TABULR(DD)
      WRITE(17,150)
      READ(17,) ANS
      IF(ANS.EQ.0.0) GO TO 5
      CALL DATXYZ(DMV)
  100 FORMAT(//,5X,'MAXIMUM DATAPOINTS ARE:',//,10X,
     1'TIMELAGS 50',/,10X,'TRACK POINTS 50',/,10X,'SENSORS 3')
```

```
  110 FORMAT(//,5X,'INPUT THE NUMBER OF DATA POINTS',/,10X,'??')
  120 FORMAT(//,5X,'INPUT THE VELOCITY OF THE MEDIUM:',//,
     110X,'AIR= 331 METERS PER SECOND',/,
     210X,'WATER= 1450 METERS PER SECOND',/,10X,'??')
  130 FORMAT(//,5X,'INPUT THE COORDINATES FOR SENSOR LOCATIONS:',//,
     210X,'XXXX.XX',5X,'YYYY.YY',5X,'ZZZZ.ZZ',
     35X,'(VALUES MUST BE SEPARATED BY SPACE OR COMMAS)',/,'??')
  140 FORMAT(//,5X,'INPUT THE COORDINATES FOR TARGET PARTICLE PATH ',
     1'POINTS:',//,10X,'XXXX.XX',5X,'YYYY.YY',5X,'ZZZZ.ZZ',//,
     25X,'(VALUES MUST BE SEPARATED BY SPACE OR COMMAS)',/,'??')
  150 FORMAT(//,10X,'IS PRINTOUT DATA CORRECT TRACK/LOCATIONS?',//,
     115X,'INPUT:   YES=1.0,   NO=0.0')
      CALL EXIT
      END

                 *MAIN*         SIZE     409    00631
                       MODULE:  *MAIN*   OPTIONS: SE LO MA
 61516-02 ENHANCED SAU FORTRAN COMPILER       REVISION LEVEL 33.071978

      SUBROUTINE PTCHK(N,USE,DX)
      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS SUBROUTINE IS FOR:
C             USE=1.0  TO CHECK SOLUTION  POINT FOR VALIDITY.
C             USE=2.0  TO PROVIDE TIME LAG SIMULATED DATA.
C
C             DX=1.0  FOR CORRECT SOLUTION.
C             DX=0.0  NO SOLUTION POINT.
C
      DX=0.0
      R1=(XTRK(N)-XSENS(1))**2+(YTRK(N)-YSENS(1))**2
      R1=SQRT(R1+(ZTRK(N)-ZSENS(1))**2)
      R2=(XTRK(N)-XSENS(2))**2+(YTRK(N)-YSENS(2))**2
      R2=SQRT(R2+(ZTRK(N)-ZSENS(2))**2)
```

```
      R3=(XTRK(N)-XSENS(3))**2+(YTRK(N)-YSENS(3))**2
      R3=SQRT(R3+(ZTRK(N)-ZSENS(3))**2)
      DT12=(R1-R2)/V
      DT13=(R1-R3)/V
      DT23=(R2-R3)/V
      IF(USE.EQ.2.0) GO TO 800
      MDT12=10000.0*DT12
      MDT13=10000.0*DT13
      MDT23=10000.0*DT23
      MDTT12=10000.0*DTT12(N)
      MDTT13=10000.0*DTT13(N)
      MDTT23=10000.0*DTT23(N)
      IF(MDT12.NE.MDTT12) GO TO 900
      IF(MDT13.NE.MDTT13) GO TO 900
      IF(MDT23.NE.MDTT23) GO TO 900
      DX=1.0
      RETURN
  800 DTT12(N)=DT12
      DTT13(N)=DT13
      DTT23(N)=DT23
  900 RETURN
      END

                PTCHK             SIZE       204     00314
                    MODULE:   *MAIN*    OPTIONS: SE LO MA
61516-02 ENHANCED SAU FORTRAN COMPILER         REVISION LEVEL 33.071978

      SUBROUTINE TABULR(DD)
      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS SUBROUTINE TABULATES VEHICLE SOURCE PATH OUTPUT
C     TO DEVICE 6 (LINE PRINTER).
C
      WRITE(6,1000) V
```

```
      DO 400 I=1,3
      WRITE(6,1100) I,XSENS(I),YSENS(I),ZSENS(I)
  400 CONTINUE
      WRITE(6,1200)
      DO 500 L=1,NPTS
      WRITE(6,1300) L,XTRK(L),YTRK(L),ZTRK(L)
  500 CONTINUE
 1000 FORMAT(1H1,5X,'VELOCITY OF MEDIUM IS ',3X,F8.3,2X,'(M/SEC)',
     1//,40X,'SENSOR POSITIONS',/)
 1100 FORMAT(5X,I2,3(10X,F10.3))
 1200 FORMAT(//,40X,'VEHICLE (SOURCE) PATH',/)
 1300 FORMAT(5X,I2,3(10X,F10.3))
      RETURN
      END

                TABULR          SIZE      169     00251
                      MODULE:   *MAIN*    OPTIONS: SE LO MA
 61516-02 ENHANCED SAU FORTRAN COMPILER       REVISION LEVEL 33.071978

      SUBROUTINE DATXYZ(DMV)
      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS  SUBROUTINE IS USED TO STORE TIME LAG, NPTS
C     AND SENSOR LOCATION DATA ON DISK FILE.
C
      REWIND 18
      WRITE(18,134) NPTS
  134 FORMAT(I2)
      DO 601 I=1, NPTS
      WRITE(18,108) DTT12(I),DTT13(I),DTT23(I)
  108 FORMAT(3(5X,F15.12))
  601 CONTINUE
      DO 48 N=1,3
```

```
      WRITE(18,110) XSENS(N),YSENS(N),ZSENS(N)
  110 FORMAT(3(5X,F10.4))
   48 CONTINUE
      REWIND 18
      RETURN
      END

                    DATXYZ          SIZE     115     00163

$CATALOG
61632-02 OVERLAY CATALOGER -- REVISION LEVEL 12.071978
TYPE=BG
DNAME SIMDAT,4
BEGIN
            P.NAME=  SIMDAT

*HIGH     11063               *START        0
*LOW          0               *LOW          0
*P.END     7650               .PASS         2
*START        0               PTCHK       631 $
.PASS         2               TABULR     1145 $
C$ERRO     7310 $             DATXYZ     1416 $
C$IOER     7321 $             F$RIFF     1601 $
C$OFLO     7312 $             F$RSFF     1601 $
C$UFLO     7316 $             F$RSSY     1602 $
DATXYZ     1416 $             F$WIFF     1604 $
DCF        6101 $             F$WSFF     1604 $
DSQRT      7303 $             F$WSSY     1605 $
ESGN$      5552 $             F$DIFF     1651 $
EXIT       7132 $             F$DSSY     1652 $
EXP10$     5551 $             F$EIFF     1654 $
F$1ISN     2265 $             F$ESSY     1655 $
F$1OSN     2464 $             F$EESY     1755 $
F$4ISN     2605 $             F$SYEE     1755 $
F$4OSN     2713 $             F$RORW     1770 $
```

F$5ISN	2603	$	F$FUNC	2073	$
F$5OSN	2711	$	F$B/E	2165	$
F$8ISN	2601	$	F$CSIO	2175	$
F$8OSN	2707	$	F$CFIO	2175	$
F$ADDR	5766	$	F$BISN	2250	$
F$AIP	3034	$	F$AISN	2261	$
F$AISN	2261	$	F$1ISN	2265	$
F$AOP	4157	$	F$BOSN	2453	$
F$B/E	2165	$	F$1OSN	2464	$
F$BBI	7624	$	F$8ISN	2601	$
F$BBJ	7555	$	F$5ISN	2603	$
F$BCC	5776	$	F$4ISN	2605	$
F$BISN	2250	$	F$8OSN	2707	$
F$BOSN	2453	$	F$5OSN	2711	$
F$BUF	5774	$	F$4OSN	2713	$
F$CFIO	2175	$	F$AIP	3034	$
F$CLOS	7207	$	F$RIP	3036	$
F$CNOP	7134	$	F$LIP	3131	$
F$CONV	6004	$	F$OIP	3163	$
F$CSIO	2175	$	F$IIP	3221	$
F$CV41	7271	$	F$GIP	3471	$
F$CV51	7271	$	F$ROP	4155	$
F$D	5771	$	F$AOP	4157	$
F$DCF	6101	$	F$IOP	4243	$
F$DCNT	6575	$	F$FOP	4537	$
F$DIFF	1651	$	F$QOP	4727	$
F$DSSY	1652	$	F$SOP	4734	$
F$DVR5	6722	$	F$O.LB	4743	$
F$E	5772	$	F$O.SN	4747	$
F$E/E	6002	$	F$OOP	4762	$
F$E/E1	6003	$	F$GOP	5110	$
F$E/O	6000	$	F$EOP	5112	$
F$E/O1	6001	$	TENTB$	5426	$
F$EESY	1755	$	EXP10$	5551	$
F$EIFF	1654	$	ESGN$	5552	$

F$EMB	7525	$	MSGN$	5553	$
F$EMBJ	7537	$	F$NORM	5556	$
F$EOP	5112	$	F$ADDR	5766	$
F$ERCD	7266	$	F$SPEC	5767	$
F$ESSY	1655	$	F$FMTC	5767	$
F$FCC	5775	$	F$W	5770	$
F$FERR	6007	$	F$D	5771	$
F$FMT	5773	$	F$E	5772	$
F$FMTC	5767	$	F$FMT	5773	$
F$FOP	4537	$	F$BUF	5774	$
F$FUNC	2073	$	F$FCC	5775	$
F$GIP	3471	$	F$BCC	5776	$
F$GOP	5110	$	F$P	5777	$
F$IBF	6005	$	F$E/O	6000	$
F$IIP	3221	$	F$E/O1	6001	$
F$IOM0	7037	$	F$E/E	6002	$
F$IOM1	7023	$	F$E/E1	6003	$
F$IOM2	7025	$	F$CONV	6004	$
F$IOM3	7135	$	F$IBF	6005	$
F$IOM4	7144	$	F$FERR	6007	$
F$IOP	4243	$	F$NCHR	6010	$
F$LIP	3131	$	F$PNL	6062	$
F$MPR5	6626	$	F$DCF	6101	$
F$NCHR	6010	$	DCF	6101	$
F$NORM	5556	$	F$DCNT	6575	$
F$O.LB	4743	$	PUTN$	6600	$
F$O.SN	4747	$	F$MPR5	6626	$
F$OIP	3163	$	F$DVR5	6722	$
F$OOP	4762	$	F$IOM1	7023	$
F$OPEN	7152	$	F$IOM2	7025	$
F$P	5777	$	F$IOM0	7037	$
F$PNL	6062	$	EXIT	7132	$
F$QOP	4727	$	F$CNOP	7134	$
F$RBM	7532	$	F$IOM3	7135	$
F$RBMI	7575	$	F$IOM4	7144	$

```
F$RIFF      1601 $         F$OPEN      7152 $
F$RIP       3036 $         F$CLOS      7207 $
F$ROP       4155 $         IOERR       7264 $
F$RORW      1770 $         F$ERCD      7266 $
F$RSFF      1601 $         INT         7267 $
F$RSSY      1602 $         IDINT       7267 $
F$SOP       4734 $         IFIX        7267 $
F$SPEC      5767 $         IFIX$       7267 $
F$SYEE      1755 $         F$CV41      7271 $
F$W         5770 $         F$CV51      7271 $
F$WIFF      1604 $         SQRT        7303 $
F$WSFF      1604 $         DSQRT       7303 $
F$WSSY      1605 $         C$ERR0      7310 $

IDINT       7267 $         C$OFLO      7312 $
IFIX        7267 $         C$UFLO      7316 $
IFIX$       7267 $         C$IOER      7321 $
INT         7267 $         F$EMB       7525 $
IOERR       7264 $         F$RBM       7532 $
MSGN$       5553 $         F$EMBJ      7537 $
PTCHK        631 $         F$BBJ       7555 $
PUTN$       6600 $         F$RBMI      7575 $
SQRT        7303 $         F$BBI       7624 $
TABULR      1145 $         *P.END      7650
TENTB$      5426 $         ZL1         7650 C
ZL1         7650 C         ZL3        10635 C
ZL2        10657 C         ZL2        10657 C
ZL3        10635 C         *HIGH      11063
$EOJ
SIMDAT TIME 00HRS 00MIN 00.000SEC
```

APPENDIX B

TWO SPACE TRACKING PROGRAM, pp. 54-77.

DESCRIPTION:

This program interprets the time-difference data and sensor locations from a disk storage file. It processes this data using principles of hyperbolic fix navigation and the Newton-Raphson algorithm in order to determine the track path of a source vehicle emanating acoustic signals. The data interpreted by this program from the disk file is found in Appendix C, page 80. The results of this tracking program are found in Appendix D.

```
$JOB TRACK DLIST4                                                     03:50:27
$OPTIONS 21,8,9
$FORTRAN
                          MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA SR
  61516-02 ENHANCED SAU FORTRAN COMPILER       REVISION LEVEL 33.071978

      :WA,SR,AB,GO
            COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
            COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
            COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
            COMMON /ZL4/ FN(25,50),IXAX(10),IYAX(5),IZAX(5),IYZAX(10)
      C***********************************************************************
      C     THIS IS THE MAIN PROGRAM FOR THE ACOUSTIC MODEL TRACKING
      C     AND PLOT PROGRAM
      C
      C     IT IS BASED ON:
      C              3-SENSORS: KNOWN COORDINATES; WITH, RELATIVE PHASE
      C                COHERENCE - OR TIME LAGS OF ARRIVING WAVEFRONTS -
      C                AS RECEIVED DATA FROM AN UNKNOWN SOURCE WITH SOME
      C                RELATIVE REFERENCE FREQUENCY USED IN DETECTION OF
      C
      C                DELTA TIME DIFFERENCES BETWEEN SENSORS/SOURCE.
      C              1-SOURCE: OF UNKNOWN POSITIONS, EMITTING SOME REFERENCE
      C                SIGNAL FOR DETECTION.
      C
      C              ASSUMPTIONS: (1) DELTA TIME DIFFERENCES HAVE BEEN
      C                PROCESSED FOR INPUT TO THIS PROGRAM BY SOME RECEIVING
      C                PROCESSOR;(2) NO PROPAGATION LOSS DUE TO MEDIUM;
      C                (3) SOUND VELOCITY OF MEDIUM IS CONSTANT AND KNOWN; AN
      C                (4) NO MULTIPATH PHENOMENA, FOR A SOURCE-SENSOR PAIR.
      C
      C
      C              SOLUTION:
      C                USING ONLY THE ABOVE SCENARIO DESCRIPTORS, TRACK AND
      C                PLOT THE SOURCE.
```

```
C
C                   APPROACH:
C                     USING DISTANCE EQUATIONS FROM THE SENSORS TO THE
C                     UNKNOWN SOURCE, AND THE VELOCITY OF THE MEDIUM:
C                       -DEVELOP THE RESULTING NON-LINEAR HYPERBOLIC LOCATOR
C                        EQUATIONS;
C                       -SOLVE ITERATIVELY WITH A NEWTON-RAPHSON NON-LINEAR,
C                        SIMULTANEOUS EQUATION METHOD.
C
C                   THIS PROGRAM WRITTEN BY:
C                         GEORGE H. FORD, JR.
C                            JULY 3, 1980
C                   IN FULFILLMENT OF A RESEARCH TOPIC/REPORT FOR AN
C                   M.S.E. DEGREE IN THE U.C.F. DEPARTMENT OF ELECTRICAL
C                   ENGINEERING.
C
C                   SOURCE PROGRAM TYPED ON CARDS.
C                   COMPUTER:
C                       NAVTRAEQUIPCEN: HARRIS DATACRAFT 6024/4
C                       WITH LINE PRINTER PLOT SOFTWARE  WHICH WAS GENERATED
C                       FOR THIS PROJECT INCLUDED HEREIN.
C
C                   REVIEW BOARD:
                          MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA SR
 61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

C                       DR. FRED SIMONS  (CHAIRMAN)
C                       DR. MIKE HARRIS
C                       DR. CHRIS BAUER
C
C**********************************************************************
      DO 3 I=1,50
      XTRK(I)=0.0
      YTRK(I)=0.0
      ZTRK(I)=0.0
    3 CONTINUE
```

```
    5 WRITE(50,99)
      READ(50,) V
      REWIND 40
      READ(40,134) NPTS
      WRITE(50,105)
      READ(50,106) IZ
C
C     IZ=1 REAL DATA
C    IZ=0 CALCULATED, FILE STORAGE DATA: DATXYZ
C
C     INPUT TIME LAGS FROM REAL DATA/FILE DATA.
C
      IF(IZ.NE.1) GO TO 10
      WRITE(50,133)
      READ(50,134) NPTS
      DO 600 I=1,NPTS
      CALL SENSOR(I,DD)
  600 CONTINUE
      GO TO 46
   10 DO 601 I=1,NPTS
      READ(40,108) DTT12(I),DTT13(I),DTT23(I)
  601 CONTINUE
   46 NGES1=0
C
C     READ FILE DATA FOR SENSORS: X,Y,Z.
C
      DO 48 N=1,3
      READ(40,110) XSENS(N),YSENS(N),ZSENS(N)
   48 CONTINUE
C
C     INPUT/CALCULATE INITIAL GUESSES FOR X,Y,Z (SOURCE)
C
      WRITE(50,109)
      READ(50,) XLAT,YLONG,ZDEPTH
```

```
      51 CALL GUESS1(NGES1,XI,YI,ZI)
         CALL TABULR(DD)
         DO 6948 IDD=1,3
      52 CALL TRKPLT(ANS,IDD)
         IF(ANS.EQ.1.0) GO TO 52
    6948 CONTINUE
         WRITE(50,111)
         READ(50,) ANS
                              MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA SR
61516-02 ENHANCED SAU FORTRAN COMPILER          REVISION LEVEL 33.071978

         IF(ANS.EQ.1.0) GO TO 5
C
C        FORMATS: MAIN.
C
      99 FORMAT(5X,'INPUT THE MAGNITUDE OF VELOCITY OF THE MEDIUM',/,
        15X,' (METERS/SEC)',/,15X,'WATER= 1450 M/SEC    AIR= 331 M/SEC')
     134 FORMAT(I2)
     105 FORMAT(5X,'INPUT THE FOLLOWING FOR TYPE PROGRAM:',/,15X,
        1'1= REAL DATA (TIME LAGS VIA PHASES PROCESSED)',/,15X,
        2'0= FILE DATA VIA DATXYZ PROGRAM')
     106 FORMAT(I1)
     133 FORMAT(5X,'HOW MANY TRACKING POINTS DESIRED? (MAX=50)')
     108 FORMAT(3(5X,F15.12))
     109 FORMAT(5X,'INPUT:',/,10X,'MAXIMUM X METERS(LATITUDE RANGE)',
        1/,10X,'MAXIMUM Y METERS(LONGITUDE RANGE)',/,10X,
        2'MAXIMUM Z METER(DEPTH)',//,10X,'XXX.XX',5X,'YYY.YY',5X,'ZZZ.ZZ')
     110 FORMAT(3(5X,F10.4))
     111 FORMAT(5X,'DO YOU WISH TO:',//,10X,
        1'1- CONTINUE PROCESSING; OR',/,10X,
        2'2- END PROCESSING',//,5X,'??')
         CALL EXIT
         END
```

```
                    *MAIN*          SIZE      665    01231
                              MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA SR
         61516-02 ENHANCED SAU FORTRAN COMPILER       REVISION LEVEL 33.071978

      SUBROUTINE GUESS1(NGES.XI,YI,ZI)
      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS SUBROUTINE IS TO SUPPLY INITIAL GUESSES FOR TRACK PLOT OF
C     THE SOURCE VEHICLE.
C     UPON SUCCESSFUL TRACKING OF THE FIRST THREE POINTS, PROCESSING
C     SHALL CONTINUE FOR ALL REQUIRED TRACK POINTS IN THE
C     SUBROUTINE TRKPT.
C
C     VARIOUS COMBINATIONS OF V*DTT FACTORS ADDED TO XI, YI, ZI.
C     ASSUMPTION: ONE TIME LAG WILL DOMINATE ONE COORDINATE.
C     ALL INITIAL GUESSES PERTURBATED ABOUT CENTER OF UNDERWATER (4)
C     QUADRANTS.
C
  700 NGES=NGES+1
      FLAG=0.0
      GO TO 900
  880 CALL TRKPT(XI,YI,ZI,FLAG)
      IF(FLAG.EQ.1.0.AND.NGES.LT.24) GO TO 700
      IF(FLAG.EQ.1.0.AND.NGES.GE.24) GO TO 900
      GO TO 1000
  900 FLAG=0.0
      WRITE(50,2000)
      READ(50,) XI,YI,ZI
 2000 FORMAT(5X,'FAILURE TO CONVERGE FOR INITIAL SOLUTION',//,
     115X,'INPUT:',5X,'XXX.XX',5X,'YYY.YY',5X,'ZZZ.ZZ',5X,'GUESS')
      GO TO 880
C
C     FLAG=0.0 CLEARED
```

```
C          =1.0 NO CONVERGENCE
C          =2.0 TRACK COMPLETED
C
  1000 RETURN
       END

                  GUESS1          SIZE      142    00216
                        MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA SR
   61516-02 ENHANCED SAU FORTRAN COMPILER      REVISION LEVEL 33.071978

       SUBROUTINE TRKPT(XI,YI,ZI,FLAG)
       COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
       COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
       COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C      THIS SUBROUTINE LOCATES TRACKING LOCATIONS FOR SPECIFIC
C      DATAPOINTS (TIME LAGS) RECEIVED.
C
       DX=0.0
       DO 200 N=1,NPTS
       TRYTWO=0.0
     5 IT=0
    10 IT=IT+1
       IF(IT.EQ.1.AND.N.EQ.1) GO TO 15
       IF(IT.NE.1) GO TO 20
       XTRK(N)=XTRK(N-1)
       YTRK(N)=YTRK(N-1)
       IF(XTRK(N-1).EQ.0.0.AND.N.GE.3) XTRK(N)=XTRK(N-2)
       IF(YTRK(N-1).EQ.0.0.AND.N.GE.3) YTRK(N)=YTRK(N-2)
       GO TO 20
    15 Y=YI
       X=XI
       XTRK(N)=X
       YTRK(N)=Y
    20 CONTINUE
```

```
      CALL SYSEQN(N,DDH,DDK,DD)
      USE=1.0
      CALL PTCHK(N,USE,DDH,DDK,DX)
C
C     DX=1.0  CORRECT SOLUTION POINT
C     DX=0.0  NO SOLUTION POINT FOUND.
C
      IF(DX.EQ.1.0) FLAG=0.0
      IF(DX.EQ.0.0) FLAG=1.0
      IF(DX.NE.1.0.AND.IT.LE.25) GO TO 10
      IF(DX.NE.1.0.AND.IT.GE.25) GO TO 150
      GO TO 180
  150 IF(TRYTWO.NE.1.0) GO TO 160
      XTRK(N)=0.0
      YTRK(N)=0.0
      ZTRK(N)=0.0
      GO TO 180
  160 TRYTWO=1.0
      XTRK(N)=XTRK(N-1)+30.0
      YTRK(N)=YTRK(N-1)+30.0
      GO TO 5
  180 DX=0.0
  200 CONTINUE
  900 RETURN
      END

                 TRKPT            SIZE     379    00573
                         MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA SR
  61516-02 ENHANCED SAU FORTRAN COMPILER      REVISION LEVEL 33.071978

      SUBROUTINE SYSEQN(N,DDH,DDK,DD)
      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
```

```
C     THIS SUBROUTINE SOLVES THE NONLINEAR SIMULTANEOUS SYSTEM
C     EQUATIONS DESCRIBING THE PARTICLE PATH DYNAMICS.  A NEWTON-RAPHSO
C     ALGORITHMIC SET OF EQUATIONS HAS BEEN IMPLEMENTED.
C
      ITSYS=0.0
  100 CONTINUE
      ITSYS=ITSYS+1.0
      DD1=(DTT12(N)*V)
      DD2=(DTT13(N)*V)
      DD3=(DTT23(N)*V)
      A=(XTRK(N)-XSENS(1))**2+(YTRK(N)-YSENS(1))**2
      B=(XTRK(N)-XSENS(2))**2+(YTRK(N)-YSENS(2))**2
      C=(XTRK(N)-XSENS(3))**2+(YTRK(N)-YSENS(3))**2
      EMF=DD1+SQRT((XTRK(N)-XSENS(2))**2+(YTRK(N)-YSENS(2))**2)
      EMF=EMF-SQRT((XTRK(N)-XSENS(1))**2+(YTRK(N)-YSENS(1))**2)
      FMF=DD2+SQRT((XTRK(N)-XSENS(3))**2+(YTRK(N)-YSENS(3))**2)
      FMF=FMF-SQRT((XTRK(N)-XSENS(1))**2+(YTRK(N)-YSENS(1))**2)
      EX=((XTRK(N)-XSENS(1))/SQRT(A))-((XTRK(N)-XSENS(2))/SQRT(B))
      EY=((YTRK(N)-YSENS(1))/SQRT(A))-((YTRK(N)-YSENS(2))/SQRT(B))
      FX=((XTRK(N)-XSENS(1))/SQRT(A))-((XTRK(N)-XSENS(3))/SQRT(C))
      FY=((YTRK(N)-YSENS(1))/SQRT(A))-((YTRK(N)-YSENS(3))/SQRT(C))
C
C     SOLVE FUNCTIONS AND PARTIALS VIA CRAMER'S RULE FOR DETERMINING
C     TRACK POINT, WITHIN A SPECIFIED TOLERANCE.
C
      XN(2)=XTRK(N)
      YN(2)=YTRK(N)
      WRITE(60,1110)
 1110 FORMAT(3X,'N',8X,'EMF',8X,'FMF',8X,'EX',8X,'EY',
     18X,'FX',8X,'FY',8X,'GX',8X,'GY',/)
      CFLAG=0.0
```

```
      WRITE(60,1111) N,EMF,FMF,EX,EY,FX,FY
 1111 FORMAT(2X,I2,2X,6(E9.1,2X))
      CALL CRAMER(N,EMF,FMF,EX,EY,FX,FY,DDH,DDK,CFLAG)
C
C     CFLAG=1.0  FOR FURTHER ACCURACY REQUIRED.
C     CFLAG=2.0  FOR WRONSKIAN=0.0.
C     CFLAG=3.0  FOR SOLUTION TOLERANCE ACHIEVED.
C
      IF(CFLAG.NE.2.0) GO TO 150
      XTRK(N)=0.0
      YTRK(N)=0.0
      GO TO 500
  150 IF(CFLAG.NE.1.0) GO TO 300
      IF(ITSYS.GE.25) GO TO 300
      XTRK(N)=XN(2)
      YTRK(N)=YN(2)
                          MODULE:  SYSEQN   OPTIONS: SE AB WA LO MA SR
   61516-02 ENHANCED SAU FORTRAN COMPILER      REVISION LEVEL 33.071978

      GO TO 100
  300 XTRK(N)=XN(2)
      YTRK(N)=YN(2)
  500 RETURN
      END

                SYSEQN          SIZE     571    01073
                          MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA SR
   61516-02 ENHANCED SAU FORTRAN COMPILER      REVISION LEVEL 33.071978

      SUBROUTINE CRAMER(N,EMF,FMF,EX,EY,FX,FY,DDH,DDK,CFLAG)
      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS SUBROUTINE SOLVES FOR THE X,Y,Z, INCREMENTS VIA CRAMER'S RUL
```

```
C      FOR THE NEWTON-RAPHSON SYSTEM EQUATIONS.  AN X,Y,Z SOLUTION IS FO
C      WITHIN SOME ACCURACY; OR WITHIN 25 ITERATIONS - WHICHEVER IS FIRS
C
       XN(1)=XN(2)
       YN(1)=YN(2)
       CALL MTX(FX,EX,FY,EY,WRON)
       IF(WRON.NE.0.0) GO TO 30
       CFLAG=2.0
       GO TO 900
    30 CALL MTX(FMF,EMF,FY,EY,DH)
       CALL MTX(FX,EX,FMF,EMF,DK)
       DDH=DH/WRON
       DDK=DK/WRON
       WRITE(60,1221) DDH,DDK
  1221 FORMAT(/,5X,'DDH= ',E12.4,5X,'DDK= ',E12.4,//)
       XN(2)=XN(1)+DDH
       YN(2)=YN(1)+DDK
       WRITE(60,4321) XN(2),YN(2)
  4321 FORMAT(10X,'X= ',E12.4,10X,'Y= ',E12.4,10X)
       XH=ABS(DDH)
       YK=ABS(DDK)
       IF(XH.LE.0.1.AND.YK.LE.0.1) GO TO 800
       CFLAG=1.0
       GO TO 900
   800 CFLAG=3.0
   900 RETURN
       END

                 CRAMER          SIZE      196     00304
                        MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA SR
   61516-02 ENHANCED SAU FORTRAN COMPILER      REVISION LEVEL 33.071978

       SUBROUTINE MTX(A,B,C,D,S)
       COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
       COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),VV
```

```
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS ROUTINE FINDS THE DETERMINANT OF A 2X2 MATRIX.
C
C     S=SOLUTION
C
C   A   C
C   B   D
C
      S=(A*D)-(B*C)
      WRITE(60,112) S
  112 FORMAT(10X,E12.4,'=S IN MTX')
      RETURN
      END

               MTX                SIZE      53    00065
                        MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA SR
 61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

      SUBROUTINE PTCHK(N,USE,DDH,DDK,DX)
      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS SUBROUTINE IS FOR:
C             USE=1.0  TO CHECK SOLUTION  POINT FOR VALIDITY.
C             USE=2.0  TO PROVIDE TIME LAG SIMULATED DATA.
C
C             DX=1.0  FOR CORRECT SOLUTION.
C             DX=0.0  NO SOLUTION POINT.
C
      DX=0.0
      R1=(XTRK(N)-XSENS(1))**2+(YTRK(N)-YSENS(1))**2
      R2=(XTRK(N)-XSENS(2))**2+(YTRK(N)-YSENS(2))**2
      R3=(XTRK(N)-XSENS(3))**2+(YTRK(N)-YSENS(3))**2
```

```
          DT12=(R1-R2)/V
          DT13=(R1-R3)/V
          DT23=(R2-R3)/V
          IF(USE.EQ.2.0) GO TO 800
          MDT12=1000.0*DT12
          MDT13=1000.0*DT13
          MDT23=1000.0*DT23
          MDTT12=1000.0*DTT12(N)
          MDTT13=1000.0*DTT13(N)
          MDTT23=1000.0*DTT23(N)
          IF(IABS(MDT12-MDTT12).GE.5) GO TO 885
          IF(IABS(MDT13-MDTT13).GE.5) GO TO 885
          IF(IABS(MDT23-MDTT23).GE.5) GO TO 885
          DX=1.0
      885 SS=SQRT(DDH**2+DDK**2)
          IF(0.1.LT.SS) GO TO 900
          DX=1.0
          RETURN
      800 DTT12(N)=DT12
          DTT13(N)=DT13
          DTT23(N)=DT23
      900 RETURN
          END
```

```
                    PTCHK            SIZE      312    00470
                            MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA SR
   61516-02 ENHANCED SAU FORTRAN COMPILER       REVISION LEVEL 33.071978
```

```
          SUBROUTINE TABULR(DD)
          COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
          COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
          COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
   C
   C      THIS SUBROUTINE TABULATES VEHICLE SOURCE PATH OUTPUT
   C      TO DEVICE 60 (LINE PRINTER).
```

```
C
      WRITE(60,1000) V
      DO 400 I=1,3
      WRITE(60,1100) I,XSENS(I),YSENS(I),ZSENS(I)
  400 CONTINUE
      WRITE(60,1200)
      DO 500 L=1,NPTS
      WRITE(60,1300) L,XTRK(L),YTRK(L),ZTRK(L)
  500 CONTINUE
 1000 FORMAT(1H1,5X,'VELOCITY OF MEDIUM IS ',3X,F8.3,2X,'(M/SEC)',
     1//,40X,'SENSOR POSITIONS',/)
 1100 FORMAT(5X,I2,3(10X,F10.3))
 1200 FORMAT(//,40X,'VEHICLE (SOURCE) PATH',/)
 1300 FORMAT(5X,I2,3(10X,F10.3))
      RETURN
      END

                  TABULR          SIZE      280     00430
                        MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA SR
   61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

      SUBROUTINE SENSOR(I,DD)
      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS SUBROUTINE IS TO BE MADE TO COLLECT REAL DATA FOR TIME LAGS
C     VIA PHASE COHERENCE DISCRIMINATION/ CROSS CORRELATION/ OR TIME
C     DIFFERENCE MEASUREMENTS.  DATA TO BE INPUT VIA A/D CONVERTOR TO
C     TRACKING COMPUTER.
C
      WRITE(50,100)
  100 FORMAT(20X,'***** WARNING *****',//,15X,
     1'PROGRAM TRYING TO ACCESS THE REAL DATA.',/,15X,
     2'THIS PART OF THE PROGRAM TO REMAIN INCOMPLETE AT THIS TIME',15X,
```

```
      3/,15X,'DUE TO LACK OF PROPER HARDWARE IMPLEMENTATION.')
       RETURN
       END

                    SENSOR          SIZE      101     00145
                          MODULE:   *MAIN*   OPTIONS: SE AB WA LO MA SR
  61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

       SUBROUTINE TRKPLT(ANS,IDD)
       COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
       COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
       COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
       COMMON /ZL4/ FN(25,50),IXAX(10),IYAX(5),IZAX(5),IYZAX(10)
C
C      THIS ROUTINE DETERMINES THE GRAPHIC DISPLAY REQUIREMENTS FOR
C      THE LINE PRINTER DISPLAY OF THE TRACKING SCENARIO.
C
C      INPUT GRAPH SPECIFICATIONS
C
       WRITE(50,100)
       READ(50,) SF
       WRITE(60,50505) SF
       WRITE(50,115)
       READ(50,) WX
       WRITE(60,40404) WX
       WRITE(50,125)
       READ(50,) HY
       WRITE(60,30303) HY
       WRITE(50,135)
       READ(50,) BZ
       WRITE(60,20202) BZ
C
C      SET OUTPUT TO BLANKS
C
       DO 8150 I=1,25
```

```
      DO 8100 J=1,50
      FN(I,J)=' '
 8100 CONTINUE
 8150 CONTINUE
C
C     SET X-AXIS:  FOR Y VS X, OR Z VS X.
C
      FAKE=5.0
      XADIV=WX/50.0
      DO 1000 I=1,10
      IXAX(I)=(FAKE*XADIV/SF)
      FAKE=FAKE+5.0
 1000 CONTINUE
C
C     SET Y-AXIS:  FOR Y VS X.
C
      FAKE=5.0
      YADIV=HY/25.0
      DO 1010 I=1,5
      IYAX(I)=FAKE*YADIV/SF
      FAKE=FAKE+5.0
 1010 CONTINUE
C
C     STORE STANDARD AXIS GRID.
C
```



```
      DO 1045 J=5,45,5
      DO 1040 L=1,25
      FN(L,J)=':'
 1040 CONTINUE
 1045 CONTINUE
      DO 1055 J=5,20,5
      DO 1050 L=1,50
```

```
              FN(J,L)='-'
       1050 CONTINUE
       1055 CONTINUE
            DO 1060 M=1,25
            FN(M,1)='I'
            FN(M,50)='I'
       1060 CONTINUE
            DO 1070 M=1,50
            FN(1,M)='I'
            FN(25,M)='I'
       1070 CONTINUE
      C
      C     STORE PARTICLE PATH.
      C
            DO 1080 MI=1,NPTS
            IX=((XTRK(MI)/XADIV)*SF)+0.5
            IY=((YTRK(MI)/YADIV)*SF)+0.5
            IF(IX.LE.50.AND.IY.LE.25.AND.IDD,EQ.1) FN(IY,IX)='T'
       1080 CONTINUE
      C
      C     STORE SENSOR GRID POSITIONS.
      C
            DO 1090 LI=1,3
            LX=((XSENS(LI)/XADIV)*SF)+0.5
            LY=((YSENS(LI)/YADIV)*SF)+0.5
            IF(LX.LE.50.AND.LY.LE.25.AND.LI.EQ.1) FN(LY,LX)='1'
            IF(LX.LE.50.AND.LY.LE.25.AND.LI.EQ.2) FN(LY,LX)='2'
            IF(LX.LE.50.AND.LY.LE.25.AND.LI.EQ.3) FN(LY,LX)='3'
       1090 CONTINUE
      C
      C     DUMP GRAPHS AND APPROPRIATE LABELS.
      C
      C     FIRST:  Y VS X:
      C
      C     GT=1 FOR Y VS X
```

```
C
      F=0.0
      DO 2100 I=1,25
      CALL GRAPHO(I,F,1,SF)
 2100 CONTINUE
      WRITE(50,140)
      READ(50,) ANS
  100 FORMAT(10X,'SCALE FACTOR=?')
50505 FORMAT(1H1,10X,'SCALE FACTOR= ',F10.5)
  115 FORMAT(/,10X,'X MAX GRAPH= ',5X,'??')
                         MODULE:  TRKPLT    OPTIONS: SE AB WA LO MA SR
 61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

40404 FORMAT(/,10X,'X MAX= ',F10.5)
  125 FORMAT(/,10X,'Y MAX GRAPH= ',5X,'??')
30303 FORMAT(1H@,T35,'Y MAX= ',F10.5)
  135 FORMAT(/,10X,'Z MAX GRAPH= ',5X,'??')
20202 FORMAT(1H@,T60,'Z MAX= ',F10.5)
  140 FORMAT(/,10X,'DO YOU WANT TO RUN ADDITIONAL PLOTS?',/,
     120X,'1= YES',/,20X,'2= NO',/,'??')
      RETURN
      END

                  TRKPLT          SIZE      907     01613
                         MODULE:  *MAIN*    OPTIONS: SE AB WA LO MA SR
 61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

      SUBROUTINE GRAPHO(I,F,NI,SF)
      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
      COMMON /ZL4/ FN(25,50),IXAX(10),IYAX(5),IZAX(5),IYZAX(10)
C
C     GRAPH FUNCTION AND OUTPUT LABELS.
C
```

```
      II=26-I
      IF(F.NE.0.0) GO TO 10
      IF(NI.NE.3) WRITE(60,100) (IXAX(L),L=1,10)
  100 FORMAT(1H1,T49,'X AXIS (METERS)',//,T32,10(I3,2X))
      F=1.0
   10 IF(I.EQ.18.OR.I.EQ.11) GO TO 131
      GO TO 133
  131 WRITE(60,212)
      GO TO 22
  133 IF(I.NE.17) GO TO 13
      WRITE(60,2133)
 2133 FORMAT(T20,'R')
      GO TO 22
   13 IF(I.NE.15) GO TO 14
      WRITE(60,203)
      GO TO 22
   14 IF(I.EQ.14.OR.I.EQ.16) GO TO 155
      GO TO 15
  155 WRITE(60,204)
      GO TO 22
   15 IF(I.NE.13) GO TO 16
      WRITE(60,205)
      GO TO 22
   16 IF(I.NE.10) GO TO 17
      WRITE(60,206)
      GO TO 22
   17 IF(I.NE.9) GO TO 18
      WRITE(60,207)
      GO TO 22
   18 IF(I.NE.8) GO TO 19
      WRITE(60,208)
      GO TO 22
   19 IF(NI.EQ.1.AND.I.EQ.6) GO TO 202
      GO TO 20
  202 WRITE(60,209)
```

```
          GO TO 22
       20 IF((NI.EQ.2.OR.NI.EQ.3).AND.I.EQ.6) GO TO 211
          GO TO 21
      211 WRITE(60,210)
          GO TO 22
       21 WRITE(60,219)
       22 INX=II/5
          IF(MOD(II,5).EQ.0.AND.NI.EQ.1) WRITE(60,300) IYAX(INX)
          WRITE(60,400) (FN(II,J),J=1,50)
                          MODULE:  GRAPHO   OPTIONS: SE AB WA LO MA SR
   61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

          IF(II.NE.1) GO TO 555
          WRITE(60,700) SF,XLAT,YLONG,ZDEPTH
      555 CONTINUE
      212 FORMAT(T20,'S')
      203 FORMAT(T20,'T')
      204 FORMAT(T20,'E')
      205 FORMAT(T20,'M')
      206 FORMAT(T20,'I')
      207 FORMAT(T20,'X')
      208 FORMAT(T20,'A')
      209 FORMAT(T20,'Y')
      210 FORMAT(T20,'Z')
      219 FORMAT(T20,' ')
      300 FORMAT(1H@,T25,I3)
      400 FORMAT(1H@,T30,50(1A1))
      700 FORMAT(//,T22,'SF= ',F6.2,5X,'XMAX= ',F6.2,5X,'YMAX= ',F6.2,5X,
         1'ZMAX= ',F6.2)
          RETURN
          END

                GRAPHO          SIZE      570     01072

$CATALOG
```

```
61632-02 OVERLAY CATALOGER -- REVISION LEVEL 12.071978
TYPE=BG
DNAME ATRCK,4
BEGIN
            P.NAME=  ATRCK

*HIGH      25006              *START          0
*LOW           0              *LOW            0
*P.END     16631              .PASS           2
*START         0              GUESS1       1231 $
.PASS          2              TRKPT        1447 $
C$ERRO     16271 $            SYSEQN       2242 $
C$IOER     16302 $            CRAMER       3335 $
C$MOD      16256 $            MTX          3641 $
C$OFLO     16273 $            PTCHK        3726 $
C$UFLO     16277 $            TABULR       4416 $
CRAMER      3335 $            SENSOR       5046 $
DCF        14522 $            TRKPLT       5213 $
DSQRT      16264 $            GRAPHO       7026 $
ESGN$      14173 $            F$SERA      10142 $
EXIT       16101 $            F$SERI      10147 $
EXP10$     14172 $            F$SERJ      10154 $
F$1ISN     10706 $            F$SERK      10161 $
F$1OSN     11105 $            F$RIFF      10222 $
F$4ISN     11226 $            F$RSFF      10222 $
F$4OSN     11334 $            F$RSSY      10223 $
F$5ISN     11224 $            F$WIFF      10225 $
F$5OSN     11332 $            F$WSFF      10225 $
F$8ISN     11222 $            F$WSSY      10226 $
F$8OSN     11330 $            F$DIFF      10272 $
F$ADDR     14407 $            F$DSSY      10273 $
F$AIP      11455 $            F$EIFF      10275 $
F$AISN     10702 $            F$ESSY      10276 $
F$AOP      12600 $            F$EESY      10376 $
F$B/E      10606 $            F$SYEE      10376 $
```

F$BBI	16605	$	F$RORW	10411	$
F$BBJ	16536	$	F$FUNC	10514	$
F$BCC	14417	$	F$B/E	10606	$
F$BISN	10671	$	F$CSIO	10616	$
F$BOSN	11074	$	F$CFIO	10616	$
F$BUF	14415	$	F$BISN	10671	$
F$CFIO	10616	$	F$AISN	10702	$
F$CLOS	16156	$	F$1ISN	10706	$
F$CNOP	16103	$	F$BOSN	11074	$
F$CONV	14425	$	F$1OSN	11105	$
F$CSIO	10616	$	F$8ISN	11222	$
F$CV41	16240	$	F$5ISN	11224	$
F$CV51	16240	$	F$4ISN	11226	$
F$D	14412	$	F$8OSN	11330	$
F$DCF	14522	$	F$5OSN	11332	$
F$DCNT	15216	$	F$4OSN	11334	$
F$DIFF	10272	$	F$AIP	11455	$
F$DSSY	10273	$	F$RIP	11457	$
F$DVR5	15343	$	F$LIP	11552	$
F$E	14413	$	F$OIP	11604	$
F$E/E	14423	$	F$IIP	11642	$
F$E/E1	14424	$	F$GIP	12112	$
F$E/O	14421	$	F$ROP	12576	$
F$E/O1	14422	$	F$AOP	12600	$
F$EESY	10376	$	F$IOP	12664	$
F$EIFF	10275	$	F$FOP	13160	$
F$EMB	16506	$	F$QOP	13350	$
F$EMBJ	16520	$	F$SOP	13355	$
F$EOP	13533	$	F$O.LB	13364	$
F$ERCD	16235	$	F$O.SN	13370	$
F$ESSY	10276	$	F$OOP	13403	$
F$FCC	14416	$	F$GOP	13531	$
F$FERR	14430	$	F$EOP	13533	$
F$FMT	14414	$	TENTd$	14047	$

```
F$FMTC    14410 $        EXP10$    14172 $
F$FOP     13160 $        ESGN$     14173 $
F$FUNC    10514 $        MSGN$     14174 $
F$GIP     12112 $        F$NORM    14177 $
F$GOP     13531 $        F$ADDR    14407 $
F$IBF     14426 $        F$SPEC    14410 $
F$IIP     11642 $        F$FMTC    14410 $
F$IOM0    16006 $        F$W       14411 $
F$IOM1    15772 $        F$D       14412 $
F$IOM2    15774 $        F$E       14413 $
F$IOM3    16104 $        F$FMT     14414 $
F$IOM4    16113 $        F$BUF     14415 $
F$IOP     12664 $        F$FCC     14416 $
F$LINE    15537 $        F$BCC     14417 $
F$LIP     11552 $        F$P       14420 $
F$MPR5    15247 $        F$E/0     14421 $
F$NAME    15444 $        F$E/01    14422 $
F$NCHR    14431 $        F$E/E     14423 $
F$NORM    14177 $        F$E/E1    14424 $
F$NTRY    15447 $        F$CONV    14425 $
F$0.LB    13364 $        F$IBF     14426 $
F$0.SN    13370 $        F$FERR    14430 $
F$0IP     11604 $        F$NCHR    14431 $
F$00P     13403 $        F$PNL     14503 $
F$OPEN    16121 $        F$DCF     14522 $
F$P       14420 $        DCF       14522 $
F$PNL     14503 $        F$DCNT    15216 $
F$QOP     13350 $        PUTN$     15221 $
F$RBM     16513 $        F$MPR5    15247 $
F$RBMI    16556 $        F$DVR5    15343 $
F$RIFF    10222 $        F$NAME    15444 $
F$RIP     11457 $        F$NTRY    15447 $
F$ROP     12576 $        F$STMT    15510 $
F$RORW    10411 $        F$RSD0    15514 $
F$RSD0    15514 $        F$SRTN    15517 $
```

F$RSFF	10222	$	F$LINE	15537	$
F$RSSY	10223	$	F$IOM1	15772	$
F$SERA	10142	$	F$IOM2	15774	$
F$SERI	10147	$	F$IOM0	16006	$
F$SERJ	10154	$	EXIT	16101	$
F$SERK	10161	$	F$CNOP	16103	$
F$SOP	13355	$	F$IOM3	16104	$
F$SPEC	14410	$	F$IOM4	16113	$
F$SRTN	15517	$	F$OPEN	16121	$
F$STMT	15510	$	F$CLOS	16156	$
F$SYEE	10376	$	IOERR	16233	$
F$W	14411	$	F$ERCD	16235	$
F$WIFF	10225	$	INT	16236	$
F$WSFF	10225	$	IDINT	16236	$
F$WSSY	10226	$	IFIX	16236	$
GRAPHO	7026	$	IFIX$	16236	$
GUESS1	1231	$	F$CV41	16240	$
IDINT	16236	$	F$CV51	16240	$
IFIX	16236	$	MOD$	16252	$
IFIX$	16236	$	C$MOD	16256	$
INT	16236	$	SQRT	16264	$
IOERR	16233	$	DSQRT	16264	$
MOD$	16252	$	C$ERR0	16271	$
MSGN$	14174	$	C$OFLO	16273	$
MTX	3641	$	C$UFLO	16277	$
PTCHK	3726	$	C$IOER	16302	$
PUTN$	15221	$	F$EMB	16506	$
SENSOR	5046	$	F$RBM	16513	$
SQRT	16264	$	F$EMBJ	16520	$
SYSEQN	2242	$	F$BBJ	16536	$
TABULR	4416	$	F$RBMI	16555	$
TENTB$	14047	$	F$BBI	16605	$
TRKPLT	5213	$	*P.END	16631	
TRKPT	1447	$	ZL1	16631	C
ZL1	16631	C	ZL4	17616	C

```
ZL2        24602 C          ZL3        24560 C
ZL3        24560 C          ZL2        24602 C
ZL4        17616 C          *HIGH      25006
$EOJ
```

APPENDIX C

TWO SPACE SIMULATION DATA, pp. 79-80.

DESCRIPTION:

This appendix contains a tabular listing and disk file data dump for data generated as a result of the program of Appendix A.

VELOCITY OF MEDIUM IS 1450.000 (M/SEC)

SENSOR POSITIONS

1	75.000	100.000	0.000
2	175.000	75.000	0.000
3	200.000	200.000	0.000

VEHICLE (SOURCE) PATH

1	80.000	10.000	0.000
2	100.000	40.000	0.000
3	110.000	70.000	0.000
4	110.000	100.000	0.000
5	100.000	120.000	0.000
6	80.000	140.000	0.000
7	90.000	165.000	0.000
8	110.000	175.000	0.000
9	130.000	180.000	0.000
10	155.000	170.000	0.000
11	175.000	150.000	0.000
12	200.000	135.000	0.000
13	225.000	150.000	0.000
14	239.000	175.000	0.000
15	250.000	210.000	0.000
16	239.000	239.000	0.000
17	250.000	245.000	0.000
18	265.000	255.000	0.000
19	280.000	245.000	0.000
20	310.000	235.000	0.000
21	340.000	225.000	0.000
22	365.000	220.000	0.000
23	375.000	210.000	0.000
24	390.000	200.000	0.000
25	410.000	180.000	0.000

25

-0.017220594790	-0.092816047347	-0.075595452558
-0.012251535714	-0.085296291476	-0.073044755762
-0.013168483976	-0.077252524636	-0.064084040660
-0.023890994059	-0.068645683083	-0.044754689024
-0.038240453266	-0.066239216249	-0.027998762983
-0.051584383166	-0.064726062005	-0.013141678839
-0.039369561179	-0.033603892059	0.005765669121
-0.025175088022	-0.007339987362	0.017835100660
-0.011830453375	0.016745751932	0.028576205307
0.006357944092	0.036012544135	0.029654600043
0.025381654396	0.038552896164	0.013171241767
0.044694861970	0.044694861970	0.000000000000
0.046879380084	0.070491161084	0.023611780999
0.042489118105	0.092421281794	0.049932163689
0.036045388945	0.107386255820	0.071340866882
0.026852376460	0.110225562490	0.083373186031
0.028591256405	0.110343616070	0.081752359672
0.030315778434	0.110384182670	0.080068404239
0.035369088664	0.109866821950	0.074499733292
0.042532683105	0.107298373000	0.064765689896
0.048283224324	0.103991089610	0.055707865294
0.051612916190	0.101820272510	0.050207356323
0.053953335126	0.099479603271	0.045526268145
0.056410658679	0.096891074063	0.040480415385
0.060020033422	0.092047979923	0.032027946502
75.0000	100.0000	0.0000
175.0000	75.0000	0.0000
200.0000	200.0000	0.0000

APPENDIX D

TWO SPACE COMPUTER DATA SOLUTION, pp. 82-91.

DESCRIPTION:

This appendix contains a tabular solution and various plots of the two-space trajectory found by use of the program of Appendix B on the disk data of Appendix C. The following key has been used in the plots:

T - Target location;
1 - Sensor 1 location;
2 - Sensor 2 location; and
3 - Sensor 3 location.

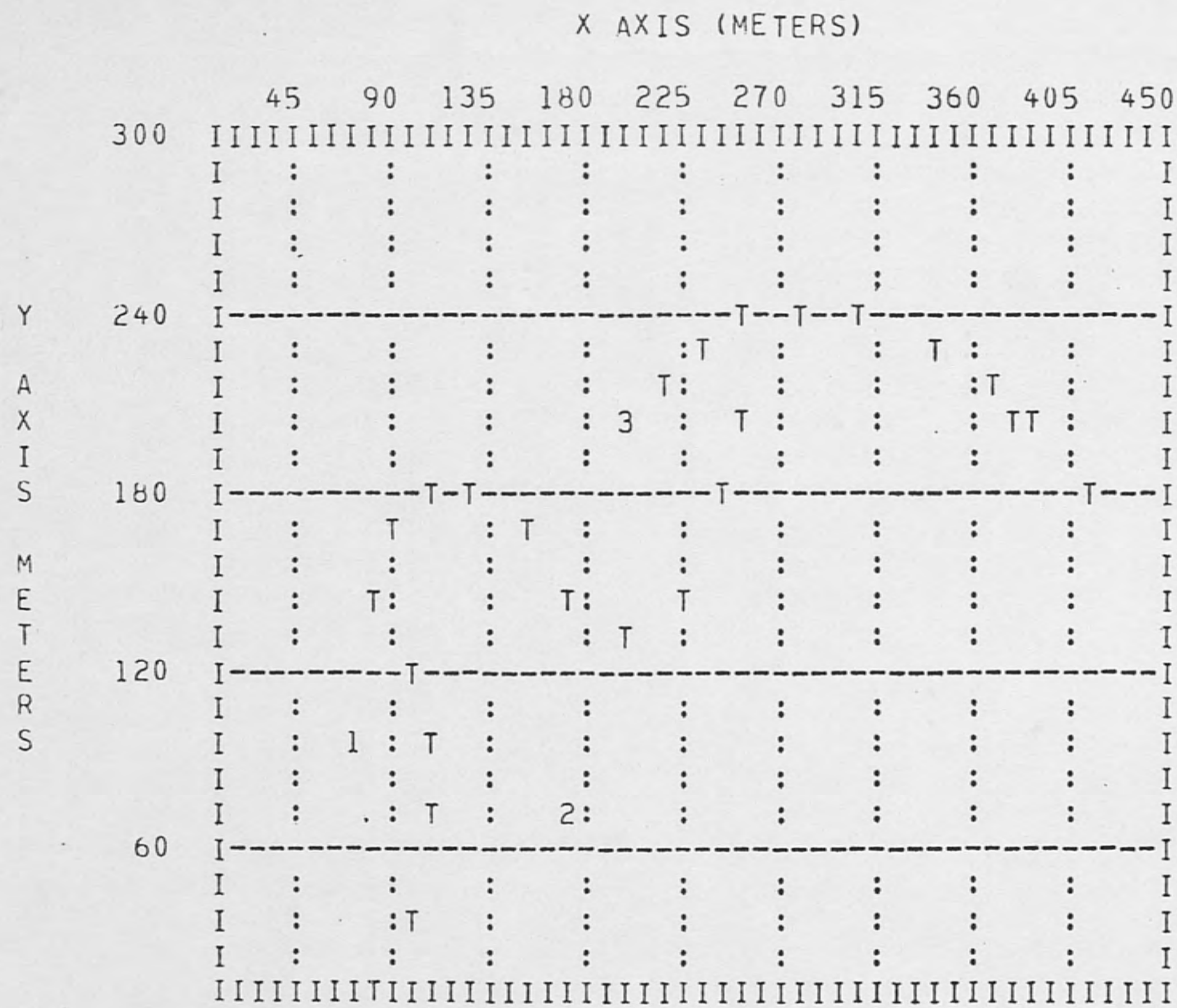
VELOCITY OF MEDIUM IS 1450.000 (M/SEC)

SENSOR POSITIONS

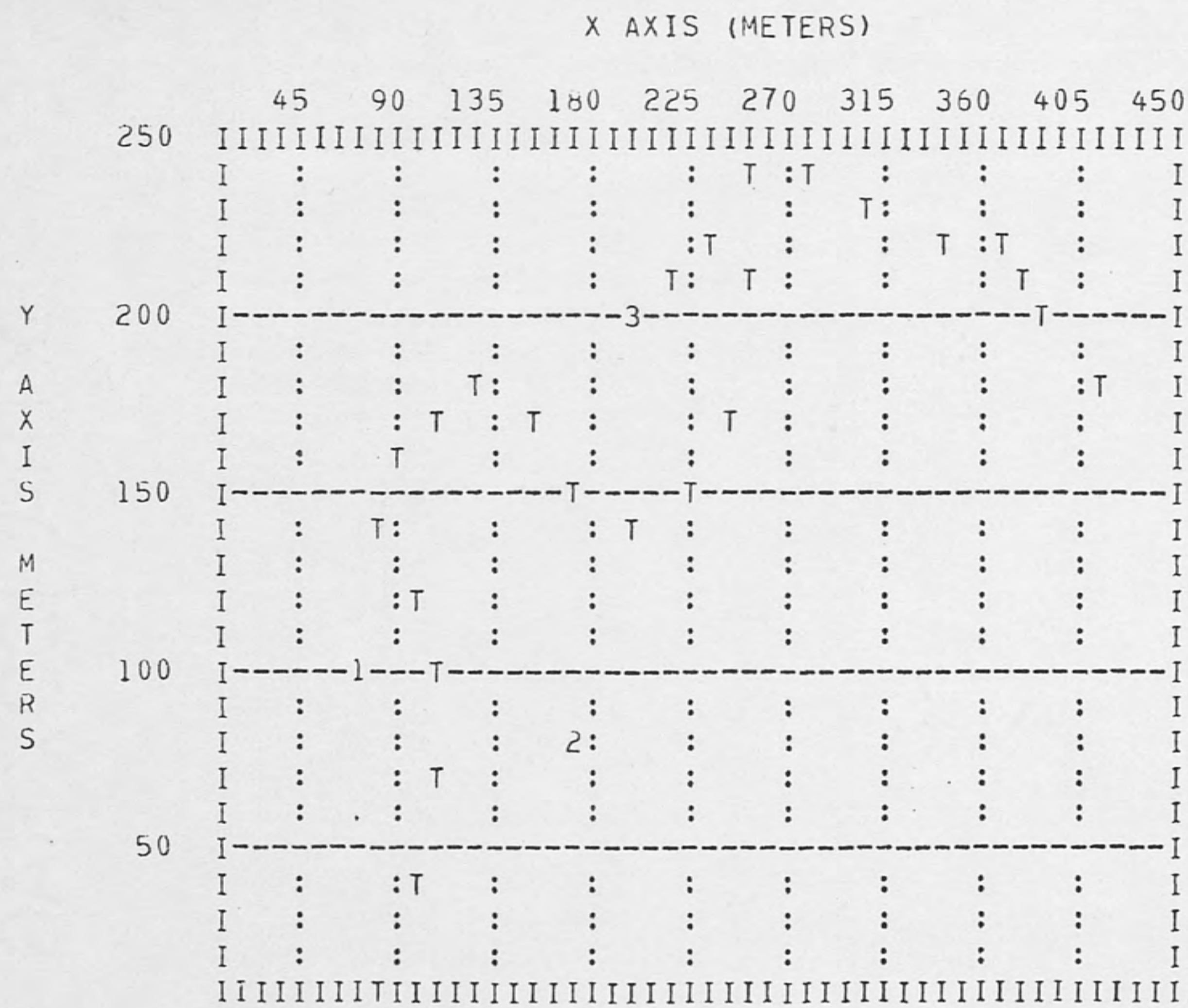
1	75.000	100.000	0.000
2	175.000	75.000	0.000
3	200.000	200.000	0.000

VEHICLE (SOURCE) PATH

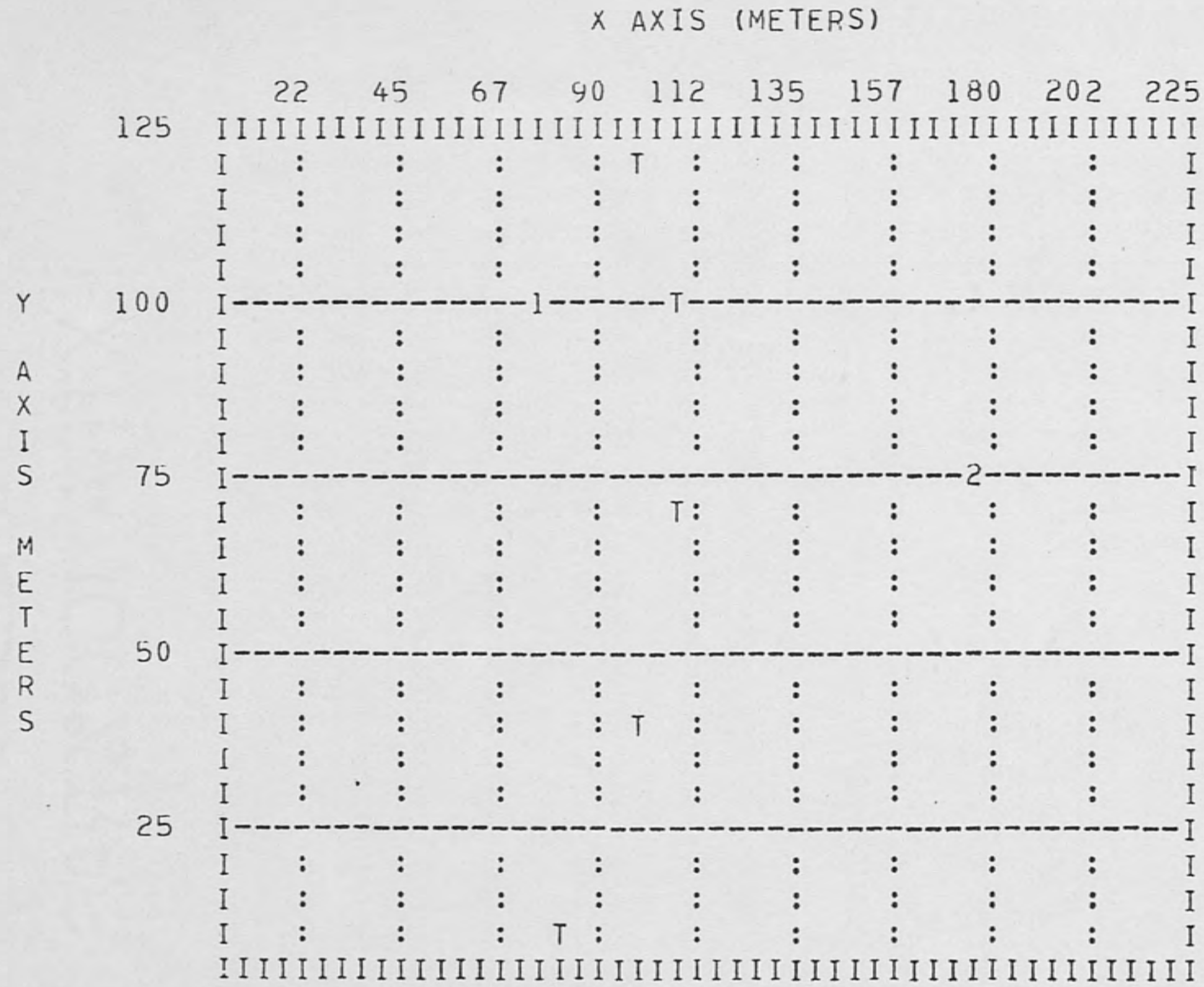
1	80.000	10.000	0.000
2	100.000	40.000	0.000
3	110.000	70.000	0.000
4	110.000	100.000	0.000
5	100.000	120.000	0.000
6	80.000	140.000	0.000
7	90.000	165.000	0.000
8	110.000	175.000	0.000
9	130.000	180.000	0.000
10	155.000	170.000	0.000
11	175.000	150.000	0.000
12	200.000	135.000	0.000
13	225.000	150.000	0.000
14	239.000	175.000	0.000
15	250.000	210.000	0.000
16	218.838	210.593	0.000
17	233.596	223.215	0.000
18	252.745	239.660	0.000
19	280.000	245.000	0.000
20	310.000	235.000	0.000
21	340.000	225.000	0.000
22	365.000	220.000	0.000
23	375.000	210.000	0.000
24	390.000	200.000	0.000
25	410.000	180.000	0.000



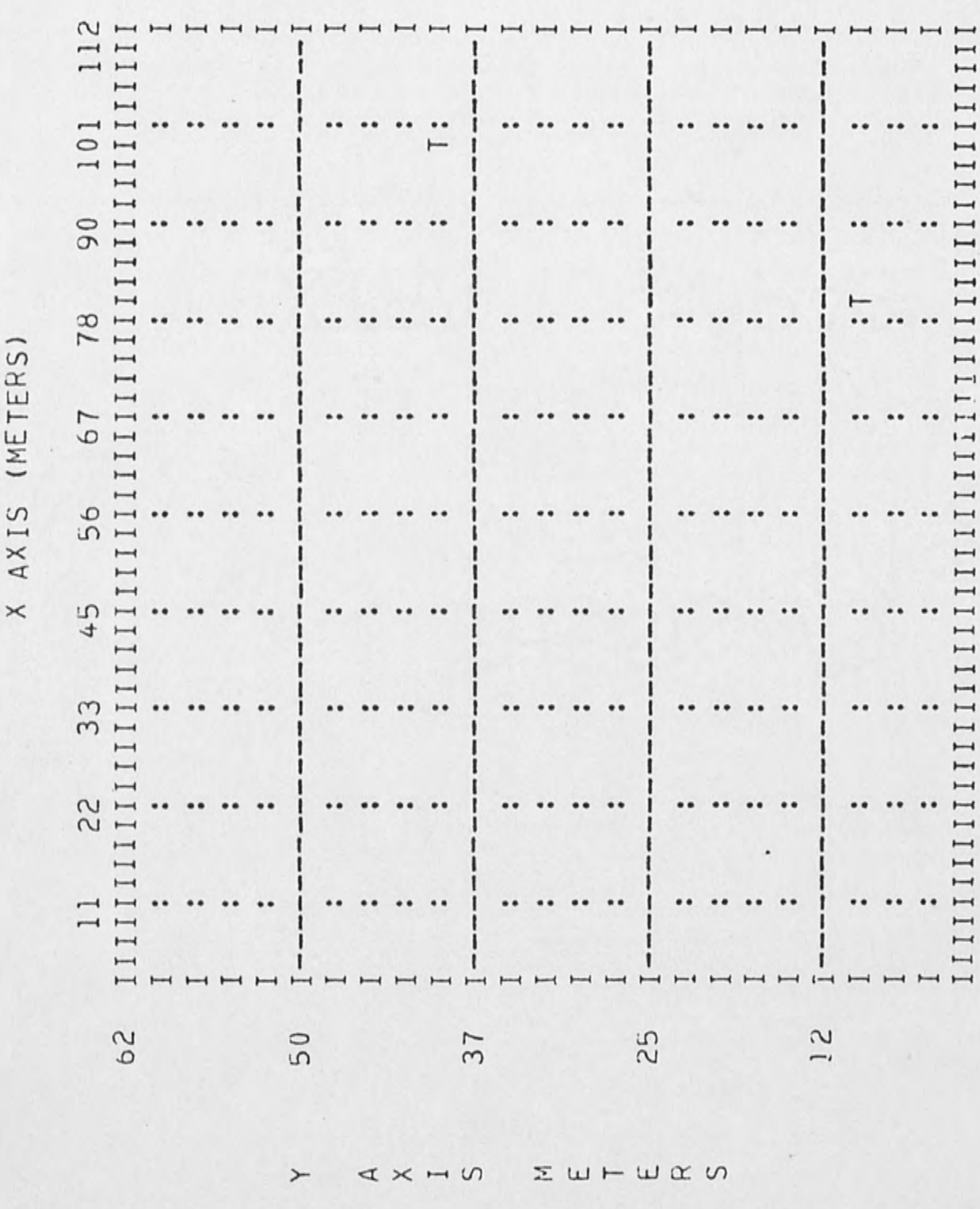
SF= 1.00 XMAX= 500.00 YMAX= 300.00 ZMAX= 300.00



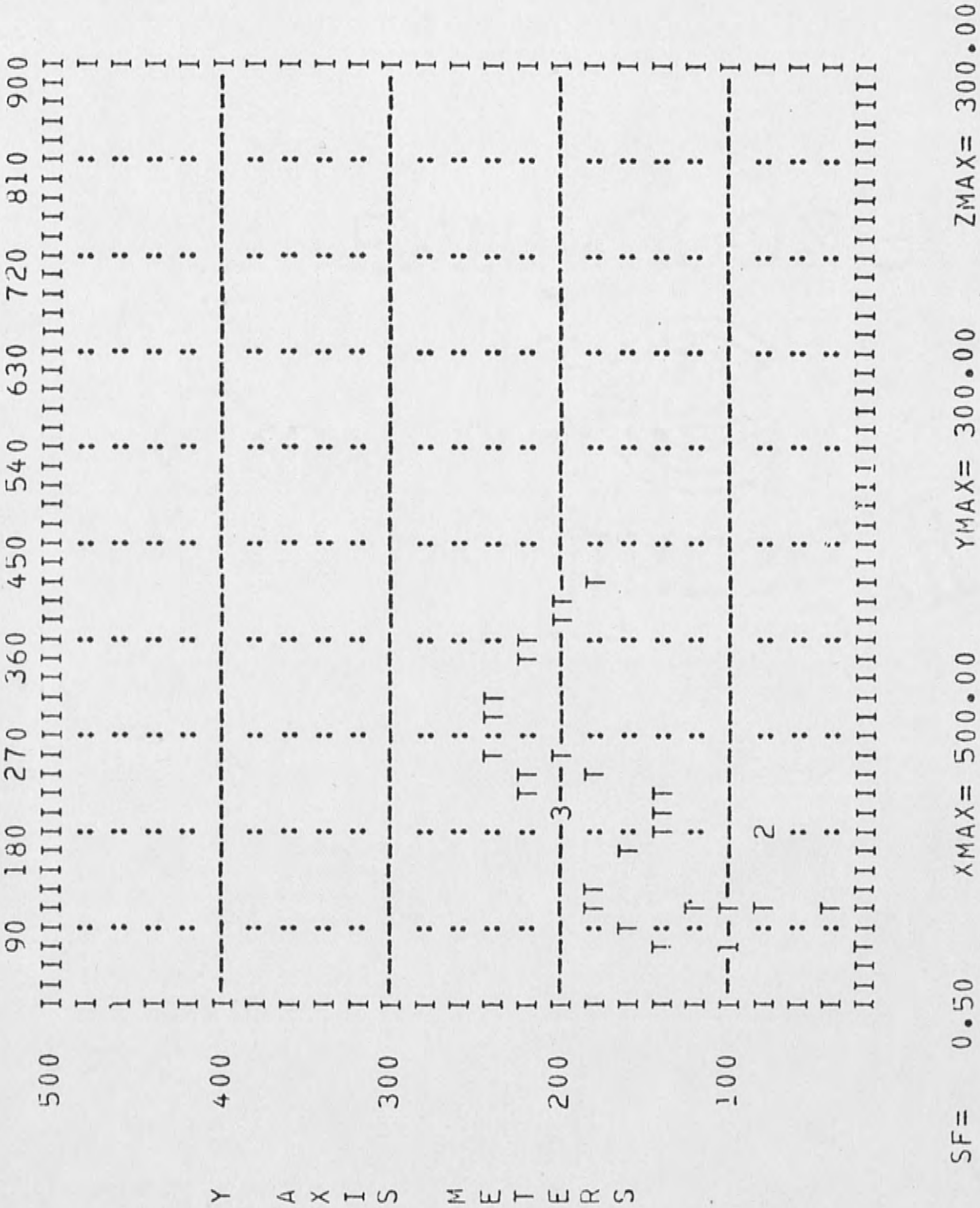
SF= 1.00 XMAX= 500.00 YMAX= 300.00 ZMAX= 300.00

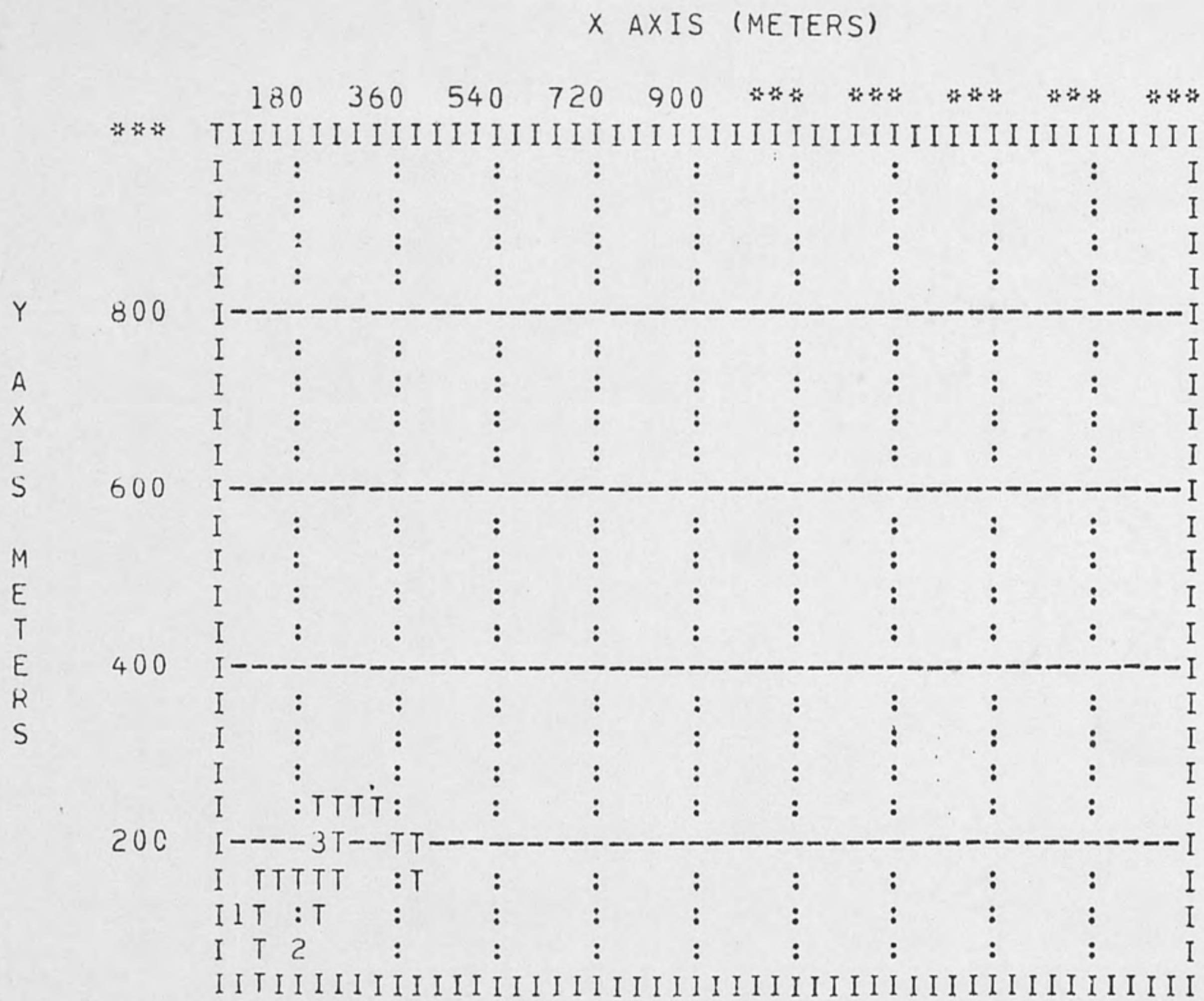


SF= 2.00 XMAX= 500.00 YMAX= 300.00 ZMAX= 300.00

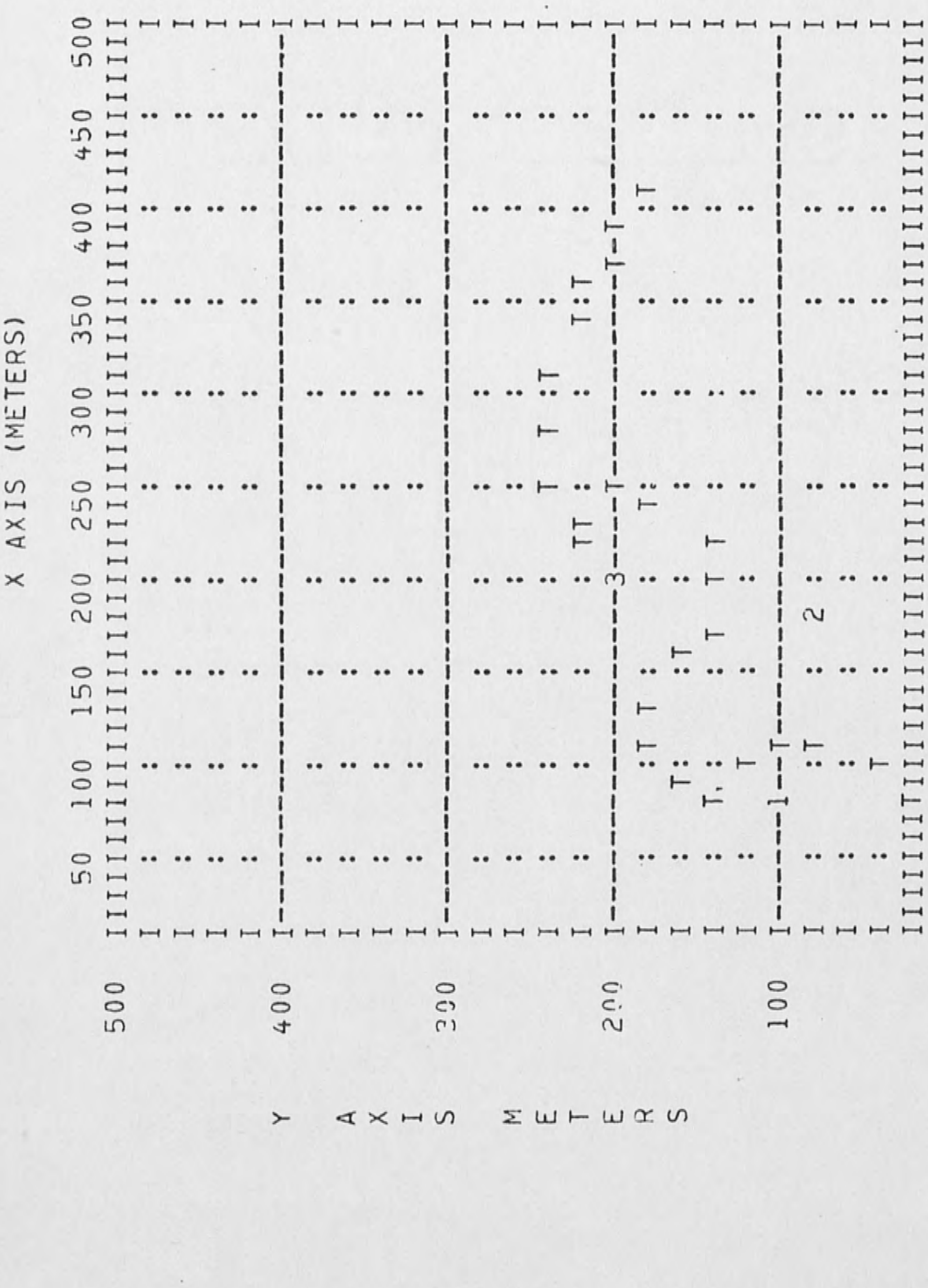


SF= 4.00 XMAX= 500.00 YMAX= 300.00 ZMAX= 300.00

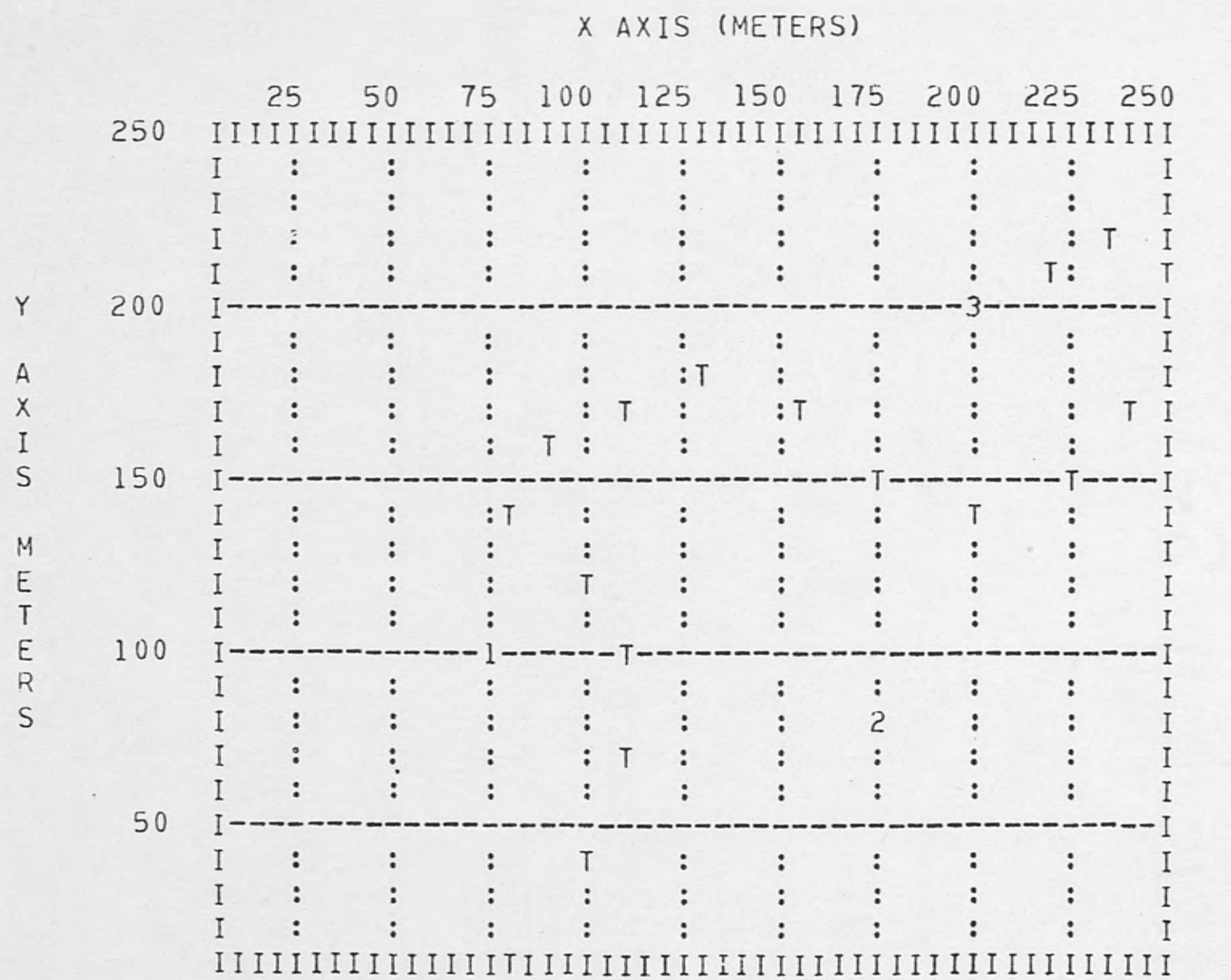

X AXIS (METERS)
90 180 270 360 450 540 630 720 810 900
Y AXIS METERS
500
400
300
200
100
SF= 0.50 XMAX= 500.00 YMAX= 300.00 ZMAX= 300.00



SF= 0.25 XMAX= 500.00 YMAX= 300.00 ZMAX= 300.00



SF= 1.00 XMAX= 500.00 YMAX= 500.00 ZMAX= 500.00



SF= 2.00 XMAX= 500.00 YMAX= 500.00 ZMAX= 500.00

```
                          X AXIS (METERS)

              100  200  300  400  500  600  700  800  900  ***
       ***  IITIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
            I   :    :    :    :    :    :    :    :    :    I
            I   :    :    :    :    :    :    :    :    :    I
            I   :    :    :    :    :    :    :    :    :    I
            I   :    :    :    :    :    :    :    :    :    I
  Y    800  I------------------------------------------------I
            I   :    :    :    :    :    :    :    :    :    I
  A         I   :    :    :    :    :    :    :    :    :    I
  X         I   :    :    :    :    :    :    :    :    :    I
  I         I   :    :    :    :    :    :    :    :    :    I
  S    600  I------------------------------------------------I
            I   :    :    :    :    :    :    :    :    :    I
  M         I   :    :    :    :    :    :    :    :    :    I
  E         I   :    :    :    :    :    :    :    :    :    I
  T         I   :    :    :    :    :    :    :    :    :    I
  E    400  I------------------------------------------------I
  R         I   :    :    :    :    :    :    :    :    :    I
  S         I   :    :    :    :    :    :    :    :    :    I
            I   :   .:    :    :    :    :    :    :    :    I
            I   :    : TTTT T  :    :    :    :    :    :    I
       200  I--------3TT-----TT------------------------------I
            I   TTTTT:TT  :    T    :    :    :    :    :    I
            I  1TT    T   :    :    :    :    :    :    :    I
            I   :T  2:    :    :    :    :    :    :    :    I
            IIIITIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII

SF=    0.50      XMAX= 500.00      YMAX= 500.00      ZMAX= 500.00
```

APPENDIX E

THREE SPACE: METHOD ONE: PROGRAM, pp. 93-117.

DESCRIPTION:

This program was designed to solve the three space tracking program, solving for all three coordinates simultaneously. This program was written to interpret disk data generated with three-space coordinates via the program of Appendix A. Typical data for this program application can be viewed in Appendix F. Problems encountered in this approach are discussed in Section V of this text.

```
$JOB TRACK DLIST4                                                  00:40:57
$OPTIONS 21,8.9,4
$FORTRAN
                           MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
  61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

   :WA,SR,AB,GO
         COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
         COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
         COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
         COMMON /ZL4/ FN(25,50),IXAX(10),IYAX(5),IZAX(5),IYZAX(10)
   C*************************************************************************
   C     THIS IS THE MAIN PROGRAM FOR THE ACOUSTIC MODEL TRACKING
   C     AND PLOT PROGRAM
   C
   C     IT IS BASED ON:
   C            3-SENSORS: KNOWN COORDINATES; WITH, RELATIVE PHASE
   C              COHERENCE - OR TIME LAGS OF ARRIVING WAVEFRONTS -
   C              AS RECEIVED DATA FROM AN UNKNOWN SOURCE WITH SOME
   C              RELATIVE REFERENCE FREQUENCY USED IN DETECTION OF
   C
   C              DELTA TIME DIFFERENCES BETWEEN SENSORS/SOURCE.
   C            1-SOURCE: OF UNKNOWN POSITIONS, EMITTING SOME REFERENCE
   C              SIGNAL FOR DETECTION.
   C
   C            ASSUMPTIONS: (1) DELTA TIME DIFFERENCES HAVE BEEN
   C              PROCESSED FOR INPUT TO THIS PROGRAM BY SOME RECEIVING
   C              PROCESSOR;(2) NO PROPAGATION LOSS DUE TO MEDIUM;
   C              (3) SOUND VELOCITY OF MEDIUM IS CONSTANT AND KNOWN; AN
   C              (4) NO MULTIPATH PHENOMENA, FOR A SOURCE-SENSOR PAIR.
   C
   C
   C            SOLUTION:
   C              USING ONLY THE ABOVE SCENARIO DESCRIPTORS, TRACK AND
   C              PLOT THE SOURCE.
```

```
C
C                   APPROACH:
C                     USING DISTANCE EQUATIONS FROM THE SENSORS TO THE
C                     UNKNOWN SOURCE, AND THE VELOCITY OF THE MEDIUM:
C                       -DEVELOP THE RESULTING NON-LINEAR HYPERBOLIC LOCATOR
C                        EQUATIONS;
C                       -SOLVE ITERATIVELY WITH A NEWTON-RAPHSON NON-LINEAR,
C                        SIMULTANEOUS EQUATION METHOD.
C
C                   THIS PROGRAM WRITTEN BY:
C                         GEORGE H. FORD, JR.
C                            JULY 3, 1980
C                   IN FULFILLMENT OF A RESEARCH TOPIC/REPORT FOR AN
C                   M.S.E. DEGREE IN THE U.C.F. DEPARTMENT OF ELECTRICAL
C                   ENGINEERING.
C
C                   SOURCE PROGRAM TYPED ON CARDS.
C                   COMPUTER:
C                       NAVTRAEQUIPCEN: HARRIS DATACRAFT 6024/4
C                       WITH LINE PRINTER PLOT SOFTWARE  WHICH WAS GENERATED
C                       FOR THIS PROJECT INCLUDED HEREIN.
C
C                   REVIEW BOARD:
                          MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
61516-02 ENHANCED SAU FORTRAN COMPILER         REVISION LEVEL 33.071978

C                       DR. FRED SIMONS  (CHAIRMAN)
C                       DR. MIKE HARRIS
C                       DR. CHRIS BAUER
C
C*******************************************************************
      DO 3 I=1,50
      XTRK(I)=0.0
      YTRK(I)=0.0
      ZTRK(I)=0.0
    3 CONTINUE
```

```
      5 WRITE(50,99)
        READ(50,) V
        REWIND 40
        READ(40,134) NPTS
        WRITE(50,105)
        READ(50,106) IZ
C
C       IZ=1 REAL DATA
C      IZ=0 CALCULATED, FILE STORAGE DATA: DATXYZ
C
C       INPUT TIME LAGS FROM REAL DATA/FILE DATA.
C
        IF(IZ.NE.1) GO TO 10
        WRITE(50,133)
        READ(50,134) NPTS
        DO 600 I=1,NPTS
        CALL SENSOR(I,DD)
    600 CONTINUE
        GO TO 46
     10 DO 601 I=1,NPTS
        READ(40,108) DTT12(I),DTT13(I),DTT23(I)
    601 CONTINUE
     46 NGES1=0
C
C       READ FILE DATA FOR SENSORS: X,Y,Z.
C
        DO 48 N=1,3
        READ(40,110) XSENS(N),YSENS(N),ZSENS(N)
     48 CONTINUE
C
C       INPUT/CALCULATE INITIAL GUESSES FOR X,Y,Z (SOURCE)
C
        WRITE(50,109)
        READ(50,) XLAT,YLONG,ZDEPTH
     51 CALL GUESS1(NGES1,XI,YI,ZI)
        CALL TABULR(DD)
```

```
      DO 6948 IDD=1,3
   52 CALL TRKPLT(ANS,IDD)
      IF(ANS.EQ.1.0) GO TO 52
 6948 CONTINUE
      WRITE(50,111)
      READ(50,) ANS
                  MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
61516-02 ENHANCED SAU FORTRAN COMPILER      REVISION LEVEL 33.071978

      IF(ANS.EQ.1.0) GO TO 5
C
C     FORMATS: MAIN.
C
   99 FORMAT(5X,'INPUT THE MAGNITUDE OF VELOCITY OF THE MEDIUM',/,
     15X,' (METERS/SEC)',/,15X,'WATER= 1450 M/SEC    AIR= 331 M/SEC')
  134 FORMAT(I2)
  105 FORMAT(5X,'INPUT THE FOLLOWING FOR TYPE PROGRAM:',/,15X,
     1'1= REAL DATA (TIME LAGS VIA PHASES PROCESSED)',/,15X,
     2'0= FILE DATA VIA DATXYZ PROGRAM')
  106 FORMAT(I1)
  133 FORMAT(5X,'HOW MANY TRACKING POINTS DESIRED? (MAX=50)')
  108 FORMAT(3(5X,F15.12))
  109 FORMAT(5X,'INPUT:',/,10X,'MAXIMUM X METERS(LATITUDE RANGE)',
     1/,10X,'MAXIMUM Y METERS(LONGITUDE RANGE)',/,10X,
     2'MAXIMUM Z METER(DEPTH)',//,10X,'XXX.XX',5X,'YYY.YY',5X,'ZZZ.ZZ')
  110 FORMAT(3(5X,F10.4))
  111 FORMAT(5X,'DO YOU WISH TO:',//,10X,
     1'1- CONTINUE PROCESSING; OR',/,10X,
     2'2- END PROCESSING',//,5X,'??')
      CALL EXIT
      END

            *MAIN*         SIZE      665    01231
                  MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
61516-02 ENHANCED SAU FORTRAN COMPILER      REVISION LEVEL 33.071978
```

```
      SUBROUTINE GUESS1(NGES,XI,YI,ZI)
      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS SUBROUTINE IS TO SUPPLY INITIAL GUESSES FOR TRACK PLOT OF
C     THE SOURCE VEHICLE.
C     UPON SUCCESSFUL TRACKING OF THE FIRST THREE POINTS, PROCESSING
C     SHALL CONTINUE FOR ALL REQUIRED TRACK POINTS IN THE
C     SUBROUTINE TRKPT.
C
C     VARIOUS COMBINATIONS OF V*DTT FACTORS ADDED TO XI, YI, ZI.
C     ASSUMPTION: ONE TIME LAG WILL DOMINATE ONE COORDINATE.
C     ALL INITIAL GUESSES PERTURBATED ABOUT CENTER OF UNDERWATER (4)
C     QUADRANTS.
C
  700 NGES=NGES+1
      FLAG=0.0
      GO TO 900
  880 CALL TRKPT(XI,YI,ZI,FLAG)
      IF(FLAG.EQ.1.0.AND.NGES.LT.24) GO TO 700
      IF(FLAG.EQ.1.0.AND.NGES.GE.24) GO TO 900
      GO TO 1000
  900 FLAG=0.0
      WRITE(50,2000)
      READ(50,) XI,YI,ZI
 2000 FORMAT(5X,'FAILURE TO CONVERGE FOR INITIAL SOLUTION',//,
     115X,'INPUT:',5X,'XXX.XX',5X,'YYY.YY',5X,'ZZZ.ZZ',5X,'GUESS')
      GO TO 880
C
C     FLAG=0.0 CLEARED
C         =1.0 NO CONVERGENCE
C         =2.0 TRACK COMPLETED
C
 1000 RETURN
      END
```

```
                  GUESS1          SIZE      142    00216
                         MODULE:  *MAIN*    OPTIONS: SE AB WA LO MA DO SR
61516-02 ENHANCED SAU FORTRAN COMPILER         REVISION LEVEL 33.071978

      SUBROUTINE TRKPT(XI,YI,ZI,FLAG)
      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS SUBROUTINE LOCATES TRACKING LOCATIONS FOR SPECIFIC
C     DATAPOINTS (TIME LAGS) RECEIVED.
C
      DX=0.0
      DO 200 N=1,NPTS
      TRYTWO=0.0
    5 IT=0
   10 IT=IT+1
      IF(IT.EQ.1.AND.N.EQ.1) GO TO 15
      IF(IT.NE.1) GO TO 20
      XTRK(N)=XTRK(N-1)
      YTRK(N)=YTRK(N-1)
      ZTRK(N)=ZTRK(N-1)
      IF(XTRK(N-1).EQ.0.0.AND.N.GE.3) XTRK(N)=XTRK(N-2)
      IF(YTRK(N-1).EQ.0.0.AND.N.GE.3) YTRK(N)=YTRK(N-2)
      IF(ZTRK(N-1).EQ.0.0.AND.N.GE.3) ZTRK(N)=ZTRK(N-2)
      GO TO 20
   15 Y=YI
      X=XI
      Z=ZI
      XTRK(N)=X
      YTRK(N)=Y
      ZTRK(N)=Z
   20 CONTINUE
      CALL SYSEQN(N,DDH,DDK,DDP,DD)
      USE=1.0
      CALL PTCHK(N,USE,DDH,DDK,DDP,DX)
```

```
C
C      DX=1.0  CORRECT SOLUTION POINT
C      DX=0.0  NO SOLUTION POINT FOUND.
C
       IF(DX.EQ.1.0) FLAG=0.0
       IF(DX.EQ.0.0) FLAG=1.0
       IF(DX.NE.1.0.AND.IT.LE.25) GO TO 10
       IF(DX.NE.1.0.AND.IT.GE.25) GO TO 150
       GO TO 180
   150 IF(TRYTWO.NE.1.0) GO TO 160
       XTRK(N)=0.0
       YTRK(N)=0.0
       ZTRK(N)=0.0
       GO TO 180
   160 TRYTWO=1.0
       XTRK(N)=XTRK(N-1)+5.0
       YTRK(N)=YTRK(N-1)+5.0
       ZTRK(N)=ZTRK(N-1)+5.0
       GO TO 5
   180 DX=0.0
                     MODULE:  TRKPT     OPTIONS: SE AB WA LO MA DO SR
 61516-02 ENHANCED SAU FORTRAN COMPILER     REVISION LEVEL 33.071978

   200 CONTINUE
   900 RETURN
       END

                TRKPT         SIZE      461    00715
                     MODULE:  *MAIN*    OPTIONS: SE AB WA LO MA DO SR
 61516-02 ENHANCED SAU FORTRAN COMPILER     REVISION LEVEL 33.071978

       SUBROUTINE SYSEQN(N,DDH,DDK,DDP,DD)
       COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
       COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
       COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
```

```
C     THIS SUBROUTINE SOLVES THE NONLINEAR SIMULTANEOUS SYSTEM
C     EQUATIONS DESCRIBING THE PARTICLE PATH DYNAMICS.  A NEWTON-RAPHSO
C     ALGORITHMIC SET OF EQUATIONS HAS BEEN IMPLEMENTED.
C
      ITSYS=0.0
  100 CONTINUE
      ITSYS=ITSYS+1.0
      DD1=(DTT12(N)*V)
      DD2=(DTT13(N)*V)
      DD3=(DTT23(N)*V)
      A=(XTRK(N)-XSENS(1))**2+(YTRK(N)-YSENS(1))**2
      A=A+(ZTRK(N)-ZSENS(1))**2
      B=(XTRK(N)-XSENS(2))**2+(YTRK(N)-YSENS(2))**2
      B=B+(ZTRK(N)-ZSENS(2))**2
      C=(XTRK(N)-XSENS(3))**2+(YTRK(N)-YSENS(3))**2
      C=C+(ZTRK(N)-ZSENS(3))**2
      EMF=DD1+SQRT(B)-SQRT(A)
      FMF=DD2+SQRT(C)-SQRT(A)
      GMF=DD3+SQRT(C)-SQRT(B)
      EX=((XTRK(N)-XSENS(1))/SQRT(A))-((XTRK(N)-XSENS(2))/SQRT(B))
      EY=((YTRK(N)-YSENS(1))/SQRT(A))-((YTRK(N)-YSENS(2))/SQRT(B))
      EZ=((ZTRK(N)-ZSENS(1))/SQRT(A))-((ZTRK(N)-ZSENS(2))/SQRT(B))
      FX=((XTRK(N)-XSENS(1))/SQRT(A))-((XTRK(N)-XSENS(3))/SQRT(C))
      FY=((YTRK(N)-YSENS(1))/SQRT(A))-((YTRK(N)-YSENS(3))/SQRT(C))
      FZ=((ZTRK(N)-ZSENS(1))/SQRT(A))-((ZTRK(N)-ZSENS(3))/SQRT(C))
      GX=((XTRK(N)-XSENS(2))/SQRT(B))-((XTRK(N)-XSENS(3))/SQRT(C))
      GY=((YTRK(N)-YSENS(2))/SQRT(B))-((YTRK(N)-YSENS(3))/SQRT(C))
      GZ=((ZTRK(N)-ZSENS(2))/SQRT(B))-((ZTRK(N)-ZSENS(3))/SQRT(C))
C
C     SOLVE FUNCTIONS AND PARTIALS VIA CRAMER'S RULE FOR DETERMINING
C     TRACK POINT, WITHIN A SPECIFIED TOLERANCE.
C
      XN(2)=XTRK(N)
      YN(2)=YTRK(N)
      ZN(2)=ZTRK(N)
      WRITE(60,1110)
```

```
      1110 FORMAT(3X,'N',8X,'EMF',8X,'FMF',8X,'GMF',8X,'EX',8X,'EY',8X,'EZ'
          18X,'FX',8X,'FY',8X,'FZ',8X,'GX',8X,'GY',/,8X,'GZ')
           CFLAG=0.0
           WRITE(60,1111) N,EMF,FMF,GMF,EX,EY,EZ,FX,FY,FZ,GX,GY,GZ
      1111 FORMAT(2X,I2,2X,11(E9.1,2X),/,3X,E9.1)
           CALL CRAMER(N,EMF,FMF,GMF,EX,EY,EZ,FX,FY,FZ,GX,GY,GZ,
          1DDH,DDK,DDP,CFLAG)
     C
     C     CFLAG=1.0  FOR FURTHER ACCURACY REQUIRED.
     C     CFLAG=2.0  FOR WRONSKIAN=0.0.
     C     CFLAG=3.0  FOR SOLUTION TOLERANCE ACHIEVED.
                          MODULE:  SYSEQN   OPTIONS: SE AB WA LO MA DO SR
  61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

     C
           IF(CFLAG.NE.2.0) GO TO 150
           XTRK(N)=0.0
           YTRK(N)=0.0
           ZTRK(N)=0.0
           GO TO 500
       150 IF(CFLAG.NE.1.0) GO TO 300
           IF(ITSYS.GE.25) GO TO 300
           XTRK(N)=XN(2)
           YTRK(N)=YN(2)
           ZTRK(N)=ZN(2)
           GO TO 100
       300 XTRK(N)=XN(2)
           YTRK(N)=YN(2)
           ZTRK(N)=ZN(2)
       500 RETURN
           END

                    SYSEQN          SIZE      757    01365
                          MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
  61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978
```

```
      SUBROUTINE CRAMER(N,EMF,FMF,GMF,EX,EY,EZ,FX,FY,FZ,GX,GY,GZ,
     1DDH,DDK,DDP,CFLAG)
      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS SUBROUTINE SOLVES FOR THE X,Y,Z, INCREMENTS VIA CRAMER'S RUL
C     FOR THE NEWTON-RAPHSON SYSTEM EQUATIONS.  AN X,Y,Z SOLUTION IS FO
C     WITHIN SOME ACCURACY; OR WITHIN 25 ITERATIONS - WHICHEVER IS FIRS
C
      XN(1)=XN(2)
      YN(1)=YN(2)
      ZN(1)=ZN(2)
      CALL MTX(FX,EX,GX,FY,EY,GY,FZ,EZ,GZ,WRON)
      IF(WRON.NE.0.0) GO TO 30
      CFLAG=2.0
      GO TO 900
   30 CALL MTX(FMF,EMF,GMF,FY,EY,GY,FZ,EZ,GZ,DH)
      CALL MTX(FX,EX,GX,FMF,EMF,GMF,FZ,EZ,GZ,DK)
      CALL MTX(FX,EX,GX,FY,EY,GY,FMF,EMF,GMF,DP)
      DDH=DH/WRON
      DDK=DK/WRON
      DDP=DP/WRON
      WRITE(60,1221) DDH,DDK,DDP
 1221 FORMAT(/,5X,'DDH= ',E12.4,5X,'DDK= ',E12.4,5X,'DDP= ',E12.4,//)
      XN(2)=XN(1)+DDH
      YN(2)=YN(1)+DDK
      ZN(2)=ZN(1)+DDP
      WRITE(60,4321) XN(2),YN(2),ZN(2)
 4321 FORMAT(10X,'X= ',E12.4,10X,'Y= ',E12.4,10X,'Z= ',E12.4)
      XH=ABS(DDH)
      YK=ABS(DDK)
      ZP=ABS(DDP)
      IF(XH.LE.0.5.AND.YK.LE.0.5.AND.ZP.LE.0.5) GO TO 800
      CFLAG=1.0
      GO TO 900
```

```
      800 CFLAG=3.0
      900 RETURN
          END

                    CRAMER              SIZE     275    00423
                           MODULE:   *MAIN*    OPTIONS: SE AB WA LO MA DO SR
 61516-02 ENHANCED SAU FORTRAN COMPILER          REVISION LEVEL 33.071978

          SUBROUTINE MTX(A,B,C,D,E,F,W,U,V,S)
          COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
          COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),VV
          COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C         THIS ROUTINE FINDS THE DETERMINANT OF A 3X3 MATRIX.
C
C         S=SOLUTION
C
C         A   D   W
C         B   E   U
C         C   F   V
C
          S=(A*E*V)+(B*F*W)+(C*D*U)-(C*E*w)-(F*U*A)-(B*D*V)
          WRITE(60,112) S
      112 FORMAT(10X,E12.4,'=S IN MTX')
          RETURN
          END

                    MTX                 SIZE      89    00131
                           MODULE:   *MAIN*    OPTIONS: SE AB WA LO MA DO SR
 61516-02 ENHANCED SAU FORTRAN COMPILER          REVISION LEVEL 33.071978

          SUBROUTINE PTCHK(N,USE,DDH,DDK,DDP,DX)
          COMMON /ZL1/ DTT12(50),DTT13(50).DTT23(50),XTRK(50),YTRK(50),NPTS
          COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
          COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
```

```
C     THIS SUBROUTINE IS FOR:
C             USE=1.0  TO CHECK SOLUTION  POINT FOR VALIDITY.
C             USE=2.0  TO PROVIDE TIME LAG SIMULATED DATA.
C
C             DX=1.0  FOR CORRECT SOLUTION.
C             DX=0.0  NO SOLUTION POINT.
C
      DX=0.0
      R1=(XTRK(N)-XSENS(1))**2+(YTRK(N)-YSENS(1))**2
      R1=SQRT(R1+(ZTRK(N)-ZSENS(1))**2)
      R2=(XTRK(N)-XSENS(2))**2+(YTRK(N)-YSENS(2))**2
      R2=SQRT(R2+(ZTRK(N)-ZSENS(2))**2)
      R3=(XTRK(N)-XSENS(3))**2+(YTRK(N)-YSENS(3))**2
      R3=SQRT(R3+(ZTRK(N)-ZSENS(3))**2)
      DT12=(R1-R2)/V
      DT13=(R1-R3)/V
      DT23=(R2-R3)/V
      IF(USE.EQ.2.0) GO TO 800
      MDT12=1000.0*DT12
      MDT13=1000.0*DT13
      MDT23=1000.0*DT23
      MDTT12=1000.0*DTT12(N)
      MDTT13=1000.0*DTT13(N)
      MDTT23=1000.0*DTT23(N)
      IF(IABS(MDT12-MDTT12).GE.5) GO TO 850
      IF(IABS(MDT13-MDTT13).GE.5) GO TO 850
      IF(IABS(MDT23-MDTT23).GE.5) GO TO 850
  850 SS=SQRT((DDH**2)+(DDK**2)+(DDP**2))
      IF(SQRT(3.0).LT.SS) GO TO 900
      DX=1.0
      RETURN
  800 DTT12(N)=DT12
      DTT13(N)=DT13
      DTT23(N)=DT23
  900 RETURN
      END
```

```
                    PTCHK            SIZE      369    00561
                         MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
   61516-02 ENHANCED SAU FORTRAN COMPILER       REVISION LEVEL 33.071978

            SUBROUTINE TABULR(DD)
            COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
            COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
            COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
      C
      C     THIS SUBROUTINE TABULATES VEHICLE SOURCE PATH OUTPUT
      C     TO DEVICE 60 (LINE PRINTER).
      C
            WRITE(60,1000) V
            DO 400 I=1,3
            WRITE(60,1100) I,XSENS(I),YSENS(I),ZSENS(I)
        400 CONTINUE
            WRITE(60,1200)
            DO 500 L=1,NPTS
            WRITE(60,1300) L,XTRK(L),YTRK(L),ZTRK(L)
        500 CONTINUE
       1000 FORMAT(1H1,5X,'VELOCITY OF MEDIUM IS ',3X,F8.3,2X,'(M/SEC)',
           1//,40X,'SENSOR POSITIONS',/)
       1100 FORMAT(5X,I2,3(10X,F10.3))
       1200 FORMAT(//,40X,'VEHICLE (SOURCE) PATH',/)
       1300 FORMAT(5X,I2,3(10X,F10.3))
            RETURN
            END

                    TABULR           SIZE      280    00430
                         MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
   61516-02 ENHANCED SAU FORTRAN COMPILER       REVISION LEVEL 33.071978

            SUBROUTINE SENSOR(I,DD)
            COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
            COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
```

```
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS SUBROUTINE IS TO BE MADE TO COLLECT REAL DATA FOR TIME LAGS
C     VIA PHASE COHERENCE DISCRIMINATION/ CROSS CORRELATION/ OR TIME
C     DIFFERENCE MEASUREMENTS.  DATA TO BE INPUT VIA A/D CONVERTOR TO
C     TRACKING COMPUTER.
C
      WRITE(50,100)
  100 FORMAT(20X,'***** WARNING *****',//,15X,
     1'PROGRAM TRYING TO ACCESS THE REAL DATA.',/,15X,
     2'THIS PART OF THE PROGRAM TO REMAIN INCOMPLETE AT THIS TIME',15X,
     3/,15X,'DUE TO LACK OF PROPER HARDWARE IMPLEMENTATION.')
      RETURN
      END

             SENSOR          SIZE      101     00145
                     MODULE:   *MAIN*   OPTIONS: SE AB WA LO MA DO SR
  61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

      SUBROUTINE TRKPLT(ANS,IDD)
      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
      COMMON /ZL4/ FN(25,50),IXAX(10),IYAX(5),IZAX(5),IYZAX(10)
C
C     THIS ROUTINE DETERMINES THE GRAPHIC DISPLAY REQUIREMENTS FOR
C     THE LINE PRINTER DISPLAY OF THE TRACKING SCENARIO.
C
C     INPUT GRAPH SPECIFICATIONS
C
      WRITE(50,100)
      READ(50,) SF
      WRITE(60,50505) SF
      WRITE(50,115)
      READ(50,) WX
```

```
      WRITE(60,40404) WX
      WRITE(50,125)
      READ(50,) HY
      WRITE(60,30303) HY
      WRITE(50,135)
      READ(50,) BZ
      WRITE(60,20202) BZ
C
C     SET OUTPUT TO BLANKS
C
      DO 8150 I=1,25
      DO 8100 J=1,50
      FN(I,J)=' '
 8100 CONTINUE
 8150 CONTINUE
C
C     SET X-AXIS:  FOR Y VS X, OR Z VS X.
C
      FAKE=5.0
      XADIV=WX/50.0
      DO 1000 I=1,10
      IXAX(I)=(FAKE*XADIV/SF)
      FAKE=FAKE+5.0
 1000 CONTINUE
C
C     SET Y-AXIS:  FOR Y VS X.
C
      FAKE=5.0
      YADIV=HY/25.0
      DO 1010 I=1,5
      IYAX(I)=FAKE*YADIV/SF
      FAKE=FAKE+5.0
 1010 CONTINUE
C
C     SET Y-AXIS:  FOR Z VS Y.
C
```

```
      FAKE=5.0
      YZADIV=HY/50.0
      DO 1020 I=1,10
      IYZAX(I)=FAKE*YZADIV/SF
      FAKE=FAKE+5.0
 1020 CONTINUE
C
C     SET Z-AXIS:  FOR Z VS X, OR Z VS Y.
C
      FAKE=5.0
      ZADIV=BZ/25.0
      DO 1030 I=1,5
      IZAX(I)=FAKE*ZADIV/SF
      FAKE=FAKE+5.0
 1030 CONTINUE
C
C     STORE STANDARD AXIS GRID.
C
      DO 1045 J=5,45,5
      DO 1040 L=1,25
      FN(L,J)=':'
 1040 CONTINUE
 1045 CONTINUE
      DO 1055 J=5,20,5
      DO 1050 L=1,50
      FN(J,L)='-'
 1050 CONTINUE
 1055 CONTINUE
      DO 1060 M=1,25
      FN(M,1)='I'
      FN(M,50)='I'
 1060 CONTINUE
```

```
      DO 1070 M=1,50
      FN(1,M)='I'
      FN(25,M)='I'
 1070 CONTINUE
C
C     STORE PARTICLE PATH.
C
      DO 1080 MI=1,NPTS
      IX=((XTRK(MI)/XADIV)*SF)+0.5
      IY=((YTRK(MI)/YADIV)*SF)+0.5
      IZ=((ZTRK(MI)/ZADIV)*SF)+0.5
      IYZ=((YTRK(MI)/YZADIV)*SF)+0.5
      IF(IX.LE.50.AND.IY.LE.25.AND.IDD.EQ.1) FN(IY,IX)='T'
      IF(IX.LE.50.AND.IZ.LE.25.AND.IDD.EQ.2) FN(IZ,IX)='T'
      IF(IYZ.LE.50.AND.IZ.LE.25.AND.IDD.EQ.3) FN(IZ,IYZ)='T'
 1080 CONTINUE
C
C     STORE SENSOR GRID POSITIONS.
C
      DO 1090 LI=1,3
```



```
      LX=((XSENS(LI)/XADIV)*SF)+0.5
      LY=((YSENS(LI)/YADIV)*SF)+0.5
      LZ=((ZSENS(LI)/ZADIV)*SF)+0.5
      LYZ=((YSENS(LI)/YZADIV)*SF)+0.5
      IF(LX.LE.50.AND.LY.LE.25.AND.LI.EQ.1) FN(LY,LX)='1'
      IF(LX.LE.50.AND.LY.LE.25.AND.LI.EQ.2) FN(LY,LX)='2'
      IF(LX.LE.50.AND.LY.LE.25.AND.LI.EQ.3) FN(LY,LX)='3'
      IF(LX.LE.50.AND.LZ.LE.25.AND.LI.EQ.1) FN(LZ,LX)='1'
      IF(LX.LE.50.AND.LZ.LE.25.AND.LI.EQ.2) FN(LZ,LX)='2'
      IF(LX.LE.50.AND.LZ.LE.25.AND.LI.EQ.3) FN(LZ,LX)='3'
      IF(LYZ.LE.50.AND.LY.LE.25.AND.LI.EQ.1) FN(LY,LYZ)='1'
      IF(LYZ.LE.50.AND.LY.LE.25.AND.LI.EQ.2) FN(LY,LYZ)='2'
```

```
      IF(LYZ.LE.50.AND.LY.LE.25.AND.LI.EQ.3) FN(LY,LYZ)='3'
 1090 CONTINUE
      IF(IDD.EQ.2) GO TO 2101
      IF(IDD.EQ.3) GO TO 2301
C
C     DUMP GRAPHS AND APPROPRIATE LABELS.
C
C     FIRST:  Y VS X:
C
C     GT=1 FOR Y VS X
C     GT=2 FOR Z VS X
C     GT=3 FOR Z VS Y
C
      F=0.0
      DO 2100 I=1,25
      CALL GRAPHO(I,F,1,SF)
 2100 CONTINUE
      GO TO 2500
 2101 CONTINUE
C
C     NEXT:  Z VS X:
C
      F=0.0
      DO 2300 I=1,25
      CALL GRAPHO(I,F,2,SF)
 2300 CONTINUE
      GO TO 2500
 2301 CONTINUE
C
C     FINALLY:  Z VS Y:
C
      F=0.0
      DO 2500 I=1,25
      CALL GRAPHO(I,F,3,SF)
 2500 CONTINUE
      WRITE(50,140)
```

```
      READ(50,) ANS
  100 FORMAT(10X,'SCALE FACTOR=?')
50505 FORMAT(1H1,10X,'SCALE FACTOR= ',F10.5)
  115 FORMAT(/,10X,'X MAX GRAPH= ',5X,'??')
                         MODULE:  TRKPLT   OPTIONS: SE AB WA LO MA DO SR
61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

40404 FORMAT(/,10X,'X MAX= ',F10.5)
  125 FORMAT(/,10X,'Y MAX GRAPH= ',5X,'??')
30303 FORMAT(1H@,T35,'Y MAX= ',F10.5)
  135 FORMAT(/,10X,'Z MAX GRAPH= ',5X,'??')
20202 FORMAT(1H@,T60,'Z MAX= ',F10.5)
  140 FORMAT(/,10X,'DO YOU WANT TO RUN ADDITIONAL PLOTS?',/,
     120X,'1= YES',/,20X,'2= NO',/,'??')
      RETURN
      END

                TRKPLT          SIZE    1435    02633
                         MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

      SUBROUTINE GRAPHO(I,F,NI,SF)
      COMMON /ZL1/ DTT12(50),DTT13(50),DTT23(50),XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
      COMMON /ZL4/ FN(25,50),IXAX(10),IYAX(5),IZAX(5),IYZAX(10)
C
C     GRAPH FUNCTION AND OUTPUT LABELS.
C
      II=26-I
      IF(F.NE.0.0) GO TO 10
      IF(NI.NE.3) WRITE(60,100) (IXAX(L),L=1,10)
      IF(NI.EQ.3) WRITE(60,110) (IYZAX(L),L=1,10)
  100 FORMAT(1H1,T49,'X AXIS (METERS)',//,T32,10(I3,2X))
  110 FORMAT(1H1,T49,'Y AXIS (METERS)',//,T32,10(I3,2X))
      F=1.0
```

```
   10 IF(I.EQ.18.OR.I.EQ.11) GO TO 131
      GO TO 133
  131 WRITE(60,212)
      GO TO 22
  133 IF(I.NE.17) GO TO 13
      WRITE(60,2133)
 2133 FORMAT(T20,'R')
      GO TO 22
   13 IF(I.NE.15) GO TO 14
      WRITE(60,203)
      GO TO 22
   14 IF(I.EQ.14.OR.I.EQ.16) GO TO 155
      GO TO 15
  155 WRITE(60,204)
      GO TO 22
   15 IF(I.NE.13) GO TO 16
      WRITE(60,205)
      GO TO 22
   16 IF(I.NE.10) GO TO 17
      WRITE(60,206)
      GO TO 22
   17 IF(I.NE.9) GO TO 18
      WRITE(60,207)
      GO TO 22
   18 IF(I.NE.8) GO TO 19
      WRITE(60,208)
      GO TO 22
   19 IF(NI.EQ.1.AND.I.EQ.6) GO TO 202
      GO TO 20
  202 WRITE(60,209)
      GO TO 22
   20 IF((NI.EQ.2.OR.NI.EQ.3).AND.I.EQ.6) GO TO 211
      GO TO 21
  211 WRITE(60,210)
```

```
        GO TO 22
     21 WRITE(60,219)
     22 INX=II/5
                          MODULE:  GRAPHO   OPTIONS: SE AB WA LO MA DO SR
  61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

        IF(MOD(II,5).EQ.0.AND.NI.EQ.1) WRITE(60,300) IYAX(INX)
        IF(MOD(II,5).EQ.0.AND.NI.NE.1) WRITE(60,300) IZAX(INX)
        WRITE(60,400) (FN(II,J),J=1,50)
        IF(II.NE.1.0) GO TO 555
        WRITE(60,700) SF,XLAT,YLONG,ZDEPTH
    555 CONTINUE
    212 FORMAT(T20,'S')
    203 FORMAT(T20,'T')
    204 FORMAT(T20,'E')
    205 FORMAT(T20,'M')
    206 FORMAT(T20,'I')
    207 FORMAT(T20,'X')
    208 FORMAT(T20,'A')
    209 FORMAT(T20,'Y')
    210 FORMAT(T20,'Z')
    219 FORMAT(T20,' ')
    300 FORMAT(1H@,T25,I3)
    400 FORMAT(1H@,T30,50(1A1))
    700 FORMAT(//,T22,'SF= ',F6.2,5X,'XMAX= ',F6.2,5X,'YMAX= ',F6.2,5X,
       1'ZMAX= ',F6.2)
        RETURN
        END

                GRAPHO          SIZE      668     01234

$CATALOG
61632-02 OVERLAY CATALOGER -- REVISION LEVEL 12.071978
TYPE=BG
DNAME BTRCK,4
BEGIN
```

```
P.NAME=   BTRCK

*HIGH     27060            *START        0
*LOW          0            *LOW          0
*P.END    20703            .PASS         2
*START        0            GUESS1     1231 $
.PASS         2            TRKPT      1447 $
C$ERRO    20343 $          SYSEQN     2364 $
C$IOER    20354 $          CRAMER     3751 $
C$MOD     20330 $          MTX        4374 $
C$OFLO    20345 $          PTCHK      4525 $
C$UFLO    20351 $          TABULR     5306 $
CRAMER     3751 $          SENSOR     5736 $
DCF       16574 $          TRKPLT     6103 $
DSQRT     20336 $          GRAPHO    10736 $
ESGN$     16245 $          F$SERA    12214 $
EXIT      20153 $          F$SERI    12221 $
EXP10$    16244 $          F$SERJ    12226 $
F$1ISN    12760 $          F$SERK    12233 $
F$10SN    13157 $          F$RIFF    12274 $
F$4ISN    13300 $          F$RSFF    12274 $
F$40SN    13406 $          F$RSSY    12275 $
F$5ISN    13276 $          F$WIFF    12277 $
F$50SN    13404 $          F$WSFF    12277 $
F$8ISN    13274 $          F$WSSY    12300 $
F$80SN    13402 $          F$DIFF    12344 $
F$ADDR    16461 $          F$DSSY    12345 $
F$AIP     13527 $          F$EIFF    12347 $
F$AISN    12754 $          F$ESSY    12350 $
F$AOP     14652 $          F$EESY    12450 $
F$B/E     12660 $          F$SYEE    12450 $
F$BBI     20657 $          F$RORW    12463 $
F$BBJ     20610 $          F$FUNC    12566 $
F$BCC     16471 $          F$B/E     12660 $
F$BISN    12743 $          F$CSIO    12670 $
```

F$BOSN	13146	$	F$CFIO	12670	$
F$BUF	16467	$	F$BISN	12743	$
F$CFIO	12670	$	F$AISN	12754	$
F$CLOS	20230	$	F$1ISN	12760	$
F$CNOP	20155	$	F$BOSN	13146	$
F$CONV	16477	$	F$1OSN	13157	$
F$CSIO	12670	$	F$8ISN	13274	$
F$CV41	20312	$	F$5ISN	13276	$
F$CV51	20312	$	F$4ISN	13300	$
F$D	16464	$	F$8OSN	13402	$
F$DCF	16574	$	F$5OSN	13404	$
F$DCNT	17270	$	F$4OSN	13406	$
F$DIFF	12344	$	F$AIP	13527	$
F$DSSY	12345	$	F$RIP	13531	$
F$DVR5	17415	$	F$LIP	13624	$
F$E	16465	$	F$OIP	13656	$
F$E/E	16475	$	F$IIP	13714	$
F$E/E1	16476	$	F$GIP	14164	$
F$E/O	16473	$	F$ROP	14650	$
F$E/O1	16474	$	F$AOP	14652	$
F$EESY	12450	$	F$IOP	14736	$
F$EIFF	12347	$	F$FOP	15232	$
F$EMB	20560	$	F$QOP	15422	$
F$EMBJ	20572	$	F$SOP	15427	$
F$EOP	15605	$	F$O.LB	15436	$
F$ERCD	20307	$	F$O.SN	15442	$
F$ESSY	12350	$	F$OOP	15455	$
F$FCC	16470	$	F$GOP	15603	$
F$FERR	16502	$	F$EOP	15605	$
F$FMT	16466	$	TENTB$	16121	$
F$FMTC	16462	$	EXP10$	16244	$
F$FOP	15232	$	ESGN$	16245	$
F$FUNC	12566	$	MSGN$	16246	$
F$GIP	14164	$	F$NORM	16251	$

F$GOP	15603	$	F$ADDR	16461	$
F$IBF	16500	$	F$SPEC	16462	$
F$IIP	13714	$	F$FMTC	16462	$
F$IOM0	20060	$	F$W	16463	$
F$IOM1	20044	$	F$D	16464	$
F$IOM2	20046	$	F$E	16465	$
F$IOM3	20156	$	F$FMT	16466	$
F$IOM4	20165	$	F$BUF	16467	$
F$IOP	14736	$	F$FCC	16470	$
F$LINE	17611	$	F$BCC	16471	$
F$LIP	13624	$	F$P	16472	$
F$MPR5	17321	$	F$E/O	16473	$
F$NAME	17516	$	F$E/O1	16474	$
F$NCHR	16503	$	F$E/E	16475	$
F$NORM	16251	$	F$E/E1	16476	$
F$NTRY	17521	$	F$CONV	16477	$
F$O.LB	15436	$	F$IBF	16500	$
F$O.SN	15442	$	F$FERR	16502	$
F$OIP	13656	$	F$NCHR	16503	$
F$OOP	15455	$	F$PNL	16555	$
F$OPEN	20173	$	F$DCF	16574	$
F$P	16472	$	DCF	16574	$
F$PNL	16555	$	F$DCNT	17270	$
F$QOP	15422	$	PUTN$	17273	$
F$RBM	20565	$	F$MPR5	17321	$
F$RBMI	20630	$	F$DVR5	17415	$
F$RIFF	12274	$	F$NAME	17516	$
F$RIP	13531	$	F$NTRY	17521	$
F$ROP	14650	$	F$STMT	17562	$
F$RORW	12463	$	F$RSDO	17566	$
F$RSDO	17566	$	F$SRTN	17571	$
F$RSFF	12274	$	F$LINE	17611	$
F$RSSY	12275	$	F$IOM1	20044	$
F$SERA	12214	$	F$IOM2	20046	$
F$SERI	12221	$	F$IOM0	20060	$

```
   F$SERJ    12226 $        EXIT      20153 $
   F$SERK    12233 $        F$CNOP    20155 $
   F$SOP     15427 $        F$IOM3    20156 $
   F$SPEC    16462 $        F$IOM4    20165 $
   F$SRTN    17571 $        F$OPEN    20173 $
   F$STMT    17562 $        F$CLOS    20230 $
   F$SYEE    12450 $        IOERR     20305 $
   F$W       16463 $        F$ERCD    20307 $
   F$WIFF    12277 $        INT       20310 $
   F$WSFF    12277 $        IDINT     20310 $
   F$WSSY    12300 $        IFIX      20310 $
   GRAPHO    10736 $        IFIX$     20310 $
   GUESS1     1231 $        F$CV51    20312 $
   IDINT     20310 $        F$CV41    20312 $
   IFIX      20310 $        MOD$      20324 $
   IFIX$     20310 $        C$MOD     20330 $
   INT       20310 $        DSQRT     20336 $
   IOERR     20305 $        SQRT      20336 $
   MOD$      20324 $        C$ERR0    20343 $
   MSGN$     16246 $        C$OFLO    20345 $
   MTX        4374 $        C$UFLO    20351 $
   PTCHK      4525 $        C$IOER    20354 $
   PUTN$-    17273 $        F$EMB     20560 $
   SENSOR     5736 $        F$RBM     20565 $
   SQRT      20336 $        F$EMBJ    20572 $
   SYSEQN     2364 $        F$BBJ     20610 $
   TABULR     5306 $        F$RBMI    20630 $
   TENTB$    16121 $        F$BBI     20657 $
   TRKPLT     6103 $        *P.END    20703
   TRKPT      1447 $        ZL1       20703 C
   ZL1       20703 C        ZL4       21670 C
   ZL2       26654 C        ZL3       26632 C
   ZL3       26632 C        ZL2       26654 C
   ZL4       21670 C        *HIGH     27060
1  $EOJ
```

APPENDIX F

THREE SPACE SIMULATION DATA, pp. 119-122.

DESCRIPTION:

This appendix contains three separate sets of tabular/ disk file data which was generated by the program of Appendix A in an attempt to utilize the tracking program of Appendix E for a three-space, simultaneous coordinate trajectory solution.

VELOCITY OF MEDIUM IS 1450.000 (M/SEC)

SENSOR POSITIONS

1	75.000	100.000	50.000
2	175.000	75.000	50.000
3	200.000	200.000	50.000

VEHICLE (SOURCE) PATH

1	80.000	10.000	50.000
2	100.000	40.000	50.000
3	110.000	70.000	50.000
4	110.000	100.000	55.000
5	100.000	120.000	55.000

5

-0.017220594790	-0.092816047347	-0.075595452558
-0.012251535714	-0.085296291476	-0.073044755762
-0.013168483976	-0.077252524636	-0.064084040660
-0.023769559420	-0.068464676634	-0.044695117213
-0.038071292309	-0.066038863923	-0.027967571614

75.0000	100.0000	50.0000
175.0000	75.0000	50.0000
200.0000	200.0000	50.0000

```
VELOCITY OF MEDIUM IS     1450.000   (M/SEC)

                                     SENSOR POSITIONS

1             900.000                150.000                200.000
2             200.000                850.000                200.000
3             180.000                160.000                200.000

                                     VEHICLE (SOURCE) PATH

1             450.000                360.000                200.000
2             455.000                365.000                200.000
3             460.000                370.000                200.000
4             460.000                375.000                200.000
5             460.000                380.000                200.000


5
   -0.036898304041      0.110746488540      0.147644792590
   -0.037058052403      0.104286271720      0.141344324130
   -0.037212970693      0.097886175896      0.135099146590
   -0.032628469459      0.097357652662      0.129986122120
   -0.028023652820      0.096826060135      0.124849712950
     900.0000        150.0000        200.0000
     200.0000        850.0000        200.0000
     180.0000        160.0000        200.0000
```

VELOCITY OF MEDIUM IS 1450.000 (M/SEC)

SENSOR POSITIONS

1	350.000	150.000	100.000
2	200.000	300.000	100.000
3	200.000	150.000	250.000

VEHICLE (SOURCE) PATH

1	20.000	10.000	10.000
2	30.000	20.000	15.000
3	40.000	25.000	20.000
4	50.000	30.000	25.000
5	50.000	35.000	30.000
6	60.000	40.000	35.000
7	60.000	40.000	40.000
8	65.000	45.000	40.000
9	75.000	50.000	45.000
10	85.000	70.000	50.000
11	95.000	90.000	60.000
12	115.000	110.000	65.000
13	135.000	120.000	70.000
14	155.000	135.000	80.000
15	175.000	145.000	100.000
16	195.000	155.000	110.000
17	215.000	165.000	120.000
18	235.000	180.000	130.000
19	245.000	195.000	140.000
20	250.000	205.000	150.000
21	260.000	215.000	160.000
22	265.000	220.000	170.000
23	275.000	225.000	175.000
24	285.000	235.000	180.000
25	295.000	245.000	180.000

25

0.011453186453	0.026576182832	0.015122996378
0.011922829695	0.026109811539	0.014186981844
0.010779781431	0.025444964131	0.014665182699
0.009555768487	0.024726842733	0.015171074245
0.011292182723	0.026745214277	0.015453031554
0.010023010273	0.026033476211	0.016010465938
0.010055859664	0.027995435876	0.017939576211
0.010271221410	0.026734110573	0.016462889163
0.008878005485	0.025962318996	0.017084313512
0.013329603884	0.025592658372	0.012263054488
0.018503374293	0.027433025800	0.008929651507
0.020597908187	0.023069473690	0.002471565503
0.017539561193	0.017539561193	0.000000000000
0.016830537198	0.013863728667	-0.002966808531
0.012460847806	0.015807016175	0.003346168369
0.006877453073	0.010499479304	0.003622026230
0.000000000000	0.003845450658	0.003845450658
-0.004119375586	-0.004119375586	0.000000000000
0.000000000000	-0.004169286829	-0.004169286829
0.004256665600	0.000000000000	-0.004256665600
0.004200119411	0.000000000000	-0.004200119411
0.004055534872	0.004055534872	0.000000000000
0.000000000000	0.000000000000	0.000000000000
0.000000000000	-0.003793369154	-0.003793369154
0.000000000000	-0.010801239313	-0.010801239313
350.0000	150.0000	100.0000
200.0000	300.0000	100.0000
200.0000	150.0000	250.0000

APPENDIX G

THREE SPACE: METHOD TWO: SIMULATION DATA PROGRAM, pp. 124-132.

DESCRIPTION:

This program is similar in purpose to that of Appendix A: it generates tabular and disk-file data. This data can be seen in Appendix I. Instead of providing simultaneous coordinate three-space time-difference data as the program of Appendix A, this program sensitizes the time-difference data to the three planes XY, YZ, and XZ. This planar data is used by the three-space program of Appendix H. This essentially results in three two-space programs to be processed to provide the three-space tracking solution.

```
$JOB SIMDAT DLIST4                                           01:42:03
$OPTIONS 21,8,9
$FORTRAN
                          MODULE:  *MAIN*   OPTIONS: SE LO MA
  61516-02 ENHANCED SAU FORTRAN COMPILER       REVISION LEVEL 33.071978


      COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
      COMMON /ZL4/ DTT12(3,50),DTT13(3,50),DTT23(3,50)
C
C     THIS MAIN ROUTINE CREATES SIMULATED TRACKING DATA (VIA TIME LAG
C     DIFFERENCES BETWEEN SENSORS): BASED UPON PROBLEM SCENARIO OF THE
C     TARGET/SENSOR LOCATIONS.
C
    5 WRITE(17,100)
      WRITE(17,110)
      READ(17,) NPTS
      WRITE(17,120)
      READ(17,) V
      DO 10 I=1,3
      WRITE(17,130) I
      READ(17,) XSENS(I),YSENS(I),ZSENS(I)
   10 CONTINUE
      DO 20 J=1, NPTS
      WRITE(17,140) J
      READ(17,) XTRK(J),YTRK(J),ZTRK(J)
      DO 15 NXYZ=1,3
      CALL PTCHK(NXYZ,J,2.0,DX)
   15 CONTINUE
   20 CONTINUE
      CALL TABULR(DD)
      WRITE(17,150)
      READ(17,) ANS
```

```
      IF(ANS.EQ.0.0) GO TO 5
      CALL DATXYZ(DMV)
  100 FORMAT(//,5X,'MAXIMUM DATAPOINTS ARE:',//,10X,
     1'TIMELAGS 50',/,10X,'TRACK POINTS 50',/,10X,'SENSORS 3')
  110 FORMAT(//,5X,'INPUT THE NUMBER OF DATA POINTS',/,10X,'??')
  120 FORMAT(//,5X,'INPUT THE VELOCITY OF THE MEDIUM:',//,
     110X,'AIR= 331 METERS PER SECOND',/,
     210X,'WATER= 1450 METERS PER SECOND',/,10X,'??')
  130 FORMAT(//,5X,'INPUT THE COORDINATES FOR SENSOR LOCATIONS:',//,
     210X,'XXXX.XX',5X,'YYYY.YY',5X,'ZZZZ.ZZ',3X,'SENSOR= ',I1,
     35X,'(VALUES MUST BE SEPARATED BY SPACE OR COMMAS)',/,'??')
  140 FORMAT(//,5X,'INPUT THE COORDINATES FOR TARGET PARTICLE PATH ',
     1'POINTS:',//,10X,'XXXX.XX',5X,'YYYY.YY',5X,'ZZZZ.ZZ',3X,I2,//,
     25X,'(VALUES MUST BE SEPARATED BY SPACE OR COMMAS)',/,'??')
  150 FORMAT(//,10X,'IS PRINTOUT DATA CORRECT TRACK/LOCATIONS?',//,
     115X,'INPUT:   YES=1.0,  NO=0.0')
      CALL EXIT
      END

                  *MAIN*         SIZE     434    00662
                       MODULE:  *MAIN*   OPTIONS: SE LO MA
 61516-02 ENHANCED SAU FORTRAN COMPILER       REVISION LEVEL 33.071978

      SUBROUTINE PTCHK(NXYZ,N,USE,DX)
      COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
      COMMON /ZL4/ DTT12(3,50),DTT13(3,50),DTT23(3,50)
C
C     THIS SUBROUTINE IS FOR:
C             USE=1.0  TO CHECK SOLUTION  POINT FOR VALIDITY.
C             USE=2.0  TO PROVIDE TIME LAG SIMULATED DATA.
C
C             DX=1.0  FOR CORRECT SOLUTION.
C             DX=0.0  NO SOLUTION POINT.
```

```
C
      DX=0.0
      IF(NXYZ.NE.1) GO TO 310
      R1=SQRT((XTRK(N)-XSENS(1))**2+(YTRK(N)-YSENS(1))**2)
      R2=SQRT((XTRK(N)-XSENS(2))**2+(YTRK(N)-YSENS(2))**2)
      R3=SQRT((XTRK(N)-XSENS(3))**2+(YTRK(N)-YSENS(3))**2)
  310 IF(NXYZ.NE.2) GO TO 320
      R1=SQRT((ZTRK(N)-ZSENS(1))**2+(YTRK(N)-YSENS(1))**2)
      R2=SQRT((ZTRK(N)-ZSENS(2))**2+(YTRK(N)-YSENS(2))**2)
      R3=SQRT((ZTRK(N)-ZSENS(3))**2+(YTRK(N)-YSENS(3))**2)
  320 IF(NXYZ.NE.3) GO TO 330
      R1=SQRT((ZTRK(N)-ZSENS(1))**2+(XTRK(N)-XSENS(1))**2)
      R2=SQRT((ZTRK(N)-ZSENS(2))**2+(XTRK(N)-XSENS(2))**2)
      R3=SQRT((ZTRK(N)-ZSENS(3))**2+(XTRK(N)-XSENS(3))**2)
  330 CONTINUE
      DT12=(R1-R2)/V
      DT13=(R1-R3)/V
      DT23=(R2-R3)/V
      IF(USE.EQ.2.0) GO TO 800
      MDT12=10000.0*DT12
      MDT13=10000.0*DT13
      MDT23=10000.0*DT23
      MDTT12=10000.0*DTT12(NXYZ,N)
      MDTT13=10000.0*DTT13(NXYZ,N)
      MDTT23=10000.0*DTT23(NXYZ,N)
      IF(MDT12.NE.MDTT12) GO TO 900
      IF(MDT13.NE.MDTT13) GO TO 900
      IF(MDT23.NE.MDTT23) GO TO 900
      DX=1.0
      RETURN
  800 IF(NXYZ.NE.1) GO TO 810
      DTT12(1,N)=DT12
      DTT13(1,N)=DT13
      DTT23(1,N)=DT23
  810 IF(NXYZ.NE.2) GO TO 820
```

```
      DTT12(2,N)=DT12
      DTT13(2,N)=DT13
      DTT23(2,N)=DT23
  820 IF(NXYZ.NE.3) GO TO 830
      DTT13(3,N)=DT13
                          MODULE:  PTCHK     OPTIONS: SE LO MA
 61516-02 ENHANCED SAU FORTRAN COMPILER          REVISION LEVEL 33.071978

      DTT12(3,N)=DT12
      DTT23(3,N)=DT23
  830 CONTINUE
  900 RETURN
      END

                    PTCHK             SIZE      361     00551
                          MODULE:  *MAIN*    OPTIONS: SE LO MA
 61516-02 ENHANCED SAU FORTRAN COMPILER          REVISION LEVEL 33.071978

      SUBROUTINE TABULR(DD)
      COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS SUBROUTINE TABULATES VEHICLE SOURCE PATH OUTPUT
C     TO DEVICE 6 (LINE PRINTER).
C
      WRITE(6,1000) V
      DO 400 I=1,3
      WRITE(6,1100) I,XSENS(I),YSENS(I),ZSENS(I)
  400 CONTINUE
      WRITE(6,1200)
      DO 500 L=1,NPTS
      WRITE(6,1300) L,XTRK(L),YTRK(L),ZTRK(L)
  500 CONTINUE
 1000 FORMAT(1H1,5X,'VELOCITY OF MEDIUM IS ',3X,F8.3,2X,'(M/SEC)',
```

```
      1//,40X,'SENSOR POSITIONS',/)
 1100 FORMAT(5X,I2,3(10X,F10.3))
 1200 FORMAT(//,40X,'VEHICLE (SOURCE) PATH',/)
 1300 FORMAT(5X,I2,3(10X,F10.3))
      RETURN
      END

                 TABULR          SIZE      169     00251
                         MODULE:  *MAIN*   OPTIONS: SE LO MA
61516-02 ENHANCED SAU FORTRAN COMPILER         REVISION LEVEL 33.071978

      SUBROUTINE DATXYZ(DMV)
      COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
      COMMON /ZL4/ DTT12(3,50),DTT13(3,50),DTT23(3,50)
C
C     THIS  SUBROUTINE IS USED TO STORE TIME LAG, NPTS
C     AND SENSOR LOCATION DATA ON DISK FILE.
C
      REWIND 18
      WRITE(18,134) NPTS
  134 FORMAT(I2)
      DO 602 L=1,3
      DO 601 I=1, NPTS
      WRITE(18,108) DTT12(L,I),DTT13(L,I),DTT23(L,I)
  108 FORMAT(3(5X,F15.12))
  601 CONTINUE
  602 CONTINUE
      DO 48 N=1,3
      WRITE(18,110) XSENS(N),YSENS(N),ZSENS(N)
  110 FORMAT(3(5X,F10.4))
   48 CONTINUE
      REWIND 18
      RETURN
```

```
152:          END

                         DATXYZ            SIZE      134      00206

$CATALOG
61632-02 OVERLAY CATALOGER -- REVISION LEVEL 12.071978
TYPE=BG
DNAME SMDT2,4
BEGIN
              P.NAME=  SMDT2

*HIGH       12524                  *START          0
*LOW            0                  *LOW            0
*P.END      10161                  .PASS           2
*START          0                  PTCHK         662 $
.PASS           2                  TABULR       1433 $
C$ERRO       7621 $                DATXYZ       1704 $
C$IOER       7632 $                F$RIFF       2112 $
C$OFLO       7623 $                F$RSFF       2112 $
C$UFLO       7627 $                F$RSSY       2113 $
DATXYZ       1704 $                F$WIFF       2115 $
DCF          6412 $                F$WSFF       2115 $
DSQRT        7614 $                F$WSSY       2116 $
ESGN$        6063 $                F$DIFF       2162 $
EXIT         7443 $                F$DSSY       2163 $
EXP10$       6062 $                F$EIFF       2165 $
F$1ISN       2576 $                F$ESSY       2166 $
F$1OSN       2775 $                F$EESY       2266 $
F$4ISN       3116 $                F$SYEE       2266 $
F$4OSN       3224 $                F$RORW       2301 $
F$5ISN       3114 $                F$FUNC       2404 $
F$5OSN       3222 $                F$B/E        2476 $
F$8ISN       3112 $                F$CSIO       2506 $
F$8OSN       3220 $                F$CFIO       2506 $
F$ADDR       6277 $                F$BISN       2561 $
```

F$AIP	3345	$	F$AISN	2572	$
F$AISN	2572	$	F$1ISN	2576	$
F$AOP	4470	$	F$BOSN	2764	$
F$B/E	2476	$	F$1OSN	2775	$
F$BBI	10135	$	F$8ISN	3112	$
F$BBJ	10066	$	F$5ISN	3114	$
F$BCC	6307	$	F$4ISN	3116	$
F$BISN	2561	$	F$8OSN	3220	$
F$BOSN	2764	$	F$5OSN	3222	$
F$BUF	6305	$	F$4OSN	3224	$
F$CFIO	2506	$	F$AIP	3345	$
F$CLOS	7520	$	F$RIP	3347	$
F$CNOP	7445	$	F$LIP	3442	$
F$CONV	6315	$	F$OIP	3474	$
F$CSIO	2506	$	F$IIP	3532	$
F$CV41	7602	$	F$GIP	4002	$
F$CV51	7602	$	F$ROP	4466	$
F$D	6302	$	F$AOP	4470	$
F$DCF	6412	$	F$IOP	4554	$
F$DCNT	7106	$	F$FOP	5050	$
F$DIFF	2162	$	F$QOP	5240	$
F$DSSY	2163	$	F$SOP	5245	$
F$DVR5	7233	$	F$O.LB	5254	$
F$E	6303	$	F$O.SN	5260	$
F$E/E	6313	$	F$OOP	5273	$
F$E/E1	6314	$	F$GOP	5421	$
F$E/O	6311	$	F$EOP	5423	$
F$E/O1	6312	$	TENTB$	5737	$
F$EESY	2266	$	EXP10$	6062	$
F$EIFF	2165	$	ESGN$	6063	$
F$EMB	10036	$	MSGN$	6064	$
F$EMBJ	10050	$	F$NORM	6067	$
F$EOP	5423	$	F$ADDR	6277	$
F$ERCD	7577	$	F$SPEC	6300	$

F$ESSY	2166	$	F$FMTC	6300	$
F$FCC	6306	$	F$W	6301	$
F$FERR	6320	$	F$D	6302	$
F$FMT	6304	$	F$E	6303	$
F$FMTC	6300	$	F$FMT	6304	$
F$FOP	5050	$	F$BUF	6305	$
F$FUNC	2404	$	F$FCC	6306	$
F$GIP	4002	$	F$BCC	6307	$
F$GOP	5421	$	F$P	6310	$
F$IBF	6316	$	F$E/O	6311	$
F$IIP	3532	$	F$E/O1	6312	$
F$IOM0	7350	$	F$E/E	6313	$
F$IOM1	7334	$	F$E/E1	6314	$
F$IOM2	7336	$	F$CONV	6315	$
F$IOM3	7446	$	F$IBF	6316	$
F$IOM4	7455	$	F$FERR	6320	$
F$IOP	4554	$	F$NCHR	6321	$
F$LIP	3442	$	F$PNL	6373	$
F$MPR5	7137	$	F$DCF	6412	$
F$NCHR	6321	$	DCF	6412	$
F$NORM	6067	$	F$DCNT	7106	$
F$O.LB	5254	$	PUTN$	7111	$
F$O.SN	5260	$	F$MPR5	7137	$
F$OIP	3474	$	F$DVR5	7233	$
F$OOP	5273	$	F$IOM1	7334	$
F$OPEN	7463	$	F$IOM2	7336	$
F$P	6310	$	F$IOM0	7350	$
F$PNL	6373	$	EXIT	7443	$
F$QOP	5240	$	F$CNOP	7445	$
F$RBM	10043	$	F$IOM3	7446	$
F$RBMI	10106	$	F$IOM4	7455	$
F$RIFF	2112	$	F$OPEN	7463	$
F$RIP	3347	$	F$CLOS	7520	$
F$ROP	4466	$	IOERR	7575	$
F$ROR	2301	$	F$ERCD	7577	$

```
F$RSFF      2112 $          INT          7600 $
F$RSSY      2113 $          IDINT        7600 $
F$SOP       5245 $          IFIX         7600 $
F$SPEC      6300 $          IFIX$        7600 $
F$SYEE      2266 $          F$CV41       7602 $
F$W         6301 $          F$CV51       7602 $
F$WIFF      2115 $          SQRT         7614 $
F$WSFF      2115 $          DSQRT        7614 $
F$WSSY      2116 $          C$ERRO       7621 $

IDINT       7600 $          C$OFLO       7623 $
IFIX        7600 $          C$UFLO       7627 $
IFIX$       7600 $          C$IOER       7632 $
INT         7600 $          F$EMB       10036 $
IOERR       7575 $          F$RBM       10043 $
MSGN$       6064 $          F$EMBJ      10050 $
PTCHK        662 $          F$BBJ       10066 $
PUTN$       7111 $          F$RBMI      10106 $
SQRT        7614 $          F$BBI       10135 $
TABULR      1433 $          *P.END      10161
TENTB$      5737 $          ZL1         10161 C
ZL1        10161 C          ZL4         10472 C
ZL2        12320 C          ZL3         12276 C
ZL3        12276.C          ZL2         12320 C
ZL4        10472 C          *HIGH       12524
$EOJ
SIMDAT TIME 00HRS 00MIN 00.000SEC
```

APPENDIX H

THREE SPACE: METHOD TWO: PLANAR PROGRAM, pp. 134-161.

DESCRIPTION:

This program interprets the disk-file three-space planar-sensitized time-difference data of Appendix I, which was generated by the program of Appendix G, and yields the tracking solution plots of Appendix J. This program essentially results in the solution of three planar two-space problems to yield the resultant three-space tracking solution.

```
$JOB TRACK DLIST4                                                   01:17:44
$OPTIONS 21,8,9,4
$FORTRAN
                         MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
  61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

:WA,SR,AB,GO
      COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
      COMMON /ZL4/ FN(25,50),IXAX(10),IYAX(5),IZAX(5),IYZAX(10)
      COMMON /ZL5/ DTT12(3,50),DTT13(3,50),DTT23(3,50)
      COMMON /ZL6/ FXTRK(50),FYTRK(50),FXSENS(3),FYSENS(3)
C*************************************************************************
C     THIS IS THE MAIN PROGRAM FOR THE ACOUSTIC MODEL TRACKING
C     AND PLOT PROGRAM
C
C     IT IS BASED ON:
C              3-SENSORS: KNOWN COORDINATES; WITH, RELATIVE PHASE
C                COHERENCE - OR TIME LAGS OF ARRIVING WAVEFRONTS -
C                AS RECEIVED DATA FROM AN UNKNOWN SOURCE WITH SOME
C                RELATIVE REFERENCE FREQUENCY USED IN DETECTION OF
C
C                DELTA TIME DIFFERENCES BETWEEN SENSORS/SOURCE.
C              1-SOURCE: OF UNKNOWN POSITIONS, EMITTING SOME REFERENCE
C                SIGNAL FOR DETECTION.
C
C              ASSUMPTIONS: (1) DELTA TIME DIFFERENCES HAVE BEEN
C                PROCESSED FOR INPUT TO THIS PROGRAM BY SOME RECEIVING
C                PROCESSOR;(2) NO PROPAGATION LOSS DUE TO MEDIUM;
C                (3) SOUND VELOCITY OF MEDIUM IS CONSTANT AND KNOWN; AN
C                (4) NO MULTIPATH PHENOMENA, FOR A SOURCE-SENSOR PAIR.
C
C
C              SOLUTION:
```

```
C                     USING ONLY THE ABOVE SCENARIO DESCRIPTORS, TRACK AND
C                     PLOT THE SOURCE.
C
C                   APPROACH:
C                     USING DISTANCE EQUATIONS FROM THE SENSORS TO THE
C                     UNKNOWN SOURCE, AND THE VELOCITY OF THE MEDIUM:
C                       -DEVELOP THE RESULTING NON-LINEAR HYPERBOLIC LOCATOR
C                        EQUATIONS;
C                       -SOLVE ITERATIVELY WITH A NEWTON-RAPHSON NON-LINEAR,
C                        SIMULTANEOUS EQUATION METHOD.
C
C                   THIS PROGRAM WRITTEN BY:
C                         GEORGE H. FORD, JR.
C                            JULY 3, 1980
C                   IN FULFILLMENT OF A RESEARCH TOPIC/REPORT FOR AN
C                   M.S.E. DEGREE IN THE U.C.F. DEPARTMENT OF ELECTRICAL
C                   ENGINEERING.
C
C                   SOURCE PROGRAM TYPED ON CARDS.
C                   COMPUTER:
C                       NAVTRAEQUIPCEN: HARRIS DATACRAFT 6024/4
C                       WITH LINE PRINTER PLOT SOFTWARE  WHICH WAS GENERATED
C                       FOR THIS PROJECT INCLUDED HEREIN.
                      .   MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
 61516-02 ENHANCED SAU FORTRAN COMPILER       REVISION LEVEL 33.071978

C
C                   REVIEW BOARD:
C                       DR. FRED SIMONS  (CHAIRMAN)
C                       DR. MIKE HARRIS
C                       DR. CHRIS BAUER
C
C*******************************************************************
            DO 3 I=1,50
```

```
      XTRK(I)=0.0
      YTRK(I)=0.0
      ZTRK(I)=0.0
    3 CONTINUE
    5 WRITE(50,99)
      READ(50,) V
      REWIND 40
      READ(40,134) NPTS
      WRITE(50,105)
      READ(50,106) IZ
C
C     IZ=1 REAL DATA
C    IZ=0 CALCULATED, FILE STORAGE DATA: DATXYZ
C
C     INPUT TIME LAGS FROM REAL DATA/FILE DATA.
C
      IF(IZ.NE.1) GO TO 10
      WRITE(50,133)
      READ(50,134) NPTS
      DO 600 I=1,NPTS
      CALL SENSOR(I,DD)
  600 CONTINUE
      GO TO 46
   10 DO 602 L=1,3
      DO 601 I=1,NPTS
      READ(40,108) DTT12(L,I),DTT13(L,I),DTT23(L,I)
  601 CONTINUE
  602 CONTINUE
   46 NGES1=0
C
C     READ FILE DATA FOR SENSORS: X,Y,Z.
C
      DO 48 N=1,3
      READ(40,110) XSENS(N),YSENS(N),ZSENS(N)
   48 CONTINUE
```

```
      WRITE(60,2222) ((MI,XSENS(MI),YSENS(MI)),MI=1,3)
 2222 FORMAT(3(5X,I2.5X,'XS= ',F8.2,15X,'YX= ',F8.2,/))
C
C     INPUT/CALCULATE INITIAL GUESSES FOR X,Y,Z (SOURCE)
C
      WRITE(50,109)
      READ(50,) XLAT,YLONG,ZDEPTH
   51 DO 80 M=1,3
      CALL GUESS1(NGES1,M,XI,YI,ZI)
                          MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
61516-02 ENHANCED SAU FORTRAN COMPILER         REVISION LEVEL 33.071978

   80 CONTINUE
      CALL TABULR(DD)
      DO 6948 IDD=1,3
   52 CALL TRKPLT(ANS,IDD)
      IF(ANS.EQ.1.0) GO TO 52
 6948 CONTINUE
      WRITE(50,111)
      READ(50,) ANS
      IF(ANS.EQ.1.0) GO TO 5
C
C     FORMATS: MAIN.
C
   99 FORMAT(5X,'INPUT THE MAGNITUDE OF VELOCITY OF THE MEDIUM',/,
     15X,' (METERS/SEC)',/,15X,'WATER= 1450 M/SEC    AIR= 331 M/SEC')
  134 FORMAT(I2)
  105 FORMAT(5X,'INPUT THE FOLLOWING FOR TYPE PROGRAM:',/,15X,
     1'1= REAL DATA (TIME LAGS VIA PHASES PROCESSED)',/,15X,
     2'0= FILE DATA VIA DATXYZ PROGRAM')
  106 FORMAT(I1)
  133 FORMAT(5X,'HOW MANY TRACKING POINTS DESIRED? (MAX=50)')
  108 FORMAT(3(5X,F15.12))
  109 FORMAT(5X,'INPUT:',/,10X,'MAXIMUM X METERS(LATITUDE RANGE)',
     1/,10X,'MAXIMUM Y METERS(LONGITUDE RANGE)',/,10X,
```

```
      2'MAXIMUM Z METER(DEPTH)',//,10X,'XXX.XX',5X,'YYY.YY',5X,'ZZZ.ZZ')
  110 FORMAT(3(5X,F10.4))
  111 FORMAT(5X,'DO YOU WISH TO:',//,10X,
      1'1- CONTINUE PROCESSING; OR',/,10X,
      2'2- END PROCESSING',//,5X,'??')
       CALL EXIT
       END

                 *MAIN*          SIZE      762     01372
                           MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
  61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

       SUBROUTINE GUESS1(NGES,M,XI,YI,ZI)
       COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
       COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
       COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
       COMMON /ZL4/ FN(25,50),IXAX(10),IYAX(5),IZAX(5),IYZAX(10)
       COMMON /ZL5/ DTT12(3,50),DTT13(3,50),DTT23(3,50)
       COMMON /ZL6/ FXTRK(50),FYTRK(50),FXSENS(3),FYSENS(3)
C
C      THIS SUBROUTINE IS TO SUPPLY INITIAL GUESSES FOR TRACK PLOT OF
C      THE SOURCE VEHICLE.
C      UPON SUCCESSFUL TRACKING OF THE FIRST THREE POINTS, PROCESSING
C      SHALL CONTINUE FOR ALL REQUIRED TRACK POINTS IN THE
C      SUBROUTINE TRKPT.
C
C      VARIOUS COMBINATIONS OF V*DTT FACTORS ADDED TO XI, YI, ZI.
C      ASSUMPTION: ONE TIME LAG WILL DOMINATE ONE COORDINATE.
C      ALL INITIAL GUESSES PERTURBATED ABOUT CENTER OF UNDERWATER (4)
C      QUADRANTS.
C
  700 NGES=NGES+1
       FLAG=0.0
       GO TO 900
  880 CALL TRKPT(XI,YI,ZI,M,FLAG)
```

```
      IF(FLAG.EQ.1.0.AND.NGES.LT.24) GO TO 700
      IF(FLAG.EQ.1.0.AND.NGES.GE.24) GO TO 900
      GO TO 1000
  900 FLAG=0.0
      WRITE(50,2000)
      READ(50,) XI,YI,ZI
 2000 FORMAT(5X,'FAILURE TO CONVERGE FOR INITIAL SOLUTION',//,
     115X,'INPUT:',5X,'XXX.XX',5X,'YYY.YY',5X,'ZZZ.ZZ',5X,'GUESS')
      GO TO 880
C
C     FLAG=0.0 CLEARED
C         =1.0 NO CONVERGENCE
C         =2.0 TRACK COMPLETED
C
 1000 RETURN
      END

               GUESS1          SIZE      144    00220
                      MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
   61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

      SUBROUTINE TRKPT(XI,YI,ZI,M,FLAG)
      COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2).XLAT,YLONG,ZDEPTH
      COMMON /ZL4/ FN(25,50),IXAX(10),IYAX(5),IZAX(5),IYZAX(10)
      COMMON /ZL5/ DTT12(3,50),DTT13(3,50),DTT23(3,50)
      COMMON /ZL6/ FXTRK(50),FYTRK(50),FXSENS(3),FYSENS(3)
C
C     THIS SUBROUTINE LOCATES TRACKING LOCATIONS FOR SPECIFIC
C     DATAPOINTS (TIME LAGS) RECEIVED.
C
      WRITE(60,8977) ((I,XSENS(I),YSENS(I)).I=1,3)
 8977 FORMAT(3(5X,I2,5X,'X= ',F8.2,10X,'Y= ',F8.2,/))
      WRITE(60,898) M
```

```
  898 FORMAT(20X,'M= ',I2,//)
      DX=0.0
      IF(M.NE.1) GO TO 2
      FYI=YI
      FXI=XI
      DO 111 K=1,3
      FYSENS(K)=YSENS(K)
      FXSENS(K)=XSENS(K)
  111 CONTINUE
      GO TO 4
    2 IF(M.NE.2) GO TO 3
      FXI=YI
      FYI=ZI
      DO 121 K=1,3
      FXSENS(K)=YSENS(K)
      FYSENS(K)=ZSENS(K)
  121 CONTINUE
      GO TO 4
    3 IF(M.NE.3) GO TO 4
      FXI=XI
      FYI=ZI
      DO 122 K=1,3
      FXSENS(K)=XSENS(K)
      FYSENS(K)=ZSENS(K)
  122 CONTINUE
    4 CONTINUE
      WRITE(60,819) ((I,FXSENS(I),FYSENS(I)),I=1,3)
  819 FORMAT(/,3(5X,I2,5X,'FXSENS(I)= ',E12.4,15X,'FYSENS(I)= ',
     1E12.4,/))
      DO 200 N=1,NPTS
      TRYTWO=0.0
    5 IT=0
   10 IT=IT+1
      IF(IT.EQ.1.AND.N.EQ.1) GO TO 15
      IF(IT.NE.1) GO TO 20
```

```
          FXTRK(N)=FXTRK(N-1)
          FYTRK(N)=FYTRK(N-1)
          IF(FXTRK(N-1).EQ.0.0.AND.N.GE.3) FXTRK(N)=FXTRK(N-2)
                         MODULE:  TRKPT    OPTIONS: SE AB WA LO MA DO SR
 61516-02 ENHANCED SAU FORTRAN COMPILER       REVISION LEVEL 33.071978

          IF(FYTRK(N-1).EQ.0.0.AND.N.GE.3) FYTRK(N)=FYTRK(N-2)
          GO TO 20
       15 Y=FYI
          X=FXI
          FXTRK(N)=X
          FYTRK(N)=Y
       20 CONTINUE
          CALL SYSEQN(N,M,DDH,DDK,DD)
          USE=1.0
          CALL PTCHK(N,M,USE,DDH,DDK,DX)
 C
 C        DX=1.0  CORRECT SOLUTION POINT
 C        DX=0.0  NO SOLUTION POINT FOUND.
 C
          IF(DX.EQ.1.0) FLAG=0.0
          IF(DX.EQ.0.0) FLAG=1.0
          IF(DX.NE.1.0.AND.IT.LE.25) GO TO 10
          IF(DX.NE.1.0.AND.IT.GE.25) GO TO 150
          GO TO 180
      150 IF(TRYTWO.NE.1.0) GO TO 160
          FXTRK(N)=0.0
          FYTRK(N)=0.0
          GO TO 180
      160 TRYTWO=1.0
          FXTRK(N)=FXTRK(N-1)+25.0
          FYTRK(N)=FYTRK(N-1)+25.0
          GO TO 5
      180 DX=0.0
      200 CONTINUE
```

```
      IF(M.NE.1) GO TO 210
      DO 205 NN=1,NPTS
      XTRK(NN)=FXTRK(NN)
      YTRK(NN)=FYTRK(NN)
  205 CONTINUE
      GO TO 900
  210 IF(M.NE.2) GO TO 220
      DO 215 NN=1,NPTS
      YTRK(NN)=FXTRK(NN)
      ZTRK(NN)=FYTRK(NN)
  215 CONTINUE
  220 IF(M.NE.3) GO TO 900
      DO 225 NN=1,NPTS
      XTRK(NN)=FXTRK(NN)
      ZTRK(NN)=FYTRK(NN)
  225 CONTINUE
  900 RETURN
      END
```

```
                  TRKPT            SIZE      922     01632
                           MODULE:   *MAIN*    OPTIONS: SE AB WA LO MA DO SR
  61516-02 ENHANCED SAU FORTRAN COMPILER          REVISION LEVEL 33.071978
```

```
      SUBROUTINE SYSEQN(N,M,DDH,DDK,DD)
      COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
      COMMON /ZL4/ FN(25,50),IXAX(10),IYAX(5),IZAX(5),IYZAX(10)
      COMMON /ZL5/ DTT12(3,50),DTT13(3,50),DTT23(3,50)
      COMMON /ZL6/ FXTRK(50),FYTRK(50),FXSENS(3),FYSENS(3)
C
C     THIS SUBROUTINE SOLVES THE NONLINEAR SIMULTANEOUS SYSTEM
C     EQUATIONS DESCRIBING THE PARTICLE PATH DYNAMICS.  A NEWTON-RAPHSO
C     ALGORITHMIC SET OF EQUATIONS HAS BEEN IMPLEMENTED.
C
```

```
      ITSYS=0.0
  100 CONTINUE
      ITSYS=ITSYS+1.0
      DD1=(DTT12(M,N)*V)
      DD2=(DTT13(M,N)*V)
      DD3=(DTT23(M,N)*V)
      A=(FXTRK(N)-FXSENS(1))**2+(FYTRK(N)-FYSENS(1))**2
      B=(FXTRK(N)-FXSENS(2))**2+(FYTRK(N)-FYSENS(2))**2
      C=(FXTRK(N)-FXSENS(3))**2+(FYTRK(N)-FYSENS(3))**2
      EMF=DD1+SQRT(B)-SQRT(A)
      FMF=DD2+SQRT(C)-SQRT(A)
      GMF=DD3+SQRT(C)-SQRT(B)
      EX=((FXTRK(N)-FXSENS(1))/SQRT(A))-((FXTRK(N)-FXSENS(2))/SQRT(B))
      EY=((FYTRK(N)-FYSENS(1))/SQRT(A))-((FYTRK(N)-FYSENS(2))/SQRT(B))
      FX=((FXTRK(N)-FXSENS(1))/SQRT(A))-((FXTRK(N)-FXSENS(3))/SQRT(C))
      FY=((FYTRK(N)-FYSENS(1))/SQRT(A))-((FYTRK(N)-FYSENS(3))/SQRT(C))
      GX=((FXTRK(N)-FXSENS(2))/SQRT(B))-((FXTRK(N)-FXSENS(3))/SQRT(C))
      GY=((FYTRK(N)-FYSENS(2))/SQRT(B))-((FYTRK(N)-FYSENS(3))/SQRT(C))
      WRITE(60,887) FXTRK(N),FYTRK(N),((FXSENS(I),FYSENS(I)),I=1,3)
  887 FORMAT(5X,'FXTRK(N)= ',E12.4,10X,'FYTRK(N)= ',E12.4,/,
     13(5X,'XSENS(I)= ',E12.4,20X,'YSENS(I)= ',E12.4,/))
      WRITE(60,888) A,B,C,EMF,FMF,GMF,EX,EY,FX,FY,GX,GY
  888 FORMAT(5X,'A= ',E12.4,/,5X,'B= ',E12.4,/,5X,'C= ',E12.4,
     1/,5X,'EMF= ',E12.4,/,5X,'FMF= ',E12.4,/,5X,'GMF= ',E12.4,
     2/,5X,'EX= ',E12.4,/,5X,'EY= ',E12.4,/,5X,'FX= ',E12.4,
     3/,5X,'FY= ',E12.4,/,5X,'GX= ',E12.4,/,5X,'GY= ',E12.4,//)
C
C     SOLVE FUNCTIONS AND PARTIALS VIA CRAMER'S RULE FOR DETERMINING
C     TRACK POINT, WITHIN A SPECIFIED TOLERANCE.
C
      XN(2)=FXTRK(N)
      YN(2)=FYTRK(N)
      CF=0.0
      IF(EX.EQ.0.0.AND.EY.EQ.0.0) CF=1.0
      IF(FX.EQ.0.0.AND.FY.EQ.0.0) GO TO 131
```

```
      IGA=EX/FX
      IGB=EY/FY
      IF(IGA.EQ.IGB) GO TO 131
      EEX=EX
      EEY=EY
                              MODULE:  SYSEQN   OPTIONS: SE AB WA LO MA DO SR
61516-02 ENHANCED SAU FORTRAN COMPILER            REVISION LEVEL 33.071978

      FEX=FX
      FEY=FY
      XMF=EMF
      YMF=FMF
      GO TO 135
  131 IF(GX.EQ.0.0.AND.GY.EQ.0.0) GO TO 132
      IF(CF.EQ.1.0) GO TO 133
      XMF=EMF
      YMF=GMF
      EEX=EX
      EEY=EY
      FEX=GX
      FEY=GY
      GO TO 135
  132 FX=0.0001
      FY=0.0001
  133 XMF=FMF
      YMF=GMF
      EEX=FX
      EEY=FY
      FEX=GX
      FEY=GY
  135 CONTINUE
      WRITE(60,1110)
 1110 FORMAT(3X,'N',8X,'EMF',8X,'FMF',8X,'GMF',8X,'EX',8X,'EY',
     18X,'FX',8X,'FY',8X,'GX',8X,'GY')
      CFLAG=0.0
      WRITE(60,1111) N,EMF,FMF,GMF,EX,EY,FX,FY,GX,GY
```

```
 1111 FORMAT(2X,I2,2X,9(E9.1,2X))
      CALL CRAMER(N,XMF,YMF,EEX,EEY,FEX,FEY,DDH,DDK,CFLAG)
C
C     CFLAG=1.0  FOR FURTHER ACCURACY REQUIRED.
C     CFLAG=2.0  FOR WRONSKIAN=0.0.
C     CFLAG=3.0  FOR SOLUTION TOLERANCE ACHIEVED.
C
      IF(CFLAG.NE.2.0) GO TO 150
      FXTRK(N)=0.0
      FYTRK(N)=0.0
      GO TO 500
  150 IF(CFLAG.NE.1.0) GO TO 300
      IF(ITSYS.GE.25) GO TO 300
      FXTRK(N)=XN(2)
      FYTRK(N)=YN(2)
      GO TO 100
  300 FXTRK(N)=XN(2)
      FYTRK(N)=YN(2)
  500 RETURN
      END

               SYSEQN          SIZE      989    01735
                      MODULE:  *MAIN*    OPTIONS: SE AB WA LO MA DO SR
 61516-02 ENHANCED SAU FORTRAN COMPILER       REVISION LEVEL 33.071978

      SUBROUTINE CRAMER(N,XMF,YMF,EEX,EEY,FEX,FEY,DDH,DDK,CFLAG)
      COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS SUBROUTINE SOLVES FOR THE X,Y,Z, INCREMENTS VIA CRAMER'S RUL
C     FOR THE NEWTON-RAPHSON SYSTEM EQUATIONS.  AN X,Y,Z SOLUTION IS FO
C     WITHIN SOME ACCURACY; OR WITHIN 25 ITERATIONS - WHICHEVER IS FIRS
C
      XN(1)=XN(2)
```

```
      YN(1)=YN(2)
      CALL MTX(EEX,FEX,EEY,FEY,WRON)
      IF(WRON.NE.0.0) GO TO 30
      CFLAG=2.0
      GO TO 900
   30 CALL MTX(XMF,YMF,EEY,FEY,DH)
      CALL MTX(EEX,FEX,XMF,YMF,DK)
      DDH=DH/WRON
      DDK=DK/WRON
      WRITE(60,1221) DDH,DDK
 1221 FORMAT(/,5X,'DDH= ',E12.4,5X,'DDK= ',E12.4,//)
      XN(2)=XN(1)+DDH
      YN(2)=YN(1)+DDK
      WRITE(60,4321) XN(2),YN(2)
 4321 FORMAT(10X,'X= ',E12.4,10X,'Y= ',E12.4,)
      XH=ABS(DDH)
      YK=ABS(DDK)
      IF(XH.LE.0.5.AND.YK.LE.0.5) GO TO 800
      CFLAG=1.0
      GO TO 900
  800 CFLAG=3.0
  900 PETURN
      END

                 CRAMER          SIZE     194    00302
                        MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
 61516-02 ENHANCED SAU FORTRAN COMPILER      REVISION LEVEL 33.071978

      SUBROUTINE MTX(A,B,C,D,S)
      COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),VV
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS ROUTINE FINDS THE DETERMINANT OF A 2X2 MATRIX.
C
```

```
C       S=SOLUTION
C
C       A   C
C       B   D
C
        S=(A*D)-(B*C)
        WRITE(60,112) S
    112 FORMAT(10X,E12.4,'=S IN MTX')
        RETURN
        END

                 MTX                 SIZE       53     00065
                               MODULE:   *MAIN*   OPTIONS: SE AB WA LO MA DO SR
  61516-02 ENHANCED SAU FORTRAN COMPILER            REVISION LEVEL 33.071978

        SUBROUTINE PTCHK(N,M,USE,DDH,DDK,DX)
        COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
        COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
        COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
        COMMON /ZL4/ FN(25,50),IXAX(10),IYAX(5),IZAX(5),IYZAX(10)
        COMMON /ZL5/ DTT12(3,50),DTT13(3,50),DTT23(3,50)
        COMMON /ZL6/ FXTRK(50),FYTRK(50),FXSENS(3),FYSENS(3)
C
C       THIS SUBROUTINE IS FOR:
C               USE=1.0  TO CHECK SOLUTION  POINT FOR VALIDITY.
C               USE=2.0  TO PROVIDE TIME LAG SIMULATED DATA.
C
C               DX=1.0  FOR CORRECT SOLUTION.
C               DX=0.0  NO SOLUTION POINT.
C
        DX=0.0
        R1=SQRT((FXTRK(N)-FXSENS(1))**2+(FYTRK(N)-FYSENS(1))**2)
        R2=SQRT((FXTRK(N)-FXSENS(2))**2+(FYTRK(N)-FYSENS(2))**2)
        R3=SQRT((FXTRK(N)-FXSENS(3))**2+(FYTRK(N)-FYSENS(3))**2)
        DT12=(R1-R2)/V
```

```
      DT13=(R1-R3)/V
      DT23=(R2-R3)/V
      IF(USE.EQ.2.0) GO TO 800
      MDT12=1000.0*DT12
      MDT13=1000.0*DT13
      MDT23=1000.0*DT23
      MDTT12=1000.0*DTT12(M,N)
      MDTT13=1000.0*DTT13(M,N)
      MDTT23=1000.0*DTT23(M,N)
      IF(IABS(MDT12-MDTT12).GE.5) GO TO 850
      IF(IABS(MDT13-MDTT13).GE.5) GO TO 850
      IF(IABS(MDT23-MDTT23).GE.5) GO TO 850
  850 SS=SQRT((DDH**2)+(DDK**2))
      IF(0.5.LT.SS) GO TO 900
      DX=1.0
      RETURN
  800 DTT12(M,N)=DT12
      DTT13(M,N)=DT13
      DTT23(M,N)=DT23
  900 RETURN
      END

                 PTCHK          SIZE      354     00542
                      MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
 61516-02 ENHANCED SAU FORTRAN COMPILER      REVISION LEVEL 33.071978

      SUBROUTINE TABULR(DD)
      COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS SUBROUTINE TABULATES VEHICLE SOURCE PATH OUTPUT
C     TO DEVICE 60 (LINE PRINTER).
C
      WRITE(60,1000) V
```

```
      DO 400 I=1,3
      WRITE(60,1100) I,XSENS(I),YSENS(I),ZSENS(I)
  400 CONTINUE
      WRITE(60,1200)
      DO 500 L=1,NPTS
      WRITE(60,1300) L,XTRK(L),YTRK(L),ZTRK(L)
  500 CONTINUE
 1000 FORMAT(1H1,5X,'VELOCITY OF MEDIUM IS ',3X,F8.3,2X,'(M/SEC)',
     1//,40X,'SENSOR POSITIONS',/)
 1100 FORMAT(5X,I2,3(10X,F10.3))
 1200 FORMAT(//,40X,'VEHICLE (SOURCE) PATH',/)
 1300 FORMAT(5X,I2,3(10X,F10.3))
      RETURN
      END

                TABULR          SIZE     280     00430
                      MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
 61516-02 ENHANCED SAU FORTRAN COMPILER      REVISION LEVEL 33.071978

      SUBROUTINE SENSOR(I,DD)
      COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
C
C     THIS SUBROUTINE IS TO BE MADE TO COLLECT REAL DATA FOR TIME LAGS
C     VIA PHASE COHERENCE DISCRIMINATION/ CROSS CORRELATION/ OR TIME
C     DIFFERENCE MEASUREMENTS.  DATA TO BE INPUT VIA A/D CONVERTOR TO
C     TRACKING COMPUTER.
C
      WRITE(50,100)
  100 FORMAT(20X,'***** WARNING *****',//,15X,
     1'PROGRAM TRYING TO ACCESS THE REAL DATA.',/,15X,
     2'THIS PART OF THE PROGRAM TO REMAIN INCOMPLETE AT THIS TIME',15X,
     3/,15X,'DUE TO LACK OF PROPER HARDWARE IMPLEMENTATION.')
      RETURN
```

```
      END

                  SENSOR            SIZE      101     00145
                           MODULE:  *MAIN*    OPTIONS: SE AB WA LO MA DO SR
 61516-02 ENHANCED SAU FORTRAN COMPILER          REVISION LEVEL 33.071978

      SUBROUTINE TRKPLT(ANS,IDD)
      COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
      COMMON /ZL4/ FN(25,50),IXAX(10),IYAX(5),IZAX(5),IYZAX(10)
C
C     THIS ROUTINE DETERMINES THE GRAPHIC DISPLAY REQUIREMENTS FOR
C     THE LINE PRINTER DISPLAY OF THE TRACKING SCENARIO.
C
C     INPUT GRAPH SPECIFICATIONS
C
      WRITE(50,100)
      READ(50,) SF
      WRITE(60,50505) SF
      WRITE(50,115)
      READ(50,) WX
      WRITE(60,40404) WX
      WRITE(50,125)
      READ(50,) HY
      WRITE(60,30303) HY
      WRITE(50,135)
      READ(50,) BZ
      WRITE(60,20202) BZ
C
C     SET OUTPUT TO BLANKS
C
      DO 8150 I=1,25
      DO 8100 J=1,50
      FN(I,J)=' '
 8100 CONTINUE
```

```
 8150 CONTINUE
C
C      SET X-AXIS:  FOR Y VS X, OR Z VS X.
C
       FAKE=5.0
       XADIV=WX/50.0
       DO 1000 I=1,10
       IXAX(I)=(FAKE*XADIV/SF)
       FAKE=FAKE+5.0
 1000 CONTINUE
C
C      SET Y-AXIS:  FOR Y VS X.
C
       FAKE=5.0
       YADIV=HY/25.0
       DO 1010 I=1,5
       IYAX(I)=FAKE*YADIV/SF
       FAKE=FAKE+5.0
 1010 CONTINUE
C
C      SET Y-AXIS:  FOR Z VS Y.
C

                      MODULE:  TRKPLT   OPTIONS: SE AB WA LO MA DO SR
61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

       FAKE=5.0
       YZADIV=HY/50.0
       DO 1020 I=1,10
       IYZAX(I)=FAKE*YZADIV/SF
       FAKE=FAKE+5.0
 1020 CONTINUE
C
C      SET Z-AXIS:  FOR Z VS X, OR Z VS Y.
C
       FAKE=5.0
```

```
      ZADIV=BZ/25.0
      DO 1030 I=1,5
      IZAX(I)=FAKE*ZADIV/SF
      FAKE=FAKE+5.0
 1030 CONTINUE
C
C     STORE STANDARD AXIS GRID.
C
      DO 1045 J=5,45,5
      DO 1040 L=1,25
      FN(L,J)=':'
 1040 CONTINUE
 1045 CONTINUE
      DO 1055 J=5,20,5
      DO 1050 L=1,50
      FN(J,L)='-'
 1050 CONTINUE
 1055 CONTINUE
      DO 1060 M=1,25
      FN(M,1)='I'
      FN(M,50)='I'
 1060 CONTINUE
      DO 1070 M=1,50
      FN(1,M)='I'
      FN(25,M)='I'
 1070 CONTINUE
C
C     STORE PARTICLE PATH.
C
      DO 1080 MI=1,NPTS
      IX=((XTRK(MI)/XADIV)*SF)+0.5
      IY=((YTRK(MI)/YADIV)*SF)+0.5
      IZ=((ZTRK(MI)/ZADIV)*SF)+0.5
      IYZ=((YTRK(MI)/YZADIV)*SF)+0.5
      IF(IX.LE.50.AND.IY.LE.25.AND.IDD.EQ.1) FN(IY,IX)='T'
```

```
      IF(IX.LE.50.AND.IZ.LE.25.AND.IDD.EQ.2) FN(IZ,IX)='T'
      IF(IYZ.LE.50.AND.IZ.LE.25.AND.IDD.EQ.3) FN(IZ,IYZ)='T'
 1080 CONTINUE
C
C     STORE SENSOR GRID POSITIONS.
C
      DO 9998 LI=1,3
                         MODULE:  TRKPLT   OPTIONS: SE AB WA LO MA DO SR
61516-02 ENHANCED SAU FORTRAN COMPILER         REVISION LEVEL 33.071978

      LX=((XSENS(LI)/XADIV)*SF)+0.5
      LY=((YSENS(LI)/YADIV)*SF)+0.5
      LZ=((ZSENS(LI)/ZADIV)*SF)+0.5
      LYZ=((YSENS(LI)/YZADIV)*SF)+0.5
      IF(LX.LE.50.AND.LY.LE.25.AND.LI.EQ.1.AND.IDD.EQ.1) FN(LY,LX)='1'
      IF(LX.LE.50.AND.LY.LE.25.AND.LI.EQ.2.AND.IDD.EQ.1) FN(LY,LX)='2'
      IF(LX.LE.50.AND.LY.LE.25.AND.LI.EQ.3.AND.IDD.EQ.1) FN(LY,LX)='3'
      IF(LX.LE.50.AND.LZ.LE.25.AND.LI.EQ.1.AND.IDD.EQ.2) FN(LZ,LX)='1'
      IF(LX.LE.50.AND.LZ.LE.25.AND.LI.EQ.2.AND.IDD.EQ.2) FN(LZ,LX)='2'
      IF(LX.LE.50.AND.LZ.LE.25.AND.LI.EQ.3.AND.IDD.EQ.2) FN(LZ,LX)='3'
      IF(LYZ.LE.50.AND.LY.LE.25.AND.LI.EQ.1.AND.IDD.EQ.3) FN(LZ,LYZ)='1
      IF(LYZ.LE.50.AND.LY.LE.25.AND.LI.EQ.2.AND.IDD.EQ.3) FN(LZ,LYZ)='2
      IF(LYZ.LE.50.AND.LY.LE.25.AND.LI.EQ.3.AND.IDD.EQ.3) FN(LZ,LYZ)='3
 9998 CONTINUE
      IF(IDD.EQ.2) GO TO 2101
      IF(IDD.EQ.3) GO TO 2301
C
C     DUMP GRAPHS AND APPROPRIATE LABELS.
C
C     FIRST:  Y VS X:
C
C     GT=1 FOR Y VS X
C     GT=2 FOR Z VS X
C     GT=3 FOR Z VS Y
C
```

```
          F=0.0
          DO 2100 I=1,25
          CALL GRAPHO(I,F,1,SF)
     2100 CONTINUE
          GO TO 2500
     2101 CONTINUE
   C
   C      NEXT:  Z VS X:
   C
          F=0.0
          DO 2300 I=1,25
          CALL GRAPHO(I,F,2,SF)
     2300 CONTINUE
          GO TO 2500
     2301 CONTINUE
   C
   C      FINALLY:  Z VS Y:
   C
          F=0.0
          DO 2500 I=1,25
          CALL GRAPHO(I,F,3,SF)
     2500 CONTINUE
          WRITE(50,140)
          READ(50,) ANS
      100 FORMAT(10X,'SCALE FACTOR=?')
    50505 FORMAT(1H1,10X,'SCALE FACTOR= ',F10.5)
      115 FORMAT(/,10X,'X MAX GRAPH= ',5X,'??')
                                MODULE:  TRKPLT   OPTIONS: SE AB WA LO MA DO SR
 61516-02 ENHANCED SAU FORTRAN COMPILER       REVISION LEVEL 33.071978

    40404 FORMAT(/,10X,'X MAX= ',F10.5)
      125 FORMAT(/,10X,'Y MAX GRAPH= ',5X,'??')
    30303 FORMAT(1H@,T35,'Y MAX= ',F10.5)
      135 FORMAT(/,10X,'Z MAX GRAPH= ',5X,'??')
    20202 FORMAT(1H@,T60,'Z MAX= ',F10.5)
```

```
  140 FORMAT(/,10X,'DO YOU WANT TO RUN ADDITIONAL PLOTS?',/,
     120X,'1= YES',/,20X,'2= NO',/,'??')
      RETURN
      END

                  TRKPLT           SIZE     1518     02756
                            MODULE:  *MAIN*   OPTIONS: SE AB WA LO MA DO SR
  61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

      SUBROUTINE GRAPHO(I,F,NI,SF)
      COMMON /ZL1/ XTRK(50),YTRK(50),NPTS
      COMMON /ZL2/ ZTRK(50),XSENS(5),YSENS(5),ZSENS(5),V
      COMMON /ZL3/ XN(2),YN(2),ZN(2),XLAT,YLONG,ZDEPTH
      COMMON /ZL4/ FN(25,50),IXAX(10),IYAX(5),IZAX(5),IYZAX(10)
C
C     GRAPH FUNCTION AND OUTPUT LABELS.
C
      II=26-I
      IF(F.NE.0.0) GO TO 10
      IF(NI.NE.3) WRITE(60,100) (IXAX(L),L=1,10)
      IF(NI.EQ.3) WRITE(60,110) (IYZAX(L),L=1,10)
  100 FORMAT(1H1,T49,'X AXIS (METERS)',//,T32,10(I3,2X))
  110 FORMAT(1H1,T49,'Y AXIS (METERS)',//,T32,10(I3,2X))
      F=1.0
   10 IF(I.EQ.18.OR.I.EQ.11) GO TO 131
      GO TO 133
  131 WRITE(60,212)
      GO TO 22
  133 IF(I.NE.17) GO TO 13
      WRITE(60,2133)
 2133 FORMAT(T20,'R')
      GO TO 22
   13 IF(I.NE.15) GO TO 14
      WRITE(60,203)
```

```
      GO TO 22
   14 IF(I.EQ.14.OR.I.EQ.16) GO TO 155
      GO TO 15
  155 WRITE(60,204)
      GO TO 22
   15 IF(I.NE.13) GO TO 16
      WRITE(60,205)
      GO TO 22
   16 IF(I.NE.10) GO TO 17
      WRITE(60,206)
      GO TO 22
   17 IF(I.NE.9) GO TO 18
      WRITE(60,207)
      GO TO 22
   18 IF(I.NE.8) GO TO 19
      WRITE(60,208)
      GO TO 22
   19 IF(NI.EQ.1.AND.I.EQ.6) GO TO 202
      GO TO 20
  202 WRITE(60,209)
      GO TO 22
   20 IF((NI.EQ.2.OR.NI.EQ.3).AND.I.EQ.6) GO TO 211
      GO TO 21
  211 WRITE(60,210)
      GO TO 22
   21 WRITE(60,219)
   22 INX=II/5
                          MODULE:  GRAPHO   OPTIONS: SE AB WA LO MA DO SR
61516-02 ENHANCED SAU FORTRAN COMPILER        REVISION LEVEL 33.071978

      IF(MOD(II,5).EQ.0.AND.NI.EQ.1) WRITE(60,300) IYAX(INX)
      IF(MOD(II,5).EQ.0.AND.NI.NE.1) WRITE(60,300) IZAX(INX)
      WRITE(60,400) (FN(II,J),J=1,50)
      IF(II.NE.1.0) GO TO 555
      WRITE(60,700) SF,XLAT,YLONG,ZDEPTH
```

```
      555 CONTINUE
      212 FORMAT(T20,'S')
      203 FORMAT(T20,'T')
      204 FORMAT(T20,'E')
      205 FORMAT(T20,'M')
      206 FORMAT(T20,'I')
      207 FORMAT(T20,'X')
      208 FORMAT(T20,'A')
      209 FORMAT(T20,'Y')
      210 FORMAT(T20,'Z')
      219 FORMAT(T20,' ')
      300 FORMAT(1H@,T25,I3)
      400 FORMAT(1H@,T30,50(1A1))
      700 FORMAT(//,T22,'SF= ',F6.2,5X,'XMAX= ',F6.2,5X,'YMAX= ',F6.2,5X,
     1'ZMAX= ',F6.2)
          RETURN
          END
```

```
                 GRAPHO          SIZE      668     01234
$CATALOG
61632-02 OVERLAY CATALOGER -- REVISION LEVEL 12.071978
TYPE=BG
DNAME BTRCK,4
BEGIN
              P.NAME=  BTRCK

*HIGH      32103               *START          0
*LOW           0               *LOW            0
*P.END     22252               .PASS           2
*START         0               GUESS1       1372 $
.PASS          2               TRKPT        1612 $
C$ERRO     21712 $             SYSEQN       3444 $
C$IOER     21723 $             CRAMER       5401 $
C$MOD      21677 $             MTX          5703 $
C$OFLO     21714 $             PTCHK        5770 $
```

C$UFLO	21720	$	TABULR	6532	$
CRAMER	5401	$	SENSOR	7162	$
DCF	20143	$	TRKPLT	7327	$
DSQRT	21705	$	GRAPHO	12305	$
ESGN$	17614	$	F$SERA	13563	$
EXIT	21522	$	F$SERI	13570	$
EXP10$	17613	$	F$SERJ	13575	$
F$1ISN	14327	$	F$SERK	13602	$
F$1OSN	14526	$	F$RIFF	13643	$
F$4ISN	14647	$	F$RSFF	13643	$
F$4OSN	14755	$	F$RSSY	13644	$
F$5ISN	14645	$	F$WIFF	13646	$
F$5OSN	14753	$	F$WSFF	13646	$
F$8ISN	14643	$	F$WSSY	13647	$
F$8OSN	14751	$	F$DIFF	13713	$
F$ADDR	20030	$	F$DSSY	13714	$
F$AIP	15076	$	F$EIFF	13716	$
F$AISN	14323	$	F$ESSY	13717	$
F$AOP	16221	$	F$EESY	14017	$
F$B/E	14227	$	F$SYEE	14017	$
F$BBI	22226	$	F$RORW	14032	$
F$BBJ	22157	$	F$FUNC	14135	$
F$BCC	20040	$	F$B/E	14227	$
F$BISN	14312	$	F$CSIO	14237	$
F$BOSN	14515	$	F$CFIO	14237	$
F$BUF	20036	$	F$BISN	14312	$
F$CFIO	14237	$	F$AISN	14323	$
F$CLOS	21577	$	F$1ISN	14327	$
F$CNOP	21524	$	F$BOSN	14515	$
F$CONV	20046	$	F$1OSN	14526	$
F$CSIO	14237	$	F$8ISN	14643	$
F$CV41	21661	$	F$5ISN	14645	$
F$CV51	21661	$	F$4ISN	14647	$
F$D	20033	$	F$8OSN	14751	$
F$DCF	20143	$	F$5OSN	14753	$

F$DCNT	20637 $	F$40SN	14755 $
F$DIFF	13713 $	F$AIP	15076 $
F$DSSY	13714 $	F$RIP	15100 $
F$DVR5	20764 $	F$LIP	15173 $
F$E	20034 $	F$OIP	15225 $
F$E/E	20044 $	F$IIP	15263 $
F$E/E1	20045 $	F$GIP	15533 $
F$E/O	20042 $	F$ROP	16217 $
F$E/O1	20043 $	F$AOP	16221 $
F$EESY	14017 $	F$IOP	16305 $
F$EIFF	13716 $	F$FOP	16601 $
F$EMB	22127 $	F$QOP	16771 $
F$EMBJ	22141 $	F$SOP	16776 $
F$EOP	17154 $	F$O.LB	17005 $
F$ERCD	21656 $	F$O.SN	17011 $
F$ESSY	13717 $	F$OOP	17024 $
F$FCC	20037 $	F$GOP	17152 $
F$FERR	20051 $	F$EOP	17154 $
F$FMT	20035 $	TENT8$	17470 $
F$FMTC	20031 $	EXP10$	17613 $
F$FOP	16601 $	ESGN$	17614 $
F$FUNC	14135 $	MSGN$	17615 $
F$GIP	15533 $	F$NORM	17620 $
F$GOP	17152 $	F$ADDR	20030 $
F$IBF	20047 $	F$SPEC	20031 $
F$IIP	15263 $	F$FMTC	20031 $
F$IOM0	21427 $	F$W	20032 $
F$IOM1	21413 $	F$D	20033 $
F$IOM2	21415 $	F$E	20034 $
F$IOM3	21525 $	F$FMT	20035 $
F$IOM4	21534 $	F$BUF	20036 $
F$IOP	16305 $	F$FCC	20037 $
F$LINE	21160 $	F$BCC	20040 $
F$LIP	15173 $	F$P	20041 $
F$MPRS	20670 $	F$E/O	20042 $

F$NAME	21065	$
F$NCHR	20052	$
F$NORM	17620	$
F$NTRY	21070	$
F$O,LB	17005	$
F$O.SN	17011	$
F$OIP	15225	$
F$OOP	17024	$
F$OPEN	21542	$
F$P	20041	$
F$PNL	20124	$
F$QOP	16771	$
F$RBM	22134	$
F$RBMI	22177	$
F$RIFF	13643	$
F$RIP	15100	$
F$ROP	16217	$
F$RORW	14032	$
F$RSDO	21135	$
F$RSFF	13643	$
F$RSSY	13644	$
F$SERA	13563	$
F$SERI	13570	$
F$SERJ	13575	$
F$SERK	13602	$
F$SOP	16776	$
F$SPEC	20031	$
F$SRTN	21140	$
F$STMT	21131	$
F$SYEE	14017	$
F$W	20032	$
F$WIFF	13646	$
F$WSFF	13646	$
F$WSSY	13647	$
GRAPHO	12305	$
F$E/O1	20043	$
F$E/E	20044	$
F$E/E1	20045	$
F$CONV	20046	$
F$IBF	20047	$
F$FERR	20051	$
F$NCHR	20052	$
F$PNL	20124	$
F$DCF	20143	$
DCF	20143	$
F$DCNT	20637	$
PUTN$	20642	$
F$MPR5	20670	$
F$DVR5	20764	$
F$NAME	21065	$
F$NTRY	21070	$
F$STMT	21131	$
F$RSDO	21135	$
F$SRTN	21140	$
F$LINE	21160	$
F$IOM1	21413	$
F$IOM2	21415	$
F$IOM0	21427	$
EXIT	21522	$
F$CNOP	21524	$
F$IOM3	21525	$
F$IOM4	21534	$
F$OPEN	21542	$
F$CLOS	21577	$
IOERR	21654	$
F$ERCD	21656	$
INT	21657	$
IDINT	21657	$
IFIX	21657	$
IFIX$	21657	$

```
GUESS1     1372 $        F$CV51    21661 $
IDINT     21657 $        F$CV41    21661 $
IFIX      21657 $        MOD$      21673 $
IFIX$     21657 $        C$MOD     21677 $
INT       21657 $        DSQRT     21705 $
IOERR     21654 $        SQRT      21705 $
MOD$      21673 $        C$ERRO    21712 $
MSGN$     17615 $        C$OFLO    21714 $
MTX        5703 $        C$UFLO    21720 $
PTCHK      5770 $        C$IOER    21723 $
PUTN$     20642 $        F$EMB     22127 $
SENSOR     7162 $        F$RBM     22134 $
SQRT      21705 $        F$EMBJ    22141 $
SYSEQN     3444 $        F$BBJ     22157 $
TABULR     6532 $        F$RBMI    22177 $
TENTB$    17470 $        F$BBI     22226 $
TRKPLT     7327 $        *P.END    22252
TRKPT      1612 $        ZL1       22252 C
ZL1       22252 C        ZL6       22563 C
ZL2       31677 C        ZL5       23107 C
ZL3       31655 C        ZL4       24713 C
ZL4       24713 C        ZL3       31655 C
ZL5       23107 C        ZL2       31677 C
ZL6     . 22563 C        *HIGH     32103
$EOJ
TRACK  TIME 00HRS 00MIN 00.000SEC
```

APPENDIX I

THREE SPACE: METHOD TWO: PLANAR SIMULATION DATA, pp. 163-166.

DESCRIPTION:

This appendix contains two sets of data generated in tabular/ disk-file format as a result of the program of Appendix G. The disk data is input to the program of Appendix H; and, the results are the two sets of plot solutions of Appendix J.

VELOCITY OF MEDIUM IS 1450.000 (M/SEC)

SENSOR POSITIONS

1	200.000	180.000	140.000
2	500.000	250.000	450.000
3	300.000	400.000	350.000

VEHICLE (SOURCE) PATH

1	100.000	500.000	110.000
2	120.000	470.000	110.000
3	140.000	440.000	110.000
4	150.000	410.000	110.000
5	150.000	380.000	110.000

5

-0.094095147804	0.077002961749	0.171098109550
-0.095350199221	0.074275931014	0.169626130240
-0.096709908591	0.070282107765	0.166992016360
-0.103079711840	0.058647642852	0.161727354700
-0.115315732940	0.037812290880	0.153128023820
-0.069390195393	0.042347016275	0.111737211660
-0.078221751644	0.028653513614	0.106875265250
-0.088111582976	0.012699686452	0.100811269420
-0.099184561700	-0.005696533031	0.093488028669
-0.111564145030	-0.026616832700	0.084947312335
-0.290050430930	-0.143453049620	0.146597381310
-0.292732354420	-0.147972387960	0.144759966450
-0.295237721930	-0.152663491520	0.142574230410
-0.296307170350	-0.154972355180	0.141334815160
-0.296307170350	-0.154972355180	0.141334815160

200.0000	180.0000	140.0000
500.0000	250.0000	450.0000
300.0000	400.0000	350.0000

VELOCITY OF MEDIUM IS 1450.000 (M/SEC)

SENSOR POSITIONS

1	350.000	150.000	100.000
2	200.000	300.000	100.000
3	200.000	150.000	250.000

VEHICLE (SOURCE) PATH

1	20.000	10.000	10.000
2	30.000	20.000	15.000
3	40.000	25.000	20.000
4	50.001	30.000	25.000
5	50.000	35.000	30.000
6	60.001	40.000	35.000
7	60.000	40.000	39.000
8	65.000	45.000	40.000
9	75.001	50.000	45.000
10	85.001	70.000	50.000
11	95.000	90.000	60.000
12	115.000	110.000	65.000
13	135.000	117.000	70.000
14	155.000	135.000	80.000
15	174.671	144.745	99.721

15

0.011826209146	0.089954401074	0.078128191929
0.012297609448	0.090613086341	0.078315476892
0.011099401978	0.090492064292	0.079392662314
0.009821368412	0.090355973314	0.080534604902
0.011571501350	0.091224755044	0.079653253695
0.010251577365	0.091114696714	0.080863119349
0.010251577365	0.091114696714	0.080863119349
0.010480062101	0.091517611866	0.081037549765
0.009040690055	0.091406477125	0.082365787070
0.013561650552	0.094291692035	0.080730041483
0.018742482973	0.097261980499	0.078519497525
0.020850597142	0.099612743227	0.078762146086
0.016080612748	0.099738363784	0.083657751036
0.016930867856	0.102166832580	0.085235964732
0.012460847806	0.103156080290	0.090695232491
-0.094628523616	-0.076838420703	0.017790102914
-0.094686380756	-0.078095678876	0.016590701881
-0.095166792437	-0.078182469970	0.016984322468
-0.095664419587	-0.078269160703	0.017395258885
-0.096179504341	-0.078354973495	0.017824530847
-0.096712098119	-0.078438875947	0.018273222173
-0.097433352892	-0.077358702821	0.020074650072
-0.097261980499	-0.078519497525	0.018742482973
-0.097828546632	-0.078595015485	0.019233531148
-0.097263610067	-0.083494331699	0.013769278368
-0.097699695003	-0.087681077892	0.010018617112
-0.096583490897	-0.093878760858	0.002704730038
-0.097134142954	-0.095449512937	0.001684630018
-0.097384621645	-0.100455504590	-0.003070882947
-0.103448275860	-0.100057455300	0.003390820554
0.097107937623	0.029001812088	-0.068106125535
0.097262678990	0.028312801639	-0.068949877351
0.097428119773	0.027570825757	-0.069857294016
0.097605395592	0.026770052594	-0.070835342998
0.098295849960	0.028819239242	-0.069476610717
0.098511539055	0.028021648152	-0.070489890903
0.099057885480	0.029741265057	-0.069316620424
0.098975483141	0.028687974433	-0.070287508710
0.099228324420	0.027822020122	-0.071406304298
0.099500944250	0.026876062967	-0.072624881283
0.100522182550	0.028300176911	-0.072222005648
0.100460831620	0.023447832203	-0.077012999421
0.100340571130	0.017728511290	-0.082612059844
0.101226668150	0.013908899234	-0.087317768924
0.103448275860	0.015814439133	-0.087633836729

350.0000	150.0000	100.0000
200.0000	300.0000	100.0000
200.0000	150.0000	250.0000

APPENDIX J

THREE SPACE: METHOD TWO: TRACKING SOLUTION, pp. 168-179.

DESCRIPTION:

This appendix contains two separate tracking solution tabular/ plot sets which are a result of the disk-file data sets of Appendix I which serve as time-difference data inputs to the program of Appendix H.

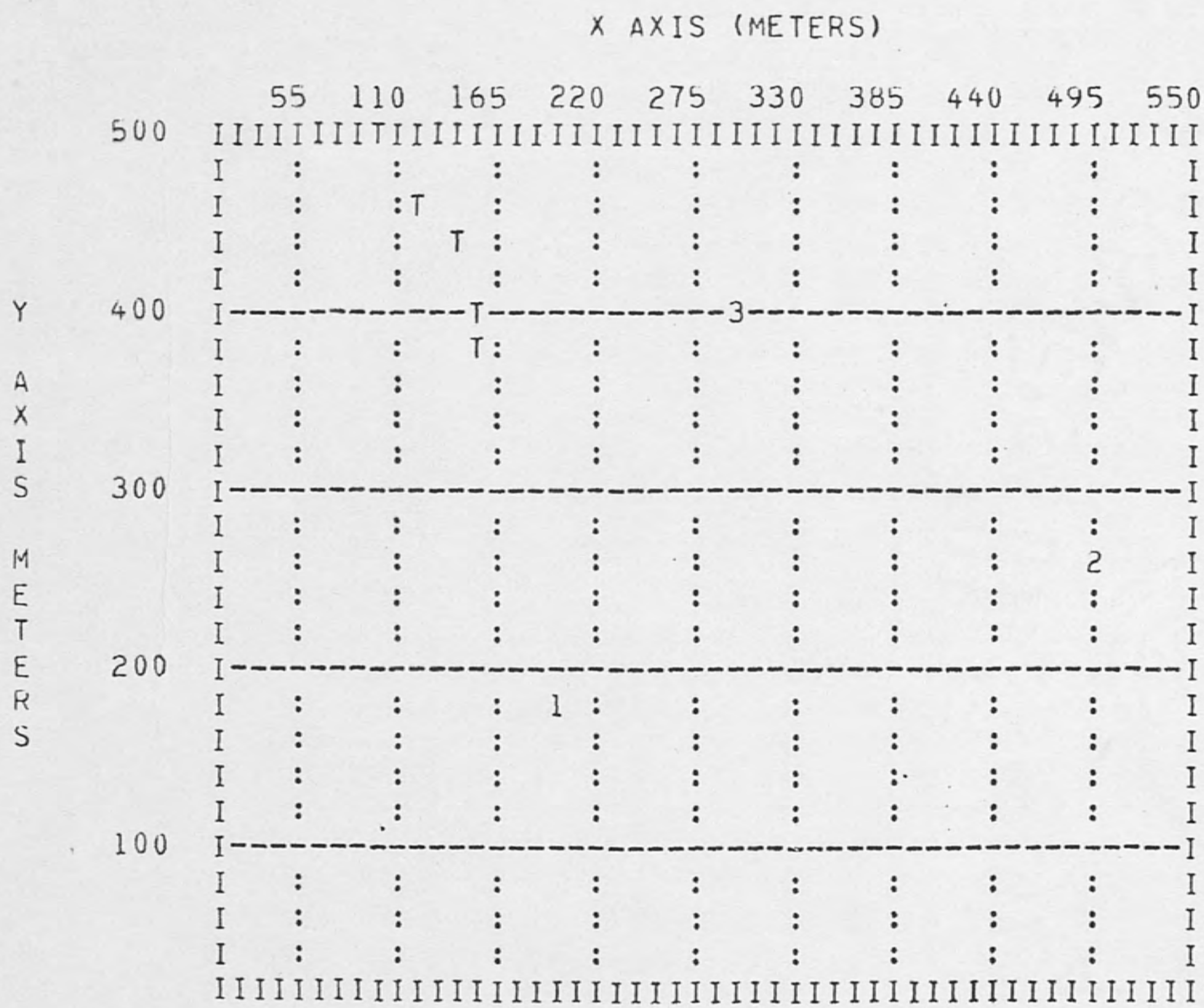
VELOCITY OF MEDIUM IS 1450.000 (M/SEC)

SENSOR POSITIONS

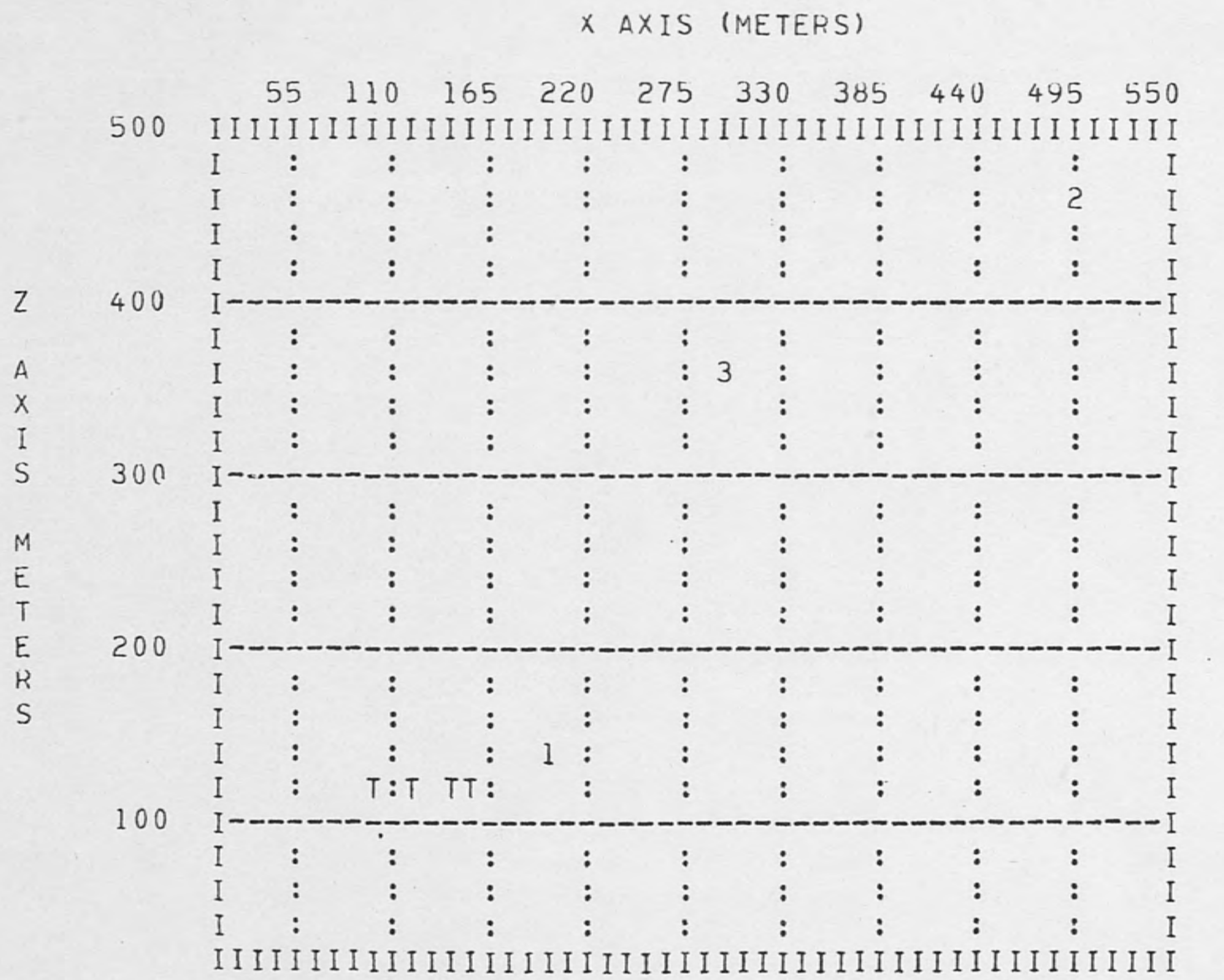
1	200.000	180.000	140.000
2	500.000	250.000	450.000
3	300.000	400.000	350.000

VEHICLE (SOURCE) PATH

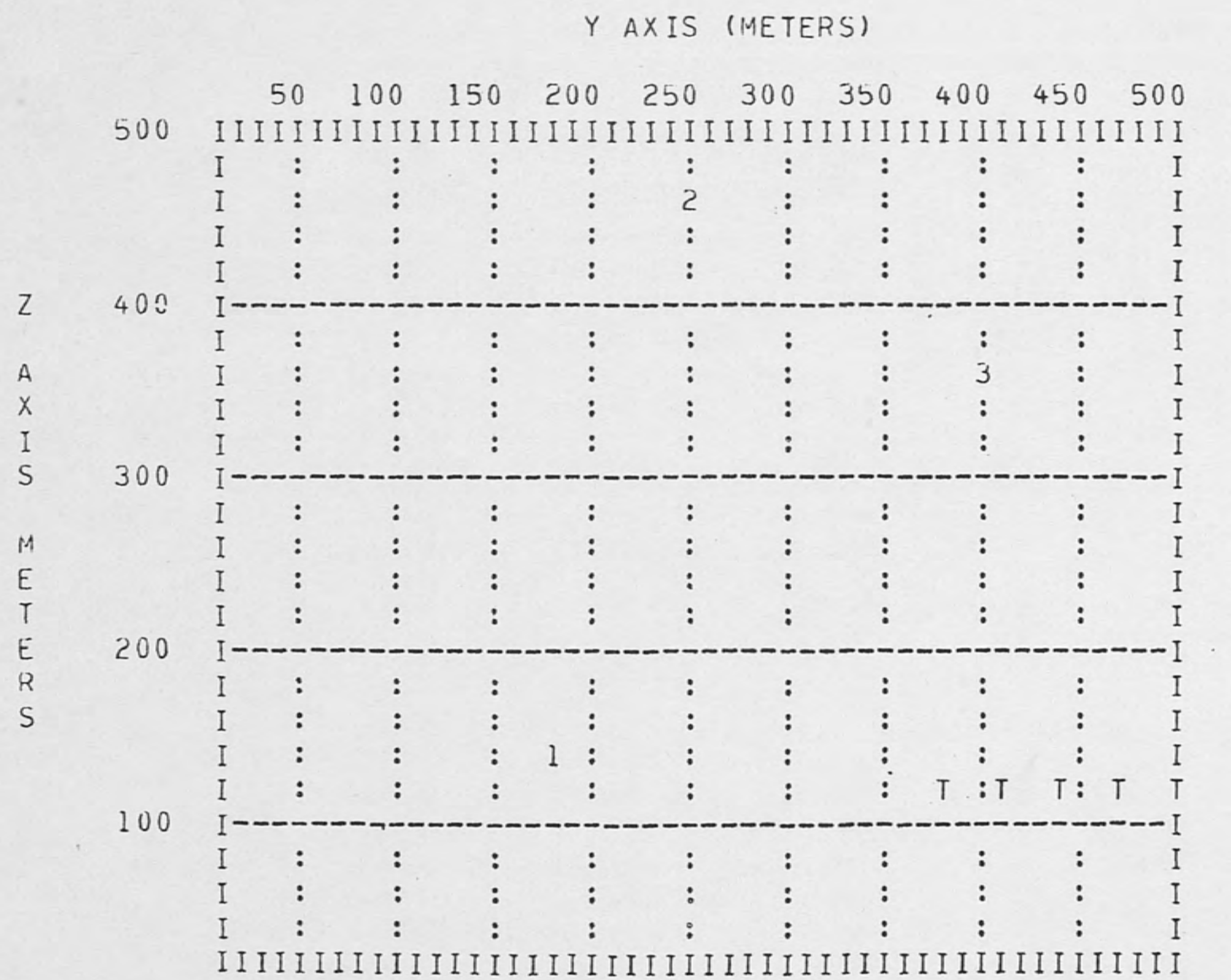
1	100.000	500.000	110.000
2	120.000	470.000	110.000
3	140.000	440.000	110.000
4	150.000	410.000	110.000
5	150.000	380.000	110.000



SF= 1.00 XMAX= 550.00 YMAX= 500.00 ZMAX= 500.00



SF= 1.00 XMAX= 550.00 YMAX= 500.00 ZMAX= 500.00



SF= 1.00 XMAX= 550.00 YMAX= 500.00 ZMAX= 500.00

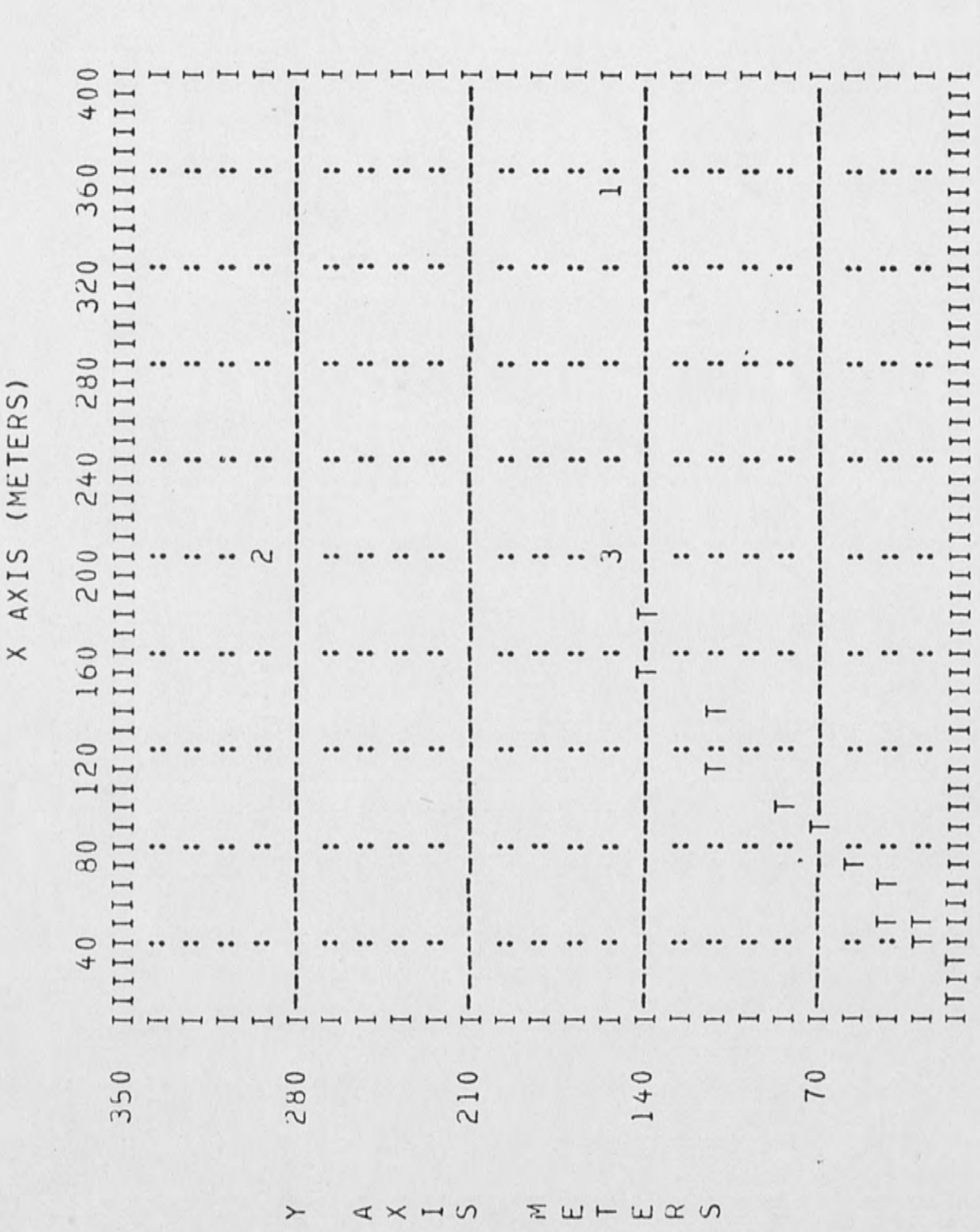
VELOCITY OF MEDIUM IS 1450.000 (M/SEC)

SENSOR POSITIONS

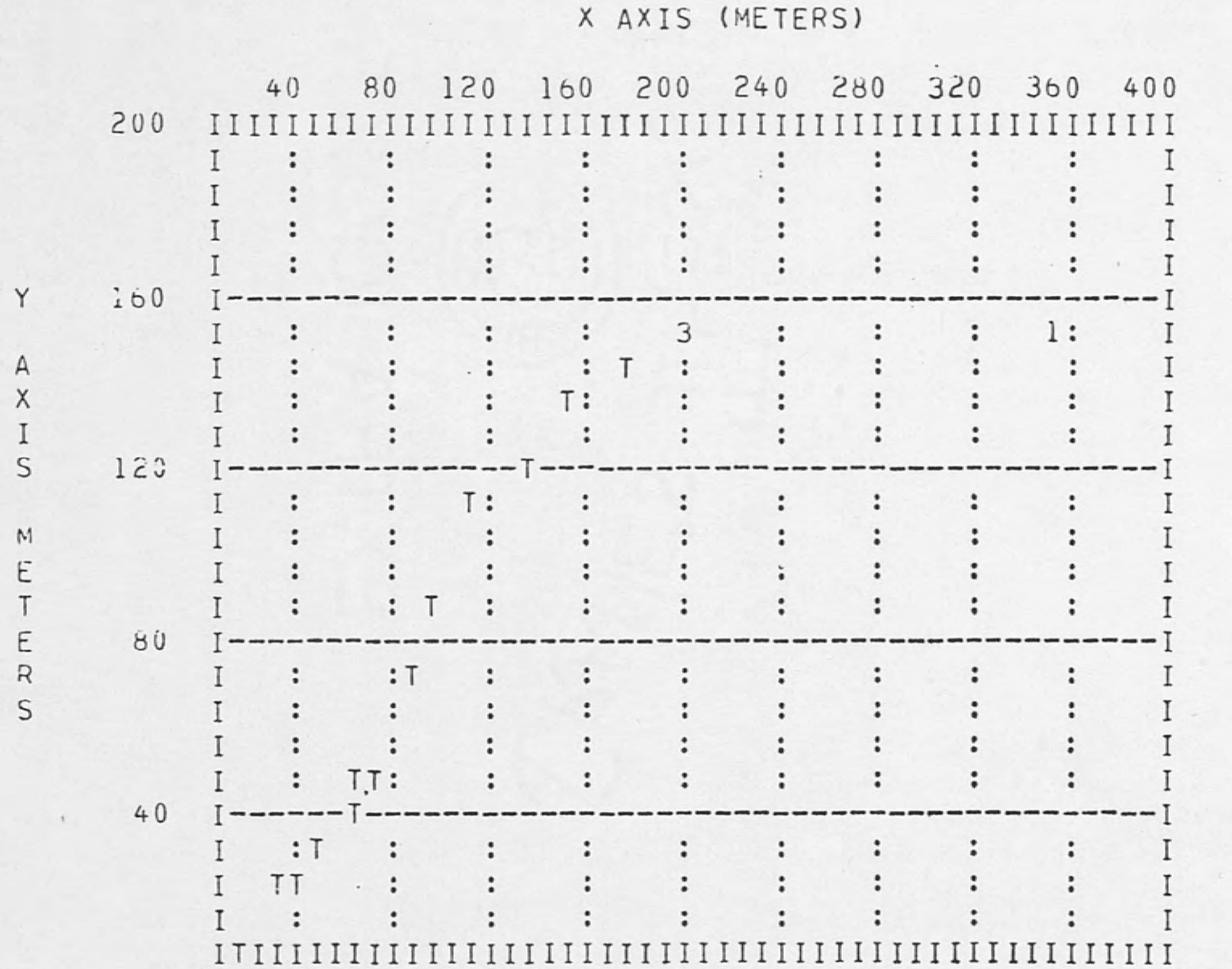
1	350.000	150.000	100.000
2	200.000	300.000	100.000
3	200.000	150.000	250.000

VEHICLE (SOURCE) PATH

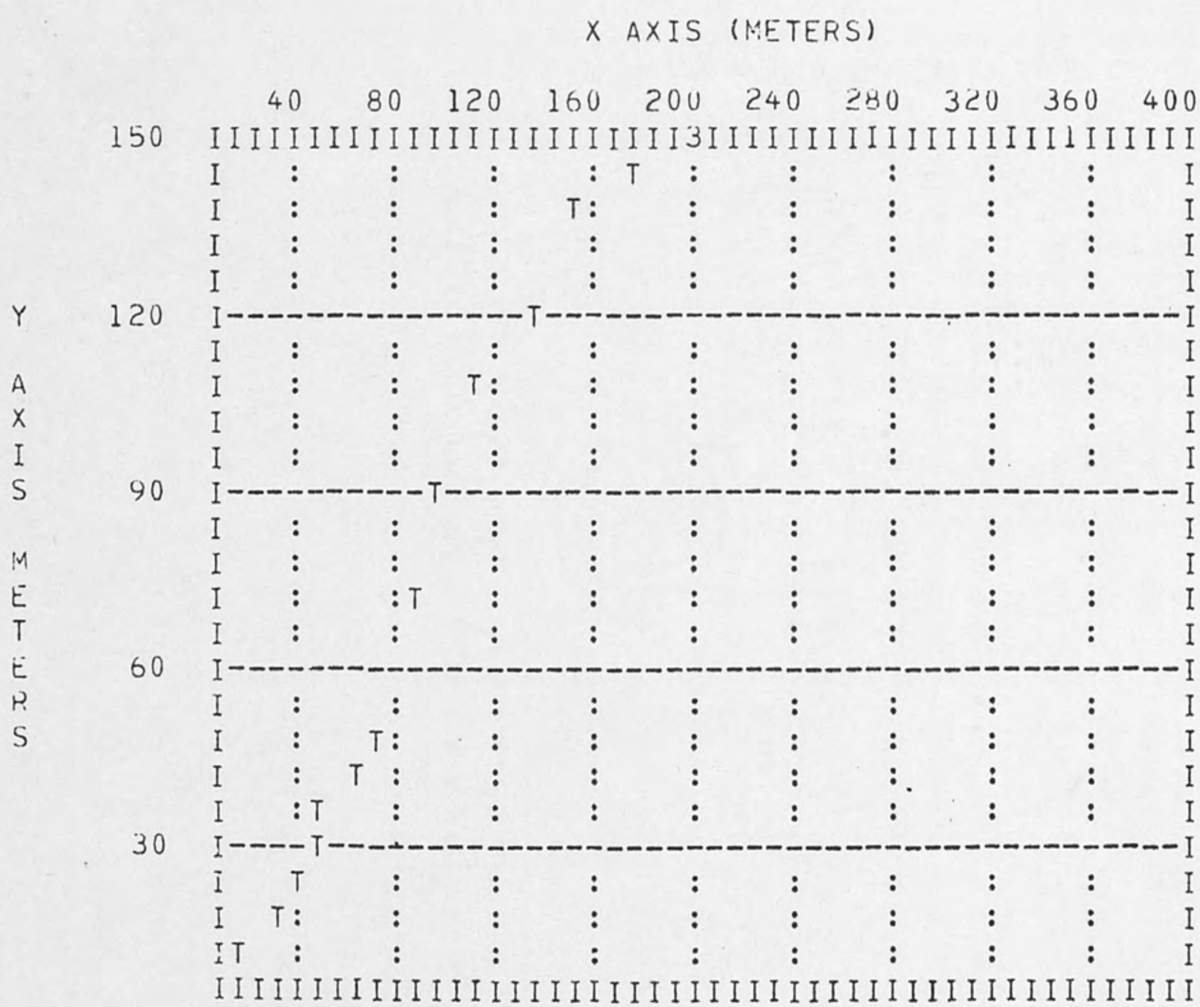
1	20.000	10.000	10.000
2	30.000	20.000	15.000
3	40.000	25.000	20.000
4	50.000	30.000	25.000
5	50.000	35.000	30.000
6	60.000	40.000	35.000
7	60.000	40.000	39.000
8	65.000	45.000	40.000
9	75.000	50.000	45.000
10	85.000	70.000	50.000
11	95.000	90.000	60.000
12	115.000	110.000	65.000
13	135.000	117.000	70.000
14	155.000	135.000	80.000
15	175.000	145.000	100.000



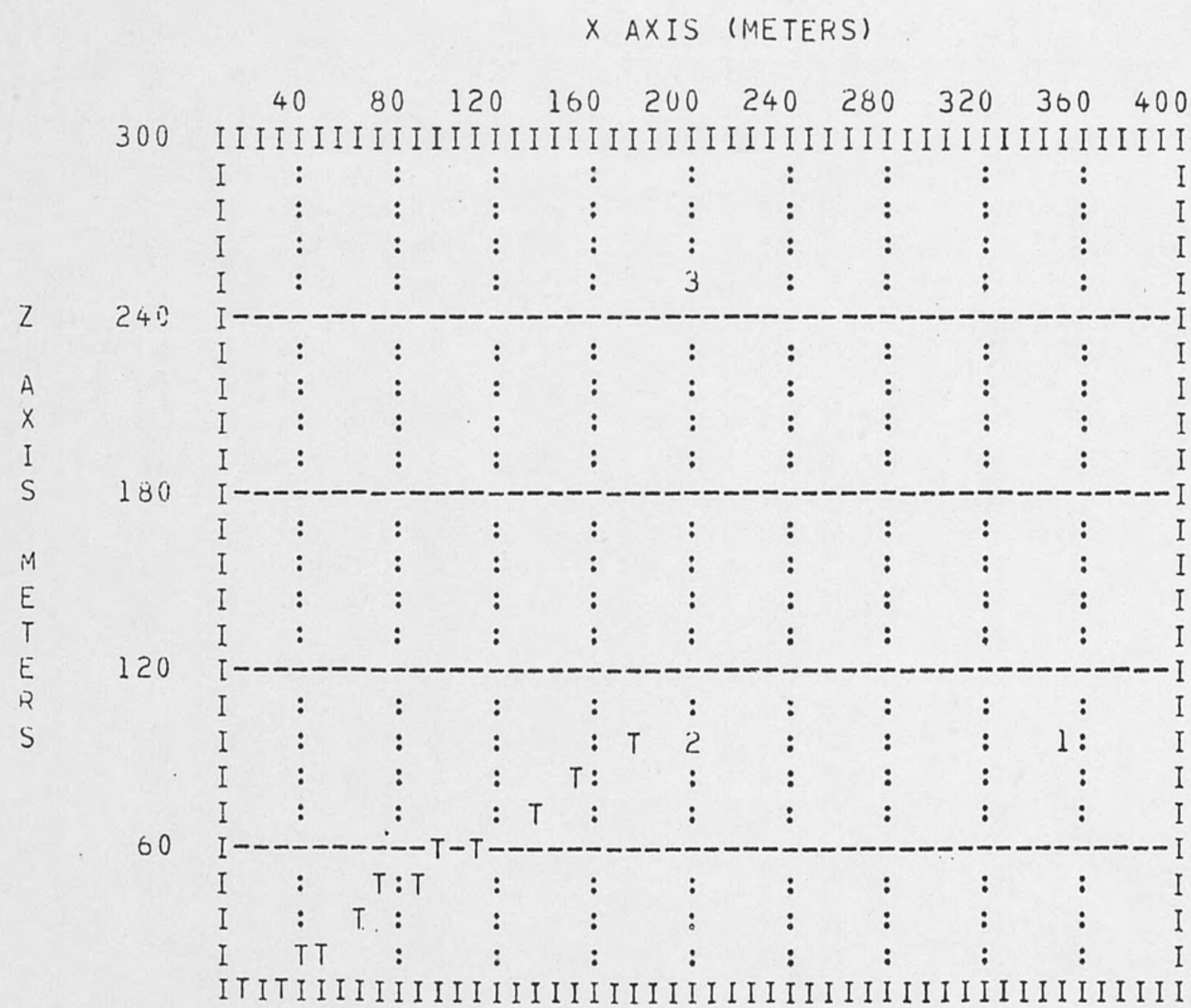
SF= 1.00 XMAX= 400.00 YMAX= 350.00 ZMAX= 300.00



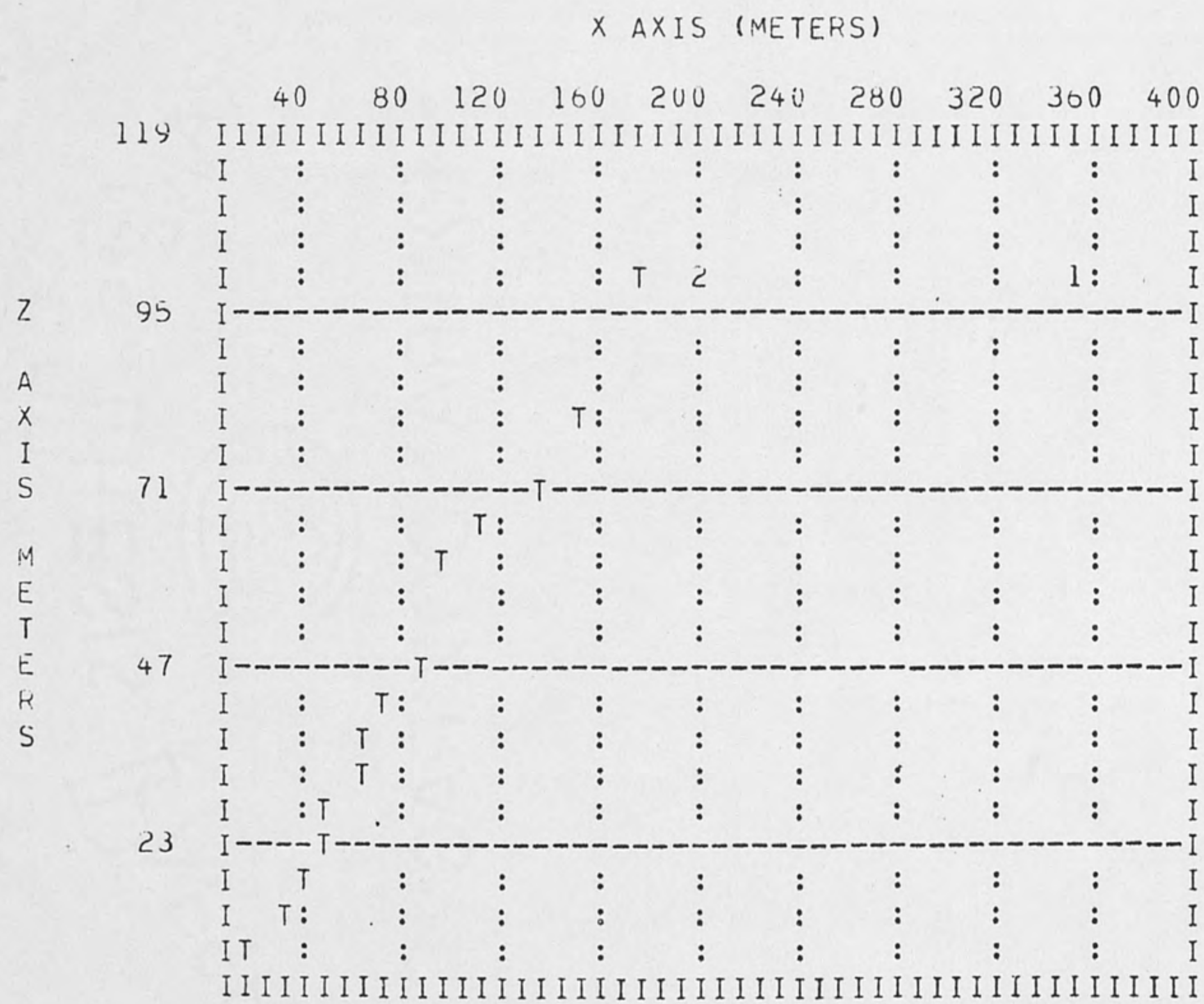
SF= 1.00 XMAX= 400.00 YMAX= 350.00 ZMAX= 300.00



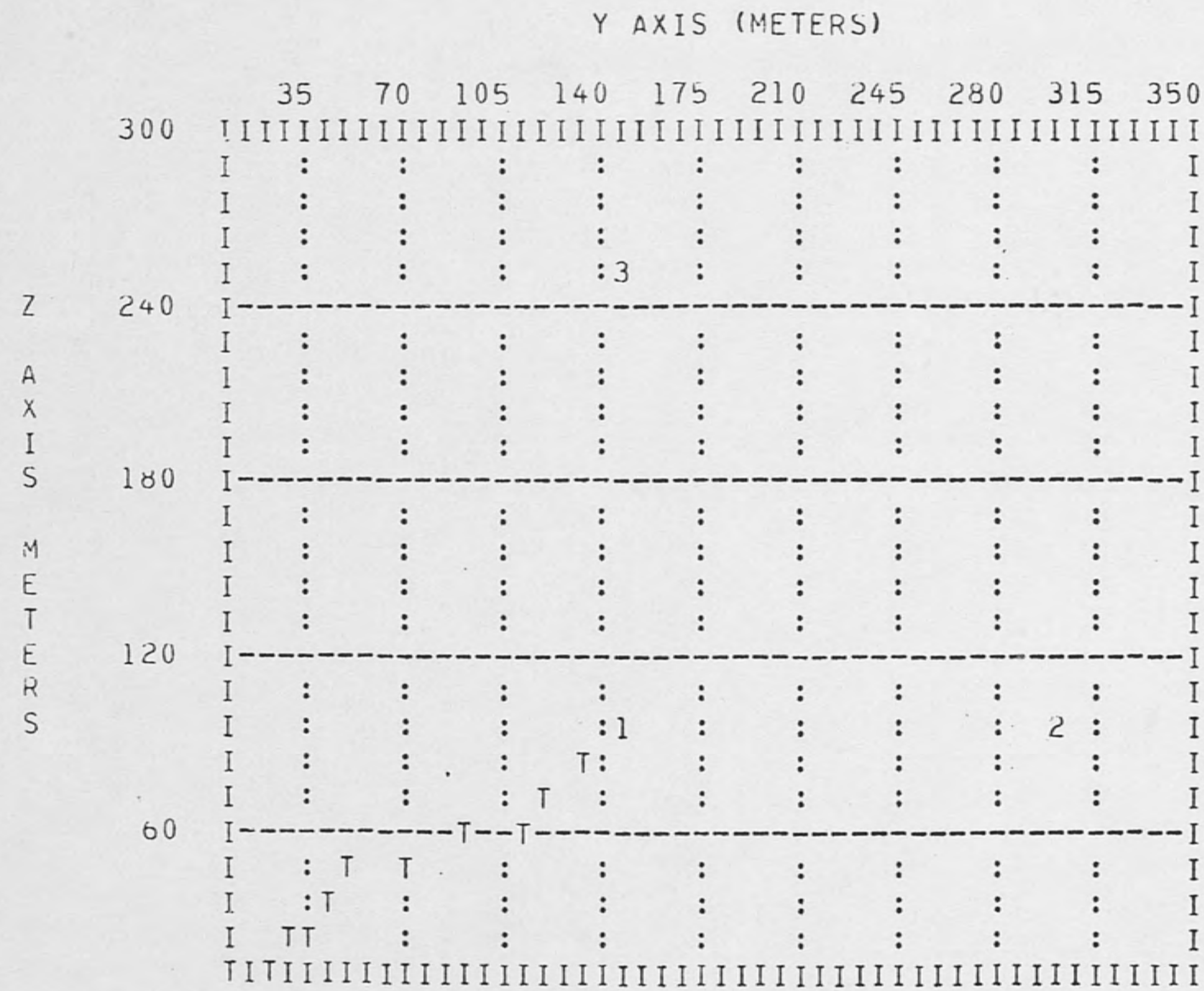
SF= 1.00 XMAX= 400.00 YMAX= 350.00 ZMAX= 300.00



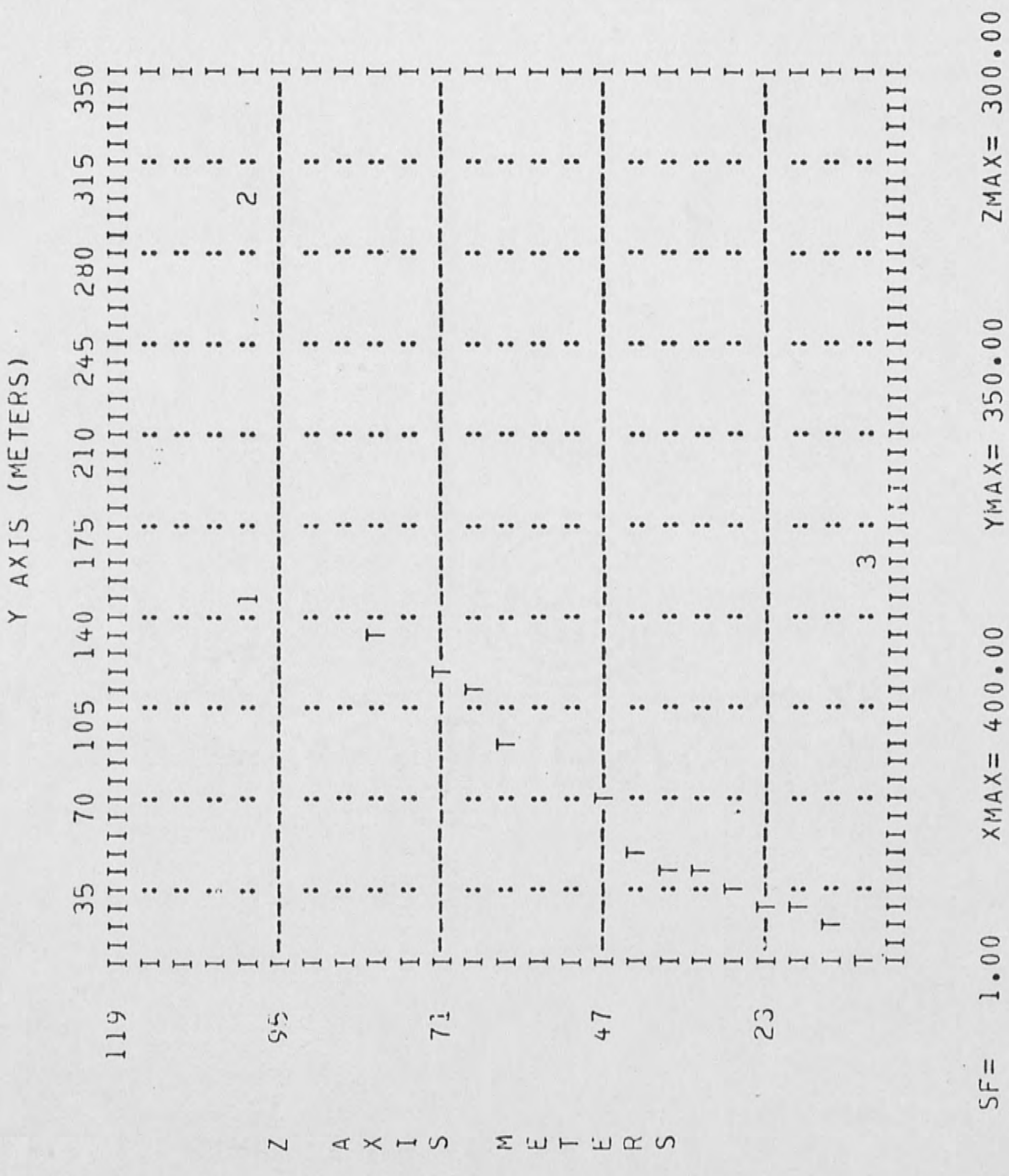
SF= 1.00 XMAX= 400.00 YMAX= 350.00 ZMAX= 300.00



SF= 1.00 XMAX= 400.00 YMAX= 350.00 ZMAX= 300.00



SF= 1.00 XMAX= 400.00 YMAX= 350.00 ZMAX= 300.00


Y AXIS (METERS)
35 70 105 140 175 210 245 280 315 350
Z AXIS METERS
119 95 71 47 23
SF= 1.00 XMAX= 400.00 YMAX= 350.00 ZMAX= 300.00

APPENDIX K

COMPUTER OBJECT-CODE RUN-TIME PROGRAMS, p. 181.

DESCRIPTION:

This appendix contains the various programs required to execute the object-code programs in the BACKGROUND PROCESSING MODE for the programs contained in the various appendices of this report.

```
$JOB SIM DLIST4
$ASSIGN 21=21,6=6,22=DATA
$SIMDAT
$EOJ
EOF..
  $SIMDAT
```

```
$JOB TRK DLIST
$ASSIGN 62=21,74=6,50=B3DATA
$BTRCK
$EOJ
EOF..
```

```
$JOB TRK DLIST
$ASSIGN 62=21,74=6,50=BDATA
$BTRCK
$EOJ
EOF..
```

```
$JOB TRK DLIST
$ASSIGN 62=21,74=6,50=DATA
$ATRCK
$EOJ
EOF..
```

SELECTED BIBLIOGRAPHY

Beer, Ferdinand P., and Johnston, E. Russell. Vector Mechanics for Engineers: Statics and Dynamics. New York: McGraw-Hill Book Co., 1972.

Carlson, A. Bruce. Communication Systems. New York: McGraw-Hill Book Co., 1975.

Halliday, David, and Resnick, Robert. Fundamentals of Physics. New York: John Wiley and Sons, Inc., 1970.

Harris Communications and Information Handling. Computer Systems Division. Fortran IV Manual. Ft. Lauderdale, Fla.: Harris Corp., 1974.

Harris Communications and Information Handling. Computer Systems Division. General Specification Alphanumeric Re-entrant On-line Interactive Manager (ACRONIM). Revision E. Ft. Lauderdale, Fla.: Harris Corp., 1974.

Horton, C. W. Signal Processing of Underwater Acoustic Waves. Washington, D.C.: U.S. Government Printing Office, 1969.

Hoyt, William H., Jr. Engineering Electromagnetics. New York: McGraw-Hill Book Co., 1974.

International Telephone and Telegraph Corp. Reference Data for Radio Engineers. Indianapolis: Howard W. Sams and Co., Inc., 1975.

James, M. L.; Smith, G. M.; and Wolford, J. C. Applied Numerical Methods for Digital Computation with Fortran. Scranton, Penn.: International Textbook Co., 1967.

Kaplan, Wilfred. Advanced Calculus. Reading, Mass.: Addison-Wesley Publishing Co., 1973.

McGillem, Clare D., and Cooper, George R. Continuous and Discrete Signal and System Analysis. New York: Holt, Rinehart, and Winston, Inc., 1974.

Nielsen, Kaj L. Methods in Numerical Analysis. 2nd ed. New York: The MacMillan Co., 1967.

Rule, W.; Finkenauer, R.; and Patrick, F. Fortran IV Programming. Boston, Mass.: Prindle, Weber, and Schmidt, Inc., 1973.

Schied, Francis. Numerical Analysis. Schaum's Outline Series in Mathematics. New York: McGraw-Hill Book Co., 1968.

Schwartz, Mischa, and Shaw, Leonard. Signal Processing: Discrete Spectral Analysis, Detection, and Estimation. New York: McGraw-Hill Book Co., 1975.

Tabak, Daniel, and Kuo, Benjamin C. Optimal Control by Mathematical Programming. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1971.

Tolstoy, Ivan, and Clay, C. S. Ocean Acoustics: Theory and Experiment in Underwater Sound. New York: McGraw-Hill Book Co., 1966.

U.S. Department of the Navy. Bureau of Ships. Introduction to Sonar Technology. NAVSHIPS Publication 0967-129-3010. Washington, D.C.: Government Printing Office, 1966.

U.S. Department of the Navy. Navy Hydrographic Office. Air Navigation. Hydrographic Office Publication No. 216. Washington, D.C.: Government Printing Office, 1967.

Urick, Robert J. Principles of Underwater Sound for Engineers. New York: McGraw-Hill Book Co., 1967.