NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE s.

1 .1

Shuttle Orbit IMU Alinement

Single-Precision Computational Error

(NASA-TM-80793) SHUTTLE CREIT INU ALINEMENT. SINGLE-PRECISICN COMPUTATION ERROR (NASA) 31 p HC A03/MF A01 N80-22379

Unclas G3/12 18480

Mission Planning and Analysis Division March 1980



Lyndon B. Johnson Space Center Houston, Texas



String and the state

JSC-16468

80-FM-18

SHUTTLE PROGRAM

SHUTTLE ORBIT IMU ALINEMENT

SINGLE-PRECISION COMPUTATIONAL ERRUR

By C. R. McClain, McDonnell Douglas Technical Services Co.

JSC Task Monitor: T. J. Blucker, Mathematical Physics Branch

Approved: Emil R. Schiesser, Chief

Mathematical Physics Branch

Approved: Ronald L. Berry, Chief Mission Planning and Analysis Vision

and the state

Mission Planning and Analysis Division National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, Texas March 1980

÷.

CONTENTS

e i

C

A DESCRIPTION OF A DESC

ļ

1.1.1

1.1

Section		Page
1.0	<u>SUMMARY</u>	, 1
2.0	INTRODUCTION	. 2
3.0	DISCUSSION	, 3
3.1	DETERMINATION OF THE MEASURED PRESENT-CLUSTER- TO-M50-MATRIX	. 3
3.2	COMPUTATION OF THE MISALINEMENT MATRIX AND ASSOCIATED ERROR QUATERNION	, 4
3.3	DESCRIPTION OF EIGEN RUTATION ANGLE EXTRACTION METHODS	5
3.3.1 3.3.2	Arc Cosine Method	5 6
3.4	DESCRIPTION OF TEST SETUP AND RUNS	6
4.0	<u>RESULTS</u>	. 8
5.0	CONCLUSIONS AND RECOMMENDATIONS	, 9
APPENDIX	A - HAL/S CODE IN SINGLE PRECISION	13
APPENDIX	B - HAL/S CODE IN DOUBLE PRECISION	, 20

THE WITCH MAY BLANK

iii

- MARINE -

TABLES

. . .

Table									Page
1.0	SUMMARY	0F	IMU	ONORBIT	ALINEMENT	HAL/S	SIMULATION		
	RESULTS	•	• •			• • •		• • • • •	. 10

1985

in the

and a summer of the second second

FIGURES

August Martin

-101-1

直日間に

Ċ

1

1.1

Figure											I	Page	
1.0	Misalinement	error	versus	input	misalinement	ang le	•	•	•	•	•	12	

۷

Section Section

БÀ.

195

and the second second

1.0 SUMMARY

This paper presents the results of a study to determine the source of computational error in the IMU onorbit alignment software as specified by the Level C IMU SOP FSSR. Simulation runs were made on the IBM 360/70 computer with the IMU onorbit alignment software coded in HAL/S. The results indicate that for small IMU misalignment angles (less than 600 arc seconds), single precision computations in combination with the arc cosine method of eigen rotation angle extraction introduces an additional misalignment error of up to 230 arc seconds per axis. Use of the arc sine method, however, produced negligible misalignment error. As a result of this study, the arc sine method has been recommended by means of a software change request (CR) for use in the IMU onorbit alignment software. This CR has been approved for STS-1 and beyond.

2.0 INTRODUCTION

During IBM's Level 2 IMU onorbit alignment testing, it was first suspected that the largest source of software produced alignment error for small torquing angle misalignments might be the flight control utility routine MAT_TO_QUAT. After review of these initial test results, it was decided by JSC, IBM and MDTSCO that detailed studies of this problem be made at MDTSCO to confirm the initial conclusion. The proposed studies were aimed at determining whether the additional computational error was in fact caused by MAT_TO_QUAT or by the IMU alignment software design. The questionable design centered around the use of single precision computations in combination with the arc cosine method for extracting the eigen rotation angle from the error quaternion, from which torquing angles were computed. The results from these studies indicated that the latter was the cause of the error. Based on this conclusion, change request IS12BA was submitted and approved to change the eigenaxis rotation angle extraction method from the arc cosine method to the arc sine method for STS missions number 1 and beyond.

Section 3.0 contains a discussion of the equations which define the IMU onorbit alignment software and a description of the simulation runs actually made. Section 4.0 contains a discussion of the results, and Section 5.0 presents the conclusions and recommendations.

3.0 DISCUSSION

The IBM 360/70 computer was used in simulating the HAL/S version of the IMU onorbit alignment torquing angle software for this study. This computer was used for the following reasons: (1) the IBM 360/70 computer implements a HAL/S compiler which enables the simulation software to be coded in HAL/S, (2) this computer emulates the word size of the AP101 flight computer. Although the word size is equivalent for both computers, AP101 double precision computations are not as accurate as the IBM 360/70 computer; therefore, the double precision results presented in this study are not representative of AP101 double precision accuracy.

The following sections describe the simulation software and test runs for determining the source of computational error in the onorbit alignment software design.

3.1 DETERMINATION OF THE MEASURED PRESENT-CLUSTER-TO-M50-MATRIX

In order to determine the measured transformation matrix from present cluster to M50 coordinates, the onorbit alignment software obtains measurements of two star LOS (Line of Sight) unit vectors in present cluster coordinates by means of the STAR TRACK or the COAS SIGHT mode, that is

$$\overline{S}_{m} = S_{xm}\overline{I}_{pc} + S_{ym}\overline{J}_{pc} + S_{zm}\overline{K}_{pc}$$
(1)

$$\overline{T}_{m} = T_{xm}\overline{I}_{pc} + T_{vm}\overline{J}_{pc} + T_{zm}\overline{K}_{pc}$$
(2)

In addition, the computer has stored in memory the exact LOS unit vectors in M50 coordinates of these two stars as follows:

$$\overline{S}_{e} = S_{xe}\overline{i}_{M50} + S_{ye}\overline{j}_{M50} + S_{ze}\overline{k}_{M50}$$
(3)

 $\overline{T}_{e} = T_{xe}\overline{i}_{M50} + T_{ye}\overline{j}_{M50} + T_{ze}\overline{k}_{M50}$ (4)

These two sets of unit vectors, measured and exact, are used to compute the desired transformation matrix.

First, the exact LOS unit vectors given in M50 coordinates are used to form three unit vectors defining a star coordinate system expressed in the M50 coordinate system as follows:

$$\overline{U}_{xe} = \overline{S}_{e} \tag{5}$$

$$\overline{U}_{VP} = UNIT \left(\overline{S}_{P} \times \overline{T}_{P}\right)$$
(6)

$$\overline{U}_{ze} = \text{UNIT} \left(\overline{U}_{xe} \times \overline{U}_{ve} \right) \tag{7}$$

Next, the three unit vectors are used to form the columns of the transformation matrix from star to M50 coordinates:

国際の法律

Ċ

3

12.00

$$\begin{bmatrix} T_{S}^{M50} \end{bmatrix} = \begin{bmatrix} U_{xe}(1) & U_{ye}(1) & U_{ze}(1) \\ U_{xe}(2) & U_{ye}(2) & U_{ze}(2) \\ U_{xe}(3) & U_{ye}(3) & U_{ze}(3) \end{bmatrix}$$
(8)

In a similar manner, the measured LOS unit vectors (given in present cluster coordinates) are used to form this same star coordinate system expressed in the present cluster coordinate system as follows:

$$\overline{U}_{xm} = \overline{S}_m \tag{9}$$

$$\overline{U}_{vm} = UNIT (\overline{S}_m \times \overline{T}_m)$$
(10)

$$\overline{U}_{zm} = UNIT (\overline{U}_{xm} \times \overline{U}_{ym})$$
(11)

These unit vectors are then used to form the rows of the measured^{*} transformation matrix from present cluster to star coordinates:

$$\begin{bmatrix} \nabla S \\ T_{PC} \end{bmatrix} = \begin{bmatrix} U_{xm}(1) & U_{xm}(2) & U_{xm}(3) \\ U_{ym}(1) & U_{ym}(2) & U_{ym}(3) \\ U_{zm}(1) & U_{zm}(2) & U_{zm}(3) \end{bmatrix}$$
(12)

Finally, the measured transformation matrix from present cluster to M50 coordinates is formed by the following matrix multiplication:

$$\begin{bmatrix} \sim M50 \\ T_{PC} \end{bmatrix} = \begin{bmatrix} M50 \\ T_{S} \end{bmatrix} \begin{bmatrix} \sim S \\ T_{PC} \end{bmatrix}$$
(13)

3.2 COMPUTATION OF THE MISALIGNMENT MATRIX AND ASSOCIATED ERROR QUATERNION

The misalignment matrix is computed as a function of the stored transformation matrix from M50 coordinates to desired cluster coordinates (desired REFSMMAT) and the measured transformation matrix from present cluster to M50 coordinates, that is

$$\begin{bmatrix} \nabla DC \\ T_{PC} \end{bmatrix} = \begin{bmatrix} DC \\ T_{M50} \end{bmatrix} \begin{bmatrix} \nabla M50 \\ T_{PC} \end{bmatrix}$$
(14)

* The tilde over the matrix indicates that the matrix is derived using onboard measured quantities and is to be distinguished from matrices without the tilde which are considered exact.

The error quaternion ΔQ is defined by the following convention

$$\Delta Q = \begin{bmatrix} \Delta QS \\ \overline{\Delta} QV \end{bmatrix} = \begin{bmatrix} \cos \omega/2 \\ - \hat{u} \sin \omega/2 \end{bmatrix}$$
(15)

where ΔQS is the scalar part of the quaternion, $\overline{\Delta} QV$ the vector part, ω

the eigen rotation angle about the eigenaxis, and u the eigenaxis unit vector. In order to compute the error quaternion associated with the misalignment matrix (14), the utility routine MAT TO QUAT is called, that is

CALL MAT_TO_QUAT
$$(\tilde{T}PC)$$

ASSIGN (ΔO)

This error quaternion is the basis for determining the torquing angles for platform realignment. The eigenaxis direction is obtained by extracting a unit vector* from the vector part of ΔQ . The eigen rotation angle can theoretically be obtained either from the scalar part of ΔQ or the magnitude of the vector part of ΔQ . The methods of extracting the eigenaxis rotation angle are described in more detail in the next section. At the time this study was performed, the eigen rotation angle was extracted by use of the scalar of the error quaternion.

3.3 DESCRIPTION OF EIGEN ROTATION ANGLE EXTRACTION METHODS

There exist two methods for extracting the eigenaxis rotation angle: (1) Arc cosine method (using the error quaternion scalar) and (2) Arc sine method (using the error quaternion vector). The equations for both methods are now presented.

3.3.1 Arc Cosine Method

The first method initially used by the IMU alignment software design is the arc cosine method. This method uses the scalar part of the error quaternion ΔQS from equation (15), that is

$$\cos \omega/2 = \Delta QS$$

(16)

* Although this paper does not discuss problems associated with this extraction process, it should be mentioned that for values of misalignment eigen rotation angles less than 10 arc seconds, the unit vector associated with the eigenaxis direction can be in error as much as 100 degrees. However, for such small eigenaxis rotation angles the direction of rotation is irrelevant. After taking arc cosine of both sides of this equation and multiplying by two, the eigenaxis rotation angle is given by

$$\omega = 2 \cos^{-1} (\Delta QS) \tag{17}$$

3.3.2 Arc Sine Method

The second method for extracting the eigenaxis rotation angle \simeq is the arc sine method. This method uses the magnitude of the vector part of the quaternion ΔQV from equation (15), that is

$$\sin \omega/2 = |\overline{\Delta}QV| \tag{18}$$

After taking arc sine of both sides of the equation and multiplying by two, the eigenaxis rotation angle is given by

$$\omega = 2 \sin^{-1} |\Delta QV| \tag{19}$$

3.4 DESCRIPTION OF TEST SETUP AND RUNS

For simulation purposes, the exact LOS unit vectors are arbitrarily defined as follows: \overline{S}_e is defined by an azimuth angle of 60 degrees and an elevation angle of 30 degrees, T_e is defined by an azimuth angle of -60 degrees and an elevation angle of 30 degrees. The measured LOS unit vectors are derived by rotating the exact LOS unit vectors (\overline{S}_e , \overline{T}_e) from the M50 frame to actual platform coordinates, that is

$$\overline{S}_{m} = \begin{bmatrix} T & DC \\ T & M50 \end{bmatrix} \begin{bmatrix} T & M5C (MEASURED) \\ T & M50 (ACTUAL) \end{bmatrix} \begin{bmatrix} \overline{S}_{e} \end{bmatrix}$$
(20)

$$\overline{T}_{m} = \begin{bmatrix} T & M50 (MEASURED) \\ T & M50 \end{bmatrix} \begin{bmatrix} T & M50 (MEASURED) \\ T & M50 (ACTUAL) \end{bmatrix} \begin{bmatrix} T & e \\ T & e \end{bmatrix}$$
(21)

The input misalignment eigenaxis direction is defined by an azimuth angle of 45 degrees, and an elevation angle of 30 degrees. The eigenaxis rotation or input misalignment for each specific test case is given in Table 1. For each case, an input misalignment quaternion is formed from these two input quantities and then a call to QUAT TO MAT results in the input misalignment matrix. Also, for each case, the measured and exact LOS unit vectors are unitized to IBM 360/70 single and double precision since they were generated using HP9825A precision. The stored REFSMMAT is defined by a quaternion which has an eigenaxis direction and eigenaxis rotation angle. The eigenaxis direction is arbitrarily defined by an azimuth angle of 45 degrees and an elevation angle of -30 degrees. The eigenaxis rotation angle between the M50 frame and desired cluster coordinates is taken to be 90 degrees. A call to QUAT_TO_MAI results in the stored transformation matrix from M50 to desired cluster coordinates. For each case, this REFSMMAT is orthogonalized in IBM 360/70 double precision before being used in the probit alignment flight software.

1

Each test run used the onorbit alignment software design as described in sections 3.1 and 3.2. For each input misalignment angle case, both single and double precision versions of the HAL/S code were exercised (Appendix A and Appendix B, respectively) and an error computed by taking the difference between the input misalignment angle and the output misalignment angle for both eigenaxis rotation angle extraction methods. This error became the standard of measure for determining the best method.

4.0 RESULTS

A summary of the IMU onorbit alignment HAL/S simulation results is presented in Table 1. The output misalignment angle ω_0 and the misalignment error $\delta\omega$ are given as functions of the input misalignment angle ω_{τ} for both precision versions (single, double) and both angle extraction methods (arc cosine, arc sine). The range of $\omega_{\rm I}$ is from 0 to 648000 arc seconds (180 degrees). As indicated in the last column of the table, both extraction methods result in zero misalignment software error over the entire range of input rotations when double precision is used. The single precision error column of $\delta \omega$ indicates that the arc cosine method has a sizeable error for small misalignment angles. In order to visually show the amount of misalignment error for both methods when coded in single precision, a graph of misalignment error $\delta \omega$ versus input misalignment ω_T for the arc cosine and arc sine rotation angle extraction methods is presented in Figure 1. For the arc cosine method, Figure 1 shows approximately 400 arc seconds of misalignment error for input misalignment angles near O arc seconds. This 400 arc seconds of misalignment error would result in approximately 230 arc seconds per axis of additional misalignment error to the current IMU alignment error budget. For increasing input misalignment angles, a gradual decrease in misalignment error can be seen. For the arc sine method, Figure 1 shows that for input misalignment angles near C arc seconds, the misalignment error is negligible (i.e., .03 arc seconds). For large input misalignment angles (near 648000 arc seconds), the maximum misalignment error is 340 arc seconds. This amount of misalignment error would result in approximately 196 arc seconds (per axis) of additional misalignment error to the IMU alignment error budget.

5.0 CONCLUSIONS AND RECOMMENDATIONS

For small IMU misalignment angles (less than 600 arc seconds), single precision computations in combination with the arc cosine method introduces additional misalignment error of up to 230 arc seconds per axis. The arc sine method, on the other hand, produces results equivalent to double precision accuracy. The misalignment error $\delta \omega$, obtained during this study, is clearly a function of the eigenaxis rotation angle extraction method and the precision used in the HAL/S code. The presence of MAT_TO_QUAT did not significantly influence the amount of misalignment error.

It is therefore recommended that the arc sine method of eigen rotation angle extraction be implemented into the onorbit alignment software at the earliest possible date. This software change would eliminate a known computationally-induced bias error.

*1.111

INPUT MISALIGN MENT ANGLE. ω.	- OUT	PUT MISALIC (arc se	inment ANGLE, ພ _ບ conds)	MISALIGNMENT (arc se	ERROR, δω conds)
(arc seconds) ¹	Single	Precision	Double Precision	Single Precision	Double Precision
0	cos-1.	402.86	0.0	$\cos^{-1} = 402.86$	0.0
	sin ⁻¹ =	0.03	0.0	$\sin^{-1} = 0.03$	0.0
5	cos-1.	4112.86	5.0	cos-1 = 397,86	0.0
	sin ⁻¹ =	5.00	5.0	$sin^{-1} = 0.00$	0.0
15	cos-1=	402.86	15.0	$\cos^{-1} = 387.80$	0.0
	sin-1-	15.01	15.0	$\sin^{-1} = 0.01$	0.0
25	cos-1-	402.86	25.0	cos-1 = 377.86	0.0
	sin ⁻¹ =	25.02	25.0	$\sin^{-1} = 0.02$	0.0
50	cos-1=	402.86	50.0	cos-1= 352.86	0.0
	_s1n ⁻¹ -	50.02	50.0	sin ⁻¹ = 0.02	0.0
75	cos-1-	407.86	75.0	cos-1 = 327.86	0.0
	_sin =	74.99	75.0	$\sin^{-1} = 0.01$	0.0
100	cos-l=	402.86	100.0	$\cos^{-1} = 302.86$	0.0
	sin ⁻¹ =	99.99	100.0	sin ⁻¹ = 0.01	0.0
125	cos-1	402.86	125.0	cos-1= 277.86	0.0
	sin ⁻¹ ≖	125.01	125.0	$\sin^{-1} = 0.01$	0.0
150	cos-1-	402.86	150.0	cos-1= 252.86	0.0
	sin-1=	150.02	150.0	sin ⁻¹ = 0.02	0.0
200	cos-1=	402.86	200.0	cos=1= 202.86	0.0
	sin ⁻¹ *	200.00	200.0	sin ⁻¹ = 0.00	0.0
300	cos-l=	569.73	300.0	cos-1 = 269.73	0.0
	sin-1=	300.02	300.0	$\sin^{-1} = 0.02$	0.0
600	cos-I.	697.17	600.0	cus-1= 97.77	0.0
	sin ¹ -	600.01	600.0	$\sin^{-1} = 0.01$	0.0
900	cos-1-	986.80	900.0	cos-1 = 86.80	0.0
	sin_!-	900.00	900.0	$\sin^{-1} = 0.00$	0.0
1800	cos-l=	1846.14	1800.0	cos-1= 46.14	0.0
	sin-l=	1800.01	1800.0	$\sin^{-1} = 0.01$	0.0
2700	cos-T.	2732.33	2700.0	$\cos^{-1} = 32.33$	0.0
	sin-!*	2700.01	2700.0	$\sin^{-1} = 0.01$	0.0
3600	COS-1=	3625.75	3600.0	cos-1= 25.75	0.0
	sin ⁻¹ =	3600.01	3600.0	$\sin^{-1} = 0.01$	0.0
640800	cos-I=	640799.75	640800.0	cos-1= 0.25	0.0
	sin-l=	640790.43	640800.0	sin ⁻¹ 9.57	0.0
644400	cos-T=	644399.56	644400.0	$\cos^{-1} = 0.44$	0.0
	sin ⁻¹ =	644376.81	644400.0	sin ⁻¹ ~ 23.19	0.0
645300	cos-l=	645299.75	645300.0	cos ⁻¹ = 0.25	0.0
	sin ⁻¹ =	645267.50	645300.0	sin ⁻¹ 32.50	0.0
646200	cos-1=	646199.87	646200.0	cos-1= 0.13	0.0
	sin"!=	646153.50	646200.0	sin ⁻¹ : 46.50	0.0
647100	cos-1 =	647099.62	647100.0	$\cos^{-1} = 0.38$	0.0
	sin-1	647023.31	64/100.0	$rin^{-1} = 76.69$	0.0
647400	cos ⁻¹ =	647399.75	647400.0	cos ⁻¹ - 0.25	0.0
	sin ⁻¹ =	647246.06	647400.0	$\sin^{-1} = 153.94$	0.0
647700	cos-I=	647699.68	647700.0	$\cos^{-1} = 0.32$	0.0
	sin ⁻¹ =	647527.31	647700.0	sin ⁻¹ = 172.69	0.0

TABLE 1.0 - SUMMARY OF IMU ONORBIT ALIGNMENT HAL/S SIMULATION RESULTS

energy and that the state is not the set of the state of the

INPUT MISALIGN MENT ANGLE, ω_I	- OUTP	PUT MISALIG	NMENT ANGLE, ω _o conds)	MISALIGNM (ar	MENT ERROR, δω rc seconds)
(arc seconds)	Single	Frec 13100	Double riecision	Single Freuis	ston bouble receision
647800	cos-1=	647799.93	647800.0	$\cos^{-1} = 0.07$	0.0
	sin ⁻¹ =	647549.37	647800.0	$sin^{-1} = 250.63$	0.0
647850	COS-I=	647849.81	647850.0	$\cos^{-1} = 0.19$	0.0
	sin ⁻¹ =	647549.37	647850.0	$sin^{-1} = 300.63$	3 0.0
647870	COS-1=	647869.56	647870.0	$\cos^{-1} = 0.44$	0.0
	sin ⁻¹ =	647549.37	647870.0	$sin^{-1} = 320.63$	0.0
647890	cos-1=	647889.68	647890.0	$\cos^{-1} = 0.32$	2 0.0
	sin ⁻¹ =	647549.37	647890.0	$sin^{-1} = 340.63$	3 0.0
647915	cos-1=	647914.75	647915.0	$\cos^{-1} = 0.25$	0.0
	sin ⁻¹ =	647715.00	647915.0	$sin^{-1} = 200.00$	0.0
647965	cos-1=	647964.62	647965.0	$\cos^{-1} = 0.38$	0.0
	sin ^{-l} =	647715.00	647965.0	$sin^{-1} = 250.00$	0.0
648000	cos-1=	647999.81	648000.0	$\cos^{-1} = 0.19$	0.0
	sin-1=	647753.18	648000.0	$\sin^{-1} = 246.82$	0.0

TABLE 1.0 - SUMMARY OF IMU ONORBIT ALIGNMENT HAL/S SIMULATION RESULTS - Concluded

All and a

and the second

1.10



100 m

A 1212

FIGURE 1.0 MISALIGNMENT ERROR VERSUS INPUT MISALIGNMENT ANGLE

}

12

×4

APPENDIX A

(HAL/S CODE IN SINGLE PRECISION)

100

at . desident

4	SUDACE	CURRENT SCOPI
	11-2-0MT:	- TANT_2_OUAT
a. T	ROGRA4:	I MAT_2_QUAT
: F	DECLARE TABLE VECTOR INITIALS .971526492,233043752,0427420731;	-1 MAT_2_OUAT
7	DECLARE TABLE1 VECTOR INTIAL(+.340969046,792263238,535322797);	1 MAT_2_0UAT
. I H	DECLARE STAR VECTCR INITIAL (.433012702; .7500000000; .500000000);	TAUD_STAN
Ŧ.	DECLARE STARL VECTOR INITIAL (.433012702,750000000, .50000000);	I MAT_2_DJAT
I IN	" DECLARE IM50DC %ATRIX INTIAL(.37500000, .87500000, ".306186218,125000000, ".375300900, "	-1 4AT_2_QUAT
Ŧ	•913558654,918558654, .306186218, .2500000001;	I MAT_2_QUAT
H	"DECLARE'X MATRIX; "TO THE PARTY OF THE	T NAT_2_OUAT
Ŧ	DECLARE QBAS SCALAR:	I H11_2_0J4F
I IN	DECLARE 3BAV VECTOR;	I HAT 2 QUAT
Ŧ	DECLARE QO SCALAR;	I MAT_2_QUAT
H	DECLARE Q VECTCF;	- [MT_2_QUAT
Ŧ	DECLARE T SCALAR:	1 MAT_2_0UAT
I H	DECLARE ANG SCALAR3	
-	DECLARE ANGV VECTOR:	I MAT_2_QUAT
1	DECLARE RD SCALAR DOU BLE TNIT TAL (57.2957795130823208767981551;	MAT_2_OUAT
Ī	DECLARE I INTEGER:	1 MAT_2_QJAT
L.	DECLARE WINTEGER ;	TANT_2_2UAT
ī	DECLARE J INTEGER;	1 441_2_QUAS
-	DECLARE K INTEGER;	1 MATOUAT
Ŧ	DECLARE H INTEGER;	TALC_2_TAN 1
Ŧ	DECLARE ANGI SCALAR;	"I MAT_2_QUAT !"
Ŧ	DECLARE TSM50 MATRIX;	LI MAT_2_QUAT
I F	DECLARE TPCS MATRIX ;	-1 HAT_2_QUAT
Ŧ	DECLARE TPCM50 MATRIX;	1 MAT_2_3JAF
- X		

の語を

a straight and the state

Υ,

× ľ

1 m CURRENT SCOPE P 46E AAT_2_QUAT HAT_2_QUAT NAT_2_OUAT MAT_2_QUAT 41_2_QUAT W1_2_2UAF MT_2_0UAT NAT_2_QUAT MAT_2_QUAT 41_2_0UAT AA 1_2_0UAT MAT_2_0UAT MT_2_QUAT NAT_2_0J NT HAT_2_OUAT 141_2_0UAT 44 F_2_QUAT TA UD_2_TAH AAT_Z_QJAT MT_2_OUAT 14:22:17.20 İ NUVENBER 11, 1978 : WRITE(6) LINE(1), CULUMN(1), "MAT_2_QUAT BENCH PROGRAM RESULTS", SKIP(4); i ì i i SOURCE CP 1 20 600.APPL . SRC FBDR I VER) RVL = AA I DECLARE C VECTOR DOURLE: DECLARE & VECTOR DOUBLE: DECLARE & VECTOF DCUBLE; DECLARE U MATRIX DCLBLE; TABLEL = UNITI TABLE 1) ; STARL = UNIT(STARL); TABLE = INITI TABLE); DECLARE X1 VECTOR; STAR = INI TISTAR DECLARE Y VESTOR: DECLARE V VECTOR: DUCLARE M VECTOP; DECLARE 7 VECTCR: = UNIT (A + B); A = UNIT(A); = 14500C ; ••3 N 10 1 CC - # . < 80 362-22.13 ; 30 M 26 M 27 M 28 M 14 16 **JM SE IK 6**6 **14 SE** 39 4E E 40 M 34 M 36 A 3.7 M 38 M WE \$ M 62. UTN WE IN JY.S U.R. 14 43 ÷5 5141 . 42

> ORIGINAL PAGE IS OF POOR QUALITY

15

F

0 • • • • • • • • • • • • • • • • • • •	SOURCE		CURRENT SCO
01 1 1 1 1.2.0urt 02 3 7 1.2.0urt 03 3 7 1.2.0urt 13 3 7 1.2.0urt 13 3 7 1.2.0urt 13 3 7 1.2.0urt 13 1.1 1.2.0urt 1.2.0urt 13 1.1 1.2.0urt 1.2.0urt 13 1.1 1.2.0urt 1.2.0urt 11 1.1 1.1 1.2.0urt 11 1.1 1.1 1.2.0urt 1.2.0urt 11 <td< th=""><th>B = C + A;</th><th></th><th>HAT 2 GUAT</th></td<>	B = C + A;		HAT 2 GUAT
D _{1,2} = B ₁ w ₁ 2_0urr D _{1,3} = <i>C</i> ₁ w ₁ 2_0urr W = STAR: W ₁ 2_0urr W = UNITISTAR & STARIJ; W ₁ 2_0urr W = STAR W ₁ 2_0urr W = UNITISTAR & TREELI; W ₁ 2_0urr W = STAR W ₁ 2_0urr W = STAR W ₁ 2_0urr W = STAR <		1	WT 2 QUAT
•.2 •.2 •.1 •		•	
$ \begin{split} \tilde{D}_{\bullet,3} = \tilde{C} : & MT_2 = 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2$	(*,2		TAULZ_DUAT
14500 = 0; w1_20urt		1	
WIT_2_0UNT V = UNIT(IJ + VI); V = UNIT(II + VI); V = UNIT(IATE = TABLEII);			
U = STAR; V = UNIT(STAR + STAR1); 447.2.0447 V = UNIT(STAR + STAR1); 447.2.0447 N = UNIT(ToTE + TABLE1); 447.2.0447 X1 = TAR F; 447.2.0447 Y = UNIT(TATE + TABLE1); 447.2.0447 Y = UNIT(TATE + TABLE1); 447.2.0447 Y = UNIT(TATE + TABLE1); 447.2.0447 Y = UNIT(X1 + V1); 447.2.0447 Y = Y; 1560, 2 = V; Y = Xi; 1565, 4 = X; Y = Y; 175.2, 4 = Y;			TAT_2 QUAT
V = UNIT(STAR + STAR1); +4r_2_0UAT H = UNIT(U + V); +4r_2_0UAT X1 = TAR F; +4r_2_0UAT Y = UNIT(TAR F + TABLET); +4r_2_0UAT S = UNIT(TAR - V); +4r_2_0UAT 15%0, +, 1 -U *15%0, +, 1 -U 15%0, +, 2 V; 15%0, -, 3 V; 15%0, -, 3 V; 15%0, -, 3 V; 15%0, -, 4 V; 15%0, -, 5 V; 15%0, -, 5 V; 17_2_0UAT V; 17_2_0UAT V; 15%0, -, 5 V; 15%0, -, 5 <td>U = STAR;</td> <td></td> <td>AAT_2_QUAT</td>	U = STAR;		AAT_2_QUAT
M = UNIT(<u>U</u> + V); X 1 = TAR F; M 1 2_0UNT Y = UNIT(TARE + TABLELL; M 1 2_0UNT Z = UNIT(TARE + TABLELL; M 1 2_0UNT M 1 2_0UNT M1 2_0UNT M1 2_0UNT M1 2_0UNT M 1 2_0UNT 	V = UNIT(STAR * STARI);		111 2 2 111
x1 = Tañi F: Mar_2_0041 $\vec{Y} = UNITITARE + TABLEIJ;$ Mar_2_0041 $\vec{Z} = UNITIXI + \vec{Y}J;$ Mar_2_0041 $\vec{Z} = UNITIXI + \vec{Y}J;$ Mar_2_0041 $\vec{X} = \vec{U};$ Mar_2_0041 $\vec{X} = \vec{V};$ Mar_2_0041 $\vec{Y} = \vec{X};$ Mar_2_0041 $\vec{Y} = \vec{X};$ Mar_2_0041	H = UNIT(Ū + V);		
	XI = TARIE;		1 4412_QUAF
Ž = UNIT(XI + Y); MAT_2_0UAT TSV50 +, I = Ū; MAT_2_0UAT TSV50 +, I = Ū; MAT_2_0UAT TSV50 +, 2 = V; MAT_2_0UAT TSV50 +, 3 = V; MAT_2_0UAT TPC S_+ = Xi: MAT_2_0UAT TPC S_+ = Xi: MAT_2_0UAT	$\vec{Y} = UNIT(TARTE + TARTER)$		NAT_2_QUAT
$15\overline{450}_{*,1} = \overline{0}i;$ $15\overline{450}_{*,2} = \overline{V}i;$ $15\overline{450}_{*,2} = \overline{V}i;$ $15\overline{450}_{*,3} = \overline{V}i;$ $15\overline{450}_{*,3} = \overline{V}i;$ $15\overline{450}_{*,3} = \overline{V}i;$ $17\overline{550}_{*,3} = \overline{V}i;$ $17\overline{550}_{*,3} = \overline{V}i;$ $17\overline{550}_{*,3} = \overline{V}i;$ $17\overline{550}_{*,3} = \overline{V}i;$		•	HAT_2_QUAT
$15^{450} \bullet_{*1} = 0;$ $15^{450} \bullet_{*2} = \tilde{V};$ $15^{450} \bullet_{*2} = \tilde{V};$ $15^{450} \bullet_{*3} = \tilde{V};$ $15^{450} \bullet_{*3} = \tilde{V};$ $15^{450} \bullet_{*3} = \tilde{V};$ $17^{2}C_{*1} = \tilde{V};$ $17^{2}C_{*2} = \tilde{V};$			HAT_2_QUAT
15450 = V; 15850 * 2 15850 * 3 15850 * 3 15850 * 3 15850 * 3 15850 * 3 15850 * 3 15850 * 3 17653 * 7	13450 = U; * 1		HAT_2_QUAT
15M50, = H; MAT_2_OUAT TPCS, = Xi; MAT_2_OUAT	$15\overline{4}$ = 1	 -	MAT 2 DELAT
1000, = 46 1PCS, = X1; 1PCS, = Y1; 1PCS, = Y1; 447,2,0047 447,2,0047 444,2,0047	True > + +		
IPCS = xi; HAT_2_OUAT IPCS * = vi;	. ;H = C4+	e'	HAT_2_QUAT
	$\tilde{TPCS}_{l,*} = \tilde{X}_{l,*}$	·	HAT_2_0UAT
	$\overline{12}CS = \overline{7}$		NAT 2 QUAT

ORIGINAL PAGE IS OF POOR QUALITY ()

(

(

1

for the black and the state

1 -----1 1 ! 1 i ! 5 CURRENT SCOPE PAGE 4A 1_2_QUA 1 MAT_2_QUAT 441_2_JUAT 7 AUC_2_1 AM 441_2_0UAT 4AT_2_3UAT 14 LO_2_ TAP MAT_2_QUAT HAT_2_OUAT MT_2_0UAT A 1_2_QUAT 447_2_0UAT MAT_2_QUAT TA 1_2_0UAT 10.00_5_167 MAT_2_QUAT HAT_2_OUAT 41_2_QUAT 44 T_2_0UAT MI_2_0UAT 1112_3111 44 T_2_QUAT 441_2_QUAT 14:22:17.20 NJ VEMBER 11, 1978 1.1 i ; i ł ł i i i SOURCE 0P1 20600 . APPL . 5 PC (RDR IV ER)RVL = AA 1 i I ļ ۱ ۰۰ ł T = SQRI(1 + 2 40 - 1); IFH = 2 THEN TPC450 = LSM50 1PCS; IF X > QO THEN * * * X = 14500C TPCM50; K = 3; 00 FOR K = 1 TC 3; DO FOR H = 1 TO 3; :-* × 00. T = TRACE(X); . "IF H = 1 THEN 1 = X; X = 30 ¥ ELSE TPC S_{3, *} = *I*: \$2,= ¥ I. N = 0; Et.O: EN ::0 :00 ELSE END; 360-22.13 2 15 12 ----2 14 02 1 In 24 --l |n 69 82 MI 1 81 MI 2 68 M 76 41 63 H 65 M] 67 H IN 11 62 ¥ 64 E 66 ¥ 14 41 75 H 78 MI 14 62 80 M 12 62 78 M 81 M 5747

(

And the state of t

ļ

ł

;

J = 6 - N - K;		MT_2_0UA
	-	MAT_2_0UA
IF I ($V = 1$) = 0 THEN	-	MAT_2_0U1
00° € = (X - X) / 1; N - J•K Ke,J		MAT_2_0UA
ELSE ELSE		MAT 2 D. A
g = (x + x) / 1;		
EVD;	-	HAT_2_QUA
IF I = 0 THEN		MAT_2_QUA
	-	HAT_2_QUA
ELSE		441 2 9J4
		4AT_2_0UA
	-	•
T = SIGN(0.01 / 2;	-	4AT_2_0UA
C34.5 = 1 C0;	-	HAT_2_OUA
081V = 7 0:	-	
AVG = 2 ARCCCS (GA35);	•	MAT 2 DIA
ANG = 36JO. ANG KD:		
ANGI = 2 AKES INGARANI ANG	• •	
		100-2- IFF
Arigv = 1'IG UNIT (PBAV);		MAT_2_0J A
MRITEI6) SKIPI4), COLUMNII), "INPUT MATRIX ", Å, SKIP(2), COLUMRI5)	TS450 44 TKI X., TS450, SKIP	MAT_2_CUA
[2], CGLJ4R4(5], TPCS MATRIX ., T ^{\$} , SKIP12], COLJMN(5], TPCM53 4	11X', TPCH50, SKIP(2), .	41_2_0UA
CJLUMN(5), "TM56DC MATPIX", T#50DC, SKIP(2), COLUMN(5), "QBAS ", QB	SKIPICI COLUANISI. "DAY	AAT 2 DUA

Ô

Ū

 \bigcirc

()

()

1. 35% | - - -• • • • 1 ! ļ 1 CURRENT SCOPE ł ì (5). [•]U VECTORº, Ū. SKIPIZ), CCLUMNIS), •V VECTOR•, Ū. SKIPIZ), COLUMNISI, •W VECTOR•, Ā. SKIPIZ | MAT_2_QUAT 99 4 LE-ARCSIN LARC SEC) + 3600. 2061, SKIPL21, CJLUNNESI, "CGMV_AN" LARC SECI +, ANGV, SKIPL21, CGLUMV | MAL_2_0JAF HAT_2_QJAT I HAT_2_JUNT HAT_2_OUAT • i ; 1. COLUMNISI, "XI VECTOR", XI, SKIPI21, COLUMNISI, "Y VECTOR", Y. SKIPI21, COLUMNISI, "Z VECTUR" 14:22:17.20 NOV EMBER 11, 1976 ÷ i 1 SOURCE ł ł 0F 120600.APPL. SRC (BOR I VER)RVL=AA ł 1 ł 1 i : . 4 100 M1 CLOSE 447_2_QUAT; • 2 • 363-22.13 ł 39 ¥ 99 M 99 F ÷ **THT**

APPENDIX B (HAL/S CODE IN DOUBLE PRECISION)

Ó

Ü

()

12 Can

RIGINAL PAGE IS

ţ

N CURRENT SCOPE PAGE MAI_2_OUAT MAT_2_OUAT MAT_2_DUAT MAT_2_OUAT **HAI_2_0UAT** M1_2_0LAT NAT_2_OUAT NAT_2_OLAT MAI_2_OUAT MAT_2_0UAT MA 1_2_0UAT HAL_2_OUAT NAT_2_GUAT MAT_20UAT TAUC_2_TAN **MAL_20UAT** NA 1_2_0UA 1 MAT_2_OUAT MAT_2_00AT NAT_2.0UAT **MAT_2_OLAT** TA1_2_OUAT MAT_2_0UAT MAT_2_OUAT NAT_2_0UAT HAT_2_OUAT NAT_2OUAT 1 13:39:6.31 DECLARE TM50NC MATRIX DOUGLE INITIALL.37500000. 875400000. 306166218. -.125000000. 37500000 ! NCVEMBER 15, 1970 ----t __0ECLARE 143LE1 VECTTR DOURLE INITIAL1-+340777128+ --194655915+ --50238722#1; ; DECLARE TARLE VECTOR DOURLE INITIAL(.971722872. -.232155915. -.043107901); DECLARE STAR VECTER DEUPLE INITIAL (.+33012702, .750000000, .50000001; DECLARE START VECTOR DOUBLE INITIAL (.433012762. -. 750100000. .5000000); i ! ; į DECLARE RD SCALAR DRUBLE INITIAL(57.2957795130823208767981551; • -- 918558654• -- 51 8558654• .306186218• .250003001; SCURCE • 0R1 20700. APPL . SRC 180R IVER JRVL = AA DECLARE TPC450 MATRIX DOUBLE: DECLARE TSM50 MATRIX CCUBLE: DECLARE TPCS MATRIX DOUBLE: DECLARE ORAS SCALAR DRUGLE: DECLARE OBAV VECTOR DOURLF; DECLARE X1 VECTOR DOUBLE: DFCLARE ON "CALAR DCUBLE; DECLARE U VECTOR DRUBLE: DECLARE V VECTOR DOUBLE: DECLARE W VECTOR DOUBLE: DECLARE Y VECTOR DOUBLE: DECLARE 7 VECTOR DOUBLE: DECLARE X "NATRIX DOUBLF: DECLARE 0 ECTOR DOUBLET INTEGER: DECLARE K INTEGER: DECLARE J INTEGER: DECLARE N INTEGER: I WI MAT_2_DUAT: DECLAPF I AI PROGRAM: 360-22.13 1. デ ベ・ 2 3 A A 17 6 12.4 7 IN CI 13 41 1 ī IN OI 1 H 11 W SI 15 41 Ŧ 7 24 H Ŧ 7 11. 81 20 41 7) 2 1 2 5 11 61 3 STHT i ł

Glante I scara Coules Scarte Mar.20041 H DECLARE MUSICER Mar.20041 H DECLARE AVENTERION DUBLER Mar.20041 H DECLARE C VECTOR DUBLER Mar.20041 H DECLARE C VECTOR DUBLER Mar.2.0041 H TABLE VECTOR DUBLER Mar.2.0041 H TABLE VECTOR DUBLER Mar.2.00	1. 22 - 0		IN LEGUK EVER JK VL ZAA	NEVERBER 15, 1978	13:39:6.31	PAGE
H DECLARE TSCLAR DUBLE: MAT_2001 H DECLARE MG SALLA DUBLE: MAT_2001 H DECLARE A VECTOR DUBLE: MAT_2001 H STAL = UNITIGTAL: MAT_2.001 H Talle = UNITIGTAL: MAT_2.001 H Talle = UNITIGTAL: MAT_2.001 H Talle = UNITIGTALE: MAT_2.001 H Talle: UNITIAL: H Talle: UNITIAL			SURCE			CURRENT SCOPE
MILL OLUATION MILL OLUATION I DECLARE MG SACLAR COURLE: MILL OLUATION I DECLARE AVECTOR DOUALE: MILL OLUATION I STAR = UNITISTAN: MILL OLUATION I STAR = UNITISTAN: MILL OLUATION I STAR = UNITISTAN: MILL OLUATION I TALE = UNITISTANE: MILL OLUATION I TALE = UNITISTANE: MILL OLUATION I TALE = UNITISTANE: MILL OLUATION I TALEONC: MILL OLUATION	Ŧ	DECLARE T SCALAR DOUBLE:				I MAT_2_DUAT
H DECLARE ANG SCALAR CURLE: MAL_2 OUNT 1 DECLARE ANG SCALAR CURLE: MAL_2 OUNT 1 DECLARE ANG SCALAR DUBLE: MAL_2 OUNT 1 DECLARE ANG SCALAR DUBLE: MAL_2 OUNT 1 DECLARE ANG SCALAR DUBLE: MAL_2 OUNT 1 DECLARE A VECTOR DUBLE: MAL_2 OUNT 1 DECLARE C VECTOR DUBLE: MAL_2 OUNT 1 STALE = UNITITATALEI: MAL_2 OUNT 1 Table: MAL_2 OUNT 1 A = THSODC_4, 1 MAL_2 OUNT	N.	DECLARF H INTEGER:				MAT_2_OLAT
91 DECLARE WCI SCA13R DOURE: MML_20041 11 DECLARE AVCV VETOR DOUBLE: MML_20041 12 DECLARE AVCV VETOR DOUBLE: MML_20041 13 DECLARE AVCV VETOR DOUBLE: MML_20041 14 DECLARE AVCV VETOR DOUBLE: MML_20041 15 DECLARE AVCV VETOR DOUBLE: MML_20041 16 DECLARE C VETOR DOUBLE: MML_20041 16 STAL = UNITITATION OF COUNTING MML_20041 16 STAL = UNITITATION MML_20041 17 STAL = UNITITATION MML_20041 17 Taile = UNITITATION ML_20041 18 Taile = UNITITATION ML_20041 19 Taile = UNITITATION ML_20041 11 Taile = UNITI	Ŧ	DECLARE ANG SCALAR CCUBLE:				I MALOUAT
NI DECLARE ANCY VECTOR DOUBLE: NIL_20UT NI DECLARE A VECTOR DOUBLE: NIL_20UT NI DECLARE C VECTOR DOUBLE: NIL_20UT NI STAR = UNITISTATI: NIL_20UT NI STAR = UNITISTATI: NIL_20UT NI STAR = UNITISTATI: NIL_20UT NI TAILE = UNITITATELI: NIL_20UT NI TAILE: NIL_20UT Si TAILE: NIL_20UT Si TA	Ţ,	DECLARE ANGL SCALAR DOUBLES				MAT_2_OUAT
%1 DECLARE & VECTOR DUBLE: MEL_20UT %1 STAL = UNITISTRI: MEL_20UT %1 STAL = UNITISTRI: MEL_20UT %1 STAL = UNITISTRI: MEL_20UT %1 TABLEI = UNITITATEL: MEL_20UT %2 TABLEI = UNITITATEL: MEL_20UT %1 TABLE = UNITITATEL: MEL_20UT %2 TABLE = UNITITATEL: MEL_20UT %3 TaBLE = UNITITATEL: MEL_20UT %4 TABLE MEL20UT	Ŧ	DECLARE ANGV VECTOR DOUBLE:				I MAT_2_OUAT
NI DECLARE & VECTOR DUBLE: MAT_2_OUNT NI DECLARE & VECTOR DUBLE: MAT_2_OUNT NI FILLARE UNITTISTAN: MAT_2_OUNT NI STAR = UNITTISTAN: MAT_2_OUNT NI TABLE! UNITTIABLE!: NI TABLE! UNITTIABLE!: NI TABLE! MAT_2_OUNT SI T TWECDOC	ļ	DECLARE A VECTOR DOUBLE:	and an addition statements and a second statement of a second statement of second statements and second state			1 N. 1_2_0UAT
MI DECLARE C VECTOR DUUBLE: MIL2_OUNT MIL2_OUNT MITF(s) LIVE(1). COLUMN(1). "MAL2_OUNT DENCH PROGRAM RESULTS' SKIP(A): MIL2_OUNT MI STAR = UNIT(STAR): MIL2_OUNT MI STAR = UNIT(STAR): MIL2_OUNT MI TABLE = UNIT(STAR): MIL2_OUNT MI TABLE = UNIT(TABLE): MIL2_OUNT MIL2_DODC_*2: MIL2_UNT MIL2_OUNT MIL2_UNT TABLE = UNIT(A = B): MIL2_UNT	1	DECLARE & VECTOR DOUBLE:			•	I MAT_2_OUAT
YI MRIFF(6) LINE(1). COLUMN(1). **AT_2_OUAT AERCHAR RESULTS*, SKIP(4): MAT_2_OUAT FI STAR = UNIT(STA1): MAT_2_OUAT FI STAR = UNIT(STA1): MAT_2_OUAT FI TABLE = UNIT(STA1): MAT_2_OUAT FI TABLE = UNIT(STA1): MAT_2_OUAT FI TABLE = UNIT(TABLE1): MAT_2_OUAT FI TENDIC_1 MAT_2_OUAT Si TENDIC_1 MAT_2_OUAT FI TENDIC_1	7.	DECLARE C VFCTOR DAUBLE:				I MAT_2_OUAT
I 5TAR = UNITISTARI: MT_20UNT I STAR = UNITISTARI: MT_20UNT I STAR = UNITISTARI: MT_20UNT I Table = UNITITABLEI: MT_2.0UNT I T THESONC; MT_2.20UNT I T UNITIA: MT_2.20UNT I T UNITIA: MT_2.20UNT I T UNITIA: MT_2.20UNT I T T MT_2.20UNT	7	WRITE(6) LINE(1). COLUMN(1).	"MAT_2_OUAT BENCH PROGRAM RESULTS".	SK 1P141;		1 MAT_2_0UA1
\vec{H} $\vec{S} \cdot \vec{T} \cdot \vec{R} \cdot I = UWTT (\vec{S} \cdot \vec{T} \cdot \vec{R} \cdot I) :$ $MT = 20 \text{ Mar} - 20 \text$	i TF	STAR = UNITESTARD:				MAT_2_0UAT
FI TALE = UNITTABLE1: NAL_2 OUNT FI TABLE1 = UNITTABLE1: NAL_2 OUNT SI A = TM500C_0.1: NAL_2 OUNT SI B = TM500C_0.1: NAL_2 OUNT SI B = TM500C_0.1: NAL_2 OUNT SI C = UNITTABLE1: NAL_2 OUNT FI T = IM500C_0.1: NAL_2 OUNT SI T = UNITTA + BI: NAL_2 OUNT FI A = UNITTA + BI: NAL_2 OUNT	- UN	STARI = UNITISTARI);				MAT 2 DUAT
H Tablet = UNIT(TABLEL): S = T#500C. S = T#500C. S = T#500C. S = T#500C. S = Tw500C. S = W.12.004 M = W.12.004 M = W.12.004 M = UNIT(A. M = UNIT(A. <		T Sal F = UNIT (T A B E) -				
MITABLE1 = UNITITABLE11: $\begin{bmatrix} F_1 \\ A \end{bmatrix} = T^{E_0}OC_{\bullet,1}$: $\begin{bmatrix} F_1 \\ B \end{bmatrix} = T^{E_0}OC_{\bullet,2}$: $\begin{bmatrix} F_1 \\ C \end{bmatrix} = T^{E_0}OC_{\bullet,2}$: $\begin{bmatrix} F_1 \\ A \end{bmatrix} = T^{E_0}OC_{\bullet,2}$: $\begin{bmatrix} F_1 \\ A \end{bmatrix} = UNITIA = Bit$: $\begin{bmatrix} F_1 \\ A \end{bmatrix} = UNITIA$: $\begin{bmatrix} F_1 \\ A \end{bmatrix} = UNITIA$: $\begin{bmatrix} F_1 \\ A \end{bmatrix} = UNITIA$: $\begin{bmatrix} F_1 \\ A \end{bmatrix} = T^{E_0}OC_{\bullet,1} = A$: $\begin{bmatrix} F_1 \\ A \end{bmatrix} = C = UNITIA$: $\begin{bmatrix} F_1 \\ A \end{bmatrix} = C = T^{E_0}OC_{\bullet,1} = A$: $\begin{bmatrix} F_1 \\ A \end{bmatrix} = C = A$: $\begin{bmatrix} F_1 \\ B $.					
$\begin{bmatrix} F \\ S \end{bmatrix} = TW500C_{\bullet,1};$ $\begin{bmatrix} F \\ S \end{bmatrix} = TW500C_{\bullet,2};$ $\begin{bmatrix} F \\ S \end{bmatrix} = TW500C_{\bullet,3};$ $\begin{bmatrix} F \\ S \end{bmatrix} = C = UNIT(\overline{A} + \overline{B});$ $\begin{bmatrix} F \\ A = UNIT(\overline{A});$ $\begin{bmatrix} F \\ B \end{bmatrix} = \overline{C} + \overline{A};$ $\begin{bmatrix} F \\ MT - 2 - 0UAT \\ MT - 2 - 0$	T	TABLE1 =' UNIT(TABLEL);				MAT_2_OUAT
Si B T #500C MT_2_0UAT Si T T #500C MT_2_0UAT Si T T #500C MT_2_0UAT Si T UNITEA * BI: MT_2_0UAT Fi T UNITEA * BI: MT_2_0UAT Fi T UNITEA * BI: MT_2_0UAT Fi T MT_2_0UAT MT_2_0UAT Fi E MT_2_0UAT MT_2_0UAT Fi B T MT_2_0UAT MT_200C A MT_2_0UAT MT_2_0UAT					•	MAT_2_OUAT
File T = INSOR INT_2OUAT Si T = UNITE INT_2OUAT File T = UNITE INT_2OUAT File T = UNITE INT_2OUAT File T = T INT_2OUAT		B = TM500C			•	MAT_2_OUAT
FI C = UNIT(Ā * B); FI Ā = UNIT(Ā); FI B E AI FI IN5005 = Ai	<u>1</u> 1 1 1 1 1	Č = TMSODC :			•	MAT_2_OUAT
FI A = UNIT(A): MI A = UNIT(A): MI B = C + A: MI B = C + A: MI COUNT MI COUNT MI C + A:		Č = UNIT (Ā * B):	•			NAT_2_OUAT
EI =	L I	A = UNIT(A):				NAT_2_OUAT
		6 = C + 7;				NAT_2_OUAT
	Mž N	$1450\beta C = A:$	• •	•	t	MA1_2_06AT

and a state of the second
7.13 3R120700. APR. SRC180N IVERINU 145005 3: 145005 3: 145005 3: 145005 3: 145005 3: 145005 3: 145005 3: 145005 5: 175005 5: 175005 5: 175005 5: 175005 5: 175005 5: 175005 5: 175005 10: 175005 10: 175005 10: 175005 10: 175005 10: 175005 10: 175005 10: 175005 10: 17505 10:	=4A NOVEMBER 15, 1978 13:39:6.31 PAGE 4	SOUNCE CURRENT SCOPE				MAL_201AT								TAL 2 DUL		1m1-2-001	Int. 2.0LAT		
	-7.13 GR120700.APPL.SRC(6DR[VFR]RVL=	1 T4500C = 3:	TwSobc E:	Ū = STAR:	V = UNIT(STAR + STARI):	H = UNIT(U + V);	XI. = TABLES	T - UNITITARLE + TABLELD:	$\overline{1} = \frac{1}{2} = 0.01 T (\overline{X}_1 + \overline{Y}_1)$	TS450	T \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$	Survey States	1755 = X1;	TPCS7;	TPC S = 7:			•	

and Tree Pro

. М ар

FI X = TH50DC TPE450: XI I. N = 0: XI I. N = 0: XI Directive: YI Directive:		CURRENT SCOPE MAT_2_OUAT MAT_2_OUAT
H X = IM500C TPE450: H I. N = 0: H 1 H 1 H 0: H 0: H 1 H 1 H 1 H 0: H 0: H 0:		MAT_2_0UAT T MAT_2_0UAT
H I. N = 0; H 00, I = TRACE(X); H 00, I = TRACE(X); H Dn FOR K = 1 T3 3; H I I F X, X > 00 THEN H I I F X, X H I I D0; H I = K; H 00 = X ;		T MAT_2_OUAT
FI CD, T = TRACE(X): 41 DN FOR K = 1 T3 3: 41 L IF X, K > 00 THEN 41 L DD: 41 2 00 = X :		TAIN C TAN
41 Dn FOR K = 1 T3 3: 51 1F x 51 1F x 41 1 41 00: 41 1 42 1 41 20:		
4 1 1F x,x > 00 THEN 51 1 K,x > 00 THEN 41 1 00: 1 * K; 41 2 1 * K;		1 MAT_2 DUAT
41 1 DC: M 2 1 * K; 1 2 00 = X ;		MAT_20LAT
H 2 1 = K;		I MAT_2_GUAT
<u>-1</u> 2 00 = X, 3		I MAT_2_GUAT
		HAT_2OLAT
41 1 EMD:		AAT_2 DLAT
1		1 MAT_2_OLAT
HI T = 50RT(1 + 2 00 - T);		I MAT_2_OUAT
41 _ DG FAR H = 1 TC 3;		I MAT_2_DUAT
41 1 IF H = 1 THE4		I MAT_2_DUAT
4 1 / K = 2:		1 M41_2_0L4T
MI I EI SE		I MAT_2_OUAT
411 Dr:		H MAT_2_OUAT
4 2 IF H = 2 THEN		I MAT_2.0UAT
41 2 K = 3:		1 MA1_2_0LAT
MI 2 ELSE		I MAT_2_OUAT
41 2 K = 1:	-	I MAT_2_OUAT
41 1 EHD:	-	I MAT_2_OUAT
		I MA 1_2_0LAT
4 1 - 7 - 7 - X - X - X - X - X - X - X - X	-	HAT_2_GUAT
4 1 1 1 = 1) = 0 THEN	-	I MAT_2_UUAT
H I 00.0 ± (X − X) / 1; SI N J.K K.J		MAT_2_0UAT

(

Ō

CURRENT SCOPE ... PAGE MA1_2_0LA1 NAT_2.0UAT NAT_2_OLAT MA1_2_0LAT MI_2_OUAT HAT_2_OUAT NAT_2_OUAT HAT_2.0UAT NAT_2.0LAT HAT_2_OUAT MA1_2_0UAT MAT_2_OUAT 151. "U VECTOR". U. SKIPLZJ. COLUMNISJ. "V VECTOR". V. SKIPLZJ. COLUMNISJ. "W VECTOR". W. SKIPLZ | MAT_ZOUAT 1. COLUMNIS1. *XI VECTOR". XI. SKIP[2]. COLUMNISJ. 'Y VECTOR'. Y. SKIP[2]. COLUMNISJ. 'Z VECTOR' | MI___OUAT HAT_2_OUAT MA1_2_0W1 MAT_2_OUAT MAT_2_OLAT NAT_ZOUAT95 HLE-ARCSIN LARC SEC1 **, 3600* 3NG1*_SKIP12<u>3 * COLUNUS3) * COMV_ANG LARC SEC1 ** ANGV* SKIP121</u>+ COLUNN <mark>(MAJ_2.DUAJ</mark> WALTELGS SKIPISS. COLUMNILS. "INPUT MATRIX ... X. SKIPISJ. COLUMNISS. "TSMSO MATRIX". TSMSU. SKIP COLUMNISI. "TH5OAC MATRIX". IMSOUC. SKIPI2J. CALUMNISJ. "OBAS ". OBAS. SKIPI2J. COLUMNISJ. "OBAY 13:39:6.31 (2). COLUMMIS). "TPCS MATRIX ". TPCS: SKIP(2). COLUMNIS). "TPCMSO MATRIX", TPCMSC. SKIP(2). NCVEMBER 15. 1978 i SOURCE JR 1 20 700. APPL. SRC 1 RDR 1 VER J RVL = AA 0 = [X + X] / T: J+K-[= J,K + K,J ANTI = 2 ARCSIMIABVALIONAVI) RD: AVGV = ANG UNITIORAVI: ANG = 2 ARCCUSIOBAS): ANG = 3600. ANG RD: ; . T = SIGN(00) / 2: IF I = 0 THFN 034,5 = T 031 08AV = T 0: 00 = 1: : . EL SE a THE FLSE END: EL. 77-13E 95 AL 13 26 - H 56 - 3. 2 87 MI 14 14 16 13.46 2 گر 8 Ŧ 85 41 15 15 17 8 |k (6 93 H 82 41 ŝ STAT Ì

) ()

C

9 YI CICKE WI-2-QUATI FORE	360-22.13 G	R1 20700. APPL. SRC 1 30R1 VER) RVL = AA	NGVENARO 16 1070		
9 11 CICRE MAL 2 OUNT IN L 2 OUT		SOURCE		15:39:65:61	PAGE 7
9 41 CISE MIL_JOURT	96 FI . Z:				CURRENT SCOPE
	97 41 CLCSE MAT_2_0UAT;				MAT_Z_OUAT
				-	1 MAT_2_OUNT
		, ;;			• •
	i		•		
	:		:	•	
			;		
	•				
			2 		
	-		•	•	
	• •				
	· · ·		•	•	
			ı	4	
					!
				•	
				•	
				1	
			•	ı	

+ + 13 (y).

☆ U.S. GOVERNMENT PRINTING OFFICE: 1980-671-099/643

and in

26

ORIGINAL PAGE IS OF POOR QUALITY]