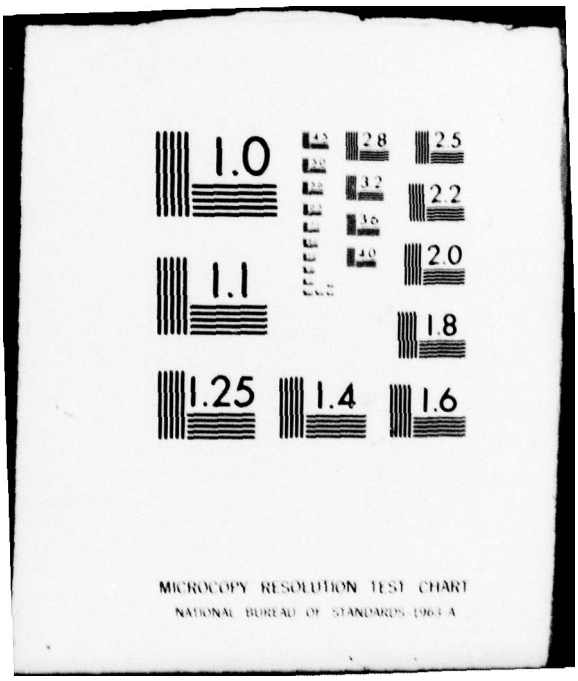
AD-A072 221 NAVY UNDERWATER SOUND LAB NEW LONDON CT F/G 17/1
SEVERAL METHODS FOR COMPUTING TRANSDUCER HEAD VELOCITIES FOR LA--ETC(U)
JUN 66 D A STREMSKY

UNCLASSIFIED USL-TM-960-68-66 NL

1 OF 1
AD A072221

END
DATE FILMED
9-79
DDC


1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

(9) Technical memo.,

Copy No. 23

USL Project No.
6-1-452-00-00

U. S. NAVY UNDERWATER SOUND LABORATORY
FORT TRUMBULL, NEW LONDON, CONNECTICUT

(6) SEVERAL METHODS FOR COMPUTING TRANSDUCER HEAD VELOCITIES FOR LARGE ARRAYS,

(10) Donald A. Stremsky

USL Technical Memorandum No. 960-68-66

17 June ~~1966~~

(12) 12 P.

(11) INTRODUCTION

Large transducer arrays are used in active sonar systems to emit powerful, directional sound beams. The transducer head velocities must be known in order to accurately determine the steady state source level and directivity patterns of these arrays. The values for these velocities depend upon the electrical driving conditions, the equivalent circuit of the transducers, and the geometry of the array. In addition, the calculation of these velocities generally requires the inversion of a complex matrix of order N, where N is the number of transducers in the array. For the case where the array has no symmetry and possesses more than 56 elements, direct inversion of the entire matrix is not practical, and due to storage limitations may not be possible. Several methods are presented here for computing transducer velocities for very large arrays when the self and mutual impedance coefficients and the driving forces are known for all the array elements.

"PATCH METHOD" (14) USL-TM-960-68-66

The "Patch Method" (references (a) and (b)) divides the array into overlapping patches and computes the velocities by doing the matrix inversion problem for each patch. This method, which has been incorporated into USL-IBM-704 Program #0717, is based on the following:

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. 960-68-66	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SEVERAL METHODS FOR COMPUTING TRANSDUCER HEAD VELOCITIES FOR LARGE ARRAYS		5. TYPE OF REPORT & PERIOD COVERED Tech Memo
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Stremsky, DonaldA.		8. CONTRACT OR GRANT NUMBER(s) Nonr-266(84)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Underwater Systems Center New London, CT		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research, Code 220 800 North Quincy St. Arlington, VA 22217		12. REPORT DATE 17 JUN 66
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

USL Tech Memo
No. 960-68-66

Consider the Thevenin equivalent circuit, as depicted in Figure 1, for the jth element, in an array of N transducers.

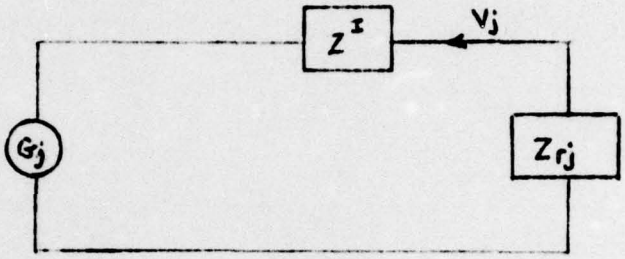


Figure 1

Thevenin Equivalent Circuit for jth Element in an Array

G_j is the Thevenin equivalent driving force; Z^I is the Thevenin equivalent internal impedance, which will depend on the operating frequency, tuning reactor, source impedance, and components of the transducer; Z_{rj} is the total radiation impedance, which is discussed in reference (c); V_j is the transducer head velocity. According to Figure 1 we can write,

$$G = (Z^I + Z_{rj})\, V_j . \tag{1}$$

If we define Z_{ij} as the mutual radiation impedance coefficient between the ith and jth elements and Z_{jj} as the self radiation impedance of the jth element, we can write,

$$G_j = \sum_{i=1}^{N} Z_{ij}^{*}\, V_i , \tag{2}$$

$$\text{since } Z_{rj} = \sum_{i=1}^{N} Z_{ij}\, \frac{V_i}{V_j} , \tag{3}$$

$$\text{and } Z_{ij}^{*} = Z_{ij},\ i \neq j \qquad = Z_{jj} + Z^I,\ i=j. \tag{4}$$

In terms of matrices equation (2) becomes

$$\begin{bmatrix} G \end{bmatrix} = \begin{bmatrix} Z \end{bmatrix} \begin{bmatrix} V \end{bmatrix}, \tag{5}$$

and multiplying by $\left[Z \right]^{-1}$, we find that

$$\left[V\right] = \left[Z \right]^{-1} \left[G\right]. \tag{6}$$

A method for obtaining the inverse of a complex matrix in terms of real matrices is given in Appendix A. If we now divide the array into patches each containing K elements, we can determine the velocities in each succeeding double patch by using the following equations:

$$Gj = \sum_{i=1}^{M} Zij^* \; Vi \text{ (known)} + \sum_{i=M+1}^{M+2K} Zij^* \; Vi \text{ (unknown)}, \tag{7}$$

which can be written as,

$$Gj = Fj + \sum_{i=M+1}^{M+2K} Zij^* \; Vi \text{ (unknown)}, \tag{8}$$

or as,

$$Hj = Gj - Fj = \sum_{i=M+1}^{M+2K} Zij^* \; Vi \text{ (unknown)}, \tag{9}$$

where M is the number of velocities which have been computed and Hj is the driving force minus the acoustic load from the elements of known velocity.

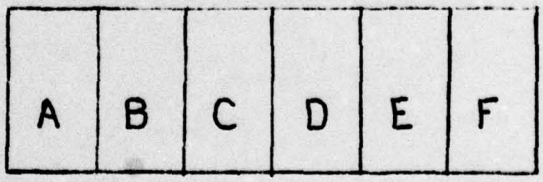


Figure 2

Consider now a large planar array as depicted in Figure 2. The array is separated into patches each containing K elements. It is necessary for the user of this method to have a computer and program which can invert a complex matrix of order 2K. The procedure incorporated in Program #0717 is as follows:

1. The coordinates are generated for each piston in the entire array.

2. The real and imaginary parts of the G forces are generated for all the array elements.

3a. The "patch coordinates" XKP(J) and YKP(J) are established by setting XKP(J) = XK(J+M) and YKP(J) = YK(J+M), for J=1, 2K.

3b. The "patch driving forces" are established by setting GP(J) = G(J+M).

3c. The Fj are computed for the 2K elements in the two adjacent patches under consideration. For [A+B] , M=0; there are no Fj.

3d. The Fj is subtracted from GPj to get Hj.

3e. The Zij* matrix and its inverse are computed for the double patch. After the first double patch, there is an option to reuse the initial Zij* inverse matrix or to calculate a new Zij* matrix and its inverse each time. The Zij* matrix will usually be the same for all the double patches in the same array. The values of the mutual radiation impedance coefficients depend upon the piston separations and the geometry of the array, and the distance between two elements in one double patch will usually be the same as the distance between two correspondingly placed elements in another double patch.

3f. The "patch velocities", VP(J), are computed using the matrix equation $[VP] = [Zij^*]^{-1} [H]$.

To compute the 2K velocities in [A+B] , the remaining velocities are assumed to be zero. M is then set equal to M+K, and the 2K velocities in [B+C] are computed assuming the velocities in [A] are as just determined and the remaining velocities are zero. The two sets of velocities on left end column of [B] are then compared. If the velocities of the elements tested do not agree within a given tolerance, the program prints out which pistons failed but continues the calculations and testing for each succeeding double patch until all of the velocities have been computed.

At this point, if any of the comparison tests failed, the user of the program has the option to return to step #3a and begin recomputing the patch velocities or to continue with the remainder

of the program. If the choice is to return to step 3a, the velocities in any double patch are obtained by considering the remaining velocities to be as last computed. This process of re-computing the patch velocities may be repeated until all of the comparison tests are successfully passed.

The "Patch Method" has been tested successfully for an array of 378 elements with 54 elements in each of 13 overlapping double patches. The results are given in Appendix B. Also, for smaller arrays ($N \leq 54$) the time for the velocity computation could be reduced by using this method. For example, the computation time for a 54 element case was reduced from nine to four minutes by dividing the array into three patches each containing 18 elements.

However, there are arrays for which the "Patch Method" in its present form fails. Correct values for the transducer head velocities could not be obtained for a 54 element planar, unshaded, close-packed array of circular pistons, with steering angles of 0^{o} and 60^{o}, $R^{I} = 0$, $X^{I} = -Xself$, and $Ka = 1.0$.

Here R^{I} and X^{I} are the real and imaginary parts respectively of Z^{I}, Xself is the imaginary part of the self radiation impedance, K is the wave number ($2\pi/\lambda$), and a is the radius of the pistons. It is possible that the badly behaved cases may be solved by increasing the overlap from 1 patch to M patches ($1 < M < 2$) without increasing the number of elements in a double patch.

"ITERATIVE METHOD"

The second method for computing transducer head velocities has been incorporated in USL IBM-704 Program #0697 and does not involve matrix inversion. It has a much simpler computation scheme than the "Patch Method". The calculations can be made by a self-correcting process.

Consider once again equation #2.

$$Gj = \sum_{i=1}^{N} Zij^{*} Vi ,$$

which can be written as

$$Gj = Vj \sum_{i=1}^{N} Zij^{*} \frac{Vi}{Vj}. \qquad (10)$$

Dividing both sides of equation #10 by $\sum_{i=1}^{N} Zij^* \frac{Vi}{Vj}$ we obtain

$$Vj = \frac{Gj}{\sum_{i=1}^{N} Z_{ij}^* \frac{Vi}{Vj}} , \qquad (11)$$

or

$$Vjnew = \frac{Gj}{\sum_{i=1}^{N} Z_{ij}^* \frac{Viold}{Vjold}} . \qquad (12)$$

Equation #12 provides the motivation for Program #0697.

This procedure is as follows:

1. Initial values are chosen for the V's.

2. These values are inserted into equation #12 to obtain new values for the V's.

3. These last values are used to obtain still other values for the V's.

4. The process may be continued until it becomes stabilized.

5. There is an option to begin averaging over the iterations at any stage of the iterative process or to continue without averaging. If averaging, the Vjold are determined by the following equation.

$$Vjold = (W)(Vj)+(1-W)(Vjavg), \qquad (13)$$

Where W is a weighting function, the Vj are the current values for the velocities, and the Vjavg are the averages of the velocity values computed for the j elements. However, averaging may not necessarily increase the rate of convergence or yield correct solutions to cases which converge.

In cases where the array permits one to have confidence that convergence will be sufficiently rapid, this method may well be preferred to all others.

However, there are arrays for which the iterative method converges too slowly, converges to the wrong limit or even diverges. Some of the elements became acoustically blocked in the cases where the velocity sequences converged to the wrong limit. Charts Numbers 1-3 show the convergence for 80 element unsteered arrays with four quadrant symmetry. Other iteration methods such as the one introduced inreference (d), pp 117-127, may produce better results.

Reference (e) describes the iteration method incorporated in Program #0077. This program is used for finding the magnitudes of the V's and the phases of the G's when the phases of the V's and the magnitudes of the G's are given. Convergence was much quicker in Program #0077, but blocked elements did occur in rare instances.

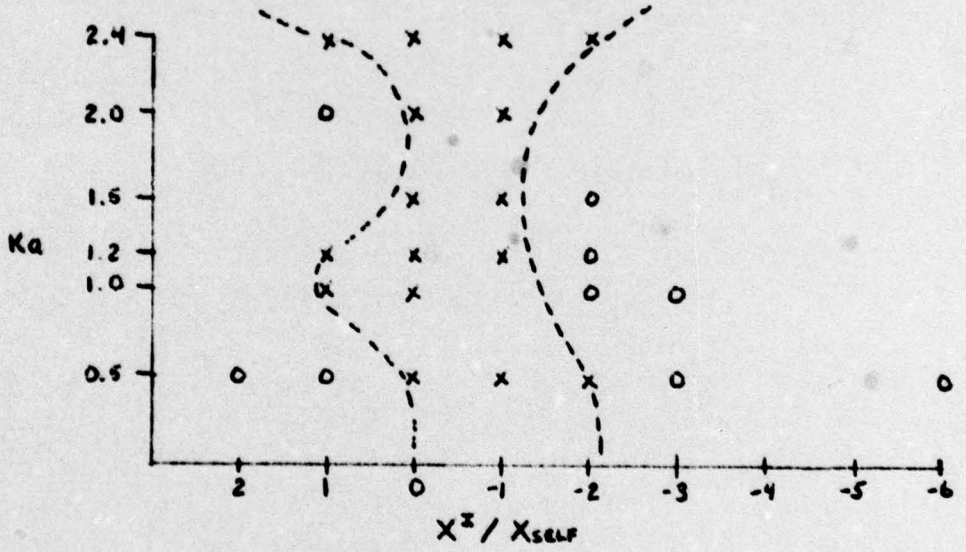


Chart Number 1
Close-packed Arrays, R' = 0.

O denotes convergent case
X denotes divergent case
Cases inside area bounded by dotted lines diverge

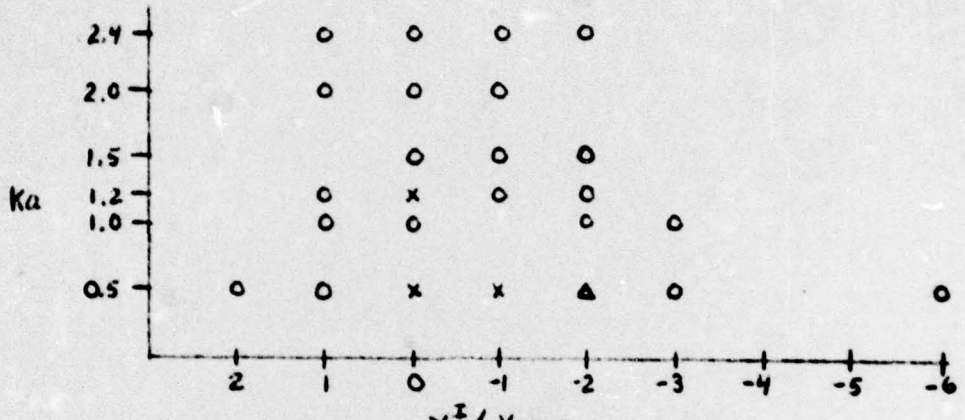


Chart Number 2

Close-packed Arrays, $R^I = R_{SELF}$

O denotes convergent case
X denotes divergent case
Δ denotes case which converged to wrong limit

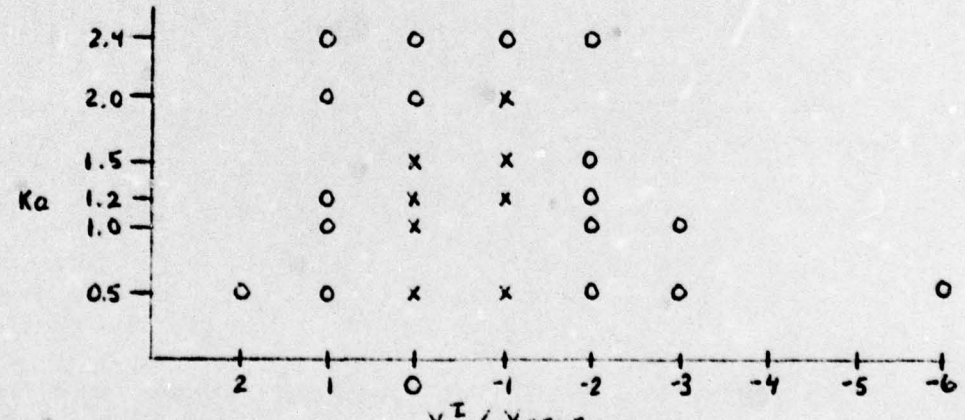


Chart Number 3

Increased Separation between Elements (3Ka), $R^I = 0$

O denotes convergent case
X denotes divergent case

"BORDERING METHOD"

The Bordering Method which at the present time is not incorporated in any USL computer program, is described in reference (d), pp 105-111. It is basically a method for inverting the Zij* matrix by successive borderings. Consider the matrix partitioned in the following manner:

$$Z_n = \left(\begin{array}{cccc|c} z_{11} & z_{12} & \cdots & z_{1,n-1} & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2,n-1} & z_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ z_{n-1,1} & z_{n-1,2} & \cdots & z_{n-1,n} & z_{n-1,n} \\ \hline z_{n1} & z_{n2} & \cdots & z_{n,n-1} & z_{nn} \end{array}\right), \qquad (14)$$

which can be written as

$$Z_n = \begin{pmatrix} Z_{n-1} & U_n \\ V_n & Z_{nn} \end{pmatrix}, \qquad (15)$$

Where Z_{n-1} is a matrix of the (n-1)th order the inverse matrix of which is considered to be known, $V_n = (Z_{n1}, \ldots, Z_{n,n-1})$, and

$$U_n = \begin{pmatrix} z_{1n} \\ \vdots \\ z_{n-1,n} \end{pmatrix}.$$

According to reference (d)

$$Z_n^{-1} = \begin{pmatrix} Z_{n-1}^{-1} + \dfrac{Z_{n-1}^{-1} U_n V_n Z_{n-1}^{-1}}{\alpha_n}, & -\dfrac{Z_{n-1}^{-1} U_n}{\alpha_n} \\ -\dfrac{V_n Z_{n-1}^{-1}}{\alpha_n}, & \dfrac{1}{\alpha_n} \end{pmatrix}, \qquad (16)$$

where $\alpha_n = Z_{nn} - V_n Z_{n-1}^{-1} U_n$.

Now, the Z inverse matrix can be obtained by constructing in succession

$$(Z_{11})^{-1}, \begin{pmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{pmatrix}^{-1}, \begin{pmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & Z_{22} & Z_{23} \\ Z_{31} & Z_{32} & Z_{33} \end{pmatrix}, \ldots.$$

where each succeeding one is obtained from the preceeding by bordering. Each step of this process is accomplished through substitution into equation #16.

The Bordering Method does not require the storage of the Zij* matrix although it does require the eventual storage of its inverse. Also, this method may be used to considerably reduce the time for the velocity computation in cases where the Z_{ij}* inverse matrix is known for an array which differs from the array under study only in the respect that the former has fewer elements.

Donald A. Stremsky
DONALD A. STREMSKY

REFERENCES

(a) D. T. Porter, "Special Problems in Composite Amplifier-Transducer Array Behavior", Talk presented at 1 June 1966 Workshop at Acoustical Society Meeting, Boston, Massachusetts.
(b) D. T. Porter, "Computation of Transducer Velocities for Very Large Transmitting Arrays", USL Report No. 741, 23 March 1966.
(c) D. T. Porter, "Effect of Thevenin Equivalent Internal Impedance or Velocity Control and Acoustic Power for Planar Broadside Arrays for Different Driving Level Limitations", USL Report No. 648, 28 April 1965.
(d) V. N. Fadeeva, "Computational Methods of Linear Algebra", 1959.
(e) D. T. Porter and R. D. Whittaker, "Four IBM 704 Programs for Sound Radiation Computations of an Array of Circular Pistons on a Rigid Sphere", USL Tech. Memo. No. 911-10-63, 15 February 1963.
(f) Notes from consultation with D. T. Porter.

APPENDIX A

The inversion of a complex matrix only requires a method of inverting a real matrix. If we are given a complex matrix $Z = R + jx$, its inverse can be shown to be

$$[R + jX]^{-1} = C + jD$$

where

$$C = [R + XR^{-1} X]^{-1}$$

and

$$D = - CXR^{-1}$$

Therefore, to invert a complex matrix, one needs to invert two real matrices.

APPENDIX B

The following 378 element array took two hours and five minutes of IBM-704 Computer time to solve for the transducer head velocities. Two passes were made. None of the comparison tests had failed on the second pass.