UNIVERSITY OF ILLINOIS

DIGITAL COMPUTER

LIBRARY ROUTINE M 20 - 234

By C. W. Gear

TITLE	Eigenvalues of a Symetric Matrix by Givens' Method (SADOI Only)
TYPE	Automatic
NUMBER OF WORDS	507 + Routines N12, R1, P16 and Y1
MAXIMUM SIZE OF MATRIX	128 x 128
DURATION	Approximately $n^2/160 + n^3/20{,}000$ minutes
ACCURACY	Depends on matrix. So far as known, good to 11 decimal places for n less than 40 and good to 10 otherwise.
SCALING	The matrix must be scaled so that its norm is less than 1. $\sum_{ij} a_{ij}^2 < 1$
CODED STOPS	FF 036 in loc 322 on R.H.S. Sum check failed on read in. FF 037 in loc 1KK Norm of Matrix has exceeded 1. FF 038 in loc 074 Sum of squares check has failed. (see description)
FINAL STOP	24 014 on R.H.S. of loc 1ON. Routine is now ready to read in another matrix.
INPUT	Normal input the program tape (it uses SADOI). It will stop on the word 34 011 FF 036 in loc 322 (16), on the L.H.S. if the tape has read in correctly, on the R.H.S. if there has been a reader error. Black switch the data tape in which should be written as follows: Consider the matrix in its upper triangular form as

$$a_{11}\ a_{12}\ a_{13}\cdots\cdots\cdots\cdots\cdots a_{1n}$$
$$a_{22}\ a_{23}\cdots\cdots\cdots\cdots\cdots a_{2n}$$
$$\cdots\cdots\cdots\cdots\cdots$$
$$a_{n-1,n-1}\ a_{n-1,n}$$
$$a_{n,n}$$

Each row should be punched, element by element as a sign followed by p decimal digits ($p \leq 12$) of the element, and

at the end of each row the character N should be punched. Thus the data tape will have the following sequence on it: n signed numbers, N, n-1 signed number, N, ... 3 signed numbers, N, 2 signed numbers, N, 1 signed number, N. The master routine will read in the first row of the matrix and plant n, the order of the matrix. It will then read in up to each N character, and store this as the next row of the triangular part of the matrix. If there are more than n-1 numbers in this row it will ignore all those after the n-1th, if there are less than n-1, say p, the p+1th will be the p+2th of the preceeding row, and so on to the n-1th interpreted as the nth of the preceeding row. It will stop reading in after it has sensed n N's.

MODIFICATION

This routine uses N12 to input the matrix. If it is desired to calculate the matrix elements within the Illiac prior to finding the eigenvalues the routine may be modified in the following way:

Replace N12 by an auxiliary subroutine to calculate the elements in the order in which they are read in. N12 is entered with the orders

```
q    50  p
     50  q
     ------
     26 365F
```

It occupies locations 365 through 403. These and the additional locations 670 through 869 may be used by the auxiliary subroutine. At no time must the auxiliary routine use other locations (except 0, 1, and 2.) Successive rows of the upper triangular form should be calculated and stored in Williams memory locations p through 870+n-1. p is 870, 871, 872, ..., 870+n-1; on each successive entry. After the first entry to the auxiliary subroutine location 386 must contain 870 + n as a left hand address, where n is the order of the matrix.

If required, library routine R1 is in location 356, P16 in location 277 and Y1 in location 23. Drum locations 11720 through 12799 may be used.

To input the auxiliary subroutine delete the sum check at the end of the master routine, i.e. copy up to and including the word 00F 00F in location 668, and delete the short last section starting with 00800K. Follow this with the auxiliary routine with a suitable directive.

Entry should be made to the left hand side of location 17. The read-in is under the control of SADOI. The only floating addresses used in the master routine are (P16), (N12), (R1) and (Y1). It will finally stop, as before, on a 24 014, and if this is obeyed it will re-enter the main routine, while the auxiliary routine may specify a new matrix (of any order less than 129).

DESCRIPTION

For full details see the paper by Wallace Givens on this subject in the library files.

The program falls into two parts, the reduction by a series of rotations into Jacobi (bordered diagonal or Tri-diagonal) form, and the subsequent calculation of the eigenvalues from this form by the use of a binary chopping procedure involving the calculation of the signature of the matrix at each point of the chop.

<u>Reduction to Jacobi Form.</u>

Consider a rotation of the matrix about the (i,j) element defined by $\underline{P}'_{ij}\underline{A}\,\underline{P}_{ij} = \bar{\bar{A}}$: Where $\underline{P}_{ij}$ =

$$
\begin{array}{c}
\\ \\ \\ i \\ \\ \\ \\ j \\ \\ \\ \\
\end{array}
\left\Vert
\begin{array}{ccccccccccccc}
1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 & & & c_{ij} & \cdot & \cdot & \cdot & \cdot & \cdot & -s_{ij} & & & \\
 & & & & 1 & & & & & & & & \\
 & & & & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & & & \\
 & & & & & & & 1 & & & & & \\
 & & & s_{ij} & & & & & & c_{ij} & & & \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 1 & 0 \\
 & & & & & & & & & & & 0 & 1
\end{array}
\right\Vert
\qquad c_{ij}^{2} + s_{ij}^{2} = 1
$$

The only elements affected in this transformation are those in the ith and jth rows and columns of $\underline{A}$. Since the matrix is symetric, and this property is retained under an orthogonal rotation, we need only consider the columns of $\underline{A}$.

The elements of the rotated matrix $\bar{\underline{A}}$ are thus:

$$\bar{a}_{p,q} = a_{p,q} \quad p, q \neq i, j.$$

$$\bar{a}_{i,p} = \bar{a}_{p,i} = c_{ij} \cdot a_{pi} + s_{ij} \cdot a_{pj} \quad p \neq i,j.$$

$$\bar{a}_{j,p} = \bar{a}_{p,j} = c_{ij} \cdot a_{pj} + s_{ij} \cdot a_{pi} \quad p \neq i,j.$$

$$\bar{a}_{i,i} = c_{ij}(c_{ij} \cdot a_{ii} + s_{ij} \cdot a_{ij}) + s_{ij}(c_{ij} \cdot a_{ij} + s_{ij} \cdot a_{jj})$$

$$\bar{a}_{ij} = \bar{a}_{ji} = c_{ij}(c_{ij} \cdot a_{ij} + s_{ij} \cdot a_{ii}) + s_{ij}(c_{ij} \cdot a_{jj} - s_{ij} \cdot a_{ij})$$

Since the trace of a matrix is invariant under an orthogonal transformation,

$$\bar{a}_{jj} = a_{ii} + a_{jj} - \bar{a}_{ii} .$$

The values of c_{ij} and s_{ij} have not yet been assigned. In the Jacobi method, also in the library, they are chosen so as to make the (i,j) element zero. However, future rotations about other elements alter this, and iteration is necessary.

Suppose that rotations are made successively about the elements (2,3):(2,4):...(2,n):(3,4):(3,5)...(3,n):(4,5)... (n-1,n); making at each step the elements (1,3):(1,4):...(1,n):(2,4):(2,5):...(2,n):(3,5)...(n-2,n) zero. At any step, elements made zero in previous rotations are left zero.

Thus, all the elements of the matrix are reduced to zero except the diagonal and those adjacent to the diagonal.

This leaves the matrix

$$\begin{Vmatrix} a_1 & b_1 & 0 & \cdots & 0 \\ b_1 & a_2 & b_2 & & \\ 0 & b_2 & a_3 & b_3 & \\ & \cdots & \cdots & \cdots & \\ & & b_{n-2} & a_{n-1} & b_{n-1} \\ 0 & \cdots & 0 & b_{n-1} & a_n \end{Vmatrix}$$

with eigenvalues equal to those of the original matrix. Denote this matrix by B.

The eigenvalues of this are the roots of the polynomial

$$\det(B - uI) = f_n(u) = 0$$

From the form of the matrix the polynomial can be defined by the sequence

$$f_0(u) = 1$$

$$f_1(u) = a_1 - u$$

$$f_2(u) = (a_2 - u)f_1(u) - b_1^2 f_0(u)$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \qquad (1)$$

$$f_k(u) = (a_k - u)f_{k-1}(u) - b_{k-1}^2 f_{k-2}(u)$$

$$k = 2,3,\ldots,n.$$

If two successive f_i's vanish then $f_n = 0$.
To overcome this, set $f_k = -b_{k-1}^2$ if $f_{k-1} = f_{k-2} = 0$ and $b_{k-1} \neq 0$ and $f_k = a_k - u$ if $f_{k-1} = b_{k-1} = 0$.
Then the number of roots of $f_n(x) = 0$ greater than u is

equal to the number of agreements in sign (zero considered as positive) in the sequence $f_0(u)$, $f_1(u)$, $f_2(u)$, $\ldots\ldots f_n(u)$. For accuracy the equations are formed in semi-floating point.

b_i^2 is formed and scaled so that:

$$0 \leq b_i^2 \cdot 2^{-\epsilon i} q_i < 2^{-40}$$

or $q_i = 0$ and $\epsilon_i = 79$ if $b_i \neq 0$

$$\frac{1}{2} \leq q_i < 1$$

and $a_i \cdot u$ is scaled so that:

$$a_i \cdot u = p_i 2^{-\delta i}, \quad 0 \leq \delta_i \leq 39; \quad \frac{1}{2} \leq p_i < 1$$

and $f_i(u) = 2^{-\pi i} n_i$

If π_0 is set as 0, and $v_i = \pi_i - \pi_{i-1}$ the machine equations are in the form:

$$m_i = 2^{\Sigma i}(p_i m_{i-1} - 2^{\Delta i} q_{i-1} m_{i-2}) \quad \text{if } \Delta_i < 0$$

$$\Sigma_i = v_i - \delta_i$$

$$= 2^{\Sigma i}(2^{-\Delta i} p_i m_{i-1} - q_i m_{i-2}) \quad \text{if } \Delta_i \geq 0$$

$$\Sigma_i = v_i + v_{i-1} - \epsilon_{i-1}$$

where $\Delta_i = v_{i+1} + \delta_i - \epsilon_{i-1}$

Σ_i is found by scaling the quantity in parenthesis to lie between 1/2 and 1, and v_i is found from the appropriate equation.

THE CODE

The program is read into the memory and stored on the drum for future use as temporary storage overwrites much of it. The triangular part of the matrix is read in row by row and packed on the drum according to the following arrangement: Consider a track of length 65 words,

location	+ 0	1	2	3	..... n-64	64
2560	a_{11}	0	$a_{66,66}$	$a_{66,67}$	$a_{66,n}$	0
2625	a_{12}	a_{22}	0	$a_{67,67}$	$a_{67,n}$	0
2690	a_{13}	a_{23}	a_{33}	0		
	. . . . .					
2560 + 64x65	$a_{1,65}$	$a_{2,65}$	. . . . .			$a_{65,65}$
2560 + 65x65	$a_{1,66}$	$a_{2,66}$	. . . . .			$a_{65,66}$
	. . . . .					
2560 + (n-1)65	$a_{1,n}$	$a_{2,n}$	.....		.....	$a_{65,n}$

If two locations A and B are such that loc(B) - loc(A) = 1(mod 64) then B can be utilised in the minimum access time after A has been referred to.

If p is greater than 65, then $a_{65,p}$ is in location 2560 + 64x66 + 65(p - 65), and $a_{66,p}$ is in location p - 64 + 2560.

Therefore $\text{loc}(a_{66,p}) - \text{loc}(a_{65,p}) \bmod 64$

$$= -64\text{x}66 - 65(p-65) + p - 64 \ (64)$$

$$= 65^2 - 64p \ (64)$$

$$= (64 + 1)^2 \ (64)$$

$$= 1 \ (64)$$

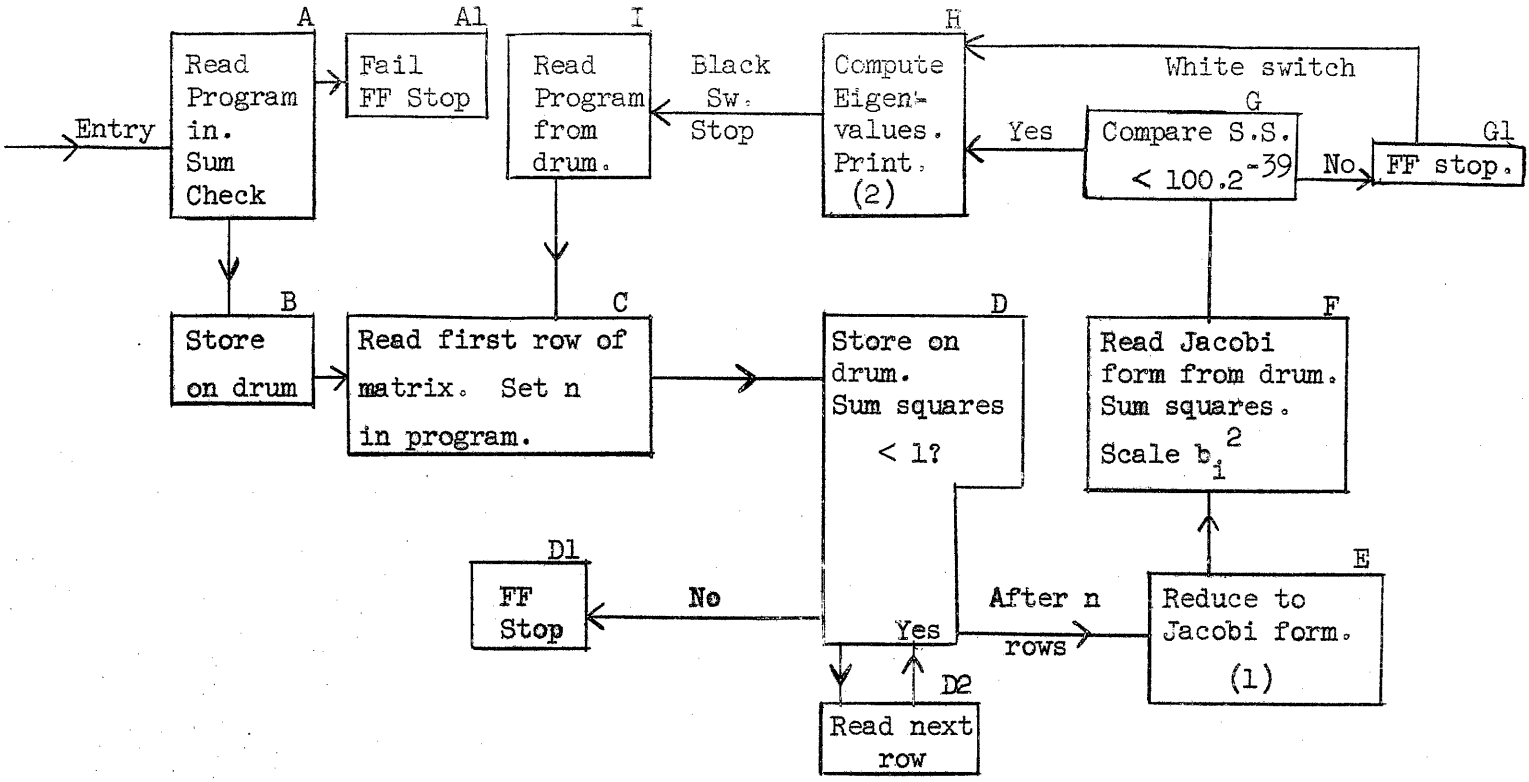
Therefore the rows and columns of the matrix can be referred to in minimum access time.
The sum of the squares of the elements of the matrix is calculated as it is read in, and stored as a check against the sum after the reduction to Jacobi form. After this reduction, the program will hang up if the difference exceeds 100×2^{-39}.
The eigenvalues are then found from the reduced form, by storing the square root R of the sum of squares of the original matrix as an upper and lower bound on the absolute size of the eigenvalues and adjusting these until they converge with information gained about the position of the roots by the method outlined.
Taking P(u) to be the number of eigenvalues greater than u, the method can best be seen from a flow chart.

STORAGE ALLOCATION

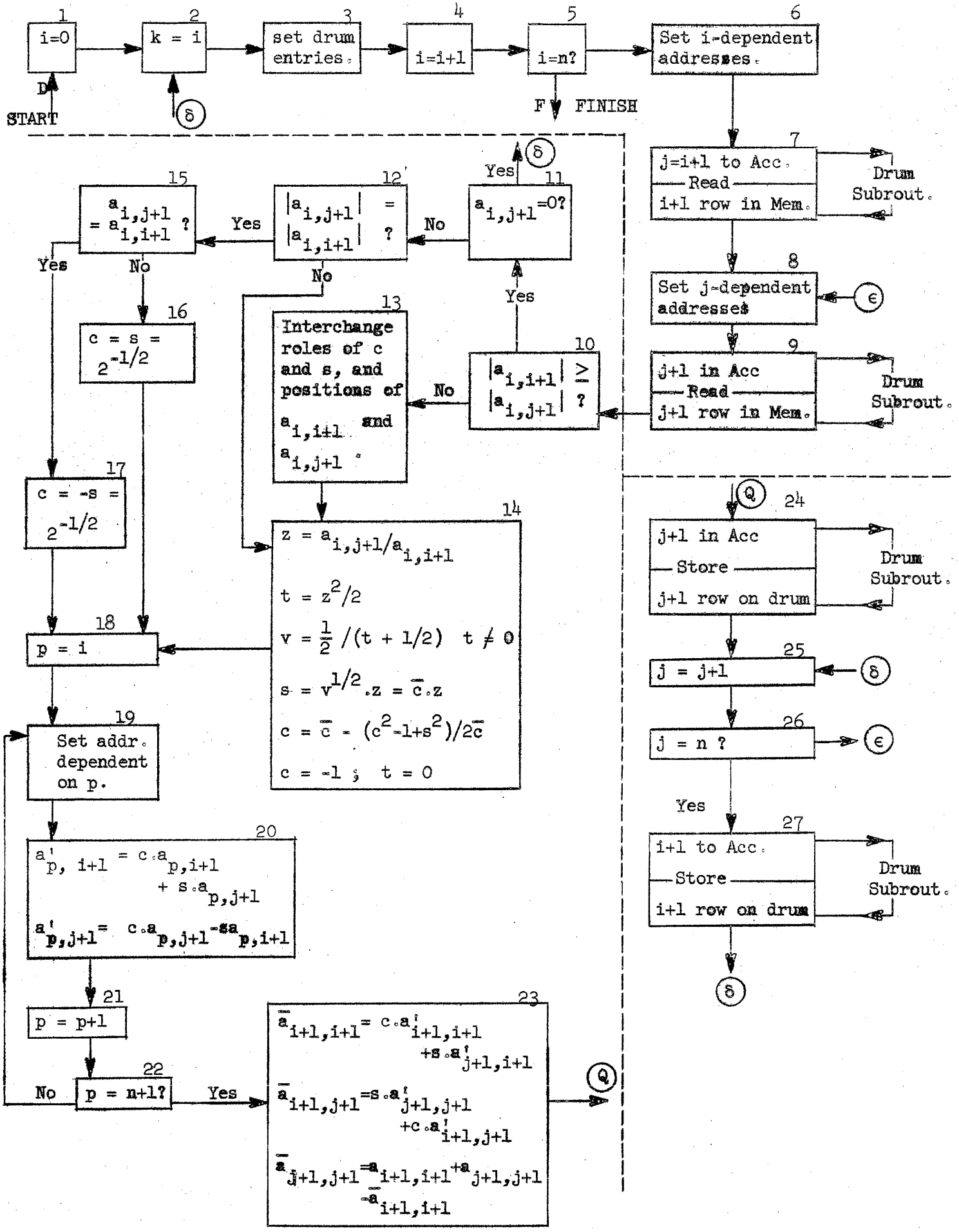
The program occupies locations 17 through 668 and also on the drum. During input of the matrix locations 870 - 997 are used to store the rows. During reduction to Jacobi form locations 742 - 869 and 870 - 997 are used as temporary storage for the i+1 and j+1th rows of the matrix which are altered in a rotation about the i+1, j+1 element. During the calculation of the eigenvalues the a_i's are stored in 742-869, the scaled b_i^2 in locations 870 - 997 with their scaling factors in 614 - 741. Locations 358 - 485 hold the lower bounds of the eigenvalues, while locations 486 - 613 hold the upper values.

DATE June 27, 1957 RT:1/23/59
PROGRAMMED BY R. W. Gear
APPROVED BY D. E. Muller

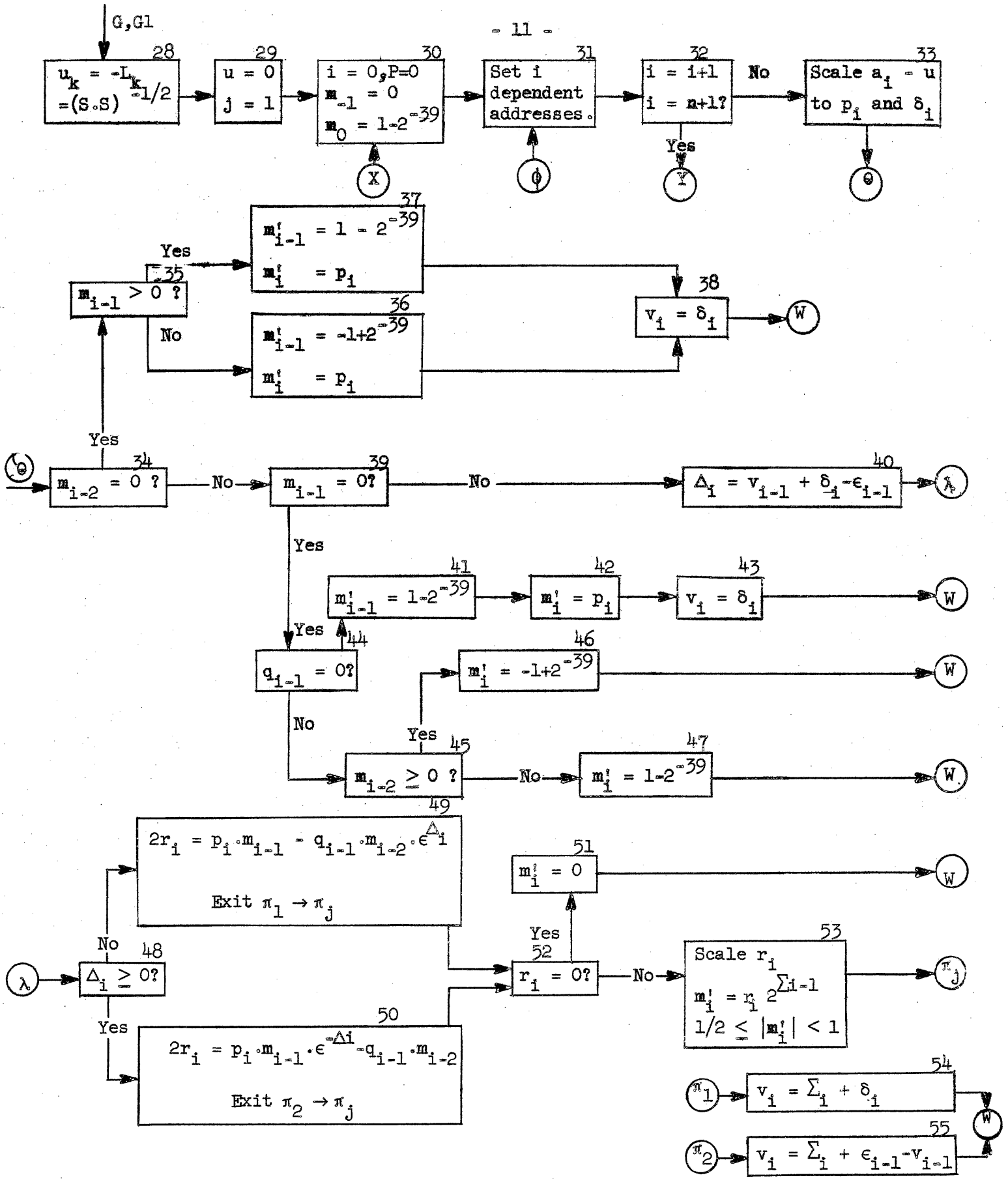
lgr



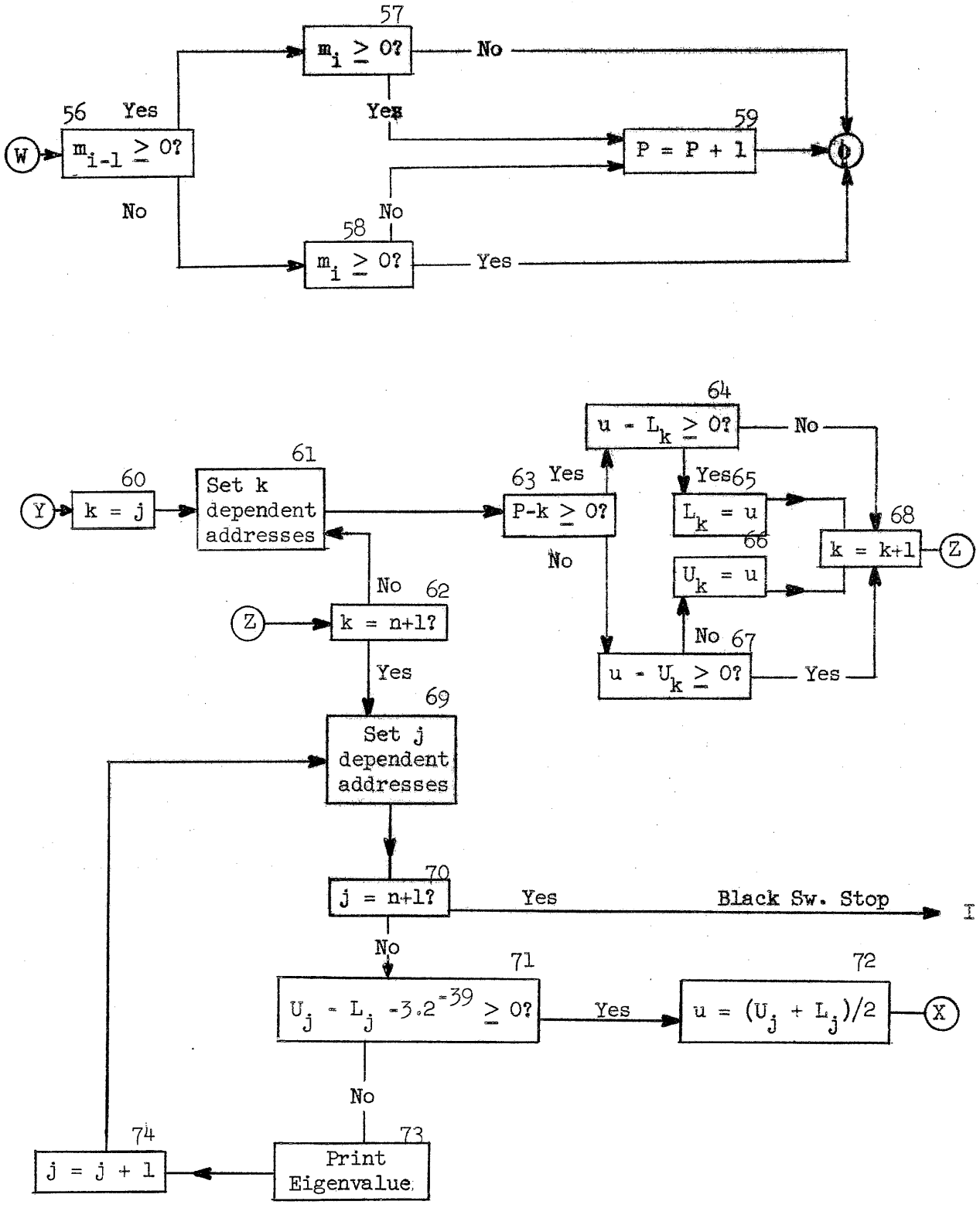
MAIN FLOW CHART



(1) Reduction to Jacobi Form



(2) Eigenvalues. Part I.



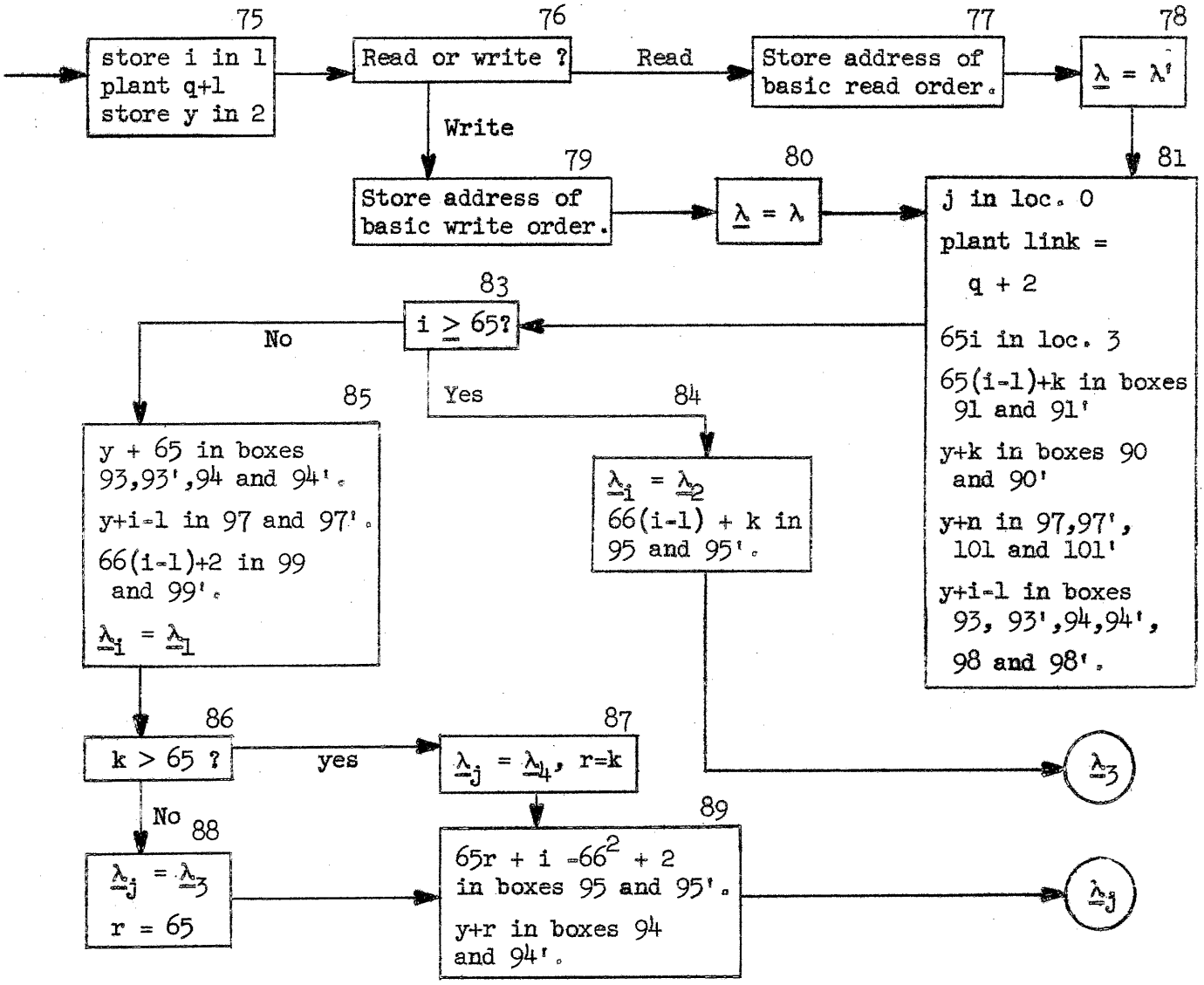
(2) Eigenvalues. Part II.

The Drum Subroutine

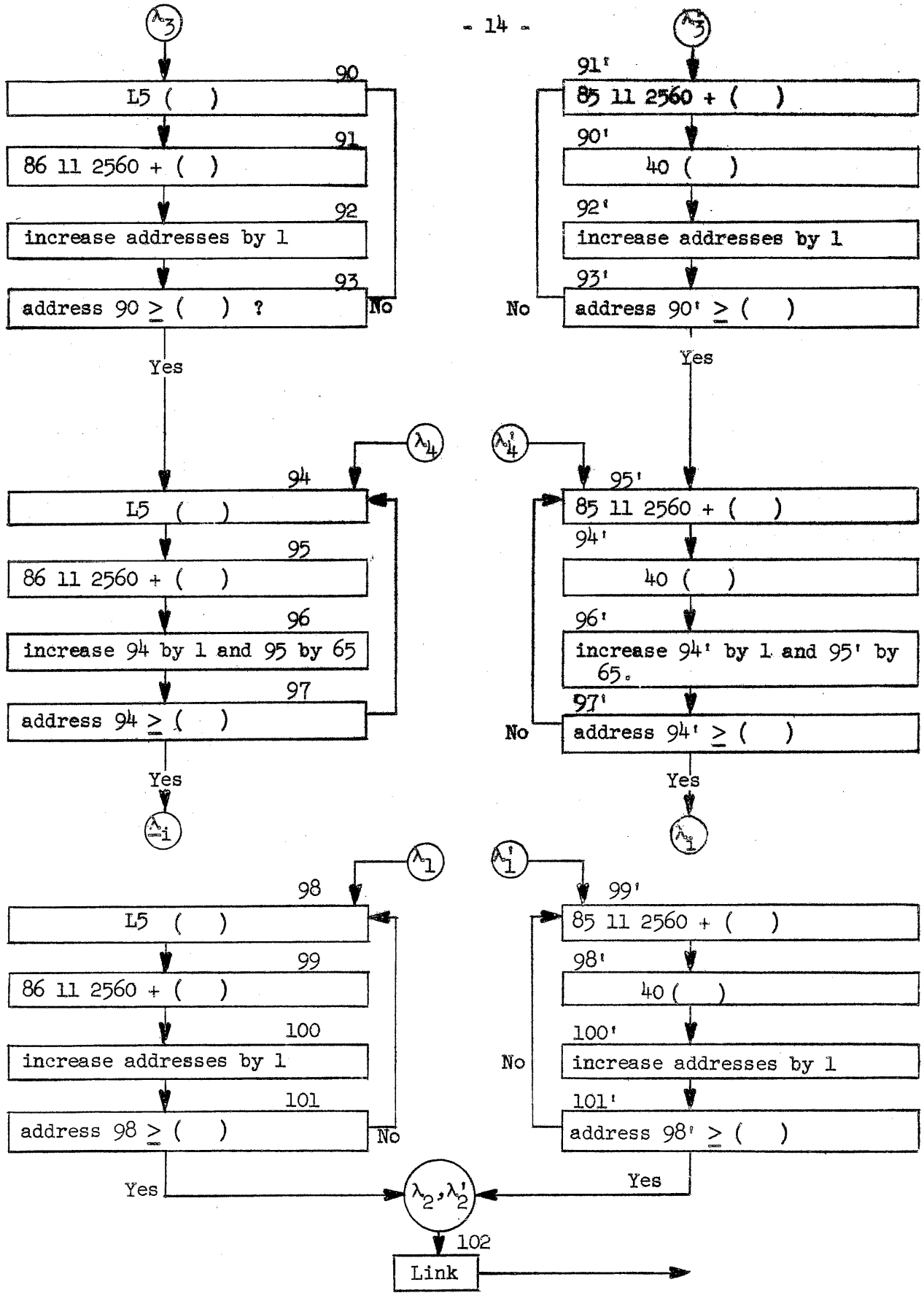
This is an independent subroutine which stores the triangular part of a symetric matrix on the drum as outlined above. Entry is made to read (write) the ith row except for the first j elements into (from) Williams memory locations y+j through y+n with the following orders:

q	5(J)0	y
	50	q
q+1	26	--
	00	k

with i in the accumulator.



The Drum Subroutine. Part I.



The Drum Subroutine. Part II.

LOCATION	ORDER	BOX NO.	NOTES	PAGE 1
	K0 017K			
17	J0 100F	B	Store program on drum.	
	50 17F			
18	26 23F			
	02 2688F			
19	00 840F		Exit to C.	
	26 404F			
20	50 100F	I	Read program from drum.	
	50 20F			
21	26 23F			
	02 2688F			
22	00 840F		Exit to C.	
	26 404F			
	(Y1) 00K			
63	F5 339F		Prepare to read Jacobi form from drum.	
	42 84F			
64	42 85F			
	42 72F			
65	F4 341F			
	42 91F			
66	42 105F			
	42 78F			
67	42 92F			
	F5 84F			
68	L0 341F			
	42 107F			
69	L5 336F			
	40 71F			
70	L4 334F			
	40 77F			
71	00 0F		Read from drum locations 2560, 2625, 2626, 2690,	
	00 0F		2691, ...	
72	22 72F			
	40 0F			

LOCATION	ORDER	BOX NO.	NOTES PAGE 2
73	L5 71F		After 130 elements have been read reset loop
	F4 334F		
74	40 71F		and read 2562, 2563, 2628, 2629, ... to end of
	F4 72F		
75	42 72F		matrix.
	L0 347F		
76	36 83F		Test for end.
	26 77F		
77	00 0F		
	00 0F		
78	22 78F		
	40 0F		
79	L5 77F		
	F4 334F		
80	40 77F		
	F5 78F		
81	42 78F		
	L0 662F		
82	36 71F		Test for 130 elements.
	26 663F		
83	41 10F		Prepare to sum squares and to scale b_i^2's.
	41 11F		
84	41 12F		
	50 0F		
85	L5 11F		
	74 0F		Sum squares of a_i's.
86	L4 10F		
	40 10F		
87	S5 0F		
	40 11F		
88	F5 84F		
	42 84F		
89	42 85F		
	L0 346F		

LOCATION	ORDER	BOX NO.	NOTES
90	32 84F		
	41 13F		
91	22 91F		Square b_i's and scale.
	50 0F		
92	22 92F		Also sum squares.
	75 0F		
93	40 0F		
	S5 0F		
94	40 1F		
	L4 13F		
95	10 39F		
	40 14F		
96	L7 14F		
	L4 0F		
97	L4 12F		
	40 12F		
98	S5 0F		
	40 13F		
99	L3 0F		
	36 128F		
100	41 2F		
	50 1F		
101	L5 0F		
	00 1F		
102	40 0F		
	F5 2F		
103	42 2F		
	L5 0F		
104	32 101F		
	10 1F		
105	L4 338F		
	40 0F		
106	L5 2F		
	F0 335F		

LOCATION	ORDER	BOX NO.	NOTES PAGE 4
107	22 107F		
	40 0F		
108	F5 107F		
	42 107F		
109	F5 91F		
	42 91F		
110	42 92F		
	42 105F		
111	L0 345F		
	32 91F		
112	L5 10F		Sum a_i^2 + twice sum b_i^2 - previous sum of
	L4 12F		squares.
113	L4 12F		
	L0 351F		
114	40 10F		
	L7 354F		
115	L2 10F		
	32 116F		
116	FF 56F		Test to see if it exceeds 100.2^{-39}
	L5 127F		
117	40 614F		Set $b_0^2 = 0$, scaling factor = 79.
	41 870F		
118	F5 269F		
	42 120F		
119	F5 270F		
	42 121F		
120	L5 353F	28	Set upper bound for eigenvalues = square root
	40 0F		of sum of
121	L1 353F		squares, and lower bound = - square root.
	40 0F		
122	F5 120F		
	42 120F		
123	F5 121F		
	42 121F		

LOCATION	ORDER	BOX NO.	NOTES
124	L0 274F		
	36 120F		
125	41 273F	29	u = 0
	F5 273F		
126	40 272F		j = 1
	26 131F		
127	00 0F		Const.
	00 79F		
128	L3 1F		If b_i^2 = 0 set scaling factor = 79.
	32 129F		
129	26 100F		
	L5 127F		
130	40 2F		
	23 105F		
131	41 221F	30	i = 0
	41 5F		m_{-1} = 0
132	41 222F		
	41 3F		
133	L5 352F		
	40 4F		
134	F5 222F	31	$M_0 = 1 - 2^{-39}$, i = 1
	42 222F		
135	L4 339F		
	42 142F		
136	L0 341F		
	42 161F		Set i dependent addresses
137	42 198F		
	L4 341F		
138	L4 341F		
	42 158F		
139	42 206F		
	L0 349F		
140	36 231F	32	i = n+1 ?
	L5 273F		

LOCATION	ORDER	BOX NO.	NOTES PAGE 6
141	10 1F	33	Scale a_i-u
	40 6F		
142	43 150F		
	L5 0F		
143	10 1F		
	L0 6F		
144	40 1F		
	L3 1F		
145	36 177F		
	L5 1F		
146	40 6F		
	F5 150F		
147	42 150F		
	L7 6F		
148	00 1F		
	36 146F		
149	L5 1F		
	50 335F		
150	10 1F		
	00 0F		
151	40 1F		
	L5 150F		
152	L0 275F		
	40 2F		
153	L3 3F	34	m_{i-2} = 0?
	36 216F		
154	L3 4F	39	m_{i-1} = 0?
	32 206F		
155	51 1F		Form $q_{i-1} \cdot m_{i-2}$. Store double precision in
	74 4F		6 and 11
156	10 1F		
	40 6F		
157	S5 0F		
	40 11F		

LOCATION	ORDER	BOX NO.	NOTES PAGE 7
158	51 3F		Form $p_i \cdot m_{i-1}$. Store double precision in
	74 0F		7 and 12
159	10 1F		
	40 7F		
160	S5 0F		
	40 12F		
161	L5 5F	40	$\Delta_i = v_{i-1} + \delta_i + \epsilon_{i-1}$
	L0 0F		
162	L4 2F		
	40 8F		
163	43 193F		
	L7 8F		
164	42 226F		Store as scaling factor
	42 227F		
165	L0 230F		$\Delta_i \geq 40$?
	36 225F		
166	L5 227F		
	40 171F		
167	40 182F		
	L5 8F		
168	36 179F	48	$\Delta_i \geq 0$?
	L5 229F		
169	46 198F		
	L1 12F		
170	10 39F	49	Calculate r_i if $\Delta_i < 0$
	L0 7F		
171	00 0F		
	00 0F		
172	L4 6F		
	40 6F		
173	S5 0F		
	L4 11F		
174	10 39F		
	40 0F		

LOCATION	ORDER	BOX NO.	NOTES
175	L1 0F		
	L4 6F		
176	40 6F		
	26 186F		
177	L5 276F		
	40 2F		
178	41 1F		
	26 153F		
179	L5 228F		
	46 198F		
180	F5 182F		
	42 182F		
181	50 11F	50	Calculate r_i if $\Delta_i \geq 0$
	L5 6F		
182	00 0F		
	00 0F		
183	00 1F		
	L0 7F		
184	40 6F		
	85 0F		
185	L0 12F		
	26 174F		
186	L3 6F	52	$r_i = 0$?
	36 223F		
187	S5 0F		
	40 0F		
188	L5 6F	53	Scale r_i
	40 1F		
189	F5 193F		
	42 193F		
190	L7 1F		
	00 1F		
191	40 1F		
	36 189F		

LOCATION	ORDER	BOX NO.	NOTES
192	L5 6F		
	50 0F		
193	10 1F		
	00 0F		
194	40 3F		
	50 3F		
195	L5 4F		
	40 3F		
196	S5 0F		
	40 4F		
197	L5 193F		
	L0 275F		
198	22 0F		
	L4 0F		
199	L0 5F	55	$v_i = \Sigma_i + \epsilon_{i-1} - v_{i-1}$
	40 5F		
200	22 201F		
	L4 2F		
201	40 5F	54	$v_i = \Sigma_i + \delta_i$
	L5 3F	56	$m_{i-1} \geq 0$?
202	36 205F		
	L5 4F	58	$m_i \geq 0$?
203	36 134F		
	F5 221F	59	P = P + 1
204	42 221F		
	26 134F		
205	L5 4F	57	$m_i \geq 0$?
	32 203F		
206	26 134F		
	L3 0F	44	$q_{i-1} = 0$?
207	32 212F		
	L5 3F		
208	32 210F	45	$m_{i-2} \geq 0$?
	41 3F		

LOCATION	ORDER	BOX NO.	NOTES PAGE 10
209	L5 352F	47	$m_i' = 1-2^{-39}$
	40 4F		
210	22 201F		
	41 3F		
211	L1 352F	46	$m_i' = -1+2^{-39}$
	40 4F		
212	22 201F		
	L5 1F	42	$m_i' = p_i$
213	40 4F		
	L5 352F	41	$m_{i-1}' = 1-2^{-39}$
214	40 3F		
	L5 2F	43	$v_i = \delta_i$
215	40 5F		
	22 201F		
216	L1 4F	35	$m_{i-1} > 0$?
	32 217F		
217	22 212F		
	L1 352F		
218	40 3F		
	L1 1F		
219	40 4F	36	$m_{i-1}' = -1+2^{-39}$; $m_i' = -p_i$
	L5 2F	38	$v_i = \delta_i$
220	40 5F		
	22 201F		
221	00 0F		P
	00 0F		
222	00 0F		Counter for sturm sequence. (i)
	00 0F		
223	L5 4F	51	$m_i' = 0$
	40 3F		
224	41 4F		
	22 201F		
225	L5 226F		If scaling factor of Δ_i exceeds 40 avoid
	22 166F		subtraction in boxes 49 and 50

LOCATION	ORDER	BOX NO.	NOTES
226	41 335F		
	50 0F		
227	40 0F		Constants
	10 0F		
228	00 198F		
	00 0F		
229	00 200F		
	00 0F		
230	00 0F		
	00 45F		
231	L5 272F	60	K = j
	40 271F		
232	L5 271F	61	Set K dependent addresses.
	L4 269F		
233	42 237F		
	42 262F		
234	L5 271F		
	L4 270F		
235	42 263F		
	42 266F		
236	L5 221F	63	$P - k \geq 0$?
	L0 271F		
237	32 263F	67	$u - U_k \geq 0$?
	L5 0F		
238	F0 273F		
	36 262F		
239	10 1F		
	F5 271F	68	k = k + 1
240	40 271F		
	L0 348F	62	k = n + 1 ?
241	36 232F		
	L5 272F		
242	L4 270F	69	Set j dependent addresses
	42 246F		

LOCATION	ORDER	BOX NO.	NOTES
243	42 257F		
	L5 272F		
244	L4 269F		
	42 245F		
245	42 259F		
	L5 0F		
246	L0 355F		
	L0 0F	71	$U_j - L_j - 3.2^{-39} \geq 0$?
247	32 257F		
	92 131F	73	Print eigenvalues and number of root (j).
248	92 515F		
	92 967F		
249	L5 272F		
	22 250F		
250	J2 3F		
	50 250F		
251	26 277F		
	92 67F		
252	92 963F		
	L5 273F		
253	50 12F		
	50 253F		
254	26 277F		
	F5 272F		
255	40 272F		
	L0 348F	74	j = j + 1
256	32 241F	70	j = n + 1 ?
	22 268F		
257	00 0F		
	L5 0F	72	$u = (U_j + L_j)/2$
258	10 1F		
	40 0F		
259	22 259F		
	L5 0F		

LOCATION	ORDER	BOX NO.	NOTES
260	10 1F		
	L4 0F		
261	40 273F		
	26 131F		
262	L5 273F	66	$U_k = u$
	40 0F		
263	22 239F		
	L1 0F		
264	L4 273F		
	36 266F	64	$u - L_k \geq 0$?
265	22 239F		
	00 0F		
266	L5 273F	65	$L_k = u$
	40 0F		
267	22 239F		
	00 0F		
268	22 268F		Useful constants
	24 20F		
269	00 0F		
	00 485F		
270	00 0F		
	00 357F		
271	00 0F		
	00 0F		
272	00 0F		
	00 0F		
273	00 0F		
	00 0F		
274	71 353F		
	40 486F		
275	10 1F		
	00 2F		
276	00 0F		
	00 79F		

LOCATION	ORDER	BOX NO.	NOTES
	(P16) 00K		
333	00 0F		More useful constants
	00 0F		
334	00 0F		
	00 65F		
335	00 0F		
	00 0F		
336	85 11F		
	00 2560F		
337	86 11F		
	00 2560F		
338	80 0F		
	00 0F		
339	00 0F		
	00 741F		
340	50 0F		
	00 0F		
341	00 128F		
	00 128F		
342	00 0F		
	00 0F		
343	80 0F		
	00 0F		
344	40 1F		
	50 0F		
345	42 91F		
	50 0F		
346	41 12F		
	50 0F		
347	22 72F		
	40 0F		
348	80 0F		
	00 0F		

LOCATION	ORDER	BOX NO.	NOTES
349	00 128F		
	00 0F		
350	50 0F		
	26 424F		
351	00 0F		
	00 0F		
352	7L 4095F		$1 \sim 2^{-39}$
	LL 4095F		
353	00 0F		
	00 0F		
354	00 0F		
	00 100F		
355	00 0F		
	00 3F		
356	(R1) 00K		
365	(N12) 00K		
			Program starts here
404	41 9F		
	41 10F		
405	41 4F		
	22 406F		
406	50 870F		
	50 406F		
407	26 365F		Read in the first row into locations
	L5 386F		870 - 870+n-1.
408	46 350F		870 + n in location (N12) + 21.
	L0 341F		
409	46 340F		Set the test constants dependent on n.
	10 20F		
410	42 347F		
	42 346F		
411	L4 341F		
	42 349F		

LOCATION	ORDER	BOX NO.	NOTES
412	42 345F		
	L0 341F		
413	L0 339F		
	42 348F		
414	F0 335F		
	42 343F		
415	42 333F		
	F0 335F		
416	42 342F		
	L5 446F		and arrange for program to skip this in future
417	40 407F		
	L5 4F		
418	42 420F		
	F5 4F		
419	J0 870F		Store using drum subroutine
	50 419F		
420	26 555F		
	00 0F		
421	L5 447F		
	40 424F		
422	L5 406F		
	46 423F		
423	50 0F		Sum squares, taking twice the sum of the off
	46 424F		diagonal elements.
424	75 0F		
	00 0F		
425	L4 9F		
	40 9F		
426	36 427F		Stop if norm exceeds 1.
	FF 55F		
427	S5 0F		
	L4 10F		
428	10 39F		
	40 0F		

LOCATION	ORDER	BOX NO.	NOTES
429	L7 0F		
	L4 9F		
430	40 9F		
	S5 0F		
431	40 10F		
	L5 448F		
432	40 424F		
	L5 423F		
433	L4 445F		
	46 423F		
434	L0 350F		
	36 436F		
435	L5 423F		
	26 423F		
436	40 10F		
	L5 406F		Increase addresses and test.
437	L4 445F		
	46 406F		
438	F5 4F		
	42 4F		
439	L0 343F		
	32 406F		
440	L5 443F		Reset routine
	40 407F		
441	L5 444F		
	40 406F		
442	41 0F		
	26 449F		
443	26 365F		Constants
	L5 386F		
444	50 870F		
	50 406F		
445	00 1F		
	00 0F		

LOCATION	ORDER	BOX NO.	NOTES
446	26 365F		
	22 417F		
447	75 0F		
	26 425F		
448	75 0F		
	00 1F		
449	L5 9F		Compute square root of sum or squares.
	50 449F		
450	26 356F		Jacobi reduction
	40 353F		
451	L5 9F		
	40 351F		
452	43 554F	1	i = 0
	L5 554F		k = i
453	42 468F	3	Set drum subroutine entries
	42 475F		
454	42 543F		
	42 548F		
455	F5 554F	4	i = i+1
	42 554F		
456	L0 342F	5	i = n ?
	36 63F		
457	L5 339F	6	Set i dependent addresses
	L4 554F		
458	42 476F		
	L4 341F		
459	42 477F		
	42 344F		
460	00 20F		
	46 541F		
461	F5 476F		
	00 20F		
462	46 532F		
	46 535F		

LOCATION	ORDER	BOX NO.	NOTES
463	46 540F		
	46 516F		
464	L4 341F		
	46 539F		
465	46 536F		
	46 515F		
466	F5 554F	7	j = i+1
	42 552F		
467	50 742F		
	50 467F		
468	26 555F		Enter drum subroutine to read i+1 th row.
	00 0F		
469	F5 339F	8	Set j dependent address
	L4 552F		
470	42 514F		
	42 533F		
471	42 538F		
	L4 341F		
472	42 515F		
	42 537F		
473	42 540F		
	F5 552F	9	Enter drum subroutine to read (j+1)th row
474	50 870F		
	50 474F		
475	26 555F		
	00 0F		
476	41 0F		
	L5 0F		
477	40 3F		
	L5 0F		
478	40 4F		
	L3 4F	11	$a_{i,j+1} = 0$?
479	36 544F		
	L7 3F	10	$\lvert a_{i,j+1} \rvert \geq \lvert a_{i,j+1} \rvert$?

LOCATION	ORDER	BOX NO.	NOTES PAGE 20
480	L2 4F		
	36 506F		
481	50 3F	13	Interchange roles of c and s and positions
	L5 4F		of $a_{i,i+1}$ and $a_{i,j+1}$
482	40 3F		
	S5 0F		
483	40 4F		
	L5 551F		
484	42 497F		
	46 504F		
485	50 335F	14	Calculate c and s
	L5 4F		
486	66 3F		
	S5 0F		
487	40 5F		
	7J 5F		
488	10 1F		
	40 1F		
489	L3 1F		
	36 505F		
490	LJ 1F		
	40 1F		
491	50 335F		
	6L 1F		
492	K5 0F		
	36 494F		
493	26 505F		
	00 0F		
494	S5 0F		
	50 494F		
495	26 356F		
	40 1F		
496	50 1F		
	7J 5F		

LOCATION	ORDER	BOX NO.	NOTES
497	40 OF		
	40 OF		
498	50 OF		
	79 OF		
499	40 OF		
	50 1F		
500	79 1F		
	L0 338F		
501	L4 OF		
	50 335F		
502	66 1F		
	S5 OF		
503	10 1F		
	L4 1F		
504	40 OF		
	22 514F		
505	L5 338F		
	22 495F		
506	40 1F	12	$\lvert a_{i,j+1} \rvert = \lvert a_{i,i+1} \rvert$?
	L3 1F		
507	36 509F		
	26 508F		
508	L5 550F		
	26 484F		
509	L5 3F	15	$a_{i,j+1} = a_{i,i+1}$?
	L4 4F		
510	40 OF		
	L5 553F		
511	40 5F	16	$c = s = 2^{-\frac{1}{2}}$
	40 6F		
512	L3 OF		
	32 513F		
513	22 514F		
	L1 553F	17	$c = -s = 2^{-\frac{1}{2}}$

LOCATION	ORDER	BOX NO.	NOTES
514	40 6F		
	L5 0F		
515	40 0F		
	L5 0F		
516	L4 0F		
	40 0F		
517	L5 344F		
	42 526F	18	p = i
518	42 530F	19	Set p dependent addresses
	42 524F		
519	L0 341F		
	42 529F		
520	00 20F		
	46 528F		
521	46 523F		
	L0 340F		
522	36 532F	22	p = n+1 ?
	26 523F		
523	50 0F	20	Rotate matrix, multiplication loop
	7J 6F		
524	40 1F		
	50 0F		
525	7J 5F		
	L0 1F		
526	40 1F		
	50 0F		
527	7J 6F		
	40 2F		
528	50 0F		
	7J 5F		
529	L4 2F		
	40 0F		
530	L5 1F		
	40 0F		

LOCATION	ORDER	BOX NO.	NOTES
531	F5 526F	21	p → p+1
	22 517F		
532	50 0F	23	
	7J 5F		
533	40 1F		
	50 0F		
534	7J 6F		
	L4 1F		
535	40 0F		
	50 5F		
536	7J 0F		
	40 1F		
537	50 6F		
	7J 0F		
538	L4 1F		
	40 0F		
539	40 0F		
	L5 0F		
540	L0 0F		
	40 0F		
541	41 0F		
	F5 552F	24	Store j+1 th row on drum
542	J0 870F		
	50 542F		
543	26 555F		
	00 0F		
544	F5 552F	25	j = j+1
	42 552F		
545	L0 343F	26	j = n ?
	36 469F		
546	F5 554F		
	22 547F		
547	J0 742F	27	Store i+1 th on drum
	50 547F		

LOCATION	ORDER	BOX NO.	NOTES
548	26 555F		
	00 0F		
549	22 452F		
	00 0F		
550	00 5F		**Constants**
	00 6F		
551	00 6F		
	00 5F		
552	00 0F		
	00 0F		
553	5K 2087F		
	99 2557F		
554	00 0F		
	00 0F		
555	40 1F		Start of drum subroutine
	K5 5F		
556	42 564F	75	Store i, q+1 and y
	10 20F		
557	42 2F	76	Read or write ?
	32 658F		
558	F5 624F	77	Store address of basic read order
	42 568F		
559	42 589F		
	42 596F		
560	42 607F		
	L5 623F	78	$\underline{\lambda}_0 = \lambda'$
561	42 618F		
	42 610F		
562	L4 555F		
	42 621F		
563	F4 555F		
	42 620F		
564	41 0F	81	Form addresses for read (write) loops
	L5 0F		

LOCATION	ORDER	BOX NO.	NOTES
565	42 0F		
	F5 564F		
566	42 641F		
	50 1F		
567	75 334F		
	S5 636F		
568	40 3F		
	L5 0F		
569	L0 334F		
	L4 0F		
570	L4 3F		
	40 625F		
571	40 643F		
	L4 2F		
572	L4 0F		
	42 626F		
573	42 642F		
	L5 333F		
574	L4 2F		
	42 613F		
575	42 614F		
	42 615F		
576	42 616F		
	L5 2F		
577	L4 1F		
	F0 335F		
578	42 611F		
	42 612F		
579	42 631F		
	42 647F		
580	42 637F		
	42 653F		

LOCATION	ORDER	BOX NO.	NOTES
581	L1 1F	83	$i \geq 65$?
	L4 334F		
582	32 605F		
	L5 2F	85	Form addresses for read (write) loops
583	L4 334F		$\underline{\lambda}_i = \underline{\lambda}_1$
	42 611F		
584	42 612F		
	42 631F		
585	42 647F		
	L5 637F		
586	42 613F		
	42 614F		
587	L5 620F		
	42 650F		
588	L5 567F		
	42 633F		
589	L5 3F		
	L4 0F		
590	L4 1F		
	L0 619F		
591	40 636F		
	40 654F		
592	L5 0F	86	$k > 65$
	L0 334F		
593	36 603F		
	L5 618F	88	$\underline{\lambda}_j = \underline{\lambda}_3$
594	42 602F		$r = 65$
	L5 617F		
595	46 597F		
	46 601F		
596	50 334F	89	Construct addresses
	L5 0F		
597	74 0F		
	85 641F		

LOCATION	ORDER	BOX NO.	NOTES
598	L4 338F		
	L4 1F		
599	L0 667F		
	40 630F		
600	40 648F		
	L5 2F		
601	L4 0F		
	42 631F		
602	42 647F		
	26 0F		
603	L5 621F	87	$\underline{\lambda}_j = \underline{\lambda}_4$, $r = k$
	42 602F		
604	47 597F		
	47 601F		
605	26 596F		
	L5 597F	84	$\underline{\lambda}_1 = \underline{\lambda}_2$
606	42 633F		
	42 650F		
607	L5 3F		
	L4 0F		
608	L4 1F		
	F0 334F		
609	40 630F		
	40 648F		
610	22 610F		
	26 0F		
611	K2 626F		Test constants used in boxes 93, 97, 101,
	40 0F		93', 97', and 101'.
612	K2 642F		
	L5 0F		
613	22 631F		
	40 0F		
614	22 647F		
	L5 0F		

LOCATION	ORDER	BOX NO.	NOTES
615	K2 637F		
	40 OF		
616	K2 653F		
	L5 OF		
617	00 334F		Constants
	00 OF		
618	00 OF		
	00 OF		
619	00 OF		
	01 258F		
620	00 OF		
	00 OF		
621	00 OF		
	00 OF		
622	00 OF		
	00 625F		
623	00 OF		
	00 642F		
624	00 OF		
	00 336F		
625	00 OF	91'	
	00 OF		
626	22 626F	90'	
	40 OF		
627	F5 625F	92'	
	40 625F		
628	F5 626F		Read loops
	42 626F		
629	L0 611F	93'	
	36 625F		
630	00 OF	95'	
	00 OF		
631	22 631F	94'	
	40 OF		

LOCATION	ORDER	BOX NO.	NOTES
632	F5 631F	96'	
	42 631F		
633	L0 613F	97'	
	36 0F		
634	L5 630F		
	L4 334F		
635	40 630F		
	26 630F		
636	00 0F	99'	
	00 0F		
637	22 637F	98'	
	40 0F		
638	F5 636F	100'	
	40 636F		
639	F5 637F		
	42 637F		
640	L0 615F	101'	
	36 636F		
641	22 641F	102	
	26 0F		
642	22 642F	90	
	L5 0F		
643	00 0F	91	
	00 0F		
644	F5 643F	92	Write loops
	40 643F		
645	F5 642F		
	42 642F		
646	L0 612F	93	
	32 642F		
647	22 647F	94	
	L5 0F		
648	00 0F	95	
	00 0F		

LOCATION	ORDER	BOX NO.	NOTES
649	F5 647F	96	
	42 647F		
650	L0 614F	97	
	32 0F		
651	L5 648F		
	L4 334F		
652	40 648F		
	22 647F		
653	22 653F	98	
	L5 0F		
654	00 0F	99	
	00 0F		
655	F5 654F	100	
	40 654F		
656	F5 653F		
	42 653F		
657	L0 616F	101	
	32 653F		
658	22 641F		
	L5 624F	79	Store address of basic write order
659	42 568F		
	42 589F		
660	42 596F		
	42 607F		
661	L5 622F	80	$\underline{\lambda} = \lambda$
	26 561F		
662	K2 78F		Test constant used in read back of Jacobi form
	40 936F		
663	F5 336F		Reset same after 130 elements have been read,
	F4 335F		and adjust test const so this block is disobeyed
664	40 71F		in future.
	F5 71F		
665	40 77F		
	L5 668F		

LOCATION	ORDER	BOX NO.	NOTES PAGE 31
666	40 662F		
	26 71F		
667	00 F		
	01 193F		
668	00 F		
	00 F		
	00 800K		Sum check on Routine. If it is to be modified, delete from here on.
800	81 40F		
	L2 F		
801	40 F		
	L3 F		
802	34 17F		
	FF 54F		
	26 800N		
	58 7543NL4N		Sum check const.