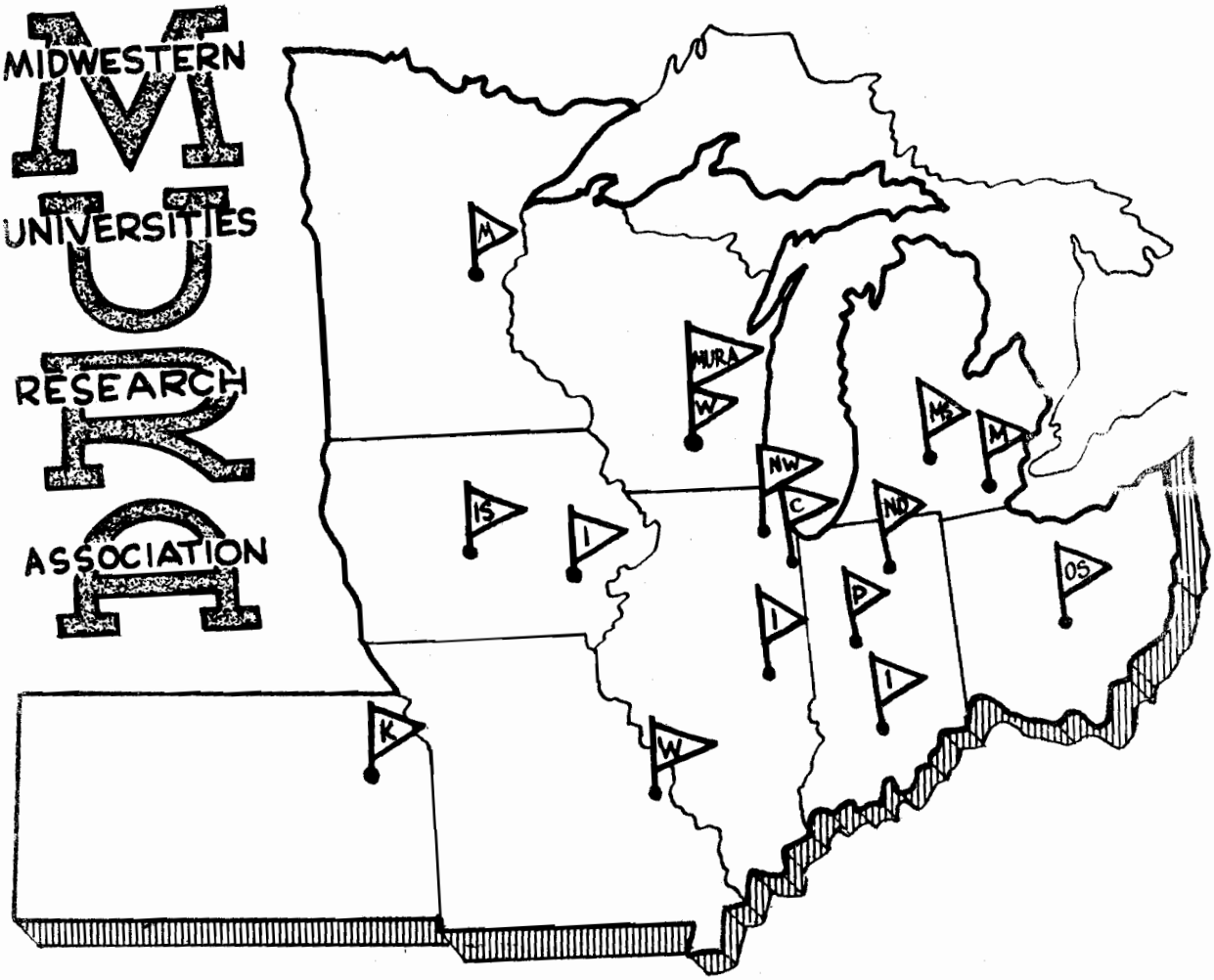
Fermilab Library
0 1160 0038707 0

392

MIDWESTERN UNIVERSITIES RESEARCH ASSOCIATION



REPORT GENDARME
(Programme 124)
March, 1958

NUMBER 392

(IBM Program)
Internal

GENDARME
(Programme 124)

James Kister

This routine diagonalizes a real symmetric matrix, using the Jacobi method, and prints out the pertinent information.

There are two forms of the code to handle two methods of input.

CODE S

The elements of the matrix to be diagonalized are assumed to be stored at the end of the memory and N, the size of the matrix, is suitably located. To be precise suppose the matrix is given in the form:

$$\begin{array}{llll} a_{00} & & & \\ a_{10} & a_{11} & & \\ a_{20} & a_{21} & a_{22} & \\ \vdots & \vdots & \vdots & \\ a_{N-1,0} & a_{N-1,1} & \cdots & a_{N-1,N-1} \end{array}$$

then a_{jk} is stored in $(17777)_8 - (\frac{j(j+1)}{2} + k)$ and N should be stored in $(1234)_8$. The code starts at $(310)_8$.

CODE R

The non-zero matrix elements are to be punched out in decimal form on cards which will be read in by the routine. Cards are prepared by punching the decimal address of the element starting in column 1 followed by a space followed by the element in decimal form. The addressing scheme is the same as that in Code S. (e.g. 8191 -105 signifies $a_{00} = -.105$) The last data card should have GENDARME entered in columns 37 - 72. The data deck is inserted behind the first transfer (green) card. For additional information see RDF3.

ALTERNATIVES (Both Codes)

The Jacobi method of diagonalization also gives the eigenvectors at the cost of very little extra computing time. If the eigenvectors are desired, sense switch 2 should be down, otherwise, up.

The matrix is diagonalized by successive rotations each of which transforms a certain off-diagonal element to zero. When the sum of the squares of the off-diagonal elements is suitably small (2^{-50}), the process has converged and the results are printed. Now if the routine were always to transform the largest off-diagonal element to zero, in some sense the computation would be more efficient, and convergence would be quicker. Unfortunately the time taken for finding the largest element more than compensates for the gain due to the fewer number of rotations required. However, the routine is slightly more accurate when operating in this way and has another possible advantage to be mentioned below. When sense switch 1 is down, the matrix is scanned for the largest off-diagonal element before each rotation. When SSW 1 is up, the rotations are taken in order.

There are of course limitations on the size of the matrix. The code occupies the region $(207)_8$ - $(1437)_8$ (135_{10} - 799_{10}). This limits the size to $N \leq 121$ in case the eigenvectors are not required (SSW 2 up). If the eigenvectors are desired (SSW 2 down), the limits are $N \leq 70$.

If $70 < N \leq 121$, it is still possible to obtain the eigenvectors by punching out the rotation constants as the symmetric matrix is being diagonalized. In this case SSW 2 is <u>up</u> and SSW 3 is <u>down</u>. The punched cards are then used in a forthcoming routine which computes the eigenvectors and prints them out.

PRINTING

If the input data is scaled in such a way that $\sum_{k \leq j}^{N-1} a_{jk}^2 < 1/8$, the code does not rescale the elements. Otherwise it does before going any further and prints immediately a line of six numbers. The first indicates the amount of scaling $\times 10^{-3}$ (i.e. 06400...0 indicates the machine has scaled each element by an additional factor of 64). The second number is the sum of the squares ($1/16 \leq \sum a_{jk}^2 < 1/8$). The rest are zeroes. If no scaling is required, the printing is omitted.

At the end of the calculation the diagonalized matrix is printed out in the natural order, six elements to a line: (a_{00}, a_{10}, a_{11}, a_{20}, a_{21}, $a_{22}, \ldots a_{N-1,\ N-1}$).

If the vectors are desired, they are printed out three spaces after the eigenvalues: b_{00}, b_{10}, $b_{20}, \ldots, b_{N-1,\ 0}$, b_{10}, $b_{11}, \ldots, b_{N-1,\ N-1}$, where b_{jk} is the jth component of the kth eigenvector, and a_{kk} is the corresponding eigenvalue.

SHARE #1 board is used.

ACCURACY

For each of two matrices of order 8 the residual error was approximately 2×10^{-13}. For one matrix of order 40 the residual error was about 5×10^{-11}.

The accuracy of the eigenvalues themselves in the case of order 8 was $\pm 1 \times 10^{-10}$ while for the order 40 it was $\pm 1 \times 10^{-8}$ or better (at least as accurate as the 8-place tables used to determine the error).

TIME

The length of time the routine takes to diagonalize an nth order matrix is roughly proportional to n^3. The table below shows the times required in the various modes for an order 40 matrix:

	Just eigenvalues	Eigenvalues and eigenvectors SSW 2
Search for largest element SSW 1	10 mins. 35 secs.	12 mins. 44 secs.
Rotations taken in order	5 mins. 15 secs.	9 mins. 13 secs.

DISCRIMINATION

It has been found that when the sums of the squares of the off-diagonal elements are less than $\delta = 2^{-50}$ maximum accuracy has been obtained for medium sized matrices. One might want to lessen δ for large matrices (70 or above) where time considerations become more important and perhaps accuracy less so. Or in small matrices one might like to make $\delta = 2^{-65}$ or 2^{-70}, the limit of accuracy in the 704. $\delta \times 2^{+35}$ is located in $(246)_8$ (i.e. 2^{-15} in $(246)_8$ corresponds to a $\delta = 2^{-50}$).