NATIONAL BUREAU OF STANDARDS REPORT

NBS PROJECT 2050630

August 2, 1965

NBS REPORT 8935

THE MEASUREMENT OF PROGRAM LOOPING SPEED IN DIGITAL COMPUTERS THROUGH MATRIX INVERSION BY PARTITIONING

by

George W. Reitwiesner

Computation Laboratory

This project supported in part by NASA funds.

U. S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

THE MEASUREMENT OF PROGRAM LOOPING SPEED IN DIGITAL COMPUTERS THROUGH MATRIX INVERSION BY PARTITIONING

by

George W. Reitwiesner

Foreword

Among internal operations in an electronic digital computer, those which perform administrative or housekeeping functions must act inconspicuously, else overall performance suffers. In general, these administrative or housekeeping functions are exercised through the manipulation of integer indexing data.

The following pages describe a set of test programs which were developed to try to measure the effect of integer indexing upon overall performance. To keep other conditions as nearly equal as possible, the same problem was programmed essentially identically except for its employment of three distinct levels of integer indexing: negligible, moderate, and severe.

That the negligible case is unrealistic in practice does not detract from its validity as a reference base from which to initiate measurement. And that the severe case might be reorganized to match the moderate case in speed performance does not totally detract from its representativeness of realistic difficult indexing situations.

Abstract

The internal operating speed of an electronic digital computer is governed basically by three parameters: the speed of the circuit components, the logical structure of the computer, and the particular set of instructions being executed; and the last of these is controlled by the programming system employed.

A set of programs is described for measuring the consolidated effect of all three parameters upon internal speed. These programs all execute the same basic task: matrix inversion by partitioning, in which the inverses are computed for successively larger upper left square partitions of the matrix being inverted, as a governing index escalates along the diagonal. But they perform that task for different conditions of a particular underlying requirement: recursive loop administration.

Assuming full efficiency of payload computational operation, overall internal speed varies with the efficiency of loop administration, and a measure of that efficiency and that speed is available in comparing machine performance on the several programs of this set.

To include all three parameters (component speed, logical structure, and programming system) in the measurement, the programs are written in a proven and popular programming language.

THE MEASUREMENT OF PROGRAM LOOPING SPEED IN DIGITAL COMPUTERS THROUGH MATRIX INVERSION BY PARTITIONING

The internal operating speed of an electronic digital computer is governed basically by three parameters: the speeds of the circuit components, the logical structure of the computer, and the particular set of instructions being executed; and the last of these is controlled by the programming system employed.

This writing describes a set of programs for measuring the consolidated effect of all three of these parameters upon overall internal speed. These programs all execute the same basic computational task, but they do so for different conditions of a particular underlying requirement: recursive loop administration. Thus, measurement becomes available by comparing machine performance on the programs of this set; and the validity of the measurement is governed by the extent to which overall internal computer performance depends upon the efficiency of recursive loop administration.

This dependence follows glibly from the assumption that payload computational operation in the computer is performed at maximum efficiency and from the observation that computer programming is recursive in basic nature. Occasionally, recursive loop administration is effected through comparison of numerical values of payload data, e.g. to effect reiteration until some convergence is achieved; but in great measure recursive loop administration is effected by manipulating integer indexing data to count iterations and/or to vary addresses over recursive loop executions.

In the earliest electronic digital computers, integer index manipulation incidental to loop reiteration and address adjustment was effected perforce by full-scale arithmetical and substitution instructions written out in the mainstream of the programming. In recognition of the historical development of a popular device for considerably reducing the chore of that manipulation and adjustment, integer indicies are in this text considered to be held in B-lines. Conceding, arbitrarily, the efficiency of execution of payload computational operations per se, loop speed --- and, consequently, overall computer speed --- rises and falls with the efficiency of B-line action; and this efficiency is measured in both the power and the lack of abundance of the B-line instructions employed.

The set of programs described here afford a measure of overall internal speed, using different conditions of recursive loop administration as a basic control variable. Alternatively, individual measures could be sought for the three parameters: component speed, logical structure, and programming system; but in the order given these three parameters become increasingly more difficult to measure abstractly, whence evolves the approach of measuring all three in unison. Yet a few comments are in order on these three parameters, individually.

The store speed is by far the easiest of the three parameters to measure abstractly, and the measure usually is expressed as (core) cycle time in microseconds or (occasionally) in hundreds of nanoseconds. Since the electrical combinations of binary signals in performing computational operations can be effected at speeds considerably above that of the store cycle, the store speed is the dominant hardware feature of internal computer performance.

The logical structure of a digital computer would be a very difficult attribute to measure in abstract terms. However, arithmetical operations being performed so fast that store cycle speed essentially dominates internal machine speed, a rough measure of efficiency in logical structure is expressed in the number of store accesses required for the execution of a particular unit of computation. And with overall speed being dominated by store speed over computational speed, this measure tends to translate to the number of instructions executed per particular unit of computation, weighted by the number of store accesses per instruction. Further, with B-line arithmetical instructions largely restricted to perform but the simplest operations of augment, tally, and test for equality to or passage beyond some limit, the efficiency of the logical structure in effecting B-line action is particularly well represented by the number of B-line instructions required per unit of computation, weighted by the number of store accesses per instruction. The ideal situation occurs when all B-line action incidental to loop control is effected by a small number of powerful index manipulating instructions which make but few store access demands. A quite important aspect, then, of the logical structure of a digital computer is the dispatch with which it enables B-line manipulation to be executed in loop administration.

Programming is prepared most readily in a problem oriented language, being translated to relentlessly detailed machine language through compiling and assembling programs by the machine itself. An abstract measure of total compiling and assembling efficiency would seem fantastically difficult to express. To provide overall semantic comprehensiveness and rigor, compiling programs tend to introduce redundancies, with attendant clumsiness, into the final machine language programs. And where redundancy and clumsiness exist in B-line manipulation in the innermost recursive loops, the overall internal operating speed of the computer falls below that which is implied by the operating speeds of its basic components and by niceties in its logical structure.

The programs described here have been developed in an effort to provide a single performance measure of the internal operating speed of a digital computer, embracing store speed, logical structure, and programming system effectiveness, in terms of its execution of B-line manipulation in the solution of a reasonably standard mathematical problem: matrix inversion.

Three programs are described. In overall plan they are essentially identical. In detailed implementation they differ in such a way as to involve different amounts of B-line manipulation in the execution of essentially the same amount of payload computational activity. It is in that difference that they attempt to measure B-line performance.

Multi-dimensional arrays of data are decidedly not uncommon in programming, and in ordinary practice an array such as $A(i,j,k)$ for $i=1,2,3...I$, $j=1,2,3...J$, $k=1,2,3...K$ is stored in a set of IJK contiguously addressed store locations of relative addresses such as $JI(k-1) + I(j-1) + (i-1)$, augmentation of the indicies i,j,k by 1 effecting respective address augmentations of $1,I,IJ$. Matrix inversion programs employ this pattern in two dimensions as rather representative of B-line activity in general scientific mathematical computation, and this representativeness extends to some general data processing applications.

Each of the three programs described here inverts a matrix by the same basic method: partitioning. The partitioned matrix relationships (where the symbol $\emptyset$ denotes irrelevant)

$$\begin{pmatrix} A & B & P \\ C & D & Q \\ R & S & T \end{pmatrix} \cdot \begin{pmatrix} W & X & 0 \\ Y & Z & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} I & 0 & \emptyset \\ 0 & I & \emptyset \\ \emptyset & \emptyset & 0 \end{pmatrix} = \begin{pmatrix} W & X & 0 \\ Y & Z & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} A & B & P \\ C & D & Q \\ R & S & T \end{pmatrix} \qquad (1)$$

yield

$$\begin{aligned} AW + BY &= I \\ AX + BZ &= 0 \\ YA + ZC &= 0 \\ YB + ZD &= I \end{aligned} \qquad (2)$$

whence, assuming the existence of A^{-1}, follow

$$\begin{aligned} Z &= (D - CA^{-1}B)^{-1} \\ X &= -A^{-1}BZ \\ Y &= -ZCA^{-1} \\ W &= A^{-1} + A^{-1}BZCA^{-1} \end{aligned} \qquad (3)$$

for the expression of the inverse $\begin{pmatrix} W & X \\ Y & Z \end{pmatrix}$ to the partition $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ in terms of the smaller partitions B,C,D, and A^{-1}. Then, starting with the inverse A^{-1} of some upper left square partition A of a given matrix, one can develop iteratively, through equations (3), the inverses of successively

3

larger upper left square partitions, each such larger inverse being taken as the A^{-1} for the next iteration of (3), until the inverse of the entire given matrix is obtained. (Of course, no upper left square partition may be singular.)

The simplest procedure, clearly, is that of restricting all partitions D,Z and the initial A, A^{-1} to be of dimension 1-by-1, whence they all are scalars, and B,C,X,Y are vectors. Under this restriction, all equations (3) in the total inversion procedure involve matrix and vector multiplication and addition (and subtraction) except for the formation of the Z's and the initial A^{-1} by simple division.

A more complicated procedure, yet one providing regularity of escalation, is that of restricting all partitions D,Z to be of dimension 2-by-2, the dimension of the initial A, A^{-1} being 1-by-1 or 2-by-2 as the total matrix order is odd or even. Under this restriction, all equations (3) in the total inversion procedure involve matrix (but not vector) multiplication and addition (and subtraction) except for the formation of the Z's and the initial A^{-1} by simple 2-by-2 inversion (e.g. by adjoints), or by simple division for A^{-1} if the total matrix order is odd.

Two of the three programs described here are referred to by the terms 1-by-1 and 2-by-2; they are straightforward respective implementations of the above two procedures. The third program will be referred to by the term 0-by-0; it will be identified below.

In each of these two programs (1-by-1 and 2-by-2), the entire inversion programming becomes a set of reiterated loops embedded in an overall escalating loop which rides down the matrix diagonal. However, the second (2-by-2) program is much more complex, loopwise, than the first (1-by-1), because the second employs only matrices (with the trivial exception of the initial scalars A, A^{-1} for odd order), while the first handles not only matrices, but vectors and scalars as well, and need not employ recursion to span the unit dimensions of the vectors B,C,X,Y and the scalars D,Z. Further, the second program requires a type of indexing activity which has no counterpart in the first: in executing equations (3), the column indicies of B and X and the row indicies of C and Y and both the row and column indicies of D and Z are formed by combining the dimension of A and W with the value (1 or 2) of the recursive index which spans the width of the matrix engaged: B,X,C,Y,D, Z; in the 2-by-2 program, these combinations are embedded as deeply as possible within the echelons of looping.

4

In developing both the 1-by-1 and 2-by-2 programs, the following rules were employed:

1. The given matrix, and its inverse, must be symmetric.
2. The given matrix must remain undisturbed throughout the entire inversion operation.
3. The given matrix and each intermediate (and final) inverse developed must be fully stored as a (symmetric) square array.
4. Storage equal to that allotted for the given matrix must be allotted for its inverse (including that for the intermediate smaller inverses), but no other numerical data storage may be employed during the entire inversion operation.
5. Reiterative loops must be employed wherever reasonably possible.
6. The symmetry property must be employed wherever possible (to eliminate redundant computation) by copying elements across the diagonal.
7. All programming must be written in a proven and prominent problem oriented language.

Deliberately, the requirement has been made severe for the 2-by-2 program over the 1-by-1. That which is easy should be executed easily. What is intended here (in the 2-by-2 program) is the development of a measuring device for the execution of that which is difficult.

The 0-by-0 program is identical to the 1-by-1 in terms of payload operation, but it is significantly different in indexing requirement; in its development, the same 7 rules were employed, except that rule 5 was reversed:

5. Reiterative loops must not be employed.

Hence, wherever the 1-by-1 program employs a K-fold recursively executed single payload operation under loop control, the 0-by-0 program employs instead a set of K separate payload operations without looping.

Thus, the 0-by-0 program provides a "B-line" - less base from which to measure the effect upon internal computer speed of rather conventional B-line action in the 1-by-1 program and of more severe, yet not unreasonable, B-line action in the 2-by-2 program.

Except for such differences as have been detailed (the escalation interval for 1-by-1 vs. 2-by-2, and avoiding vs. using B-lines for 0-by-0 vs. 1-by-1), and except for such details as necessarily follow therefrom, all three total programs are alike. Each performs as follows: three parameters are read from a punched card: an interval, a maximum order, and a frequency; for each order n in increments of the interval up to the maximum, the (positive definite) symmetric matrix[(1)]

$$A_{ij} = \sin(ij/n) + (n+1)\delta_{ij}$$

is inverted as many times as prescribed by the frequency; for each such order, the total machine time for all inversions is read internally and stored; for each such order beyond the third, a least squares fit is made to the cubic polynomial which expresses performance time in terms of matrix order; and for each such order the following data are printed: order, time, polynomial coefficients, and the maximum error (departure from the identity matrix) in postmultiplying the least squares normal matrix by its computed inverse. The polynomial coefficients are normalized to represent the cubic for one iteration at each order n.

In Appendix A are derived the cubic polynomials which express, in terms of matrix order, the numbers of payload arithmetical operations which are executed in the three programs: 0-by-0, 1-by-1, and 2-by-2. In this derivation the payload classification is assigned categorically to all non-B-line operations to distinguish them from the integer index manipulating operations which are under scrutiny. Glibly summarized, the polynomials for all three programs have essentially equivalent respective coefficients. For example, within an error of at most half the order n of the matrix, the total number of multiplications (types M and V) for each program is given by $(1/2)n^3 + (1/2)n^2 - (3/2)n$. Hence, apart from B-line manipulation, the times expended in the execution of payload operations in the inversion of a matrix of order n should not differ widely among the three programs, for significantly large n. Therefore, such differences as are observed in machine running times under these three programs express a measure of the efficiency with which B-line manipulation is handled by a particular combination of machine hardware, logical structure, and programming system.

The ratio of machine running time for the 1-by-1 program to that for the 0-by-0 program offers a measure of the cost to perform conventional B-line action; and this ratio is particularly significant in that, for the same order matrix, the succession of payload operations performed for the 0-by-0 program is identical to that for the 1-by-1 program. The ratio of machine running time for the 2-by-2 program to that for the 1-by-1 program offers a measure of the increase in cost to handle complex indexing situations.

---

(1) This matrix has been chosen as one which will give reasonable assurance of freedom from singularity problems, with a rather random spattering of small elements superimposed around a dominant diagonal.

No machine performance data are given here. It is the intention of this writing not to report on measurements taken, but rather to describe a device which has been developed in an effort to make a measurement. As of this writing, machine operation has been employed only as necessary to insure that the programs perform as intended.[2]

It is acknowledged that the validity of the least-squares-generated cubic timing polynomials

$$\text{time} = a_3x^3 + a_2x^2 + a_1x + a_0$$

argues its own case as the coefficients stabilize for matrix orders which are sufficiently large to dominate the magnitudes of the ratios of successive pairs of coefficients $|a_{i-1}/a_i|$, $i=1,2,3$. It is expected that on a computer with a 32,000 word store: matrices of up to around 100th order will be accommodated under the 1-by-1 and 2-by-2 programs; but that a much lower limit (ten to twenty) will apply for the 0-by-0 program, because clearly its length increases as the 3rd power of the matrix order. The lower limit (ten to twenty) for the 0-by-0 program should not seriously impede validity, for no B-line action occurs, and the corresponding coefficient ratios for the payload operation polynomials are small. For the 1-by-1 and 2-by-2 programs, where B-line action does occur, the higher limit (around 100) permits broad variation in the ratios of the timing polynomial coefficeints as they stabilize, thus accommodating correspondingly broad effects of indexing upon problem execution time.

In blunt deference to its relative universality, Fortran has been chosen as the programming language. In particular, both programs have been written in Fortran II as embodied in the Bell operating system as employed at the National Bureau of Standards. Rewriting the programming in other problem oriented languages should be reasonably easily effected, following the logical flow displayed in the Fortran listings given here. Such changes as may be necessary to accommodate other versions of Fortran should be readily recognized and simply executed.

The three programs are exhibited in Fortran language in Appendix B. The 1-by-1 and 2-by-2 programs are displayed in full for maximum order 100. The 0-by-0 program is displayed only for the accommodation of matrices up to maximum order 5; generalization to higher maximum order is obvious.

A program used to generate the 0-by-0 program for any stipulated maximum order (up to 99) also is exhibited in Appendix B.

The contribution by Dr. R.J. Arms of suggestions re least squares fitting is acknowledged.

(2) As of September, 1964.

Postscript

As an afterthought, and just to provide completeness of reference data, the 2-by-2 program was revised to remove the combinations of indicies from the deepest echelons of looping, as had been provided in the last phrases of page 4, above. This revision is referred to here by the term D-by-D.

Illustrative of the resulting change is the simplification of a sequence of Fortran statements such as

```
1  L=M-2
2  DO 4 K=1,2
3  I=L+K
4  Payload operation involving index I
```

to

```
1  L=M-1
2  DO 4 I=L,M
4  Payload operation involving index I.
```

Implicitly, the latter sequence is implemented more readily than is the former, and consequently the very use of the 2-by-2 case begs the justification which appears readily when more complex arithmetical operations are substituted for the simple addition of statement 3, above. Indeed, the mere substitution here of subtraction for addition (representing, for example, a need to reverse the direction of procedure through an array) precludes the simplification to the D-by-D case for programming languages and compilers which cannot accommodate bidirectional incrementation.

The D-by-D program is displayed in Appendix C.

Appendix A

The number of payload operations employed in a single inversion is expressed as a cubic polynomial in which the variable is the order n of the matrix. This follows trivially from the nature of the inversion procedure and from the employment of three echelons of looping in the programming.

Six types of payload operations are recognized:
C = copy or negate,
Z = set to zero (e.g. before entering a summation loop),
T = test (for zero divisor),
D = divide,
M = multiply,
V = multiply and then either add or subtract.
Argument may be advanced that certain of these types should not have payload status, e.g. the test for zero divisor prior to division could be classified as administrative. However all these six types are here regarded in the payload category in order to isolate distinctly all remaining activity, which embodies all the B-line activity. (Indeed, the inclusion vs. exclusion of the type T operation will affect but the linear and constant coefficients of the polynomials developed.)

In this appendix the coefficients are determined for the cubic polynomials which express the numbers of each of the above six types of operations and for cubics which express the numbers of certain combinations of types, for both the 1-by-1 and 2-by-2 programs. From the technique of its development, simply that of unwinding reiterative loops, it follows that corresponding data for the polynomials for the 0-by-0 program are respectively indentical to the data given for the 1-by-1 program. Hence 0-by-0 data are given here implicitly, along with the 1-by-1 data.

The following notations are employed:

for 1-by-1	for 2-by-2
n = matrix order	n = matrix order
r = n-1	r = n-2
	m = 1-or-2 as n is odd-or-even
k = 1,2,3...n	k = m,m+2,m+4...n
ℓ = k-1 = 0,1,2...r	ℓ = k-2 = m-2,m,m+2,..r
	E = m-1 = 0-or-1 as n is odd-or-even

The notations n,m,k,ℓ correspond, respectively, to the indices NATRIX,MATRIX,KATRIX,LATRIX in the programming:

n denotes the order of the matrix being inverted.

m denotes the initial value of the running index k for 2-by-2.

k denotes the order of the upper left partition whose inverse currently is being determined in the programming.

ℓ denotes the order of the upper left partition whose inverse last was determined in the programming and limits the iteration of programming loops in developing the inverse of the upper left partition of order k.

r denotes the upper limit for the running index ℓ.

E is employed for the 2-by-2 case: to count the number of executions of payload operations at the beginning of the inversion procedure, incidental to preparing for recussion over the running index ℓ; and to distinguish between the summations of powers of even and odd alternate integers.

The following relationships are involved:

for sums of powers of consecutive integers:

$$\Sigma_1^n (i)^0 = n,$$
$$\Sigma_1^n (i) = n(n+1)/2,$$
$$\Sigma_1^n (i)^2 = n(n+1)(2n+1)/6;$$

for sums of powers of alternate[1] integers (for 2-by-2):

$$\Sigma_m^r (\ell)^0 = (r+1-E)/2,$$
$$\Sigma_m^r (\ell) = ((r+1)^2-E)/4,$$
$$\Sigma_m^r (\ell)^2 = r(r+1)(r+2)/6 \quad \text{(independent of E)}.$$

Ignoring, momentarily, the cases k=1 (ℓ=0) for 1-by-1 and k$\leq$2 ($\ell\leq$0) for 2-by-2 (for these cases are treated separately in the programming), the numbers of executions of the several types of payload operations as functions of the index ℓ are seen from the programming to be as indicated in Table I, where bracketing indicates subordination of looping, and where i is employed as a limiting index in an inner summation, but a running index in an outer summation, in the 2nd and 3rd lines from the bottom. Performance of the indicated summations yields Table II, which is consolidated by operation type to Table III. The numbers of operations in this consolidation are performed recursively for ℓ=1,2,3...r for 1-by-1 and for ℓ=m,m+2,m+4...r for 2-by-2, whence follows Table IV for the total numbers of payload operations per inversion of order n (= r+1 or r+2), omitting those deliberately ignored.

---

[1] The notation here is unconventional in that

$$\Sigma_m^r (\ell)^p = (m)^p + (m+2)^p + (m+4)^p + \ldots + (r)^p, \quad p = 0,1,2.$$

10

Those deliberately ignored are seen from the programming to be as indicated in Table V. Therefore, combining Tables IV and V and substituting n-1 and n-2 for r yields Table VI for the total number of payload operations executed in the inversion of a matrix of order n.

In relating to the time of execution of a matrix inversion, the polynomials developed in Table VI may be combined only insofar as they are weighted according to the mechanics of machine execution of the related operations. Within the limitation placed by this reservation, a crude expression of the amount of payload work expended in the execution of an inversion of order n is derived by adding the bottom three lines of Table VI to develop the total number of computational operations as

for 1-by-1 $\quad (1/2)n^3 + (1/2)n^2$

for 2-by-2 $\quad (1/2)n^3 + (1/2)n^2 - (1/2)n + (1/2)(1-E)$

and by adding all six lines to develop the total number of all recognized operations as

for 1-by-1 $\quad (2/3)n^3 + (3/2)n^2 + (5/6)n$

for 2-by-2 $\quad (7/12)n^3 + (15/8)n^2 + (29/12)n - [1+(15/8)(1-E)]$.

From the similarity of the coefficients of the higher powers of n in the six lines of Table VI and in the two sets of polynomials given above, it would appear that --- within the reservation expressed re mechanics of machine execution --- the times expended in the execution of all payload operations in the inversion of a matrix of order n should not differ widely among the cases 0-by-0 and 1-by-1 and 2-by-2, for significantly large n.

//

Line	1-by-1	Type	2-by-2	Line
72	1	C	4	310
77	Σ_1^ℓ	Z	Σ_1^ℓ [2]	317
80	Σ_1^ℓ [Σ_1^ℓ]	V	Σ_1^ℓ [2[Σ_1^ℓ]]	320
83	Σ_1^ℓ	V	Σ_1^ℓ [3]	330
		C	1	333
		C	3	337-9
		Z	1	342
		V	2	346
86	1	T	1	350
		C	1	357
87	1	D	3	358
		C	1	361
		Z	Σ_1^ℓ [2]	369
93	Σ_1^ℓ	M		
		V	Σ_1^ℓ [2[2]]	372
98	Σ_1^ℓ [Σ_1^ℓ]	V	Σ_1^ℓ [Σ_1^ℓ[2]]	380
101	Σ_1^ℓ [Σ_1^ℓ]	C	Σ_1^ℓ [Σ_1^ℓ]	383
103	Σ_1^ℓ	C	Σ_1^ℓ [2]	387

Table I

12

Line	1-by-1	Type	2-by-2	Line
72	1	C	4	310
77	ℓ	Z	2ℓ	317
80	ℓ^2	V	$2\ell^2$	320
83	ℓ	V	3ℓ	330
		C	1	333
		C	3	337-9
		Z	1	342
		V	2	346
86	1	T	1	350
		C	1	357
87	1	D	3	358
		C	1	361
		Z	2ℓ	369
93	ℓ	M		
		V	4ℓ	372
98	$\ell^2/2 + \ell/2$	V	$\ell^2 + \ell$	380
101	$\ell^2/2 + \ell/2$	C	$\ell^2/2 + \ell/2$	383
103	ℓ	C	2ℓ	387

Table II

1-by-1	Type	2-by-2
$(1/2)\ell^2 + (3/2)\ell + 1$	C	$(1/2)\ell^2 + (5/2)\ell + 10$
ℓ	Z	$4\ell + 1$
1	T	1
1	D	3
ℓ	M	
$(3/2)\ell^2 + (3/2)\ell$	V	$3\ell^2 + 8\ell + 2$

Table III

1-by-1	Type	2-by-2
$(1/6)r^3 + r^2 + (11/6)r$	C	$(1/12)r^3 + (7/8)r^2 + (77/12)r + (45/8)(1-E)$
$(1/2)r^2 + (1/2)r$	Z	$r^2 + (5/2)r + (3/2)(1-E)$
r	T	$(1/2)r + (1/2)(1-E)$
r	D	$(3/2)r + (3/2)(1-E)$
$(1/2)r^2 + (1/2)r$	M	
$(1/2)r^3 + (3/2)r^2 + r$	V	$(1/2)r^3 + (7/2)r^2 + 6r + 3(1-E)$

Table IV

14

1-by-1	Type	2-by-2
1	C	1 + 8E
	Z	E
1	T	1
1	D	1 + 2E
	V	2E

Table V

1-by-1	Type	2-by-2
$(1/6)n^3 + (1/2)n^2 + (1/3)n$	C	$(1/12)n^3 + (3/8)n^2 + (47/12)n - (27/8) + (19/8)E$
$(1/2)n^2 - (1/2)n$	Z	$n^2 - (3/2)n + (1/2) - (1/2)E$
n	T	$(1/2)n + (1/2) - (1/2)E$
n	D	$(3/2)n - (1/2) + (1/2)E$
$(1/2)n^2 - (1/2)n$	M	
$(1/2)n^3 - (1/2)n$	V	$(1/2)n^3 + (1/2)n^2 - 2n + 1 - E$

Table VI

15

Appendix B

Programming for

Case 1-by-1
Case 2-by-2
Generator
Case 0-by-0

Note: Between the date of submission and the date of publication of this report, an error of arrangement was revealed in the programming in that the main recursive loop (beginning at statement 11) had been located out of sequence from its loop control statements:

```
      DO 127 NITRTN = 1, NFRQCY
      GO TO 11
127   CONTINUE.
```

This mislocation quite apparently did not adversely affect operation under the Fortran II compiler used for checkout, but it did invalidate operation under a later Fortran IV compilation.

The correction of this deficiency and certain programming improvements, developed since initial composition, are not shown in this writing.

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTRX   1
       JOB     00655,AS,PROGRAMMER,                           (BY 1) SEPT64MTRX   2
       FOR                                                                 MTRX   3
C     *SYMMETRIC MATRIX INVERSION BY PARTITION, WITH LEAST SQUARES FIT ONMTRX   4
C     *THE CUBIC POLYNOMIAL FOR INVERSION TIME IN TERMS OF MATRIX ORDER. MTRX   5
C     *FOR CERTAIN ORDERS NORDER=1,2,3... TO A PRESCRIBED MAXIMUM MORDER,MTRX   6
C     *OMITTING ALL ORDERS NORDER NOT DIVISIBLE BY A PRESCRIBED INTERV,  MTRX   7
C     *INVERT THE (POSITIVE DEFINITE) SYMMETRIC MATRIX DEFINED AS FOLLOWSMTRX   8
C     *A(I,J) = SIN((I*J)/NORDER) + THE DIAGONAL MATRIX L(I,I) = NORDER+1MTRX   9
C     *A PRESCRIBED NUMBER NFRQCY OF TIMES, INTERNALLY MEASURING THE     MTRX  10
C     *TOTAL DURATION OF ALL NFRQCY INVERSIONS FOR EACH NORDER.          MTRX  11
C     *USING THE MATRIX INVERSION PROGRAM FOR THE LEAST SQUARES FITTING, MTRX  12
C     *FOR EACH NORDER ABOVE 3, USE LEAST SQUARES TO FIT A CUBIC TO THE  MTRX  13
C     *INTERNALLY MEASURED DURATION TIMES FOR ALL ORDERS UP TO NORDER,   MTRX  14
C     *AND RECORD THE MAXIMUM ERROR (DEPARTURE FROM IDENTITY MATRIX) IN  MTRX  15
C     *POSTMULTIPLYING EACH LEAST SQUARES NORMAL MATRIX BY ITS INVERSE.  MTRX  16
C     *FINALLY, PRINT ALL DATA APPLICABLE TO EACH NORDER, SPECIFICALLY-  MTRX  17
C     *NORDER, TIME, COEFFICIENTS (CUBIC,SQUARE,LINEAR,CONSTANT), ERROR, MTRX  18
C     *EMPLOYING (FAKED) NEGATIVE TIME TO MARK INABILITY TO GET SOLUTION.MTRX  19
C     *LMTDIM MUST RELATE TO SEVERAL DIMENSIONS AND LIMITS MORDER.       MTRX  20
C     *LMTDIM=100                                                        MTRX  21
C     *BOTH DIMENSIONS EQUAL LMTDIM                                      MTRX  22
      DIMENSION WATRIX(100,100)                                          MTRX  23
      DIMENSION VATRIX(100,100)                                          MTRX  24
C     *ONE DIMENSION EQUAL TO LMTDIM                                     MTRX  25
      DIMENSION DURATN(100)                                              MTRX  26
C     *DATA INPUT                                                        MTRX  27
C     *READ THE PARAMETERS INTERV,MORDER,NFRQCY FROM A PUNCHED CARD      MTRX  28
C     *INTERV FROM COLUMNS 11-15 IN INTEGER FORMAT, AND                  MTRX  29
C     *MORDER FROM COLUMNS 16-20 IN INTEGER FORMAT, AND                  MTRX  30
C     *NFRQCY FROM COLUMNS 21-25 IN INTEGER FORMAT.                      MTRX  31
    4 FORMAT(10H          3I5)                                           MTRX  32
      READ  4,                                 INTERV, MORDER, NFRQCYMTRX  33
      PRINT 4,                                 INTERV, MORDER, NFRQCYMTRX  34
      GO TO 110                                                          MTRX  35
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTRX  36
C     *SYMMETRIC MATRIX INVERSION BY PARTITION, RECURSIVE ENLARGEMENT = 1MTRX  37
C     *(AAABHH) (WWWXOO) (IOOOOO) (WWWXOO) (AAABHH)                     MTRX  38
C     *(AAABHH) (WWWXOO) (OIOOOO) (WWWXOO) (AAABHH)  DEVELOP  (1) AW+BY=IMTRX  39
C     *(AAABHH).(WWWXOO)=(OOIOOO)=(WWWXOO).(AAABHH)    THE    (2) AX+BZ=OMTRX  40
C     *(CCCDHH) (YYYZOO) (OOOIOO) (YYYZOO) (CCCDHH)   BASIC   (3) YA+ZC=OMTRX  41
C     *(HHHHHH) (OOOOOO) (OOOOOO) (OOOOOO) (HHHHHH) EQUATIONS (4) YB+ZD=IMTRX  42
C     *(HHHHHH) (OOOOOO) (OOOOOO) (OOOOOO) (HHHHHH)                     MTRX  43
C     *THE SLASH (/), USED AS A PREFIX, HERE DENOTES INVERSION          MTRX  44
C     *DEFINING E=/AB                                                   MTRX  45
C     *     AND F=C/A                                                   MTRX  46
C     *YIELDS 1 Z=/(D-FB)=/(D-CE)            WHERE SYMMETRY APPLIES FOR MTRX  47
C     *   AND 2 X=-EZ                                  A,W (AND FOR D,Z) MTRX  48
C     *   AND 3 Y=-ZF                     AND WHERE MUTUAL TRANSPOSES AREMTRX  49
C     *   AND 4 W=/A-EY=/A-XF                          (B,C),(E,F),(X,Y) MTRX  50
C     *HENCE THE COMPUTATIONAL PROCEDURE TO DEVELOP (WXYZ) FROM (ABCD) ISMTRX  51
C     *1. INITIALLY COMPUTE /A FROM A, AND THEN REITERATE STEPS 2-8     MTRX  52
C     *2. MOVE D TO Z                                                   MTRX  53
C     *3. COMPUTE F=C/A AND STORE AT Y                                  MTRX  54
C     *4. COMPUTE Q=(/Z)=D-FB AT Z                                      MTRX  55
C     *5. INVERT  Z=/Q IN PLACE                                         MTRX  56
C     *6. COMPUTE X=-EZ                                                 MTRX  57
C     *7. COMPUTE W=/A-XF           (COMPUTE ONE HALF, COPY OTHER HALF)MTRX  58
C     *8. COPY Y FROM X, WHENCE (WXYZ) IS /A FOR NEXT ITERATION         MTRX  59
C     *RESTRICT ALL PARTITIONS D,Z TO ORDER 1,                          MTRX  60
C     *INCREASING ORDER OF A,W BY 1 FOR EACH REITERATION OF STEPS 2-8.  MTRX  61
C     *BEGIN AND END WITH MATRIX VATRIX(-,-) OF ORDER NATRIX            MTRX  62
C     *END INVERSION WITH MATRIX WATRIX(-,-) OF ORDER NATRIX            MTRX  63
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTRX
   11 CONTINUE                                                          MTRX
C     *MAIN LOOP - EXECUTE N TIMES, ADVANCING INDEX KATRIX BY 1         MTRX
      IF(NATRIX)                                     92,    92,    28MTRX
   28 DO    90 KATRIX=1,NATRIX                                          MTRX
C     *OBSERVE KATRIX IS ORDER OF (ABCD),(WXYZ). LATRIX IS ORDER OF A,W MTRX
      LATRIX=KATRIX-1                                                   MTRX
C     *MOVE D TO Z (A TO W FOR FIRST PASS (AND INVERT A IF KATRIX=1))   MTRX
      WATRIX(KATRIX,KATRIX)= VATRIX(KATRIX,KATRIX)                      MTRX
C     *SEPARATE CASE KATRIX=1 FOR INVERSION OF A (SKIP FORMING F,Z)     MTRX
   39 IF(LATRIX)                                     90,    51,    40MTRX
C     *COMPUTE F=C/A AND Q=D-FB                                         MTRX
   40 DO    49 IATRIX=1,LATRIX                                          MTRX
      WATRIX(KATRIX,IATRIX)=0.                                          MTRX
C     *ACCUMULATE ONE ELEMENT OF F=C/A                                  MTRX
      DO    45 JATRIX=1,LATRIX                                          MTRX
   45 WATRIX(KATRIX,IATRIX)= WATRIX(KATRIX,IATRIX)+VATRIX(KATRIX,JATRIX)MTRX
  451                                          *WATRIX(JATRIX,IATRIX)MTRX
C     *FORM Q=D-FB BY ACCUMULATION TO D                                 MTRX
   49 WATRIX(KATRIX,KATRIX)= WATRIX(KATRIX,KATRIX)-WATRIX(KATRIX,IATRIX)MTRX
  491                                          *VATRIX(IATRIX,KATRIX)MTRX
C     *INVERT Q (INVERT D=A FOR KATRIX=1)                               MTRX
   51 IF(WATRIX(KATRIX,KATRIX))                      52,    92,    52MTRX
   52 WATRIX(KATRIX,KATRIX)=1./WATRIX(KATRIX,KATRIX)                    MTRX
C     *SEPARATE CASE, SKIP TO END OF LOOP AFTER INVERTING A FOR KATRIX=1 MTRX
   69 IF(LATRIX)                                     90,    90,    70MTRX
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTRX  90
C     *COMPUTE X=-EZ BY TRANSPOSING Y=-ZF                               MTRX  91
   70 DO    77 IATRIX=1,LATRIX                                          MTRX  92
   77 WATRIX(IATRIX,KATRIX)=-WATRIX(KATRIX,KATRIX)*WATRIX(KATRIX,IATRIX)MTRX  93
C     *COMPUTE W=/A-XF AND FORM Y AS TRANSPOSE OF X                     MTRX  94
      DO    88 IATRIX=1,LATRIX                                          MTRX  95
      DO    85 JATRIX=IATRIX,LATRIX                                     MTRX  96
C     *ACCUMULATE -XF INTO /A, ONLY ALONG AND ABOVE MAJOR DIAGONAL      MTRX  97
   84 WATRIX(IATRIX,JATRIX)= WATRIX(IATRIX,JATRIX)-WATRIX(IATRIX,KATRIX)MTRX  98
   841                                          *WATRIX(KATRIX,JATRIX)MTRX  99
C     *COMPLETE W BY SYMMETRY                                           MTRX 100
   85 WATRIX(JATRIX,IATRIX)= WATRIX(IATRIX,JATRIX)                      MTRX 101
C     *TRANSPOSE ONE ELEMENT OF X TO Y                                  MTRX 102
   88 WATRIX(KATRIX,IATRIX)= WATRIX(IATRIX,KATRIX)                      MTRX 103
C     *END OF MAIN LOOP                                                 MTRX 104
   90 CONTINUE                                                          MTRX 105
      GO TO (  127,  170), ISWTCH                                       MTRX 106
C     *ALARMS                                                           MTRX 107
   92 DURATN(NORDER)=-1.                                                MTRX 108
      IF(ISWTCH-2)                                    130,  210,  210MTRX 109
C     *END OF MATRIX INVERSION PROGRAMMING                              MTRX 110
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTRX 111
  110 CONTINUE                                                          MTRX 112
C     *PRINCIPAL ITERATIVE LOOP ON NORDER=1,2,3...MORDER                MTRX 113
      DO  222  NORDER=1,MORDER                                          MTRX 114
      TORDER=NORDER                                                     MTRX 115
C     *CLEAR DURATN TABLE AND SET ERROR RECORD = 0.                     MTRX 116
      ERROR=0.                                                          MTRX 117
      DURATN(NORDER)=-TORDER                                            MTRX 118
C     *SKIP CASES FOR WHICH NORDER IS NOT DIVISIBLE BY INTERV           MTRX 119
      IF(NORDER-(NORDER/INTERV)*INTERV)               222,  115,   222MTRX 120
  115 DURATN(NORDER)=0.                                                 MTRX 121
C     *CREATE A(I,J)                                                    MTRX 122
      DO    119 IORDER=1,NORDER                                         MTRX 123
      DO    119 JORDER=1,IORDER                                         MTRX 124
      KORDER=IORDER*JORDER                                              MTRX 125
      VATRIX(IORDER,JORDER)=KORDER                                      MTRX 126
      VATRIX(IORDER,JORDER)=SINF(VATRIX(IORDER,JORDER)/TORDER)          MTRX 127
      IF(IORDER-JORDER)                               119,  118,   119MTRX 128
  118 VATRIX(IORDER,JORDER)=VATRIX(IORDER,JORDER)+TORDER+1.             MTRX 129
  119 VATRIX(JORDER,IORDER)=VATRIX(IORDER,JORDER)                       MTRX 130
C     *INVERSION LOOP SETUP                                             MTRX 131
      NATRIX=NORDER                                                     MTRX 132
      ISWTCH=1                                                          MTRX 133
C     *CLOCK ON     INTERROGATE CLOCK HERE                              MTRX 134
      TIMEON=CLOCKF(DUMMY)                                              MTRX 135
C     *LOOP NFRQCY ITERATIONS                                           MTRX 136
      DO    127 NITRTN=1,NFRQCY                                         MTRX 137
      GO TO 11                                                          MTRX 138
  127 CONTINUE                                                          MTRX 139
C     *CLOCK OFF    INTERROGATE CLOCK HERE                              MTRX 140
      TIMEUP=CLOCKF(DUMMY)                                              MTRX 141
C     *COMPUTE DURATION  DURATN(NORDER) = CLOCK SPAN                    MTRX 142
      DURATN(NORDER)=TIMEUP-TIMEON                                      MTRX 143
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
C     *LEAST SQUARES COMPUTATION.
C     *DENOTE S(K) = SUM OF ((I**K)              ) OVER I = ORDERS EMPLOYED
C     *AND    T(K) = SUM OF ((I**K)*(DURATN(I))) OVER I = ORDERS EMPLOYED
C     *AND CUBIC = A(3)*(X**3) + A(2)*(X**2) + A(1)*X + A(0) WHERE THE
C     *VARIABLE X DENOTES DURATION.
C     *SOLVE       (S(0) S(1) S(2) S(3))        (A(0))         (T(0))
C     *THIS        (S(1) S(2) S(3) S(4))    *   (A(1))    =    (T(1))
C     *SYSTEM      (S(2) S(3) S(4) S(5))        (A(2))         (T(2))
C     *USING       (S(3) S(4) S(5) S(6))        (A(3))         (T(3))
C     *STORAGE   VATRIX(1,1)-VATRIX(4,4)   VATRIX(8,-)   VATRIX(-,8)
C     *CREATE NORMAL MATRIX AND VECTOR AND CLEAR COEFFICIENT VECTOR
  130 DO  133  IORDER=1,7
      VATRIX(IORDER,7)=0.
      VATRIX(IORDER,8)=0.
  133 VATRIX(8,IORDER)=0.
C     *USE KORDER TO COUNT VALID DATA CASES
      KORDER=0
      DO    159 IORDER=1,NORDER
C     *SKIP IRRELEVANT CASES (DURATN(IORDER) NEGATIVE)
      IF(DURATN(IORDER))                                159,    142,    142
  142 KORDER=KORDER+1
C     *DENOTE POWRDR = POWERS OF ORDER VARIABLE IORDER
      POWRDR=1.
C     *SCALE (NORMALIZE) INDEPENDENT TIME VARIABLE BY DIVISION BY NORDER.
      XORDER=IORDER
      YORDER=NORDER
      XORDER=XORDER/YORDER
      DO    158 JORDER=1,7
      IF(JORDER-1)                                      222,    153,    152
  152 POWRDR=POWRDR*XORDER
  153 IF(JORDER-4)                                      154,    154,    158
  154 VATRIX(JORDER,8)=VATRIX(JORDER,8)+POWRDR*DURATN(IORDER)
  158 VATRIX(JORDER,7)=VATRIX(JORDER,7)+POWRDR
  159 CONTINUE
C     *SKIP CASES WITH LESS THAN 4 DATA POINTS
      IF(KORDER-4)                                      210,    161,    161
  161 DO  168  IORDER=1,4
      DO  168  JORDER=1,IORDER
      KORDER=IORDER+JORDER
      VATRIX(IORDER,JORDER)=VATRIX(KORDER-1,7)
  168 VATRIX(JORDER,IORDER)=VATRIX(IORDER,JORDER)
C     *INVERT NORMAL MATRIX
      NATRIX=4
      ISWTCH=2
      GO TO 11
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
  170 CONTINUE
C     *OBTAIN COEFFICIENTS AND MEASURE ERROR
C     *DESCALE THE TIME VARIABLE BY DIVIDING THE COEFFICIENTS BY NORDER.
C     *NORMALIZE THE COEFFICIENTS TO REPRESENT DATA FOR NFRQCY = 1
      ZORDER=NFRQCY
      DO    200 IORDER=1,4
      LORDER=5-IORDER
      DO    199 JORDER=1,4
      XORDER=0.
      DO    193 KORDER=1,4
  193 XORDER=XORDER+VATRIX(IORDER,KORDER)*WATRIX(KORDER,JORDER)
      IF(IORDER-JORDER)                                  196,  195,  196
  195 XORDER=XORDER-1.
  196 XORDER=ABSF(XORDER)
      IF(ERROR-XORDER)                                   198,  199,  199
  198 ERROR=XORDER
  199 VATRIX(8,LORDER)=VATRIX(8,LORDER)+WATRIX(IORDER,JORDER)
 1991                                *VATRIX(JORDER,8)
 1992                                /ZORDER
  200 ZORDER=ZORDER*YORDER
C     *PRINT
  201 FORMAT(10H          7F15.9)
  210 PRINT   201, TORDER, DURATN(NORDER), (VATRIX(8,I),I=1,4), ERROR
C     *END OF PRINCIPAL ITERATIVE LOOP
  222 CONTINUE
      CALL SYSTEM
      END
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTRX 218
      LOAD BATCH                                                        MTRX 219
      REMARK  THE CLOCK INTERROGATICN PROGRAM IS LOADED HERE            MTRX 220
      TRA                                                               MTRX 221
    DATA      2   20  100                                               MTRX 222
      REMARK  END OF RUN                                                MTRX 223
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTRX 224
       JOB     00655,AS,PROGRAMMER,                        (BY 2) SEPT64MTRX 225
       FOR                                                              MTRX 226
C     *SYMMETRIC MATRIX INVERSICN BY PARTITION, WITH LEAST SQUARES FIT TOMTRX 227
C     *THE CUBIC POLYNOMIAL FOR INVERSION TIME IN TERMS OF MATRIX ORDER. MTRX 228
C     *FOR CERTAIN ORDERS NORDER=1,2,3... TO A PRESCRIBED MAXIMUM MORDER,MTRX 229
C     *OMITTING ALL ORDERS NORDER NOT DIVISIBLE BY A PRESCRIBED INTERV,  MTRX 230
C     *INVERT THE (POSITIVE DEFINITE) SYMMETRIC MATRIX DEFINED AS FOLLOWSMTRX 231
C     *A(I,J) = SIN((I*J)/NORDER) + THE DIAGONAL MATRIX L(I,I) = NORDER+1MTRX 232
C     *A PRESCRIBED NUMBER NFRQCY OF TIMES, INTERNALLY MEASURING THE     MTRX 233
C     *TOTAL DURATION OF ALL NFRQCY INVERSIONS FOR EACH NORDER.          MTRX 234
C     *USING THE MATRIX INVERSICN PROGRAM FOR THE LEAST SQUARES FITTING, MTRX 235
C     *FOR EACH NORDER ABOVE 3, USE LEAST SQUARES TO FIT A CUBIC TO THE  MTRX 236
C     *INTERNALLY MEASURED DURATION TIMES FOR ALL ORDERS UP TO NORDER,   MTRX 237
C     *AND RECORD THE MAXIMUM ERROR (DEPARTURE FROM IDENTITY MATRIX) IN  MTRX 238
C     *POSTMULTIPLYING EACH LEAST SQUARES NORMAL MATRIX BY ITS INVERSE.  MTRX 239
C     *FINALLY, PRINT ALL DATA APPLICABLE TO EACH NORDER, SPECIFICALLY-  MTRX 240
C     *NORDER, TIME, COEFFICIENTS (CUBIC,SQUARE,LINEAR,CONSTANT), ERROR, MTRX 241
C     *EMPLOYING (FAKED) NEGATIVE TIME TO MARK INABILITY TO GET SOLUTION.MTRX 242
C     *LMTDIM MUST RELATE TO SEVERAL DIMENSIONS AND LIMITS MORDER.       MTRX 243
C     *LMTDIM=100                                                        MTRX 244
C     *BOTH DIMENSIONS EQUAL LMTDIM                                      MTRX 245
      DIMENSION WATRIX(100,100)                                          MTRX 246
      DIMENSION VATRIX(100,100)                                          MTRX 247
C     *ONE DIMENSION EQUAL TO LMTDIM                                     MTRX 248
      DIMENSION DURATN(100)                                              MTRX 249
C     *DATA INPUT                                                        MTRX 250
C     *READ THE PARAMETERS INTERV,MORDER,NFRQCY FROM A PUNCHED CARD      MTRX 251
C     *INTERV FROM COLUMNS 11-15 IN INTEGER FORMAT, AND                  MTRX 252
C     *MORDER FROM COLUMNS 16-20 IN INTEGER FORMAT, AND                  MTRX 253
C     *NFRQCY FROM COLUMNS 21-25 IN INTEGER FORMAT.                      MTRX 254
    4 FORMAT(10H          3I5)                                           MTRX 255
      READ  4,                               INTERV, MORDER, NFRQCYMTRX 256
      PRINT 4,                               INTERV, MORDER, NFRQCYMTRX 257
      GO TO 110                                                          MTRX 258
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTRX 259
C     *SYMMETRIC MATRIX INVERSION BY PARTITION, RECURSIVE ENLARGEMENT = 2MTRX 260
C     *(AAABHH) (WWWXOO) (IOOOOO) (WWWXOO) (AAABHH)                      MTRX 261
C     *(AAABHH) (WWWXOO) (OIOOOO) (WWWXOO) (AAABHH)  DEVELOP  (1) AW+BY=IMTRX 262
C     *(AAABHH).(WWWXOO)=(OOIOOO)=(WWWXOO).(AAABHH)    THE    (2) AX+BZ=OMTRX 263
C     *(CCCDHH) (YYYZOO) (OOOIOO) (YYYZOO) (CCCDHH)   BASIC   (3) YA+ZC=OMTRX 264
C     *(HHHHHH) (OOOOOO) (OOOOOO) (OOOOOO) (HHHHHH) EQUATIONS (4) YB+ZD=IMTRX 265
C     *(HHHHHH) (OOOOOO) (OOOOOO) (OOOOOO) (HHHHHH)                      MTRX 266
C     *THE SLASH (/), USED AS A PREFIX, HERE DENOTES INVERSION           MTRX 267
C     *DEFINING E=/AB                                                    MTRX 268
C     *     AND F=C/A                                                    MTRX 269
C     *YIELDS 1 Z=/(D-FB)=/(D-CE)           WHERE SYMMETRY APPLIES FOR MTRX 270
C     *   AND 2 X=-EZ                                       A,D,W,Z      MTRX 271
C     *   AND 3 Y=-ZF                   AND WHERE MUTUAL TRANSPOSES AREMTRX 272
C     *   AND 4 W=/A-EY=/A-XF                       (B,C),(E,F),(X,Y)    MTRX 273
C     *HENCE THE COMPUTATIONAL PROCEDURE TO DEVELOP (WXYZ) FROM (ABCD) ISMTRX 274
C     *1. INITIALLY COMPUTE /A FROM A, AND THEN REITERATE STEPS 2-8      MTRX 275
C     *2. MOVE D TO Z                                                    MTRX 276
C     *3. COMPUTE F=C/A AND STORE AT Y                                   MTRX 277
C     *4. COMPUTE Q=(/Z)=D-FB AT Z                                       MTRX 278
C     *5. INVERT  Z=/Q IN PLACE       (INVERT ONE HALF, COPY OTHER HALF)MTRX 279
C     *6. COMPUTE X=-EZ                                                  MTRX 280
C     *7. COMPUTE W=/A-XF            (COMPUTE ONE HALF, COPY OTHER HALF)MTRX 281
C     *8. COPY Y FROM X, WHENCE (WXYZ) IS /A FOR NEXT ITERATION          MTRX 282
C     *RESTRICT ALL PARTITIONS D,Z TO ORDER 2 BY INITIAL PARTITION OF A,WMTRX 283
C     *AT ORDER 1-OR-2 AS ORDER OF TOTAL MATRIX IS ODD-OR-EVEN, THEREUPONMTRX 284
C     *INCREASING ORDER OF A,W BY 2 FOR EACH REITERATION OF STEPS 2-8.   MTRX 285
C     *BEGIN AND END WITH MATRIX VATRIX(-,-) OF ORDER NATRIX             MTRX 286
C     *END INVERSION WITH MATRIX WATRIX(-,-) OF ORDER NATRIX             MTRX 287
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTRX 288
   11 CONTINUE                                                          MTRX 289
C     *MATRIX=1,2 AS NATRIX=ODD,EVEN                                    MTRX 290
      MATRIX=2-(NATRIX-(NATRIX/2)*2)                                    MTRX 291
C     *MAIN LOOP - EXECUTE (N+2-M)/2 TIMES, ADVANCING INDEX KATRIX BY 2 MTRX 292
      IF(NATRIX)                                      92,    92,    28MTRX 293
   28 DO     90 KATRIX=MATRIX,NATRIX,2                                  MTRX 294
C     *OBSERVE KATRIX IS ORDER OF (ABCD),(WXYZ). LATRIX IS ORDER OF A,W MTRX 295
      LATRIX=KATRIX-2                                                   MTRX 296
C     *MOVE D TO Z (A TO W FOR FIRST PASS (AND INVERT A IF KATRIX=1))   MTRX 297
      DO     38 MMTRIX=1,2                                              MTRX 298
      KMTRIX=KATRIX+1-MMTRIX                                            MTRX 299
      DO     38 NNTRIX=1,2                                              MTRX 300
      KNTRIX=KATRIX+1-NNTRIX                                            MTRX 301
C     *ORDER OF MOVEMENT IS (K,K),(K,K-1),(K-1,K),(K-1,K-1)   (K=KATRIX)MTRX 302
C     *SEPARATE EXTERRANEOUS CASES (OTHER THAN K,K FOR KATRIX=1)        MTRX 303
      IF(KMTRIX*KNTRIX)                               90,    35,    38MTRX 304
C     *INVERT A FOR KATRIX=1, THEN SKIP TO END OF LOOP                  MTRX 305
   35 IF(WATRIX(1,1))                                 36,    92,    36MTRX 306
   36 WATRIX(1,1)=1./WATRIX(1,1)                                        MTRX 307
      GO TO     90                                                      MTRX 308
C     *MOVE ONE ELEMENT OF D TO Z                                       MTRX 309
   38 WATRIX(KMTRIX,KNTRIX)= VATRIX(KMTRIX,KNTRIX)                      MTRX 310
C     *SEPARATE CASE KATRIX=2 FOR INVERSION OF A (SKIP FORMING F,Z)     MTRX 311
   39 IF(LATRIX)                                      90,    51,    40MTRX 312
C     *COMPUTE F=C/A AND Q=D-FB                                         MTRX 313
   40 DO     49 IATRIX=1,LATRIX                                         MTRX 314
      DO     49 MMTRIX=1,2                                              MTRX 315
      KMTRIX=KATRIX+1-MMTRIX                                            MTRX 316
      WATRIX(KMTRIX,IATRIX)=0.                                          MTRX 317
C     *ACCUMULATE ONE ELEMENT OF F=C/A                                  MTRX 318
      DO     45 JATRIX=1,LATRIX                                         MTRX 319
   45 WATRIX(KMTRIX,IATRIX)= WATRIX(KMTRIX,IATRIX)+VATRIX(KMTRIX,JATRIX)MTRX 320
  451                                            *WATRIX(JATRIX,IATRIX)MTRX 321
C     *FORM Q=D-FB BY CONCURRENT ACCUMULATION OVER ENTIRE D             MTRX 322
      DO     49 NNTRIX=1,2                                              MTRX 323
      KNTRIX=LATRIX+NNTRIX                                              MTRX 324
C     *ORDER OF ACCUMULATION IS (K,K-1),(K,K),(K-1,K-1),(K-1,K)         MTRX 325
C     *SEPARATE CASE FOR UPPER RIGHT CORNER                             MTRX 326
      IF(KMTRIX-KNTRIX)                               48,    49,    49MTRX 327
C     *SEPARATE CASE FOR RECOVERY BY SYMMETRY ON LAST ITERATION         MTRX 328
   48 IF(LATRIX-IATRIX)                               90,    50,    49MTRX 329
   49 WATRIX(KMTRIX,KNTRIX)= WATRIX(KMTRIX,KNTRIX)-WATRIX(KMTRIX,IATRIX)MTRX 330
  491                                            *VATRIX(IATRIX,KNTRIX)MTRX 331
C     *FINISH Q BY SYMMETRY                                             MTRX 332
   50 WATRIX(LATRIX+1,KATRIX)=WATRIX(KATRIX,KATRIX-1)                   MTRX 333
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTRX 334
C    *INVERT Q (INVERT D=A FOR KATRIX=2) BY ADJOINTS                    MTRX 335
C    *PREPARE INVERSION BY INTERCHANGING DIAGONAL ELEMENTS THRU (K,K-1) MTRX 336
   51 WATRIX(KATRIX  ,KATRIX-1)=WATRIX(KATRIX  ,KATRIX  )               MTRX 337
      WATRIX(KATRIX  ,KATRIX  )=WATRIX(KATRIX-1,KATRIX-1)               MTRX 338
      WATRIX(KATRIX-1,KATRIX-1)=WATRIX(KATRIX  ,KATRIX-1)               MTRX 339
C    *STORE DETERMINANT IN LOWER LEFT CORNER AND RECOVER BY SYMMETRY    MTRX 340
C    *COMPUTE DETERMINANT AS ((K-1,K-1)(K,K)-((K-1,K)(K-1,K)-(0)))      MTRX 341
      WATRIX(KATRIX,KATRIX-1)=0.                                        MTRX 342
      DO    58 NNTRIX=1,2                                               MTRX 343
      IATRIX=KATRIX-(2-NNTRIX)                                          MTRX 344
      JATRIX=KATRIX+(1-NNTRIX)                                          MTRX 345
   58 WATRIX(KATRIX,KATRIX-1)= WATRIX(KATRIX-1,JATRIX)                  MTRX 346
  581                         *WATRIX(IATRIX,KATRIX)                    MTRX 347
  582                         -WATRIX(KATRIX,KATRIX-1)                  MTRX 348
C    *COMPUTE INVERSE (EXCEPT (K,K-1))                                  MTRX 349
      IF(WATRIX(KATRIX,KATRIX-1))                        60,   92,   60MTRX 350
   60 DO    66 MMTRIX=1,2                                               MTRX 351
      KMTRIX=LATRIX+MMTRIX                                              MTRX 352
      DO    66 NNTRIX=1,2                                               MTRX 353
      KNTRIX=KATRIX+1-NNTRIX                                            MTRX 354
C    *SEPARATE CASES FOR NEGATION FOR (K-1,K) AND RECOVERY FOR (K,K-1)  MTRX 355
      IF(KMTRIX-KNTRIX)                                  65,   66,   67MTRX 356
   65 WATRIX(KMTRIX,KNTRIX)=-WATRIX(KMTRIX,KNTRIX)                      MTRX 357
   66 WATRIX(KMTRIX,KNTRIX)= WATRIX(KMTRIX,KNTRIX)                      MTRX 358
  661                      /WATRIX(KATRIX,KATRIX-1)                     MTRX 359
C    *RECOVER BY SYMMETRY                                               MTRX 360
   67 WATRIX(KATRIX,KATRIX-1)=WATRIX(KATRIX-1,KATRIX)                   MTRX 361
C    *SEPARATE CASE, SKIP TO END OF LOOP AFTER INVERTING A FOR KATRIX=2 MTRX 362
   69 IF(LATRIX)                                         90,   90,   70MTRX 363
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
C     *COMPUTE X=-EZ BY TRANSPOSING Y=-ZF
   70 DO      77 IATRIX=1,LATRIX
      DO      77 NNTRIX=1,2
      KNTRIX=KATRIX+1-NNTRIX
      WATRIX(IATRIX,KNTRIX)=0.
      DO      77 MMTRIX=1,2
      KMTRIX=LATRIX+MMTRIX
   77 WATRIX(IATRIX,KNTRIX)= WATRIX(IATRIX,KNTRIX)-WATRIX(KNTRIX,KMTRIX)
   771                                          *WATRIX(KMTRIX,IATRIX)
C     *COMPUTE W=/A-XF AND FORM Y AS TRANSPOSE OF X
      DO      88 IATRIX=1,LATRIX
      DO      85 JATRIX=IATRIX,LATRIX
      DO      84 NNTRIX=1,2
      KNTRIX=KATRIX+1-NNTRIX
C     *ACCUMULATE -XF INTO /A, ONLY ALONG AND ABOVE MAJOR DIAGONAL
   84 WATRIX(IATRIX,JATRIX)= WATRIX(IATRIX,JATRIX)-WATRIX(IATRIX,KNTRIX)
   841                                          *WATRIX(KNTRIX,JATRIX)
C     *COMPLETE W BY SYMMETRY
   85 WATRIX(JATRIX,IATRIX)= WATRIX(IATRIX,JATRIX)
      DO      88 NNTRIX=1,2
      KNTRIX=KATRIX+1-NNTRIX
C     *TRANSPOSE ONE ELEMENT OF X TO Y
   88 WATRIX(KNTRIX,IATRIX)= WATRIX(IATRIX,KNTRIX)
C     *END OF MAIN LOOP
   90 CONTINUE
      GO TO (  127,  170), ISWTCH
C     *ALARMS
   92 DURATN(NORDER)=-1.
      IF(ISWTCH-2)                                130,   210,   210
C     *END OF MATRIX INVERSION PROGRAMMING
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
  110 CONTINUE
C     *PRINCIPAL ITERATIVE LOOP ON NORDER=1,2,3...MORDER
      DO  222  NORDER=1,MORDER
      TORDER=NORDER
C     *CLEAR DURATN TABLE AND SET ERROR RECORD = 0.
      ERROR=0.
      DURATN(NORDER)=-TORDER
C     *SKIP CASES FOR WHICH NORDER IS NOT DIVISIBLE BY INTERV
      IF(NORDER-(NORDER/INTERV)*INTERV)                 222,   115,   222
  115 DURATN(NORDER)=0.
C     *CREATE A(I,J)
      DO    119 IORDER=1,NORDER
      DO    119 JORDER=1,IORDER
      KORDER=IORDER*JORDER
      VATRIX(IORDER,JORDER)=KORDER
      VATRIX(IORDER,JORDER)=SINF(VATRIX(IORDER,JORDER)/TORDER)
      IF(IORDER-JORDER)                                 119,   118,   119
  118 VATRIX(IORDER,JORDER)=VATRIX(IORDER,JORDER)+TORDER+1.
  119 VATRIX(JORDER,IORDER)=VATRIX(IORDER,JORDER)
C     *INVERSION LOOP SETUP
      NATRIX=NORDER
      ISWTCH=1
C     *CLOCK ON     INTERROGATE CLOCK HERE
      TIMEON=CLOCKF(DUMMY)
C     *LOOP NFRQCY ITERATIONS
      DO    127 NITRTN=1,NFRQCY
      GO TO 11
  127 CONTINUE
C     *CLOCK OFF    INTERROGATE CLOCK HERE
      TIMEUP=CLOCKF(DUMMY)
C     *COMPUTE DURATION  DURATN(NORDER) = CLOCK SPAN
      DURATN(NORDER)=TIMEUP-TIMEON
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTRX 428
C     *LEAST SQUARES COMPUTATION.                                       MTRX 429
C     *DENOTE S(K) = SUM OF ((I**K)           ) OVER I = ORDERS EMPLOYEDMTRX 430
C     *AND    T(K) = SUM OF ((I**K)*(DURATN(I))) OVER I = ORDERS EMPLOYEDMTRX 431
C     *AND CUBIC = A(3)*(X**3) + A(2)*(X**2) + A(1)*X + A(0) WHERE THE   MTRX 432
C     *VARIABLE X DENOTES DURATION.                                     MTRX 433
C     *SOLVE      (S(0) S(1) S(2) S(3))       (A(0))       (T(0))       MTRX 434
C     *THIS       (S(1) S(2) S(3) S(4))   *   (A(1))   =   (T(1))       MTRX 435
C     *SYSTEM     (S(2) S(3) S(4) S(5))       (A(2))       (T(2))       MTRX 436
C     *USING      (S(3) S(4) S(5) S(6))       (A(3))       (T(3))       MTRX 437
C     *STORAGE    VATRIX(1,1)-VATRIX(4,4)   VATRIX(8,-)   VATRIX(-,8)   MTRX 438
C     *CREATE NORMAL MATRIX AND VECTOR AND CLEAR COEFFICIENT VECTOR     MTRX 439
  130 DO  133  IORDER=1,7                                               MTRX 440
      VATRIX(IORDER,7)=0.                                               MTRX 441
      VATRIX(IORDER,8)=0.                                               MTRX 442
  133 VATRIX(8,IORDER)=0.                                               MTRX 443
C     *USE KORDER TO COUNT VALID DATA CASES                             MTRX 444
      KORDER=0                                                          MTRX 445
      DO   159 IORDER=1,NORDER                                          MTRX 446
C     *SKIP IRRELEVANT CASES (DURATN(IORDER) NEGATIVE)                  MTRX 447
      IF(DURATN(IORDER))                             159,  142,   142MTRX 448
  142 KORDER=KORDER+1                                                   MTRX 449
C     *DENOTE POWRDR = POWERS OF ORDER VARIABLE IORDER                  MTRX 450
      POWRDR=1.                                                         MTRX 451
C     *SCALE (NORMALIZE) INDEPENDENT TIME VARIABLE BY DIVISION BY NORDER.MTRX 452
      XORDER=IORDER                                                     MTRX 453
      YORDER=NORDER                                                     MTRX 454
      XORDER=XORDER/YORDER                                              MTRX 455
      DO   158 JORDER=1,7                                               MTRX 456
      IF(JORDER-1)                                   222,  153,   152MTRX 457
  152 POWRDR=POWRDR*XORDER                                              MTRX 458
  153 IF(JORDER-4)                                   154,  154,   158MTRX 459
  154 VATRIX(JORDER,8)=VATRIX(JORDER,8)+POWRDR*DURATN(IORDER)           MTRX 460
  158 VATRIX(JORDER,7)=VATRIX(JORDER,7)+POWRDR                          MTRX 461
  159 CONTINUE                                                          MTRX 462
C     *SKIP CASES WITH LESS THAN 4 DATA POINTS                          MTRX 463
      IF(KORDER-4)                                   210,  161,   161MTRX 464
  161 DO  168  IORDER=1,4                                               MTRX 465
      DO  168  JORDER=1,IORDER                                          MTRX 466
      KORDER=IORDER+JORDER                                              MTRX 467
      VATRIX(IORDER,JORDER)=VATRIX(KORDER-1,7)                          MTRX 468
  168 VATRIX(JORDER,IORDER)=VATRIX(IORDER,JORDER)                       MTRX 469
C     *INVERT NORMAL MATRIX                                             MTRX 470
      NATRIX=4                                                          MTRX 471
      ISWTCH=2                                                          MTRX 472
      GO TO 11                                                          MTRX 473
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
  170 CONTINUE
C     *OBTAIN COEFFICIENTS AND MEASURE ERROR
C     *DESCALE THE TIME VARIABLE BY DIVIDING THE COEFFICIENTS BY NORDER.
C     *NORMALIZE THE COEFFICIENTS TO REPRESENT DATA FOR NFRQCY = 1
      ZORDER=NFRQCY
      DO    200 IORDER=1,4
      LORDER=5-IORDER
      DO    199 JORDER=1,4
      XORDER=0.
      DO    193 KORDER=1,4
  193 XORDER=XORDER+VATRIX(IORDER,KORDER)*WATRIX(KORDER,JORDER)
      IF(IORDER-JORDER)                                       196,   195,   196
  195 XORDER=XORDER-1.
  196 XORDER=ABSF(XORDER)
      IF(ERROR-XORDER)                                        198,   199,   199
  198 ERROR=XORDER
  199 VATRIX(8,LORDER)=VATRIX(8,LORDER)+WATRIX(IORDER,JORDER)
  1991                                *VATRIX(JORDER,8)
  1992                                /ZORDER
  200 ZORDER=ZORDER*YORDER
C     *PRINT
  201 FORMAT(10H          7F15.9)
  210 PRINT   201, TORDER, DURATN(NORDER), (VATRIX(8,I),I=1,4), ERROR
C     *END OF PRINCIPAL ITERATIVE LOOP
  222 CONTINUE
      CALL SYSTEM
      END
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTRX 502
         LOAD BATCH                                                     MTRX 503
         REMARK  THE CLOCK INTERROGATION PROGRAM IS LOADED HERE         MTRX 504
         TRA                                                            MTRX 505
       DATA      2   20  100                                            MTRX 506
         REMARK  END OF RUN                                             MTRX 507
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXG
       JOB     00655,AS,PROGRAMMER,                        (GENERATOR) SEPT64MTXG
       REMARK  THIS JOB GENERATES A FORTRAN PROGRAM FOR CASE 0-BY-0      MTXG
       REMARK  MAXIMUM ORDER IS ENTERED IN COLUMNS 7-8 OF DATA CARD      MTXG
       SPACE                                                             MTXG
       REMARK  THIS JOB WRITES ONE BCD FILE ON TAPE 5                    MTXG
       SPACE                                                             MTXG
       REMARK  PLEASE SAVE TAPE 5 AT END OF RUN                          MTXG
       SPACE                                                             MTXG
       REMARK  PAUSE TO PUSH START AFTER CHECKING AVAILABILITY OF TAPE 5MTXG
       SPACE                                                             MTXG
       SPACE                                                             MTXG
       PAUSE                                                             MTXG
       FOR                                                               MTXG
      DIMENSION CARD(80)                                                 MTXG
 1000 FORMAT(6A1,I2,72A1)                                                MTXG
 1001 FORMAT(80A1)                                                       MTXG
 1002 FORMAT(72A1,4H MTXI4)                                              MTXG
      NRCARD=0                                                           MTXG
      NATRIX=0                                                           MTXG
      NRTAPE=5                                                           MTXG
      REWIND                                                        NRTAPEMTXG
      READ                                                            1000MTXG
     1,FLAG,CLAG,(CARD(I),I=1,4),NATRIX,(CARD(I),I=9,80)                 MTXG
      WRITE OUTPUT TAPE NRTAPE,                                       1000MTXG
     1,FLAG,CLAG,(CARD(I),I=1,4),NATRIX,(CARD(I),I=9,80)                 MTXG
 1111 READ                                                            1001MTXG
 11111,(CARD(I),I=1,80)                                                  MTXG
      IF(CARD(1)-FLAG)                                    1121, 1112, 1121MTXG
 1112 IF(CARD(2)-FLAG)                                    1123, 1122, 1123MTXG
 1121 NRCARD=NRCARD+1                                                    MTXG
      WRITE OUTPUT TAPE NRTAPE,                                       1002MTXG
     1,(CARD(I),I=1,72),NRCARD                                           MTXG
      GO TO 1111                                                         MTXG
 1122 CONTINUE                                                           MTXG
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXG
C      INSERT PROGRAMMING LOOP                                          MTXG
   28 DO    90 KATRIX=1,NATRIX                                          MTXG
      LATRIX=KATRIX-1                                                   MTXG
      NRCARD=NRCARD+1                                                   MTXG
      WRITE OUTPUT TAPE NRTAPE,                                     9948MTXG
     1                                                           ,NRCARDMTXG
 9948                                            FORMAT(6H         66HMTXG
 99481KATRIX=KATRIX+1                                                   MTXG
 99482                                                      4H MTXI4)MTXG
      NRCARD=NRCARD+1                                                   MTXG
      WRITE OUTPUT TAPE NRTAPE,                                     9949MTXG
     1                                                           ,NRCARDMTXG
 9949                                            FORMAT(6H         66HMTXG
 99491LATRIX=KATRIX-1                                                   MTXG
 99492                                                      4H MTXI4)MTXG
      NRCARD=NRCARD+1                                                   MTXG
      WRITE OUTPUT TAPE NRTAPE,                                     9900MTXG
     1                                                           ,NRCARDMTXG
 9900                                            FORMAT(6HC       •66HMTXG
 99001MOVE D TO Z (A TO W FOR FIRST PASS (AND INVERT A IF KATRIX=1))    MTXG
 99002                                                      4H MTXI4)MTXG
      NRCARD=NRCARD+1                                                   MTXG
      WRITE OUTPUT TAPE NRTAPE,                                     9950MTXG
     1,KATRIX,KATRIX,KATRIX,KATRIX                                      MTXG
     2                                                           ,NRCARDMTXG
 9950                                            FORMAT(6H          7HMTXG
 99501WATRIX(I2, 1H,I2,10H)= VATRIX(I2, 1H,I2, 1H)                      MTXG
 9950238H                                                   4H MTXI4)MTXG
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXG  65
C     *SEPARATE CASE KATRIX=1 FOR INVERSION OF A (SKIP FORMING F,Z)     MTXG  66
   39 IF(LATRIX)                                          90,    51,    40MTXG  67
   40 CONTINUE                                                          MTXG  68
      NRCARD=NRCARD+1                                                   MTXG  69
      WRITE OUTPUT TAPE NRTAPE,                                     9901MTXG  70
     1                                                           ,NRCARDMTXG  71
 9901                                           FORMAT(6HC      *66HMTXG  72
 99011COMPUTE F=C/A AND Q=D-FB                                          MTXG  73
 99012                                                      4H MTXI4)MTXG  74
      DO     49 IATRIX=1,LATRIX                                         MTXG  75
      NRCARD=NRCARD+1                                                   MTXG  76
      WRITE OUTPUT TAPE NRTAPE,                                     9951MTXG  77
     1,KATRIX,IATRIX                                                    MTXG  78
     2                                                           ,NRCARDMTXG  79
 9951                                           FORMAT(6H           7HMTXG  80
 99511WATRIX(I2, 1H,I2, 4H)=0.                                          MTXG  81
 9951250H                                                   4H MTXI4)MTXG  82
      NRCARD=NRCARD+1                                                   MTXG  83
      WRITE OUTPUT TAPE NRTAPE,                                     9902MTXG  84
     1                                                           ,NRCARDMTXG  85
 9902                                           FORMAT(6HC      *66HMTXG  86
 99021ACCUMULATE ONE ELEMENT OF F=C/A                                   MTXG  87
 99022                                                      4H MTXI4)MTXG  88
      DO     45 JATRIX=1,LATRIX                                         MTXG  89
      NRCARD=NRCARD+1                                                   MTXG  90
      WRITE OUTPUT TAPE NRTAPE,                                     9952MTXG  91
     1,KATRIX,IATRIX,KATRIX,IATRIX,KATRIX,JATRIX,JATRIX,IATRIX          MTXG  92
     2                                                           ,NRCARDMTXG  93
 9952                                           FORMAT(6H           7HMTXG  94
 99521WATRIX(I2, 1H,I2,10H)= WATRIX(I2, 1H,I2, 9H)+VATRIX(I2, 1H,I2, 9H)MTXG  95
 99522*WATRIX(I2,1H,I2,1H)                                              MTXG  96
 9952310H                                                   4H MTXI4)MTXG  97
   45 CONTINUE                                                          MTXG  98
      NRCARD=NRCARD+1                                                   MTXG  99
      WRITE OUTPUT TAPE NRTAPE,                                     9903MTXG 100
     1                                                           ,NRCARDMTXG 101
 9903                                           FORMAT(6HC      *66HMTXG 102
 99031FORM Q=D-FB BY ACCUMULATION TO D                                  MTXG 103
 99032                                                      4H MTXI4)MTXG 104
      NRCARD=NRCARD+1                                                   MTXG 105
      WRITE OUTPUT TAPE NRTAPE,                                     9953MTXG 106
     1,KATRIX,KATRIX,KATRIX,KATRIX,KATRIX,IATRIX,IATRIX,KATRIX          MTXG 107
     2                                                           ,NRCARDMTXG 108
 9953                                           FORMAT(6H           7HMTXG 109
 99531WATRIX(I2, 1H,I2,10H)= WATRIX(I2, 1H,I2, 9H)-WATRIX(I2, 1H,I2, 9H)MTXG 110
 99532*VATRIX(I2,1H,I2,1H)                                              MTXG 111
 9953310H                                                   4H MTXI4)MTXG 112
   49 CONTINUE                                                          MTXG 113
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXG 114
   51 CONTINUE                                                          MTXG 115
      NRCARD=NRCARD+1                                                   MTXG 116
      WRITE OUTPUT TAPE NRTAPE,                                     9904MTXG 117
     1                                                           ,NRCARDMTXG 118
 9904                                              FORMAT(6HC     *66HMTXG 119
 99041INVERT Q (INVERT D=A FOR KATRIX=1)                                MTXG 120
 99042                                                       4H MTXI4)MTXG 121
      KKTRIX=1000+KATRIX                                                MTXG 122
      NRCARD=NRCARD+1                                                   MTXG 123
      WRITE OUTPUT TAPE NRTAPE,                                     9954MTXG 124
     1,KATRIX,KATRIX,KKTRIX,KKTRIX                                      MTXG 125
     2                                                           ,NRCARDMTXG 126
 9954                                              FORMAT(6H        10HMTXG 127
 99541IF(WATRIX(I2, 1H,I2,34H))                                         MTXG 128
 99542                                                  I5,7H,   92,I5MTXG 129
 99543,                                                      4H MTXI4)MTXG 130
      NRCARD=NRCARD+1                                                   MTXG 131
      WRITE OUTPUT TAPE NRTAPE,                                     9955MTXG 132
     1,KKTRIX,KATRIX,KATRIX,KATRIX,KATRIX                               MTXG 133
     2                                                           ,NRCARDMTXG 134
 9955                                              FORMAT(I5,       8H MTXG 135
 99551WATRIX(I2, 1H,I2,12H)=1./WATRIX(I2, 1H,[illegible]               MTXG 136
 9955236H                                                    4H MTXI4)MTXG 137
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
      NRCARD=NRCARD+1
      WRITE OUTPUT TAPE NRTAPE,                                         9906
     1                                                                ,NRCARD
 9906                                            FORMAT(6HC      •66H
 99061COMPUTE W=/A-XF AND FORM Y AS TRANSPOSE OF X
 99062                                                        4H MTXI4)
      DO     88 IATRIX=1,LATRIX
      NRCARD=NRCARD+1
      WRITE OUTPUT TAPE NRTAPE,                                         9907
     1                                                                ,NRCARD
 9907                                            FORMAT(6HC      •66H
 99071ACCUMULATE -XF INTO /A, ONLY ALONG AND ABOVE MAJOR DIAGONAL
 99072                                                        4H MTXI4)
      NRCARD=NRCARD+1
      WRITE OUTPUT TAPE NRTAPE,                                         9908
     1                                                                ,NRCARD
 9908                                            FORMAT(6HC      •66H
 99081COMPLETE W BY SYMMETRY
 99082                                                        4H MTXI4)
      DO     85 JATRIX=IATRIX,LATRIX
      NRCARD=NRCARD+1
      WRITE OUTPUT TAPE NRTAPE,                                         9957
     1,IATRIX,JATRIX,IATRIX,JATRIX,IATRIX,KATRIX,KATRIX,JATRIX
     2                                                                ,NRCARD
 9957                                            FORMAT(6H         7H
 99571WATRIX(I2, 1H,I2,10H)= WATRIX(I2, 1H,I2, 9H)-WATRIX(I2, 1H,I2, 9H)
 99572*WATRIX(I2,1H,I2,1H)
 9957310H                                                     4H MTXI4)
      NRCARD=NRCARD+1
      WRITE OUTPUT TAPE NRTAPE,                                         9958
     1,JATRIX,IATRIX,IATRIX,JATRIX
     2                                                                ,NRCARD
 9958                                            FORMAT(6H         7H
 99581WATRIX(I2, 1H,I2,10H)= WATRIX(I2, 1H,I2, 1H)
 9958238H                                                     4H MTXI4)
   85 CONTINUE
      NRCARD=NRCARD+1
      WRITE OUTPUT TAPE NRTAPE,                                         9909
     1                                                                ,NRCARD
 9909                                            FORMAT(6HC      •66H
 99091TRANSPOSE ONE ELEMENT OF X TO Y
 99092                                                        4H MTXI4)
      NRCARD=NRCARD+1
      WRITE OUTPUT TAPE NRTAPE,                                         9959
     1,KATRIX,IATRIX,IATRIX,KATRIX
     2                                                                ,NRCARD
 9959                                            FORMAT(6H         7H
 99591WATRIX(I2, 1H,I2,10H)= WATRIX(I2, 1H,I2, 1H)
 9959238H                                                     4H MTXI4)
   88 CONTINUE
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXG 138
C    *SEPARATE CASE, SKIP TO END OF LOOP AFTER INVERTING A FOR KATRIX=1 MTXG 139
   69 IF(LATRIX)                                         89,    89,    70MTXG 140
   70 CONTINUE                                                           MTXG 141
      NRCARD=NRCARD+1                                                    MTXG 142
      WRITE OUTPUT TAPE NRTAPE,                                      9905MTXG 143
     1                                                           ,NRCARDMTXG 144
 9905                                              FORMAT(6HC      *66HMTXG 145
 99051COMPUTE X=-EZ BY TRANSPOSING Y=-ZF                                 MTXG 146
 99052                                                       4H MTXI4)MTXG 147
      DO    77 IATRIX=1,LATRIX                                           MTXG 148
      NRCARD=NRCARD+1                                                    MTXG 149
      WRITE OUTPUT TAPE NRTAPE,                                      9956MTXG 150
     1,IATRIX,KATRIX,KATRIX,KATRIX,KATRIX,IATRIX                         MTXG 151
     2                                                           ,NRCARDMTXG 152
 9956                                              FORMAT(6H          7HMTXG 153
 99561WATRIX(I2, 1H,I2,10H)=-WATRIX(I2, 1H,I2, 9H)*WATRIX(I2, 1H,I2,25H)MTXG 154
 99562                                                       4H MTXI4)MTXG 155
   77 CONTINUE                                                           MTXG 156
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXG 208
   89 CONTINUE                                                          MTXG 209
      KKKRIX=1100+KATRIX                                                MTXG 210
      NRCARD=NRCARD+1                                                   MTXG 211
      WRITE OUTPUT TAPE NRTAPE,                                     9960MTXG 212
     1                                                    ,KKKRIX,NRCARDMTXG 213
 9960                                              FORMAT(6H       48HMTXG 214
 99601IF(KATRIX-NATRIX)                            I6,12H,    90,    90MTXG 215
 99602                                                         4H MTXI4)MTXG 216
      NRCARD=NRCARD+1                                                   MTXG 217
      WRITE OUTPUT TAPE NRTAPE,                                     9965MTXG 218
     1,KKKRIX                                                           MTXG 219
     2                                                           ,NRCARDMTXG 220
 9965                                              FORMAT(I5,     67H MTXG 221
 99651CONTINUE                                                          MTXG 222
 99652                                                         4H MTXI4)MTXG 223
      NRCARD=NRCARD+1                                                   MTXG 224
      WRITE OUTPUT TAPE NRTAPE,                                     9966MTXG 225
     1                                                           ,NRCARDMTXG 226
 9966                                              FORMAT(6HC /*/ 66HMTXG 227
 99661SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXG 228
 99662                                                         4H MTXI4)MTXG 229
   90 CONTINUE                                                          MTXG 230
      GO TO 1111                                                        MTXG 231
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXG 232
C     *191 STATEMENTS AND COMMENTS ARE GENERATED BY NONLOOP ACTION       MTXG 233
C     *NUMBER OF STATEMENTS AND COMMENTS GENERATED BY LOOP IS GIVEN BY   MTXG 234
C     *(2*N*N*N +12*N*N +25*N -9)/3, WHICH IS TABULATED BELOW,           MTXG 235
C     *ACCOMPANIED BY THE VALUES OF THE DATA STORAGE NEEDS (2*N*N)       MTXG 236
C      N   STMTS    D1    D2 D3       2*N*N                              MTXG 237
C      1      10    25    16  4           2                              MTXG 238
C      2      35    41    20  4           8                              MTXG 239
C      3      76    61    24  4          18                              MTXG 240
C      4     137    85    28  4          32                              MTXG 241
C      5     222   113    32  4          50                              MTXG 242
C      6     335   145    36  4          72                              MTXG 243
C      7     480   181    40  4          98                              MTXG 244
C      8     661   221    44  4         128                              MTXG 245
C      9     882   265    48  4         162                              MTXG 246
C     10    1147   313    52  4         200                              MTXG 247
C     11    1460   365    56  4         242                              MTXG 248
C     12    1825   421    60  4         288                              MTXG 249
C     13    2246   481    64  4         338                              MTXG 250
C     14    2727   545    68  4         392                              MTXG 251
C     15    3272   613    72  4         450                              MTXG 252
C     16    3885   685    76  4         512                              MTXG 253
C     17    4570   761    80  4         578                              MTXG 254
C     18    5331   841    84  4         648                              MTXG 255
C     19    6172   925    88  4         722                              MTXG 256
C     20    7097  1013    92  4         800                              MTXG 257
C     21    8110  1105    96  4         882                              MTXG 258
C     22    9215  1201   100  4         968                              MTXG 259
C     23   10416  1301   104           1058                              MTXG 260
C     24   11717  1405                 1152                              MTXG 261
C     25   13122                       1250                              MTXG 262
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
 1123 CONTINUE
C     *ADJUSTMENT OF LMTDIM TO EQUAL NATRIX AS READ ON FLAG CARD
      IF(CARD(2)-CLAG)                                1140, 1124, 1140
 1124 DO  1125 J=1,4
      IF(CARD(3)-CARD(J))                             1125, 1126, 1125
 1125 CONTINUE
 1126 NRCARD=NRCARD+1
      DO  1127 L=1,3
      K=4-L
      CARD(K  )=CARD(K+1)
 1127 CARD(K+1)=CARD(K+2)
      GO TO (1131,1130,1132),J
 1130 WRITE OUTPUT TAPE NRTAPE,                                   8800
 11301,(CARD(I),I=1,13),NATRIX,(CARD(I),I=17,72)                ,NRCARD
      GO TO 1111
 1131 WRITE OUTPUT TAPE NRTAPE,                                   8801
 11311,(CARD(I),I=1,23),NATRIX,(CARD(I),I=27,72)                ,NRCARD
      GO TO 1111
 1132 WRITE OUTPUT TAPE NRTAPE,                                   8802
 11321,(CARD(I),I=1,23),NATRIX,CARD(27),NATRIX,(CARD(I),I=31,72) ,NRCARD
      GO TO 1111
 8800 FORMAT(13A1,I3,56A1                                     ,4H MTXI4)
 8801 FORMAT(23A1,I3,46A1                                     ,4H MTXI4)
 8802 FORMAT(23A1,I3,A1,I3,42A1                               ,4H MTXI4)
 1140 CONTINUE
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXG 289
      END FILE                                                    NRTAPEMTXG 290
      UNLOAD                                                      NRTAPEMTXG 291
      CALL SYSTEM                                                       MTXG 292
      END                                                               MTXG 293
       LOAD BATCH                                                       MTXG 294
       TRA                                                              MTXG 295
FC     5       DATA DECK FLAG CARD                                      MTXG 296
       JOB     00655,AS,PROGRAMMER,                        (BY O) SEPT64MTXG 297
       FOR                                                              MTXG 298
C     *SYMMETRIC MATRIX INVERSION BY PARTITION, WITH LEAST SQUARES FIT OFMTXG 299
C     *THE CUBIC POLYNOMIAL FOR INVERSION TIME IN TERMS OF MATRIX ORDER. MTXG 300
C     *FOR CERTAIN ORDERS NORDER=1,2,3... TO A PRESCRIBED MAXIMUM MORDER,MTXG 301
C     *OMITTING ALL ORDERS NORDER NOT DIVISIBLE BY A PRESCRIBED INTERV,  MTXG 302
C     *INVERT THE (POSITIVE DEFINITE) SYMMETRIC MATRIX DEFINED AS FOLLOWSMTXG 303
C     *A(I,J) = SIN((I*J)/NORDER) + THE DIAGONAL MATRIX L(I,I) = NORDER+1MTXG 304
C     *A PRESCRIBED NUMBER NFRQCY OF TIMES, INTERNALLY MEASURING THE     MTXG 305
C     *TOTAL DURATION OF ALL NFRQCY INVERSIONS FOR EACH NORDER.          MTXG 306
C     *USING THE MATRIX INVERSION PROGRAM FOR THE LEAST SQUARES FITTING, MTXG 307
C     *FOR EACH NORDER ABOVE 3, USE LEAST SQUARES TO FIT A CUBIC TO THE  MTXG 308
C     *INTERNALLY MEASURED DURATION TIMES FOR ALL ORDERS UP TO NORDER,   MTXG 309
C     *AND RECORD THE MAXIMUM ERROR (DEPARTURE FROM IDENTITY MATRIX) IN  MTXG 310
C     *POSTMULTIPLYING EACH LEAST SQUARES NORMAL MATRIX BY ITS INVERSE.  MTXG 311
C     *FINALLY, PRINT ALL DATA APPLICABLE TO EACH NORDER, SPECIFICALLY-  MTXG 312
C     *NORDER, TIME, COEFFICIENTS (CUBIC,SQUARE,LINEAR,CONSTANT), ERROR, MTXG 313
C     *EMPLOYING (FAKED) NEGATIVE TIME TO MARK INABILITY TO GET SOLUTION.MTXG 314
       DIMENSION COEFFT(4)                                              MTXG 315
       DIMENSION WTDPOW(4)                                              MTXG 316
       DIMENSION POWERS(7)                                              MTXG 317
C     *LMTDIM MUST RELATE TO SEVERAL DIMENSIONS AND LIMITS MORDER.       MTXG 318
FCCC  *LMTDIM=100                                                       MTXG 319
C     *BOTH DIMENSIONS EQUAL LMTDIM                                     MTXG 320
FCD    DIMENSION WATRIX(100,100)                                        MTXG 321
FCD    DIMENSION VATRIX(100,100)                                        MTXG 322
C     *ONE DIMENSION EQUAL TO LMTDIM                                    MTXG 323
FCF    DIMENSION DURATN(100)                                            MTXG 324
C     *DATA INPUT                                                       MTXG 325
C     *READ THE PARAMETERS INTERV,MORDER,NFRQCY FROM A PUNCHED CARD     MTXG 326
C     *INTERV FROM COLUMNS 11-15 IN INTEGER FORMAT, AND                 MTXG 327
C     *MORDER FROM COLUMNS 16-20 IN INTEGER FORMAT, AND                 MTXG 328
C     *NFRQCY FROM COLUMNS 21-25 IN INTEGER FORMAT.                     MTXG 329
     4 FORMAT(10H          3I5)                                         MTXG 330
       READ  4,                                   INTERV, MORDER, NFRQCYMTXG 331
       PRINT 4,                                   INTERV, MORDER, NFRQCYMTXG 332
       GO TO 110                                                        MTXG 333
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXG 334
C     *SYMMETRIC MATRIX INVERSION BY PARTITION, BY NONLOOPING PROGRAMMINGMTXG 335
C     *(AAABHH) (WWWXOO) (IOOOOO) (WWWXOO) (AAABHH)                     MTXG 336
C     *(AAABHH) (WWWXOO) (OIOOOO) (WWWXOO) (AAABHH)  DEVELOP  (1) AW+BY=IMTXG 337
C     *(AAABHH).(WWWXOO)=(OOIOOO)=(WWWXOO).(AAABHH)    THE    (2) AX+BZ=OMTXG 338
C     *(CCCDHH) (YYYZOO) (OOOIOO) (YYYZOO) (CCCDHH)   BASIC   (3) YA+ZC=OMTXG 339
C     *(HHHHHH) (OOOOOO) (OOOOOO) (OOOOOO) (HHHHHH) EQUATIONS (4) YB+ZD=IMTXG 340
C     *(HHHHHH) (OOOOOO) (OOOOOO) (OOOOOO) (HHHHHH)                     MTXG 341
C     *THE SLASH (/), USED AS A PREFIX, HERE DENOTES INVERSION          MTXG 342
C     *DEFINING E=/AB                                                   MTXG 343
C     *      AND F=C/A                                                  MTXG 344
C     *YIELDS 1 Z=/(D-FB)=/(D-CE)             WHERE SYMMETRY APPLIES FOR MTXG 345
C     *    AND 2 X=-EZ                                A,W (AND FOR D,Z)    MTXG 346
C     *    AND 3 Y=-ZF                    AND WHERE MUTUAL TRANSPOSES AREMTXG 347
C     *    AND 4 W=/A-EY=/A-XF                        (B,C),(E,F),(X,Y)    MTXG 348
C     *HENCE THE COMPUTATIONAL PROCEDURE TO DEVELOP (WXYZ) FROM (ABCD) ISMTXG 349
C     *1. INITIALLY COMPUTE /A FROM A, AND THEN RECURSIVELY USE STEPS 2-8MTXG 350
C     *2. MOVE D TO Z                                                   MTXG 351
C     *3. COMPUTE F=C/A AND STORE AT Y                                  MTXG 352
C     *4. COMPUTE Q=(/Z)=D-FB AT Z                                      MTXG 353
C     *5. INVERT  Z=/Q IN PLACE                                         MTXG 354
C     *6. COMPUTE X=-EZ                                                 MTXG 355
C     *7. COMPUTE W=/A-XF              (COMPUTE ONE HALF, COPY OTHER HALF)MTXG 356
C     *8. COPY Y FROM X, WHENCE (WXYZ) IS /A FOR NEXT ITERATION         MTXG 357
C     *RESTRICT ALL PARTITIONS D,Z TO ORDER 1,                          MTXG 358
C     *INCREASING ORDER OF A,W BY 1 FOR EACH RECURSIVE USE OF STEPS 2-8. MTXG 359
C     *BEGIN AND END WITH MATRIX VATRIX(-,-) OF ORDER NATRIX            MTXG 360
C     *END INVERSION WITH MATRIX WATRIX(-,-) OF ORDER NATRIX            MTXG 361
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXG 362
   11 CONTINUE                                                         MTXG 363
C      *MAIN LOOP - EXECUTE N TIMES, ADVANCING INDEX KATRIX BY 1       MTXG 364
       IF(NATRIX)                                          92,   92,   28MTXG 365
   28 KATRIX=0                                                         MTXG 366
C      *OBSERVE KATRIX IS ORDER OF (ABCD),(WXYZ). LATRIX IS ORDER OF A,W  MTXG 367
FFF    LOOP FLAG                                                       MTXG 368
C      *END OF MAIN LOOP                                               MTXG 369
   90 CONTINUE                                                         MTXG 370
       GO TO (   127,   170), ISWTCH                                   MTXG 371
C      *ALARMS                                                         MTXG 372
   92 DURATN(NORDER)=-1.                                               MTXG 373
       IF(ISWTCH-2)                                       130.  210,  210MTXG 374
C      *END OF MATRIX INVERSION PROGRAMMING                            MTXG 375
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
  110 CONTINUE
C     *PRINCIPAL ITERATIVE LOOP ON NORDER=1,2,3...MORDER
      DO  222  NORDER=1,MORDER
      TORDER=NORDER
C     *CLEAR DURATN TABLE AND SET ERROR RECORD = 0.
      ERROR=0.
      DURATN(NORDER)=-TORDER
C     *SKIP CASES FOR WHICH NORDER IS NOT DIVISIBLE BY INTERV
      IF(NORDER-(NORDER/INTERV)*INTERV)                     222,   115,   222
  115 DURATN(NORDER)=0.
C     *CREATE A(I,J)
      DO    119 IORDER=1,NORDER
      DO    119 JORDER=1,IORDER
      KORDER=IORDER*JORDER
      VATRIX(IORDER,JORDER)=KORDER
      VATRIX(IORDER,JORDER)=SINF(VATRIX(IORDER,JORDER)/TORDER)
      IF(IORDER-JORDER)                                     119,   118,   119
  118 VATRIX(IORDER,JORDER)=VATRIX(IORDER,JORDER)+TORDER+1.
  119 VATRIX(JORDER,IORDER)=VATRIX(IORDER,JORDER)
C     *INVERSION LOOP SETUP
      NATRIX=NORDER
      ISWTCH=1
C     *CLOCK ON     INTERROGATE CLOCK HERE
      TIMEON=CLOCKF(DUMMY)
C     *LOOP NFRQCY ITERATIONS
      DO    127 NITRTN=1,NFRQCY
      GO TO 11
  127 CONTINUE
C     *CLOCK OFF    INTERROGATE CLOCK HERE
      TIMEUP=CLOCKF(DUMMY)
C     *COMPUTE DURATION  DURATN(NORDER) = CLOCK SPAN
      DURATN(NORDER)=TIMEUP-TIMEON
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXG 409
C    *LEAST SQUARES COMPUTATION.                                        MTXG 410
C    *DENOTE S(K) = SUM OF ((I**K)             ) OVER I = ORDERS EMPLOYEDMTXG 411
C    *AND    T(K) = SUM OF ((I**K)*(DURATN(I))) OVER I = ORDERS EMPLOYEDMTXG 412
C    *AND CUBIC = A(3)*(X**3) + A(2)*(X**2) + A(1)*X + A(0) WHERE THE   MTXG 413
C    *VARIABLE X DENOTES DURATION.                                      MTXG 414
C    *SOLVE      (S(0) S(1) S(2) S(3))        (A(0))        (T(0))      MTXG 415
C    *THIS       (S(1) S(2) S(3) S(4))   *    (A(1))   =    (T(1))      MTXG 416
C    *MATRIX     (S(2) S(3) S(4) S(5))        (A(2))        (T(2))      MTXG 417
C    *STSTEM     (S(3) S(4) S(5) S(6))        (A(3))        (T(3))      MTXG 418
C    *CREATE NORMAL MATRIX AND VECTOR AND CLEAR COEFFICIENT VECTOR      MTXG 419
  130 DO  133  IORDER=1,7                                               MTXG 420
      IF(IORDER-4)                                    131,   131,   133MTXG 421
  131 WTDPOW(IORDER)=0.                                                 MTXG 422
      COEFFT(IORDER)=0.                                                 MTXG 423
  133 POWERS(IORDER)=0.                                                 MTXG 424
C    *USE KORDER TO COUNT VALID DATA CASES                              MTXG 425
      KORDER=0                                                          MTXG 426
      DO   159 IORDER=1,NORDER                                          MTXG 427
C    *SKIP IRRELEVANT CASES (DURATN(IORDER) NEGATIVE)                   MTXG 428
      IF(DURATN(IORDER))                              159,   142,   142MTXG 429
  142 KORDER=KORDER+1                                                   MTXG 430
C    *DENOTE POWRDR = POWERS OF ORDER VARIABLE IORDER                   MTXG 431
      POWRDR=1.                                                         MTXG 432
C    *SCALE (NORMALIZE) INDEPENDENT TIME VARIABLE BY DIVISION BY NORDER.MTXG 433
      XORDER=IORDER                                                     MTXG 434
      YORDER=NORDER                                                     MTXG 435
      XORDER=XORDER/YORDER                                              MTXG 436
      DO   158 JORDER=1,7                                               MTXG 437
      IF(JORDER-1)                                    222,   153,   152MTXG 438
  152 POWRDR=POWRDR*XORDER                                              MTXG 439
  153 IF(JORDER-4)                                    154,   154,   158MTXG 440
  154 WTDPOW(JORDER   )=WTDPOW(JORDER   )+POWRDR*DURATN(IORDER)         MTXG 441
  158 POWERS(JORDER   )=POWERS(JORDER   )+POWRDR                        MTXG 442
  159 CONTINUE                                                          MTXG 443
C    *SKIP CASES WITH LESS THAN 4 DATA POINTS                           MTXG 444
      IF(KORDER-4)                                    210,   161,   161MTXG 445
  161 DO  168  IORDER=1,4                                               MTXG 446
      DO  168  JORDER=1,IORDER                                          MTXG 447
      KORDER=IORDER+JORDER                                              MTXG 448
      VATRIX(IORDER,JORDER)=POWERS(KORDER-1   )                         MTXG 449
  168 VATRIX(JORDER,IORDER)=VATRIX(IORDER,JORDER)                       MTXG 450
C    *INVERT NORMAL MATRIX                                              MTXG 451
      NATRIX=4                                                          MTXG 452
      ISWTCH=2                                                          MTXG 453
      GO TO 11                                                          MTXG 454
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
  170 CONTINUE
C     *OBTAIN COEFFICIENTS AND MEASURE ERROR
C     *DESCALE THE TIME VARIABLE BY DIVIDING THE COEFFICIENTS BY NORDER.
C     *NORMALIZE THE COEFFICIENTS TO REPRESENT DATA FOR NFRQCY = 1
      ZORDER=NFRQCY
      DO    200 IORDER=1,4
      LORDER=5-IORDER
      DO    199 JORDER=1,4
      XORDER=0.
      DO    193 KORDER=1,4
  193 XORDER=XORDER+VATRIX(IORDER,KORDER)*WATRIX(KORDER,JORDER)
      IF(IORDER-JORDER)                                196,  195,  196
  195 XORDER=XORDER-1.
  196 XORDER=ABSF(XORDER)
      IF(ERROR-XORDER)                                 198,  199,  199
  198 ERROR=XORDER
  199 COEFFT(  LORDER)=COEFFT(  LORDER)+WATRIX(IORDER,JORDER)
 1991                                *WTDPOW(JORDER   )
 1992                                /ZORDER
  200 ZORDER=ZORDER*YORDER
C     *PRINT
  201 FORMAT(10H          7F15.9)
  210 PRINT    201, TORDER, DURATN(NORDER), (COEFFT(  I),I=1,4), ERROR
C     *END OF PRINCIPAL ITERATIVE LOOP
  222 CONTINUE
      CALL SYSTEM
      END
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXG 483
        LOAD BATCH                                                     MTXG 484
        REMARK  THE CLOCK INTERROGATION PROGRAM IS LOADED HERE         MTXG 485
        TRA                                                            MTXG 486
      DATA      1     5 5000                                           MTXG 487
        REMARK  END OF RUN                                             MTXG 488
F       END OF DATA FLAG                                               MTXG 489
        SPACE                                                          MTXG 490
        SPACE                                                          MTXG 491
        REMARK  TAPE 5 HAS BEEN UNLOADED                               MTXG 492
        REMARK  PLEASE REMOVE, THEN PUNCH, THEN SAVE TAPE 5            MTXG 493
        SPACE                                                          MTXG 494
        REMARK  THEN PUSH START TO TERMINATE THIS JOB AND CALL NEXT JOB MTXG 495
        SPACE                                                          MTXG 496
        SPACE                                                          MTXG 497
        PAUSE                                                          MTXG 498
        REMARK  END OF RUN                                             MTXG 499
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTX
       JOB     00655,AS,PROGRAMMER,                             (BY O) SEPT64MTX
       FOR                                                               MTX
C     *SYMMETRIC MATRIX INVERSION BY PARTITION, WITH LEAST SQUARES FIT OFMTX
C     *THE CUBIC POLYNOMIAL FOR INVERSION TIME IN TERMS OF MATRIX ORDER. MTX
C     *FOR CERTAIN ORDERS NORDER=1,2,3... TO A PRESCRIBED MAXIMUM MORDER,MTX
C     *OMITTING ALL ORDERS NORDER NOT DIVISIBLE BY A PRESCRIBED INTERV,  MTX
C     *INVERT THE (POSITIVE DEFINITE) SYMMETRIC MATRIX DEFINED AS FOLLOWSMTX
C     *A(I,J) = SIN((I*J)/NORDER) + THE DIAGONAL MATRIX L(I,I) = NORDER+1MTX
C     *A PRESCRIBED NUMBER NFRQCY OF TIMES, INTERNALLY MEASURING THE     MTX
C     *TOTAL DURATION OF ALL NFRQCY INVERSIONS FOR EACH NORDER.          MTX
C     *USING THE MATRIX INVERSION PROGRAM FOR THE LEAST SQUARES FITTING, MTX
C     *FOR EACH NORDER ABOVE 3, USE LEAST SQUARES TO FIT A CUBIC TO THE  MTX
C     *INTERNALLY MEASURED DURATION TIMES FOR ALL ORDERS UP TO NORDER,   MTX
C     *AND RECORD THE MAXIMUM ERROR (DEPARTURE FROM IDENTITY MATRIX) IN  MTX
C     *POSTMULTIPLYING EACH LEAST SQUARES NORMAL MATRIX BY ITS INVERSE.  MTX
C     *FINALLY, PRINT ALL DATA APPLICABLE TO EACH NORDER, SPECIFICALLY-  MTX
C     *NORDER, TIME, COEFFICIENTS (CUBIC,SQUARE,LINEAR,CONSTANT), ERROR, MTX
C     *EMPLOYING (FAKED) NEGATIVE TIME TC MARK INABILITY TO GET SOLUTION.MTX
      DIMENSION COEFFT(4)                                                MTX
      DIMENSION WTDPOW(4)                                                MTX
      DIMENSION POWERS(7)                                                MTX
C     *LMTDIM MUST RELATE TO SEVERAL DIMENSIONS AND LIMITS MORDER.       MTX
C     *LMTDIM=  5                                                        MTX
C     *BOTH DIMENSIONS EQUAL LMTDIM                                      MTX
      DIMENSION WATRIX(  5,   5)                                         MTX
      DIMENSION VATRIX(  5,   5)                                         MTX
C     *ONE DIMENSION EQUAL TO LMTDIM                                     MTX
      DIMENSION DURATN(  5)                                              MTX
C     *DATA INPUT                                                        MTX
C     *READ THE PARAMETERS INTERV,MORDER,NFRQCY FROM A PUNCHED CARD      MTX
C     *INTERV FROM COLUMNS 11-15 IN INTEGER FORMAT, AND                  MTX
C     *MORDER FROM COLUMNS 16-20 IN INTEGER FORMAT, AND                  MTX
C     *NFRQCY FROM COLUMNS 21-25 IN INTEGER FORMAT.                      MTX
    4 FORMAT(10H          3I5)                                           MTX
      READ  4,                                   INTERV, MORDER, NFRQCYMTX
      PRINT 4,                                   INTERV, MORDER, NFRQCYMTX
      GO TO 110                                                          MTX
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTX
C     *SYMMETRIC MATRIX INVERSION BY PARTITION, BY NONLOOPING PROGRAMMINGMTX
C     *(AAABHH) (WWWXOO) (IOOOOO) (WWWXOO) (AAABHH)                     MTX
C     *(AAABHH) (WWWXOO) (OIOOOO) (WWWXOO) (AAABHH)  DEVELOP  (1) AW+BY=IMTX
C     *(AAABHH).(WWWXOO)=(OOIOOO)=(WWWXOO).(AAABHH)    THE    (2) AX+BZ=OMTX
C     *(CCCDHH) (YYYZOO) (OOOIOO) (YYYZOO) (CCCDHH)   BASIC   (3) YA+ZC=OMTX
C     *(HHHHHH) (OOOOOO) (OOOOOO) (OOOOOO) (HHHHHH) EQUATIONS (4) YB+ZD=IMTX
C     *(HHHHHH) (OOOOOO) (OOOOOO) (OOOOOO) (HHHHHH)                     MTX
C     *THE SLASH (/), USED AS A PREFIX, HERE DENOTES INVERSION          MTX
C     *DEFINING E=/AB                                                   MTX
C     *     AND F=C/A                                                   MTX
C     *YIELDS 1 Z=/(D-FB)=/(D-CE)             WHERE SYMMETRY APPLIES FOR MTX
C     *    AND 2 X=-EZ                                 A,W (AND FOR D,Z)   MTX
C     *    AND 3 Y=-ZF                     AND WHERE MUTUAL TRANSPOSES AREMTX
C     *    AND 4 W=/A-EY=/A-XF                         (B,C),(E,F),(X,Y)   MTX
C     *HENCE THE COMPUTATIONAL PROCEDURE TO DEVELOP (WXYZ) FROM (ABCD) ISMTX
C     *1. INITIALLY COMPUTE /A FROM A, AND THEN RECURSIVELY USE STEPS 2-8MTX
C     *2. MOVE D TO Z                                                   MTX
C     *3. COMPUTE F=C/A AND STORE AT Y                                  MTX
C     *4. COMPUTE Q=(/Z)=D-FB AT Z                                      MTX
C     *5. INVERT  Z=/Q IN PLACE                                         MTX
C     *6. COMPUTE X=-EZ                                                 MTX
C     *7. COMPUTE W=/A-XF            (COMPUTE ONE HALF, COPY OTHER HALF)MTX
C     *8. COPY Y FROM X, WHENCE (WXYZ) IS /A FOR NEXT ITERATION         MTX
C     *RESTRICT ALL PARTITIONS D,Z TO ORDER 1,                          MTX
C     *INCREASING ORDER OF A,W BY 1 FOR EACH RECURSIVE USE OF STEPS 2-8. MTX
C     *BEGIN AND END WITH MATRIX VATRIX(-,-) OF ORDER NATRIX            MTX
C     *END INVERSION WITH MATRIX WATRIX(-,-) OF ORDER NATRIX            MTX
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
   11 CONTINUE
C      *MAIN LOOP - EXECUTE N TIMES, ADVANCING INDEX KATRIX BY 1
       IF(NATRIX)                                          92,    92,    28
   28 KATRIX=0
C      *OBSERVE KATRIX IS ORDER OF (ABCD),(WXYZ). LATRIX IS ORDER OF A,W
       KATRIX=KATRIX+1
       LATRIX=KATRIX-1
C      *MOVE D TO Z (A TO W FOR FIRST PASS (AND INVERT A IF KATRIX=1))
       WATRIX( 1, 1)= VATRIX( 1, 1)
C      *INVERT Q (INVERT D=A FOR KATRIX=1)
       IF(WATRIX( 1, 1))                                 1001,    92, 1001
 1001 WATRIX( 1, 1)=1./WATRIX( 1, 1)
       IF(KATRIX-NATRIX)                                 1101,    90,    90
 1101 CONTINUE
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
      KATRIX=KATRIX+1
      LATRIX=KATRIX-1
C     *MOVE D TO Z (A TO W FOR FIRST PASS (AND INVERT A IF KATRIX=1))
      WATRIX( 2, 2)= VATRIX( 2, 2)
C     *COMPUTE F=C/A AND Q=D-FB
      WATRIX( 2, 1)=0.
C     *ACCUMULATE ONE ELEMENT OF F=C/A
      WATRIX( 2, 1)= WATRIX( 2, 1)+VATRIX( 2, 1)*WATRIX( 1, 1)
C     *FORM Q=D-FB BY ACCUMULATION TO D
      WATRIX( 2, 2)= WATRIX( 2, 2)-WATRIX( 2, 1)*VATRIX( 1, 2)
C     *INVERT Q (INVERT D=A FOR KATRIX=1)
      IF(WATRIX( 2, 2))                                 1002,   92, 1002
 1002 WATRIX( 2, 2)=1./WATRIX( 2, 2)
C     *COMPUTE X=-EZ BY TRANSPOSING Y=-ZF
      WATRIX( 1, 2)=-WATRIX( 2, 2)*WATRIX( 2, 1)
C     *COMPUTE W=/A-XF AND FORM Y AS TRANSPOSE OF X
C     *ACCUMULATE -XF INTO /A, ONLY ALONG AND ABOVE MAJOR DIAGONAL
C     *COMPLETE W BY SYMMETRY
      WATRIX( 1, 1)= WATRIX( 1, 1)-WATRIX( 1, 2)*WATRIX( 2, 1)
      WATRIX( 1, 1)= WATRIX( 1, 1)
C     *TRANSPOSE ONE ELEMENT OF X TO Y
      WATRIX( 2, 1)= WATRIX( 1, 2)
      IF(KATRIX-NATRIX)                                 1102,   90,   90
 1102 CONTINUE
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
      KATRIX=KATRIX+1
      LATRIX=KATRIX-1
C     *MOVE D TO Z (A TO W FOR FIRST PASS (AND INVERT A IF KATRIX=1))
      WATRIX( 3, 3)= VATRIX( 3, 3)
C     *COMPUTE F=C/A AND Q=D-FB
      WATRIX( 3, 1)=0.
C     *ACCUMULATE ONE ELEMENT OF F=C/A
      WATRIX( 3, 1)= WATRIX( 3, 1)+VATRIX( 3, 1)*WATRIX( 1, 1)
      WATRIX( 3, 1)= WATRIX( 3, 1)+VATRIX( 3, 2)*WATRIX( 2, 1)
C     *FORM Q=D-FB BY ACCUMULATION TO D
      WATRIX( 3, 3)= WATRIX( 3, 3)-WATRIX( 3, 1)*VATRIX( 1, 3)
      WATRIX( 3, 2)=0.
C     *ACCUMULATE ONE ELEMENT OF F=C/A
      WATRIX( 3, 2)= WATRIX( 3, 2)+VATRIX( 3, 1)*WATRIX( 1, 2)
      WATRIX( 3, 2)= WATRIX( 3, 2)+VATRIX( 3, 2)*WATRIX( 2, 2)
C     *FORM Q=D-FB BY ACCUMULATION TO D
      WATRIX( 3, 3)= WATRIX( 3, 3)-WATRIX( 3, 2)*VATRIX( 2, 3)
C     *INVERT Q (INVERT D=A FOR KATRIX=1)
      IF(WATRIX( 3, 3))                                1003,    92, 1003
 1003 WATRIX( 3, 3)=1./WATRIX( 3, 3)
C     *COMPUTE X=-EZ BY TRANSPOSING Y=-ZF
      WATRIX( 1, 3)=-WATRIX( 3, 3)*WATRIX( 3, 1)
      WATRIX( 2, 3)=-WATRIX( 3, 3)*WATRIX( 3, 2)
C     *COMPUTE W=/A-XF AND FORM Y AS TRANSPOSE OF X
C     *ACCUMULATE -XF INTO /A, ONLY ALONG AND ABOVE MAJOR DIAGONAL
C     *COMPLETE W BY SYMMETRY
      WATRIX( 1, 1)= WATRIX( 1, 1)-WATRIX( 1, 3)*WATRIX( 3, 1)
      WATRIX( 1, 1)= WATRIX( 1, 1)
      WATRIX( 1, 2)= WATRIX( 1, 2)-WATRIX( 1, 3)*WATRIX( 3, 2)
      WATRIX( 2, 1)= WATRIX( 1, 2)
C     *TRANSPOSE ONE ELEMENT OF X TO Y
      WATRIX( 3, 1)= WATRIX( 1, 3)
C     *ACCUMULATE -XF INTO /A, ONLY ALONG AND ABOVE MAJOR DIAGONAL
C     *COMPLETE W BY SYMMETRY
      WATRIX( 2, 2)= WATRIX( 2, 2)-WATRIX( 2, 3)*WATRIX( 3, 2)
      WATRIX( 2, 2)= WATRIX( 2, 2)
C     *TRANSPOSE ONE ELEMENT OF X TO Y
      WATRIX( 3, 2)= WATRIX( 2, 3)
      IF(KATRIX-NATRIX)                                1103,    90,   90
 1103 CONTINUE
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
      KATRIX=KATRIX+1
      LATRIX=KATRIX-1
C     *MOVE D TO Z (A TO W FOR FIRST PASS (AND INVERT A IF KATRIX=1))
      WATRIX( 4, 4)= VATRIX( 4, 4)
C     *COMPUTE F=C/A AND Q=D-FB
      WATRIX( 4, 1)=0.
C     *ACCUMULATE ONE ELEMENT OF F=C/A
      WATRIX( 4, 1)= WATRIX( 4, 1)+VATRIX( 4, 1)*WATRIX( 1, 1)
      WATRIX( 4, 1)= WATRIX( 4, 1)+VATRIX( 4, 2)*WATRIX( 2, 1)
      WATRIX( 4, 1)= WATRIX( 4, 1)+VATRIX( 4, 3)*WATRIX( 3, 1)
C     *FORM Q=D-FB BY ACCUMULATION TO D
      WATRIX( 4, 4)= WATRIX( 4, 4)-WATRIX( 4, 1)*VATRIX( 1, 4)
      WATRIX( 4, 2)=0.
C     *ACCUMULATE ONE ELEMENT OF F=C/A
      WATRIX( 4, 2)= WATRIX( 4, 2)+VATRIX( 4, 1)*WATRIX( 1, 2)
      WATRIX( 4, 2)= WATRIX( 4, 2)+VATRIX( 4, 2)*WATRIX( 2, 2)
      WATRIX( 4, 2)= WATRIX( 4, 2)+VATRIX( 4, 3)*WATRIX( 3, 2)
C     *FORM Q=D-FB BY ACCUMULATION TO D
      WATRIX( 4, 4)= WATRIX( 4, 4)-WATRIX( 4, 2)*VATRIX( 2, 4)
      WATRIX( 4, 3)=0.
C     *ACCUMULATE ONE ELEMENT OF F=C/A
      WATRIX( 4, 3)= WATRIX( 4, 3)+VATRIX( 4, 1)*WATRIX( 1, 3)
      WATRIX( 4, 3)= WATRIX( 4, 3)+VATRIX( 4, 2)*WATRIX( 2, 3)
      WATRIX( 4, 3)= WATRIX( 4, 3)+VATRIX( 4, 3)*WATRIX( 3, 3)
C     *FORM Q=D-FB BY ACCUMULATION TO D
      WATRIX( 4, 4)= WATRIX( 4, 4)-WATRIX( 4, 3)*VATRIX( 3, 4)
C     *INVERT Q (INVERT D=A FOR KATRIX=1)
      IF(WATRIX( 4, 4))                                  1004,   92, 1004
 1004 WATRIX( 4, 4)=1./WATRIX( 4, 4)
C     *COMPUTE X=-EZ BY TRANSPOSING Y=-ZF
      WATRIX( 1, 4)=-WATRIX( 4, 4)*WATRIX( 4, 1)
      WATRIX( 2, 4)=-WATRIX( 4, 4)*WATRIX( 4, 2)
      WATRIX( 3, 4)=-WATRIX( 4, 4)*WATRIX( 4, 3)
C     *COMPUTE W=/A-XF AND FORM Y AS TRANSPOSE OF X
C     *ACCUMULATE -XF INTO /A, ONLY ALONG AND ABOVE MAJOR DIAGONAL
C     *COMPLETE W BY SYMMETRY
      WATRIX( 1, 1)= WATRIX( 1, 1)-WATRIX( 1, 4)*WATRIX( 4, 1)
      WATRIX( 1, 1)= WATRIX( 1, 1)
      WATRIX( 1, 2)= WATRIX( 1, 2)-WATRIX( 1, 4)*WATRIX( 4, 2)
      WATRIX( 2, 1)= WATRIX( 1, 2)
      WATRIX( 1, 3)= WATRIX( 1, 3)-WATRIX( 1, 4)*WATRIX( 4, 3)
      WATRIX( 3, 1)= WATRIX( 1, 3)
C     *TRANSPOSE ONE ELEMENT OF X TO Y
      WATRIX( 4, 1)= WATRIX( 1, 4)
C     *ACCUMULATE -XF INTO /A, ONLY ALONG AND ABOVE MAJOR DIAGONAL
C     *COMPLETE W BY SYMMETRY
      WATRIX( 2, 2)= WATRIX( 2, 2)-WATRIX( 2, 4)*WATRIX( 4, 2)
      WATRIX( 2, 2)= WATRIX( 2, 2)
      WATRIX( 2, 3)= WATRIX( 2, 3)-WATRIX( 2, 4)*WATRIX( 4, 3)
      WATRIX( 3, 2)= WATRIX( 2, 3)
C     *TRANSPOSE ONE ELEMENT OF X TO Y
      WATRIX( 4, 2)= WATRIX( 2, 4)
C     *ACCUMULATE -XF INTO /A, ONLY ALONG AND ABOVE MAJOR DIAGONAL
C     *COMPLETE W BY SYMMETRY
      WATRIX( 3, 3)= WATRIX( 3, 3)-WATRIX( 3, 4)*WATRIX( 4, 3)
      WATRIX( 3, 3)= WATRIX( 3, 3)
C     *TRANSPOSE ONE ELEMENT OF X TO Y
      WATRIX( 4, 3)= WATRIX( 3, 4)
      IF(KATRIX-NATRIX)                                  1104,   90,   90
 1104 CONTINUE
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
      KATRIX=KATRIX+1
      LATRIX=KATRIX-1
C     *MOVE D TO Z (A TO W FOR FIRST PASS (AND INVERT A IF KATRIX=1))
      WATRIX( 5, 5)= VATRIX( 5, 5)
C     *COMPUTE F=C/A AND Q=D-FB
      WATRIX( 5, 1)=0.
C     *ACCUMULATE ONE ELEMENT OF F=C/A
      WATRIX( 5, 1)= WATRIX( 5, 1)+VATRIX( 5, 1)*WATRIX( 1, 1)
      WATRIX( 5, 1)= WATRIX( 5, 1)+VATRIX( 5, 2)*WATRIX( 2, 1)
      WATRIX( 5, 1)= WATRIX( 5, 1)+VATRIX( 5, 3)*WATRIX( 3, 1)
      WATRIX( 5, 1)= WATRIX( 5, 1)+VATRIX( 5, 4)*WATRIX( 4, 1)
C     *FORM Q=D-FB BY ACCUMULATION TO D
      WATRIX( 5, 5)= WATRIX( 5, 5)-WATRIX( 5, 1)*VATRIX( 1, 5)
      WATRIX( 5, 2)=0.
C     *ACCUMULATE ONE ELEMENT OF F=C/A
      WATRIX( 5, 2)= WATRIX( 5, 2)+VATRIX( 5, 1)*WATRIX( 1, 2)
      WATRIX( 5, 2)= WATRIX( 5, 2)+VATRIX( 5, 2)*WATRIX( 2, 2)
      WATRIX( 5, 2)= WATRIX( 5, 2)+VATRIX( 5, 3)*WATRIX( 3, 2)
      WATRIX( 5, 2)= WATRIX( 5, 2)+VATRIX( 5, 4)*WATRIX( 4, 2)
C     *FORM Q=D-FB BY ACCUMULATION TO D
      WATRIX( 5, 5)= WATRIX( 5, 5)-WATRIX( 5, 2)*VATRIX( 2, 5)
      WATRIX( 5, 3)=0.
C     *ACCUMULATE ONE ELEMENT OF F=C/A
      WATRIX( 5, 3)= WATRIX( 5, 3)+VATRIX( 5, 1)*WATRIX( 1, 3)
      WATRIX( 5, 3)= WATRIX( 5, 3)+VATRIX( 5, 2)*WATRIX( 2, 3)
      WATRIX( 5, 3)= WATRIX( 5, 3)+VATRIX( 5, 3)*WATRIX( 3, 3)
      WATRIX( 5, 3)= WATRIX( 5, 3)+VATRIX( 5, 4)*WATRIX( 4, 3)
C     *FORM Q=D-FB BY ACCUMULATION TO D
      WATRIX( 5, 5)= WATRIX( 5, 5)-WATRIX( 5, 3)*VATRIX( 3, 5)
      WATRIX( 5, 4)=0.
C     *ACCUMULATE ONE ELEMENT OF F=C/A
      WATRIX( 5, 4)= WATRIX( 5, 4)+VATRIX( 5, 1)*WATRIX( 1, 4)
      WATRIX( 5, 4)= WATRIX( 5, 4)+VATRIX( 5, 2)*WATRIX( 2, 4)
      WATRIX( 5, 4)= WATRIX( 5, 4)+VATRIX( 5, 3)*WATRIX( 3, 4)
      WATRIX( 5, 4)= WATRIX( 5, 4)+VATRIX( 5, 4)*WATRIX( 4, 4)
C     *FORM Q=D-FB BY ACCUMULATION TO D
      WATRIX( 5, 5)= WATRIX( 5, 5)-WATRIX( 5, 4)*VATRIX( 4, 5)
C     *INVERT Q (INVERT D=A FOR KATRIX=1)
      IF(WATRIX( 5, 5))                                 1005,   92, 1005
 1005 WATRIX( 5, 5)=1./WATRIX( 5, 5)
C     *COMPUTE X=-EZ BY TRANSPOSING Y=-ZF
      WATRIX( 1, 5)=-WATRIX( 5, 5)*WATRIX( 5, 1)
      WATRIX( 2, 5)=-WATRIX( 5, 5)*WATRIX( 5, 2)
      WATRIX( 3, 5)=-WATRIX( 5, 5)*WATRIX( 5, 3)
      WATRIX( 4, 5)=-WATRIX( 5, 5)*WATRIX( 5, 4)
C     *COMPUTE W=/A-XF AND FORM Y AS TRANSPOSE OF X
C     *ACCUMULATE -XF INTO /A, ONLY ALONG AND ABOVE MAJOR DIAGONAL
C     *COMPLETE W BY SYMMETRY
      WATRIX( 1, 1)= WATRIX( 1, 1)-WATRIX( 1, 5)*WATRIX( 5, 1)
      WATRIX( 1, 1)= WATRIX( 1, 1)
      WATRIX( 1, 2)= WATRIX( 1, 2)-WATRIX( 1, 5)*WATRIX( 5, 2)
      WATRIX( 2, 1)= WATRIX( 1, 2)
      WATRIX( 1, 3)= WATRIX( 1, 3)-WATRIX( 1, 5)*WATRIX( 5, 3)
      WATRIX( 3, 1)= WATRIX( 1, 3)
      WATRIX( 1, 4)= WATRIX( 1, 4)-WATRIX( 1, 5)*WATRIX( 5, 4)
      WATRIX( 4, 1)= WATRIX( 1, 4)
C     *TRANSPOSE ONE ELEMENT OF X TO Y
      WATRIX( 5, 1)= WATRIX( 1, 5)
C     *ACCUMULATE -XF INTO /A, ONLY ALONG AND ABOVE MAJOR DIAGONAL
C     *COMPLETE W BY SYMMETRY
      WATRIX( 2, 2)= WATRIX( 2, 2)-WATRIX( 2, 5)*WATRIX( 5, 2)
      WATRIX( 2, 2)= WATRIX( 2, 2)
      WATRIX( 2, 3)= WATRIX( 2, 3)-WATRIX( 2, 5)*WATRIX( 5, 3)
      WATRIX( 3, 2)= WATRIX( 2, 3)
      WATRIX( 2, 4)= WATRIX( 2, 4)-WATRIX( 2, 5)*WATRIX( 5, 4)
      WATRIX( 4, 2)= WATRIX( 2, 4)
C     *TRANSPOSE ONE ELEMENT OF X TO Y
      WATRIX( 5, 2)= WATRIX( 2, 5)
C     *ACCUMULATE -XF INTO /A, ONLY ALONG AND ABOVE MAJOR DIAGONAL
C     *COMPLETE W BY SYMMETRY
      WATRIX( 3, 3)= WATRIX( 3, 3)-WATRIX( 3, 5)*WATRIX( 5, 3)
      WATRIX( 3, 3)= WATRIX( 3, 3)
      WATRIX( 3, 4)= WATRIX( 3, 4)-WATRIX( 3, 5)*WATRIX( 5, 4)
      WATRIX( 4, 3)= WATRIX( 3, 4)
C     *TRANSPOSE ONE ELEMENT OF X TO Y
      WATRIX( 5, 3)= WATRIX( 3, 5)
C     *ACCUMULATE -XF INTO /A, ONLY ALONG AND ABOVE MAJOR DIAGONAL
C     *COMPLETE W BY SYMMETRY
      WATRIX( 4, 4)= WATRIX( 4, 4)-WATRIX( 4, 5)*WATRIX( 5, 4)
      WATRIX( 4, 4)= WATRIX( 4, 4)
C     *TRANSPOSE ONE ELEMENT OF X TO Y
      WATRIX( 5, 4)= WATRIX( 4, 5)
      IF(KATRIX-NATRIX)                                 1105,   90,   90
 1105 CONTINUE
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTX  294
C     *END OF MAIN LOOP                                                  MTX  295
   90 CONTINUE                                                           MTX  296
      GO TO (  127,  170), ISWTCH                                        MTX  297
C     *ALARMS                                                            MTX  298
   92 DURATN(NORDER)=-1.                                                 MTX  299
      IF(ISWTCH-2)                                    130,  210,  210MTX  300
C     *END OF MATRIX INVERSION PROGRAMMING                               MTX  301
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
  110 CONTINUE
C    *PRINCIPAL ITERATIVE LOOP ON NORDER=1,2,3...MORDER
      DO  222  NORDER=1,MORDER
      TORDER=NORDER
C    *CLEAR DURATN TABLE AND SET ERROR RECORD = 0.
      ERROR=0.
      DURATN(NORDER)=-TORDER
C    *SKIP CASES FOR WHICH NORDER IS NOT DIVISIBLE BY INTERV
      IF(NORDER-(NORDER/INTERV)*INTERV)                   222,  115,  222
  115 DURATN(NORDER)=0.
C    *CREATE A(I,J)
      DO    119 IORDER=1,NORDER
      DO    119 JORDER=1,IORDER
      KORDER=IORDER*JORDER
      VATRIX(IORDER,JORDER)=KORDER
      VATRIX(IORDER,JORDER)=SINF(VATRIX(IORDER,JORDER)/TORDER)
      IF(IORDER-JORDER)                                   119,  118,  119
  118 VATRIX(IORDER,JORDER)=VATRIX(IORDER,JORDER)+TORDER+1.
  119 VATRIX(JORDER,IORDER)=VATRIX(IORDER,JORDER)
C    *INVERSION LOOP SETUP
      NATRIX=NORDER
      ISWTCH=1
C    *CLOCK ON     INTERROGATE CLOCK HERE
      TIMEON=CLOCKF(DUMMY)
C    *LOOP NFRQCY ITERATIONS
      DO    127 NITRTN=1,NFRQCY
      GO TO 11
  127 CONTINUE
C    *CLOCK OFF    INTERROGATE CLOCK HERE
      TIMEUP=CLOCKF(DUMMY)
C    *COMPUTE DURATION  DURATN(NORDER) = CLOCK SPAN
      DURATN(NORDER)=TIMEUP-TIMEON
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTX
C     *LEAST SQUARES COMPUTATION.                                      MTX
C     *DENOTE S(K) = SUM OF ((I**K)              ) OVER I = ORDERS EMPLOYEDMTX
C     *AND    T(K) = SUM OF ((I**K)*(DURATN(I))) OVER I = ORDERS EMPLOYEDMTX
C     *AND CUBIC = A(3)*(X**3) + A(2)*(X**2) + A(1)*X + A(0) WHERE THE   MTX
C     *VARIABLE X DENOTES DURATION.                                    MTX
C     *SOLVE       (S(0) S(1) S(2) S(3))      (A(0))      (T(0))      MTX
C     *THIS        (S(1) S(2) S(3) S(4))  *   (A(1))  =   (T(1))      MTX
C     *MATRIX      (S(2) S(3) S(4) S(5))      (A(2))      (T(2))      MTX
C     *STSTEM      (S(3) S(4) S(5) S(6))      (A(3))      (T(3))      MTX
C     *CREATE NORMAL MATRIX AND VECTOR AND CLEAR COEFFICIENT VECTOR    MTX
  130 DO   133   IORDER=1,7                                            MTX
      IF(IORDER-4)                                  131,    131,    133MTX
  131 WTDPOW(IORDER)=0.                                                MTX
      COEFFT(IORDER)=0.                                                MTX
  133 POWERS(IORDER)=0.                                                MTX
C     *USE KORDER TO COUNT VALID DATA CASES                            MTX
      KORDER=0                                                         MTX
      DO    159 IORDER=1,NORDER                                        MTX
C     *SKIP IRRELEVANT CASES (DURATN(IORDER) NEGATIVE)                 MTX
      IF(DURATN(IORDER))                            159,    142,    142MTX
  142 KORDER=KORDER+1                                                  MTX
C     *DENOTE POWRDR = POWERS OF ORDER VARIABLE IORDER                 MTX
      POWRDR=1.                                                        MTX
C     *SCALE (NORMALIZE) INDEPENDENT TIME VARIABLE BY DIVISION BY NORDER.MTX
      XORDER=IORDER                                                    MTX
      YORDER=NORDER                                                    MTX
      XORDER=XORDER/YORDER                                             MTX
      DO    158 JORDER=1,7                                             MTX
      IF(JORDER-1)                                  222,    153,    152MTX
  152 POWRDR=POWRDR*XORDER                                             MTX
  153 IF(JORDER-4)                                  154,    154,    158MTX
  154 WTDPOW(JORDER  )=WTDPOW(JORDER  )+POWRDR*DURATN(IORDER)          MTX
  158 POWERS(JORDER  )=POWERS(JORDER  )+POWRDR                         MTX
  159 CONTINUE                                                         MTX
C     *SKIP CASES WITH LESS THAN 4 DATA POINTS                         MTX
      IF(KORDER-4)                                  210,    161,    161MTX
  161 DO   168   IORDER=1,4                                            MTX
      DO   168   JORDER=1,IORDER                                       MTX
      KORDER=IORDER+JORDER                                             MTX
      VATRIX(IORDER,JORDER)=POWERS(KORDER-1   )                        MTX
  168 VATRIX(JORDER,IORDER)=VATRIX(IORDER,JORDER)                      MTX
C     *INVERT NORMAL MATRIX                                            MTX
      NATRIX=4                                                         MTX
      ISWTCH=2                                                         MTX
      GO TO 11                                                         MTX
```

Appendix C

Programming for

Case D-by-D

Note: See Note on Appendix B cover sheet.

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXD
       JOB      00655,AS,PROGRAMMER,                            (BY D) SEPT64MTXD
       FOR                                                                   MTXD
C     *SYMMETRIC MATRIX INVERSION BY PARTITION, WITH LEAST SQUARES FIT ATMTXD
C     *THE CUBIC POLYNOMIAL FOR INVERSION TIME IN TERMS OF MATRIX ORDER. MTXD
C     *FOR CERTAIN ORDERS NORDER=1,2,3... TO A PRESCRIBED MAXIMUM MORDER,MTXD
C     *OMITTING ALL ORDERS NORDER NOT DIVISIBLE BY A PRESCRIBED INTERV,  MTXD
C     *INVERT THE (POSITIVE DEFINITE) SYMMETRIC MATRIX DEFINED AS FOLLOWSMTXD
C     *A(I,J) = SIN((I*J)/NORDER) + THE DIAGONAL MATRIX L(I,I) = NORDER+1MTXD
C     *A PRESCRIBED NUMBER NFRQCY OF TIMES, INTERNALLY MEASURING THE     MTXD
C     *TOTAL DURATION OF ALL NFRQCY INVERSIONS FOR EACH NORDER.          MTXD
C     *USING THE MATRIX INVERSION PROGRAM FOR THE LEAST SQUARES FITTING, MTXD
C     *FOR EACH NORDER ABOVE 3, USE LEAST SQUARES TO FIT A CUBIC TO THE  MTXD
C     *INTERNALLY MEASURED DURATION TIMES FOR ALL ORDERS UP TO NORDER,   MTXD
C     *AND RECORD THE MAXIMUM ERROR (DEPARTURE FROM IDENTITY MATRIX) IN  MTXD
C     *POSTMULTIPLYING EACH LEAST SQUARES NORMAL MATRIX BY ITS INVERSE.  MTXD
C     *FINALLY, PRINT ALL DATA APPLICABLE TO EACH NORDER, SPECIFICALLY-  MTXD
C     *NORDER, TIME, COEFFICIENTS (CUBIC,SQUARE,LINEAR,CONSTANT), ERROR, MTXD
C     *EMPLOYING (FAKED) NEGATIVE TIME TO MARK INABILITY TO GET SOLUTION.MTXD
C     *LMTDIM MUST RELATE TO SEVERAL DIMENSIONS AND LIMITS MORDER.       MTXD
C     *LMTDIM=100                                                        MTXD
C     *BOTH DIMENSIONS EQUAL LMTDIM                                      MTXD
      DIMENSION WATRIX(100,100)                                          MTXD
      DIMENSION VATRIX(100,100)                                          MTXD
C     *ONE DIMENSION EQUAL TO LMTDIM                                     MTXD
      DIMENSION DURATN(100)                                              MTXD
C     *DATA INPUT                                                        MTXD
C     *READ THE PARAMETERS INTERV,MORDER,NFRQCY FROM A PUNCHED CARD      MTXD
C     *INTERV FROM COLUMNS 11-15 IN INTEGER FORMAT, AND                  MTXD
C     *MORDER FROM COLUMNS 16-20 IN INTEGER FORMAT, AND                  MTXD
C     *NFRQCY FROM COLUMNS 21-25 IN INTEGER FORMAT.                      MTXD
    4 FORMAT(10H          3I5)                                           MTXD
      READ  4,                                  INTERV, MORDER, NFRQCYMTXD
      PRINT 4,                                  INTERV, MORDER, NFRQCYMTXD
      GO TO 110                                                          MTXD
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXD  36
C     *SYMMETRIC MATRIX INVERSION BY PARTITION, RECURSIVE ENLARGEMENT = 2MTXD  37
C     *(AAABHH) (WWWXOO) (IOOOOO) (WWWXOO) (AAABHH)                      MTXD  38
C     *(AAABHH) (WWWXOO) (OIOOOO) (WWWXOO) (AAABHH)  DEVELOP  (1) AW+BY=IMTXD  39
C     *(AAABHH).(WWWXOO)=(OOIOOO)=(WWWXOO).(AAABHH)    THE    (2) AX+BZ=OMTXD  40
C     *(CCCDHH) (YYYZOO) (OOOIOO) (YYYZOO) (CCCDHH)   BASIC   (3) YA+ZC=OMTXD  41
C     *(HHHHHH) (OOOOOO) (OOOOOO) (OOOOOO) (HHHHHH) EQUATIONS (4) YB+ZD=IMTXD  42
C     *(HHHHHH) (OOOOOO) (OOOOOO) (OOOOOO) (HHHHHH)                      MTXD  43
C     *THE SLASH (/), USED AS A PREFIX, HERE DENOTES INVERSION           MTXD  44
C     *DEFINING E=/AB                                                    MTXD  45
C     *    AND F=C/A                                                     MTXD  46
C     *YIELDS 1 Z=/(D-FB)=/(D-CE)           WHERE SYMMETRY APPLIES FOR MTXD  47
C     *   AND 2 X=-EZ                                    A,D,W,Z         MTXD  48
C     *   AND 3 Y=-ZF                   AND WHERE MUTUAL TRANSPOSES AREMTXD  49
C     *   AND 4 W=/A-EY=/A-XF                     (B,C),(E,F),(X,Y)      MTXD  50
C     *HENCE THE COMPUTATIONAL PROCEDURE TO DEVELOP (WXYZ) FROM (ABCD) ISMTXD  51
C     *1. INITIALLY COMPUTE /A FROM A, AND THEN REITERATE STEPS 2-8      MTXD  52
C     *2. MOVE D TO Z                                                    MTXD  53
C     *3. COMPUTE F=C/A AND STORE AT Y                                   MTXD  54
C     *4. COMPUTE Q=(/Z)=D-FB AT Z                                       MTXD  55
C     *5. INVERT  Z=/Q IN PLACE       (INVERT ONE HALF, COPY OTHER HALF)MTXD  56
C     *6. COMPUTE X=-EZ                                                  MTXD  57
C     *7. COMPUTE W=/A-XF            (COMPUTE ONE HALF, COPY OTHER HALF)MTXD  58
C     *8. COPY Y FROM X, WHENCE (WXYZ) IS /A FOR NEXT ITERATION          MTXD  59
C     *RESTRICT ALL PARTITIONS D,Z TO ORDER 2 BY INITIAL PARTITION OF A,WMTXD  60
C     *AT ORDER 1-OR-2 AS ORDER OF TOTAL MATRIX IS ODD-OR-EVEN, THEREUPONMTXD  61
C     *INCREASING ORDER OF A,W BY 2 FOR EACH REITERATION OF STEPS 2-8.   MTXD  62
C     *BEGIN AND END WITH MATRIX VATRIX(-,-) OF ORDER NATRIX             MTXD  63
C     *END INVERSION WITH MATRIX WATRIX(-,-) OF ORDER NATRIX             MTXD  64
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
   11 CONTINUE
C     *MATRIX=1,2 AS NATRIX=ODD,EVEN
      MATRIX=2-(NATRIX-(NATRIX/2)*2)
C     *MAIN LOOP - EXECUTE (N+2-M)/2 TIMES, ADVANCING INDEX KATRIX BY 2
      IF(NATRIX)                                          92,    92,    28
   28 DO     90 KATRIX=MATRIX,NATRIX,2
C     *OBSERVE KATRIX IS ORDER OF (ABCD),(WXYZ). LATRIX IS ORDER OF A,W
      LATRIX=KATRIX-2
C     *OBSERVE KLTRIX IS AVERAGE OF LATRIX AND KATRIX, BUT GREATER THAN 0
      KLTRIX=(LATRIX+KATRIX)/2
      IF(KLTRIX)                                          90,    30,    31
   30 KLTRIX=1
   31 CONTINUE
C     *MOVE D TO Z (A TO W FOR FIRST PASS (AND INVERT A IF KATRIX=1))
      DO     33 KMTRIX=KLTRIX,KATRIX
      DO     33 KNTRIX=KLTRIX,KATRIX
C     *ORDER OF MOVEMENT IS (K-1,K-1),(K-1,K),(K,K-1),(K,K)    (K=KATRIX)
   33 WATRIX(KMTRIX,KNTRIX)= VATRIX(KMTRIX,KNTRIX)
C     *INVERT A FOR KATRIX=1, THEN SKIP TO END OF LOOP
      IF(KLTRIX-KATRIX)                                   39,    35,    90
   35 IF(WATRIX(1,1))                                     36,    92,    36
   36 WATRIX(1,1)=1./WATRIX(1,1)
      GO TO     90
C     *SEPARATE CASE KATRIX=2 FOR INVERSION OF A (SKIP FORMING F,Z)
   39 IF(LATRIX)                                          90,    51,    40
C     *COMPUTE F=C/A
   40 DO     45 IATRIX=      1,LATRIX
      DO     45 KMTRIX=KLTRIX,KATRIX
      WATRIX(KMTRIX,IATRIX)=0.
C     *ACCUMULATE ONE ELEMENT OF F=C/A
      DO     45 JATRIX=      1,LATRIX
   45 WATRIX(KMTRIX,IATRIX)= WATRIX(KMTRIX,IATRIX)+VATRIX(KMTRIX,JATRIX)
  451                                          *WATRIX(JATRIX,IATRIX)
C     *COMPUTE Q=D-FB ONLY ALONG AND BELOW THE DIAGONAL
      DO     49 KNTRIX=KLTRIX,KATRIX
      DO     49 KMTRIX=KNTRIX,KATRIX
C     *ORDER OF ACCUMULATION IS (K-1,K-1),(K,K-1),(K,K)
      DO     49 IATRIX=      1,LATRIX
   49 WATRIX(KMTRIX,KNTRIX)= WATRIX(KMTRIX,KNTRIX)-WATRIX(KMTRIX,IATRIX)
  491                                          *VATRIX(IATRIX,KNTRIX)
C     *FINISH Q BY SYMMETRY
   50 WATRIX(KLTRIX   ,KATRIX)=WATRIX(KATRIX,KLTRIX   )
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXD 108
C    *INVERT Q (INVERT D=A FOR KATRIX=2) BY ADJOINTS                   MTXD 109
C    *PREPARE INVERSION BY INTERCHANGING DIAGONAL ELEMENTS THRU (K,K-1) MTXD 110
  51 WATRIX(KATRIX  ,KLTRIX  )=WATRIX(KATRIX  ,KATRIX  )               MTXD 111
     WATRIX(KATRIX  ,KATRIX  )=WATRIX(KLTRIX  ,KLTRIX  )               MTXD 112
     WATRIX(KLTRIX  ,KLTRIX  )=WATRIX(KATRIX  ,KLTRIX  )               MTXD 113
C    *STORE DETERMINANT IN LOWER LEFT CORNER AND RECOVER BY SYMMETRY    MTXD 114
C    *COMPUTE DETERMINANT AS ((K-1,K-1)(K,K)-((K-1,K)(K-1,K)-(0)))      MTXD 115
     WATRIX(KATRIX,KLTRIX  )=0.                                        MTXD 116
     DO     58 IATRIX=KLTRIX,KATRIX                                    MTXD 117
     JATRIX=KATRIX+KLTRIX-IATRIX                                       MTXD 118
  58 WATRIX(KATRIX,KLTRIX) =  WATRIX(KLTRIX,JATRIX)                    MTXD 119
  581                        *WATRIX(IATRIX,KATRIX)                    MTXD 120
  582                        -WATRIX(KATRIX,KLTRIX)                    MTXD 121
C    *COMPUTE INVERSE (EXCEPT (K,K-1))                                 MTXD 122
     IF(WATRIX(KATRIX,KLTRIX  ))                     60,   92,   60MTXD 123
C    *NEGATE (K-1,K)                                                   MTXD 124
  60 WATRIX(KLTRIX,KATRIX)=-WATRIX(KLTRIX,KATRIX)                      MTXD 125
     DO     66 KMTRIX=KLTRIX,KATRIX                                    MTXD 126
     DO     66 KNTRIX=KMTRIX,KATRIX                                    MTXD 127
  66 WATRIX(KMTRIX,KNTRIX)= WATRIX(KMTRIX,KNTRIX)                      MTXD 128
  661                      /WATRIX(KATRIX,KLTRIX  )                    MTXD 129
C    *RECOVER BY SYMMETRY                                              MTXD 130
  67 WATRIX(KATRIX,KLTRIX  )=WATRIX(KLTRIX  ,KATRIX)                   MTXD 131
C    *SEPARATE CASE, SKIP TO END OF LOOP AFTER INVERTING A FOR KATRIX=2 MTXD 132
  69 IF(LATRIX)                                      90,   90,   70MTXD 133
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXD 134
C     *COMPUTE X=-EZ BY TRANSPOSING Y=-ZF                                MTXD 135
   70 DO      77 IATRIX=      1,LATRIX                                    MTXD 136
      DO      77 KNTRIX=KLTRIX,KATRIX                                     MTXD 137
      WATRIX(IATRIX,KNTRIX)=0.                                            MTXD 138
      DO      77 KMTRIX=KLTRIX,KATRIX                                     MTXD 139
   77 WATRIX(IATRIX,KNTRIX)= WATRIX(IATRIX,KNTRIX)-WATRIX(KNTRIX,KMTRIX)MTXD 140
   771                                       *WATRIX(KMTRIX,IATRIX)MTXD 141
C     *COMPUTE W=/A-XF AND FORM Y AS TRANSPOSE OF X                       MTXD 142
      DO      88 IATRIX=1,LATRIX                                          MTXD 143
      DO      85 JATRIX=IATRIX,LATRIX                                     MTXD 144
      DO      84 KNTRIX=KLTRIX,KATRIX                                     MTXD 145
C     *ACCUMULATE -XF INTO /A, ONLY ALONG AND ABOVE MAJOR DIAGONAL        MTXD 146
   84 WATRIX(IATRIX,JATRIX)= WATRIX(IATRIX,JATRIX)-WATRIX(IATRIX,KNTRIX)MTXD 147
   841                                       *WATRIX(KNTRIX,JATRIX)MTXD 148
C     *COMPLETE W BY SYMMETRY                                             MTXD 149
   85 WATRIX(JATRIX,IATRIX)= WATRIX(IATRIX,JATRIX)                        MTXD 150
      DO      88 KNTRIX=KLTRIX,KATRIX                                     MTXD 151
C     *TRANSPOSE ONE ELEMENT OF X TO Y                                    MTXD 152
   88 WATRIX(KNTRIX,IATRIX)= WATRIX(IATRIX,KNTRIX)                        MTXD 153
C     *END OF MAIN LOOP                                                   MTXD 154
   90 CONTINUE                                                            MTXD 155
      GO TO (   127,   170), ISWTCH                                       MTXD 156
C     *ALARMS                                                             MTXD 157
   92 DURATN(NORDER)=-1.                                                  MTXD 158
      IF(ISWTCH-2)                                 130,   210,   210MTXD 159
C     *END OF MATRIX INVERSION PROGRAMMING                                MTXD 160
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
  110 CONTINUE
C    *PRINCIPAL ITERATIVE LOOP ON NORDER=1,2,3...MORDER
      DO  222  NORDER=1,MORDER
      TORDER=NORDER
C    *CLEAR DURATN TABLE AND SET ERROR RECORD = 0.
      ERROR=0.
      DURATN(NORDER)=-TORDER
C    *SKIP CASES FOR WHICH NORDER IS NOT DIVISIBLE BY INTERV
      IF(NORDER-(NORDER/INTERV)*INTERV)                 222,    115,    222
  115 DURATN(NORDER)=0.
C    *CREATE A(I,J)
      DO    119 IORDER=1,NORDER
      DO    119 JORDER=1,IORDER
      KORDER=IORDER*JORDER
      VATRIX(IORDER,JORDER)=KORDER
      VATRIX(IORDER,JORDER)=SINF(VATRIX(IORDER,JORDER)/TORDER)
      IF(IORDER-JORDER)                                 119,    118,    119
  118 VATRIX(IORDER,JORDER)=VATRIX(IORDER,JORDER)+TORDER+1.
  119 VATRIX(JORDER,IORDER)=VATRIX(IORDER,JORDER)
C    *INVERSION LOOP SETUP
      NATRIX=NORDER
      ISWTCH=1
C    *CLOCK ON     INTERROGATE CLOCK HERE
      TIMEON=CLOCKF(DUMMY)
C    *LOOP NFRQCY ITERATIONS
      DO    127 NITRTN=1,NFRQCY
      GO TO 11
  127 CONTINUE
C    *CLOCK OFF    INTERROGATE CLOCK HERE
      TIMEUP=CLOCKF(DUMMY)
C    *COMPUTE DURATION  DURATN(NORDER) = CLOCK SPAN
      DURATN(NORDER)=TIMEUP-TIMEON
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
C     *LEAST SQUARES COMPUTATION.
C     *DENOTE S(K) = SUM OF ((I**K)              ) OVER I = ORDERS EMPLOYED
C     *AND    T(K) = SUM OF ((I**K)*(DURATN(I))) OVER I = ORDERS EMPLOYED
C     *AND CUBIC = A(3)*(X**3) + A(2)*(X**2) + A(1)*X + A(0) WHERE THE
C     *VARIABLE X DENOTES DURATION.
C     *SOLVE       (S(0) S(1) S(2) S(3))       (A(0))       (T(0))
C     *THIS        (S(1) S(2) S(3) S(4))   *   (A(1))   =   (T(1))
C     *SYSTEM      (S(2) S(3) S(4) S(5))       (A(2))       (T(2))
C     *USING       (S(3) S(4) S(5) S(6))       (A(3))       (T(3))
C     *STORAGE   VATRIX(1,1)-VATRIX(4,4)   VATRIX(8,-)   VATRIX(-,8)
C     *CREATE NORMAL MATRIX AND VECTOR AND CLEAR COEFFICIENT VECTOR
  130 DO  133  IORDER=1,7
      VATRIX(IORDER,7)=0.
      VATRIX(IORDER,8)=0.
  133 VATRIX(8,IORDER)=0.
C     *USE KORDER TO COUNT VALID DATA CASES
      KORDER=0
      DO   159 IORDER=1,NORDER
C     *SKIP IRRELEVANT CASES (DURATN(IORDER) NEGATIVE)
      IF(DURATN(IORDER))                                  159,  142,  142
  142 KORDER=KORDER+1
C     *DENOTE POWRDR = POWERS OF ORDER VARIABLE IORDER
      POWRDR=1.
C     *SCALE (NORMALIZE) INDEPENDENT TIME VARIABLE BY DIVISION BY NORDER.
      XORDER=IORDER
      YORDER=NORDER
      XORDER=XORDER/YORDER
      DO   158 JORDER=1,7
      IF(JORDER-1)                                        222,  153,  152
  152 POWRDR=POWRDR*XORDER
  153 IF(JORDER-4)                                        154,  154,  158
  154 VATRIX(JORDER,8)=VATRIX(JORDER,8)+POWRDR*DURATN(IORDER)
  158 VATRIX(JORDER,7)=VATRIX(JORDER,7)+POWRDR
  159 CONTINUE
C     *SKIP CASES WITH LESS THAN 4 DATA POINTS
      IF(KORDER-4)                                        210,  161,  161
  161 DO  168  IORDER=1,4
      DO  168  JORDER=1,IORDER
      KORDER=IORDER+JORDER
      VATRIX(IORDER,JORDER)=VATRIX(KORDER-1,7)
  168 VATRIX(JORDER,IORDER)=VATRIX(IORDER,JORDER)
C     *INVERT NORMAL MATRIX
      NATRIX=4
      ISWTCH=2
      GO TO 11
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM
  170 CONTINUE
C     *OBTAIN COEFFICIENTS AND MEASURE ERROR
C     *DESCALE THE TIME VARIABLE BY DIVIDING THE COEFFICIENTS BY NORDER.
C     *NORMALIZE THE COEFFICIENTS TO REPRESENT DATA FOR NFRQCY = 1
      ZORDER=NFRQCY
      DO   200 IORDER=1,4
      LORDER=5-IORDER
      DO   199 JORDER=1,4
      XORDER=0.
      DO   193 KORDER=1,4
  193 XORDER=XORDER+VATRIX(IORDER,KORDER)*WATRIX(KORDER,JORDER)
      IF(IORDER-JORDER)                                    196,  195,  196
  195 XORDER=XORDER-1.
  196 XORDER=ABSF(XORDER)
      IF(ERROR-XORDER)                                     198,  199,  199
  198 ERROR=XORDER
  199 VATRIX(8,LORDER)=VATRIX(8,LORDER)+WATRIX(IORDER,JORDER)
  1991                            *VATRIX(JORDER,8)
  1992                            /ZORDER
  200 ZORDER=ZORDER*YORDER
C     *PRINT
  201 FORMAT(10H           7F15.9)
  210 PRINT   201, TORDER, DURATN(NORDER), (VATRIX(8,I),I=1,4), ERROR
C     *END OF PRINCIPAL ITERATIVE LOOP
  222 CONTINUE
      CALL SYSTEM
      END
```

```
C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM MTXD 268
       LOAD BATCH                                                     MTXD 269
       REMARK  THE CLOCK INTERROGATICN PROGRAM IS LOADED HERE         MTXD 270
       TRA                                                            MTXD 271
     DATA     10  100     1                                           MTXD 272
       REMARK  END OF RUN                                             MTXD 273
```