

# THE UNIVERSITY OF MISSOURI

ENGINEERING REPRINT SERIES

Reprint Number 68

❖ ❖ ❖

Engineering Experiment Station  
Columbia, Missouri

## A METHOD OF DATA LIST PROCESSING WITH APPLICATION TO EEG ANALYSIS

By C. M. Philpott, Control Data Corporation, St. Paul, Minnesota, and  
G. V. Lago, Professor of Electrical Engineering, University of Missouri

Reprinted from Communications  
of the ACM  
1965



COLLEGE OF ENGINEERING  
THE ENGINEERING EXPERIMENT STATION

The Engineering Experiment Station was organized in 1909 as a part of the College of Engineering. The staff of the Station includes all members of the Faculty of the College of Engineering, together with Research Assistants supported by the Station Funds.

The Station is primarily an engineering research institution engaged in the investigation of fundamental engineering problems of general interest, in the improvement of engineering design, and in the development of new industrial processes.

The Station desires particularly to co-operate with industries of Missouri in the solution of such problems. For this purpose, there is available not only the special equipment belonging to the Station but all of the equipment and facilities of the College of Engineering not in immediate use for class instruction.

Inquiries regarding these matters should be addressed to :

The Director  
Engineering Experiment Station  
University of Missouri  
Columbia, Missouri



# A Method of Data List Processing With Application to EEG Analysis

C. M. PHILPOTT\* AND G. V. LAGO  
*University of Missouri,† Columbia, Missouri*

A set of subroutines is discussed, which is designed to aid in the programming of computations on indexed lists of numbers using machine language or a symbolic assembly system. The most commonly performed list operations are outlined, and logically arranged into five groups. As an example, the computation of power spectral density from the autocovariance function is discussed for a class of EEG signals.

## 1. Introduction

A large class of problems require operations on indexed lists of numbers.<sup>1</sup> While this presents no great difficulty using FORTRAN, ALGOL, or some of the other algebraic compilers, it can be very troublesome when writing in machine language or using some symbolic assembly system, such as IBM's SPS. As a result of the large speed gain of machine language or SPS over FORTRAN (some 5-15 times) and because storage savings can be correspondingly great, machine language or SPS is often more desirable, especially if a well-developed method is to be used on large amounts of data.

A brief inspection of many programs indicates that only a few basic operations are continually utilized. The purpose here is to discuss methods of performing these basic operations, using as an example an algorithm for computing power spectral density from autocovariance functions for a class of electroencephalogram (EEG) signals.

*Notation.* A list is a group of  $N$  numbers, either fixed or floating point, which are stored sequentially in memory in some particular order.

Let these lists be represented by  $A, B, C \ni A = \{A_i \mid i \in I(N)\}$ . Let  $^\circ$  represent the four ordinary arithmetic operations ( $+$ ,  $-$ ,  $\times$ ,  $\div$ ) and let  $K$  be a constant of the same mode as  $A, B$  and  $C$ . Note that this notation

This work has been supported by the National Institutes of Health, Grant No. MH-025-53.

\* Present address: Control Data Corporation, St. Paul, Minn.

† Department of Electrical Engineering.

<sup>1</sup> Numerical integration, normalizing, etc.

indicates that, for example, the equation  $C = A + B$  implies  $C_i = A_i + B_i \nexists i \in I(N)$ .

*List Operations.* The normal operations performed on a list may be divided into five groups.

### GROUP I. ARITHMETIC

- |                      |                          |
|----------------------|--------------------------|
| (i) $C = A^\circ B$  | list-list arithmetic     |
| (ii) $C = A^\circ K$ | list-constant arithmetic |

### GROUP II. SPECIAL ARITHMETIC

- |                            |  |
|----------------------------|--|
| (i) $C = 1/A$              | list inversion   |
| (ii) $K = \text{ACC}(A)$   | $K = \sum_{i=1}^N A_i$   |
| (iii) $K = \text{PROD}(A)$ | $K = \prod_{i=1}^N A_i$  |
| (iv) $A = 0$               | clear list to zero   |
| (v) $C = f(A)$             | function evaluation $f$ : sin, cos, tan <sup>-1</sup> ,<br>√, etc. |

### GROUP III. INTERNAL DATA TRANSMISSION

- |                |                                       |
|----------------|---------------------------------------|
| (i) $C = A$    | Move List                             |
| (ii) File (A)  | Put list in aux. storage <sup>2</sup> |
| (iii) Load (A) | Get list from aux. storage            |

### GROUP IV. SELECTION

- |   |                               |
|---|-------------------------------|
| (i) $K = A_i \mid i$                      | fetch $i$ th element          |
| (ii) $A_i = K \mid i$                     | Enter $K$ as $i$ th element   |
| (iii) $K = \max(A)$                       | $K = A_i \geq A_j \nexists j$ |
| (iv) $K = \min(A)$                        | $K = A_i \leq A_j \nexists j$ |
| (v) $A_i = A_{i+1}, i < N$<br>$A_n = A_1$ | Shift list up, end around     |

### GROUP V. INPUT/OUTPUT

- |                |                             |
|----------------|-----------------------------|
| (i) Read (A)   | Read list from input device |
| (ii) Write (A) | Write list on output device |

Other list operations are performed, but the above seem to be the most common.

## 2. List Subroutines

Since these operations are relatively frequent in certain classes of problems, it is useful to develop a set of subroutines to perform them. Such a set must be written with due consideration of three factors: minimum execution time, minimum required storage and simplest calling procedure. When writing a subroutine, the programmer must decide the order of importance of these factors.

*Programming the Routines.* Each of the list operations can be written as a separate subroutine, but considerable storage may be saved (at a slight cost in speed) by com-

<sup>2</sup> Such as magnetic tape or disk file.



pressing the routines, that is, by writing only a few basic indexing routines, each of which has multiple entries.

For example, consider the group I operations. A flowchart is shown in Figure 1 which requires eight entry points, but which will perform all group I operations.

At the first level, the operation performed in the indexing loop is determined, perhaps by instruction modification.

Note that while the first operand ( $A$ ) is always indexed, the second operand ( $B$  or  $K$ ) is indexed for list-list operation but not for list-constant. This decision is made at level 2.

Thus this routine compresses eight slightly different operations into a single routine.

Another method of compressing would be to set the mode of the routine before entry by external instructions. This results in a simpler flowchart and is, in fact, equivalent to entering the chart in Figure 1 at the point  $X$ . Using this method, however, the calling instructions become more complex. Hence, if the routine is called at many different points in the mainline program, space may be wasted.

In addition, it is the authors' opinion that a simple calling macroinstruction is of basic importance.

Similar procedures for compressing can be applied to the other groups. The amount of compression possible depends to some extent on the command structure of the particular machine.

*Version for IBM 1620.* These routines have been programmed for an IBM 1620 Model I with the following special features and peripheral equipment: (1) indirect addressing, (2) automatic division, (3) automatic floating point, (4) additional instructions (MF, TNS, TNF), (5) 1311 Disk File, (6) 1422 Card Read/Punch, Model 2, and (7) expanded memory (60K characters).

Table I is a table of the routines showing symbolic name, function and execution time. A typical execution time for 50 number lists of 10-digit words is also shown. It should be noted that the I/O routines are limited in speed by the speed of the I/O devices.

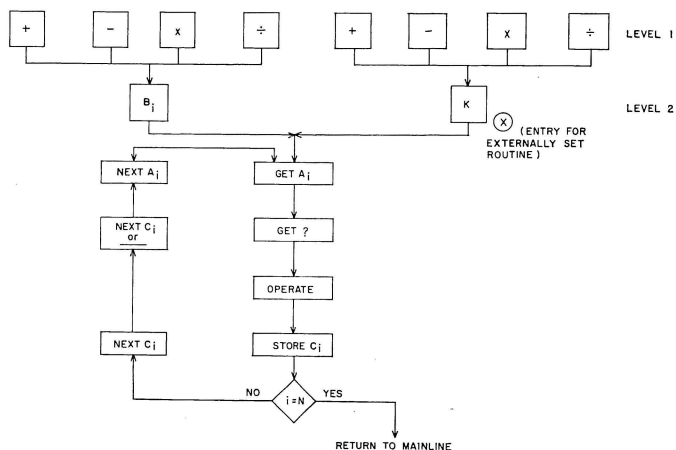


FIG. 1

The routines for the 1620 are floating point with variable mantissa length (2-20 digits). List length is specified in a

TABLE I. DATA-LIST SUBROUTINES FOR IBM 1620  
( $N$ : number of elements in list.  $L$ : element (word) length.)

Symbolic Name	Function	Execution Time ( $\mu$ S)
GROUP I. ARITHMETIC		
LADD	$C = A + B$	$560 + N(3960 + 240L)$
LSUB	$C = A - B$	$560 + N(3960 + 240L)$
LMUL	$C = A \cdot B$	$560 + N(4680 + 160L + 168L^2)$
LDIV	$C = A/B$	$560 + N(4440 + 1020L + 520L^2)$
LADDK	$C = A + K$	$560 + N(3960 + 240L)$
LSUBK	$C = A - K$	$560 + N(3960 + 240L)$
LMULK	$C = A \cdot K$	$560 + N(4680 + 160L + 168L^2)$
LDIVK	$C = A/K$	$560 + N(4440 + 1020L + 520L^2)$
GROUP II. SPECIAL ARITHMETIC		
LINV	$C = 1/A$	$560 + N(3720 + 80L)$
LACCUM	$K = \sum_{i=1}^N A_i$	$760 + 40L + N(1720 + 500L)$
LPROD	$K = \prod_{i=1}^N A_i$	$760 + 40L + N(1680 + 80L + 168L^2)$
LCLR	$A = 0$	$560 + N(1560 + 40L)$
LFNC	$C = F(A)$	$560 + N(2280 + 40L + X)$
	$F: \sin$	$X = 168L^3 + 3792L^2 + 13340L + 4708$
	$\cos$	$X = 168L^3 + 3792L^2 + 13340L + 5228$
	$\tan^{-1}$	$X = 168L^3 + 2996L^2 + 7792L + 7260$
	$\text{EXP}_e$	$X = 168L^3 + 3582L^2 + 15890L + 26418$
	$\text{EXP}_{10}$	$X = 168L^3 + 3440L^2 + 10530L + 12180$
	$\log_e$	$X = 168L^3 + 3656L^2 + 15414L + 24538$
	$\log_{10}$	$X = 168L^3 + 3608L^2 + 11610L + 15108$
		$620L^2 + 9776L + 5328$
GROUP III. INTERNAL DATA TRANSMISSION		
LMOVE	$C = A$	$560 + N(3720 + 80L)$
LFILE	Write $A$ on disk	$41180 + 20NL$
LLOAD	Read $A$ from disk	$41180 + 20NL$
GROUP IV. SELECTION		
LFETCH	$K = A_i i$	$7680 + 40L$
LINSRT	$A_i = Ki$	$7680 + 40L$
LMXSKP	$K = \max(A)$	$1520 + 40L + N(2530 + 80L)$
LMXSKN	$K = \min(A)$	$1520 + 40L + N(2530 + 80L)$
LSHIFT	$A_i = A_{i+1} i N$	$2280 + 40L + N(2120 + 40L)$
	$A_N = A_1$	
GROUP V. INPUT/OUTPUT		
LREADE	Read $E$ -Format Cards to $A$	$3.4N$ ms
LWRITE	Punch $E$ -Format Cards or type $E$ -Format from $A$	$3.4N$ ms



single location so that it may be changed easily by the mainline program.

For function evaluation, the standard SPS function subroutines have been used in such a way that they are still available for use in the ordinary manner.

The entire set, excluding the SPS functions requires less than 3600 core positions (this corresponds to 300 instructions).

### 3. Example Use of the List Subroutines

The example to be discussed concerns computation of power spectral density from the autocovariance (or autocorrelation) function of EEG data.

*Electroencephalogram (EEG).* The EEG is a continuous, nearly periodic, randomly fluctuating voltage resulting from the electrical activity of the cortex of the brain. The voltage is of 50 to 200 microvolts amplitude at the scalp surface, and under certain conditions, changes in this spontaneous activity may represent a response to some stimulus. Thus, from a psychological point of view, the EEG may represent a behavior. Classically the method of analysis has been visual inspection. Modern computing tools and methods of analysis may be used to develop better quantifications. Two measures of particular interest are autocovariance and power spectral density.

*Autocovariance.* The autocovariance of a stationary, random process  $e(t)$  is

$$R(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T e(t)e(t + \tau) dt. \quad (1)$$

For finite samples of time data,  $e^1(t)$ ,  $0 \leq t \leq T$ , and if  $T$  is large enough, an approximation suffices, namely,

$$R(\tau) = \frac{1}{T} \int_0^T e^1(t)e^1(t + \tau) dt, \quad \tau \ll T. \quad (2)$$

This function may be computed by analog methods. The result of the method used is a set of  $N$  points, equally

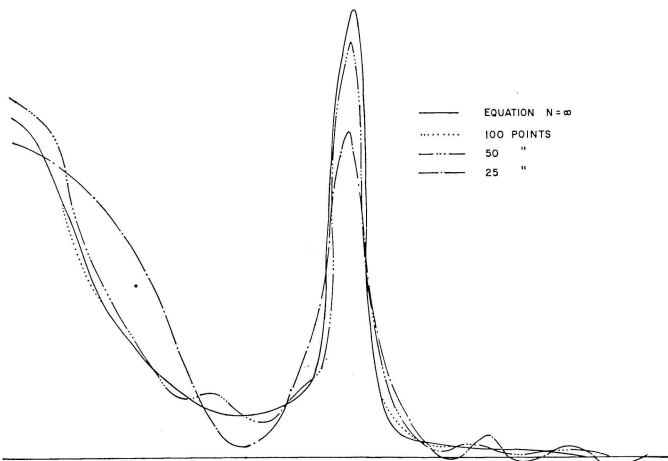


FIG. 2

spaced on  $\tau$  at intervals  $\Delta\tau$ . For the type of data normally analyzed, the autocovariance function closely resembles the curve.

$$A e^{-a\tau} + B e^{-b\tau} \cos \omega_i \tau. \quad (3)$$

An example of a typical autocovariance curve is shown in Figure 2.

*Power Spectral Density.* The power spectral density of a time function  $f(t)$  may be written as

$$P(\omega) = \lim_{T \rightarrow \infty} \frac{1}{T} \left| \int_{-T/2}^{T/2} f(t) e^{-i\omega t} dt \right|^2. \quad (4)$$

If the autocovariance function is available, a well-known relation may be applied, namely, the Fourier cosine transform

$$P(\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} R(\tau) \cos \omega \tau d\tau. \quad (5)$$

It is the latter relation which is to be studied here. It is important to note that if the autocovariance exists, the transform (5) can always be taken. Hereafter, all necessary properties of the time function are assumed to exist.

*The Algorithm.* For the case of discrete autocorrelation data, consider the following approximation to the integral (5)

$$P(\omega_j) = \frac{1}{2\pi} \sum_{\tau=-\infty}^{\infty} R(\tau) \cos \omega_j \tau. \quad (6)$$

For the finite set of points on  $R_i$  this reduces to

$$P_N(\omega_j) = \frac{R_0}{2\pi} + \frac{1}{\pi} \sum_{i=1}^N R(i\Delta\tau) \cos(\omega_j i\Delta\tau).$$

The approximation to the Fourier Transform will be found satisfactory if it converges to the power spectral density.

An indication of the answer to the question of convergence may be shown by the following experiment.

First, recall that the functions of interest resemble  $Ae^{-a\tau} + Be^{-b\tau} \cos \omega_i \tau$  whose power spectral density can be shown to be

$$\frac{aA}{a^2 + \omega^2} + Bb \frac{b^2 + \omega_1^2 + \omega^2}{\omega^4 + 2\omega^2(b^2 - \omega_1^2) + (b^2 + \omega^2)^2}.$$

Suppose that  $M$  points of this function are computed and the transform approximation applied to the result. Figure 2 shows a family of computed spectral density curves using  $M$  as a parameter. Clearly,  $P_N(\omega) \rightarrow P(\omega)$  as  $M \rightarrow \infty$ .

For  $M > 100$ , the spectral density is quite accurate, at least in the frequency range of interest.

*Weighting Functions.* A number of weighting functions have been suggested to account for the finite length of correlation data. These "lag windows" are continuous functions of  $\tau$ , constrained by  $D_i(0) = 1$ ,  $D_i(\tau) = 0$ ,  $\tau < \tau_{\max}$ .

Several of these functions have been tried, but the resulting change in the spectral density was never observed to be more than 0.5 percent. It seems, therefore, that for



the particular type of data being studied, the weighting functions are not worthwhile.

*Programming the Algorithm.* The frequencies for which the spectral densities are computed were chosen between 0.625 and 100cps, spaced at  $\frac{1}{3}$  octave. This results in 23 frequencies. Since these are fixed, and for fixed  $\Delta\tau$ , it is possible to compute all the  $\cos(\omega_j i \Delta\tau)$  terms and store them in a table.

With such a scheme, using FORTRAN, the spectral density of a sample of EEG autocovariance data can be computed in about  $1\frac{1}{2}$  minutes. A flowchart for this computation is shown in Figure 3. If the cosines must be generated, the computations take nearly 20 minutes.

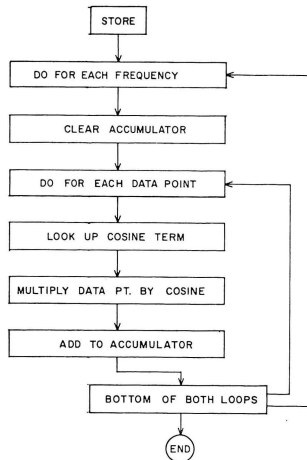


FIG. 3. Computation of spectral density (FORTRAN)

Suppose the cosines terms are stored in the following way: first a list of the  $M$  points for the first frequency, then for the second frequency, etc. Then the list subroutines may be used for this computation as shown in the flowchart of Figure 4. Execution time for this program is about 20 seconds.

This is a speed gain of more than 4 over FORTRAN. The amount of storage space required is also much less.

#### 4. Conclusion

*Power of the Method.* This programming procedure lends itself well to a certain large class of problems. For these types of problems, the routines have proven very

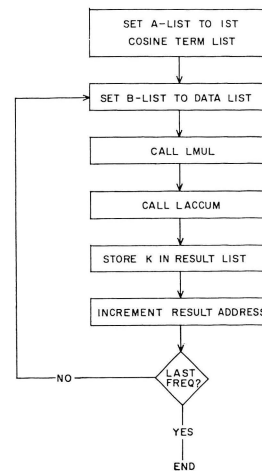


FIG. 4. Computation of spectral density

useful. Programming in machine language or SPS is greatly simplified, yet the power and speed of machine language is essentially preserved.

It is the authors' opinion that a set of routines such as described would form a useful addition to many program libraries.

*Extending the Method.* Depending upon the type of work, other routines may be written to perform special types of data handling common to the particular installation. The input and output routines especially should be adapted to the particular format commonly used. Other special machine functions, such as analog/digital conversion, online plotting, special readout devices, etc. can be handled by such routines.

*The Algorithm.* Computation of other types of spectral densities may be possible with this method. However, convergence at high frequencies has not been investigated, so one must be careful in using it. For the type of data under study, however, the method appears quite satisfactory.

*Acknowledgments.* The authors wish to express thanks for the assistance and encouragement of many people, especially Dr. S. J. Dwyer, III, for his suggestions and encouragement; Dr. Roy Keller, for providing the computer time; and The Staff of the EEG project.

RECEIVED SEPTEMBER, 1964

## PUBLICATIONS OF THE ENGINEERING REPRINT SERIES

Copies of publications may be secured from the Director of the Engineering Experiment Station, University of Missouri. Single copies may be obtained free unless otherwise indicated until the supply is exhausted. Requests for additional copies will be considered upon further inquiry.

### Reprint No.

58. Controlled-Deflection Design Method for Reinforced Concrete Beams and Slabs by Adrian Pauw, Professor of Civil Engineering, University of Missouri. Reprinted from Journal of The American Concrete Institute, Vol. 59, No. 5, May 1962.
59. The Impedance of a Coil Placed on a Conducting Plane by T. J. Russell, The Bendix Corporation, Kansas City, Missouri, V. E. Schuster, Motorola, Inc., Phoenix, Arizona and D. L. Waidelich, Professor of Electrical Engineering, University of Missouri. Reprinted from AIEE Winter General Meeting, New York, N. Y., January 28 - February 2, 1962.
60. Discussion of an Article by G. S. Ramaswamy and M. Ramaiah: Characteristic Equation of Cylindrical Shells--A Simplified Method of Solution by Adrian Pauw, Professor of Civil Engineering, University of Missouri, and W. M. Sangster, Professor of Civil Engineering, University of Missouri. Reprinted from Journal of The American Concrete Institute Concrete Briefs, pages 1505-1509, October 1962.
61. Dissolution Rates of Uranium Dioxide Sintered Pellets in Nitric Acid Systems by R. F. Taylor, Chemical Engineering Division, AERE, Harwell, England, E. W. Sharratt, Chemical Engineering Division, AERE, Harwell, England, L. E. M. de Chazal, Associate Professor Chemical Engineering and D. H. Logsdail, Chemical Engineering Division, AERE, Harwell, England. Reprinted from Journal of Applied Chemistry, Published by the Society of Chemical Industry, Volume 13, No. 1, January 1963, Pages 32-40.
62. Circuit Waveforms for Periodic Waves by D. L. Waidelich, Professor, Electrical Engineering, University of Missouri. Reprinted from Communication and Electronics, March 1963.
63. Transmission Losses and Economy Loading by the Use of Admittance Constants by J. R. Tudor, Associate Professor, Electrical Engineering, University of Missouri and W. A. Lewis, Research Professor, Electrical Engineering, Illinois Institute of Technology, Chicago, Illinois. Reprinted from Power Apparatus and Systems, IEEE, October, 1963.
64. Applicability of Thermoacoustic Phenomena to MHD Conversion Systems by R. L. Carter, Professor, Electrical Engineering, University of Missouri, K. T. Feldman, Jr., NDEA Fellow, Mechanical Engineering, University of Missouri, and C. N. McKinnon, Jr., Instructor, Mechanical Engineering, University of Missouri. Reprinted from Fifth Symposium on the Engineering Aspects of Magnetohydrodynamics MIT, April 1 & 2, 1964.
65. Three and Four Coil Systems for Homogeneous Magnetic Fields by M. E. Pittman, Research Assistant, Physics Department, University of Maryland, D. L. Waidelich, Professor, Electrical Engineering, University of Missouri. Reprinted from IEEE Transactions on Aerospace, February, 1964.
66. Variable-Mesh Difference Equation for the Stream Function in Axially Symmetric Flow by J. C. Lysen, Associate Professor, Mechanical Engineering, University of Missouri. Reprinted from AIAA Journal, 1964.
67. Creep of Concrete: Influencing Factors and Prediction by A. M. Neville, Chairman, Division of Engineering, University of Alberta, Calgary, and B. L. Meyers, Assistant Professor of Civil Engineering, University of Missouri.  
  
Effect of Creep and Shrinkage on the Behavior of Reinforced Concrete Members by A. Pauw, Professor and Chairman of Civil Engineering, and B. L. Meyers, Assistant Professor of Civil Engineering, University of Missouri. Reprinted from Symposium on Creep of Concrete, Publication SP-9, The American Concrete Institute.
68. A Method of Data List Processing with Application to EEG Analysis by C. M. Philpott, Control Data Corporation, St. Paul, Minnesota, and G. V. Lago, Professor of Electrical Engineering, University of Missouri. Reprinted from Communications of the ACM, Volume 8, Number 5, May, 1965.





University of Missouri Libraries  
University of Missouri

MU Engineering Experiment Station Series

Local Identifier Philpott1965

Capture information

Date captured 2018 January  
Scanner manufacturer Ricoh  
Scanner model MP C4503  
Scanning software  
Optical resolution 600 dpi  
Color settings Grayscale, 8 bit; Color 24 bit  
File types Tiff

Source information

Format Book  
Content type Text  
Notes Digitized duplicate copy not retained in collection.

Derivatives - Access copy

Compression LZW  
Editing software Adobe Photoshop  
Resolution 600 dpi  
Color Grayscale, 8 bit; Color, 24 bit  
File types Tiffs converted to pdf  
Notes Greyscale pages cropped and canvassed. Noise removed  
from background and text darkened.  
Color pages cropped.