Annual Report on

Computer-Aided Circuit Analysis

Submitted to

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Office of Grants and Research Contracts

Washington D. C 20546

This work was done under the NASA grant NGR-39-023-004, during the period May 15, 1965 to May 14, 1966, at the Electrical Engineering Department, Villanova University, Villanova, Pennsylvania.

RESEARCH AND DEVELOPMENT DIVISION

VILLANOVA UNIVERSITY

VILLANOVA, PENNSYLVANIA

Annual Report on

Computer-Aided Circuit Analysis

Submitted to

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Office of Grants and Research Contracts

Washington D. C 20546

This work was done under the NASA grant NGR-39-023-004, during the period May 15, 1965 to May 14, 1966, at the Electrical Engineering Department, Villanova University, Villanova, Pennsylvania.

N66 31224

PRINCIPAL INVESTIGATOR

Tsute Yang

TSUTE YANG
Professor of Electrical Engineering

Henry T. Koonce

HENRY T. KOONCE
Director of Research and Development

RESEARCH AND DEVELOPMENT DIVISION

VILLANOVA UNIVERSITY

VILLANOVA, PENNSYLVANIA

The contributors to this report are

Tsute Yang

J. J. Hicks

R. M. Jansen

C. P. Rich, Jr.

J. J. Perkowski

Table of Contents

Annual Report of the Research on

Digital Computer-Aided Circuit Analysis

under the NASA Grant NGR-39-023-004

covering period from May 15, 1965 to May 14, 1966.

Section

ERRATA

page	line	printed as	should read
1	3	catagories	categories
28	10	fortet	forte
32	1	Machine	Machines
39	10	j	J
41	2 from bottom	C_4	ωC_4
42	7, 8, 9	L_3	ωL_3
51	Fig. 8	L_3	L_3 8
		C_4	C_4 3
55	1	Caption "Table 1" to be added.	
73	5	Matrix equation to be labelled as (2)	
77	18	10	16
78	11	1, 2 1, 3	3, 2 3, 3
85	5, 11	(2) and (4) to be interchanged.	
106	This page is redundant.		
107	19	NPS	NBS
114	11	x	x^2
116	Equation for sigma to be labelled as (2.2).		

Section I. Literature Search

One of the initial activities of computer-aided analysis of electric circuits consisted of a literature search in the subject area. Publications in the professional and technical journals may be classified into two catagories: those dealing with the general approach of analyzing circuit performance of any configuration within given sets of constraints, and those tailored for the design of specific types of networks. The primary interest of this project has been in the investigation and development of techniques and programs for the analytic solution of circuit problems in general. There is evidence of growing interest and activity both in industry and research institutions in exploiting digital computers as an aid in system design and reliability evaluation [1], [2]. It gradually comes to the realization that progress and practicality of the concepts involves more than the mathematical model formulation and digital algorithm applied to circuit solutions. Various other facets such as man-machine interface, time cost of operation, etc. will influence and contribute significantly to the success of the program [3].

In order to provide the uninitiated with a starting base to get into the area of computer-aided analysis and design and the specialists in the field with a ready reference which would reflect the current development in research and industry, a comprehensive bibliography is attempted and completed with 205 entries as included in this report. It is expected that the Bibliography will be revised and brought up to date and distributed when necessary.

One difficulty encountered in the preparation of the Bibliography is to arrive at a balanced medium between indiscriminative exhaustiveness which may tend blurring the significant contributions, and unintentional or misjudged omissions which would adversely affect its usefulness as a source of reference. Since the interest of the research activity weighs heavily on the analysis and design of conventional electronic circuits, many important titles which may be excellent background literature in the application of digital computer for circuit analysis are not contained in the Bibliography. Specific examples include Kron's method of large system analysis, Monte Carlo sampling technique in general digital simulation, computer solution of matrix functions and of nonlinear differential equations, and the design of graphic display as computer output. Another notable missing segment is concerned with the use of digital computers in electric power machinery design, power system distribution and transmission. There is a great wealth of literature in that area published in the IEEE Transactions on Power Apparatus and Systems and during the Power Industry Computer Application Conferences.

An early effort in analyzing the potentialities of automatic digital computers to research seems to be the technical paper in six parts by Clippinger, Dimsdale, and Levin [4] published in the Journal of the Society of Industrial and Applied Mathematics in 1953-54. Although the possible use of computers in the analysis of electric circuits has been recognized for some time, the first recorded arrangement in technical meetings on the subject is the session on "Computers in Network Synthesis" in 1957 WESCON Convention at which time three papers were presented.

T. R. Bashkow and C. A. Desoer, "Digital Computers and Network Theory"
D. T. Bell, "Digital Computers as Tools in Designing Transmission Networks"
W. Mayeda and M. E. Van Valkenberg, "Network Analysis and Synthesis by Digital Computers"

In 1961 the IRE Transactions on Circuit Theory issued a special number on Network Design by Computers, including the following papers:

G. M. Cohen and D. Plantnick, "The Design of Transistor IF Using an IBM 650 Digital Computer"
C. A. Desoer and S. K. Mitra, "Design of Lossy Ladder Filters by Digital Computer"
D. C. Fiedler, "A Combinatorial-Digital Computation of a Network Parameter"
S. Hellerstein, "Synthesis of All-Pass Delay Equilizers"
K. Yamanoto, K. Fujimoto, and H. Watanabe, "Programming the Minimum Inductance Transformation"

A Computer Program Reviews Department has since been inaugurated to the Transactions under the editorship of P. R. Geffe, which collects and publishes titles and reviews of available programs on circuit theory problems. There was a symposium on the Design of Networks with a Digital Computer at 1962 IRE International Convention when four papers were presented.

F. H. Branin, Jr., "D-C and Transient Analysis of Networks Using a Digital Computer"
O. P. Clark, "Design of Transistor Feedback Amplifiers and Automatic Control Circuits with the Aid of a Digital Computer"
C. L. Semmelman, "Experience with a Steepest Decent Computer Program for Designing Delay Networks"
G. C. Temes, "Filter Synthesis Using a Digital Computer"

In 1963 Lockheed Missiles and Space staff prepared an annotated bibliography on computer-aided analysis and design with 63 entries.

C. M. Pierce, "The Design and Analysis of Electrical and Electronic Systems by Means of Digital Computers: An Annotated Bibliography", Lockheed Missiles and Space Co., September, 1963; SB-63-65; ASTIA Document AD 439 440.

More recently the Third Allerton Conference on Circuit and System Theory, October 20-22, 1965, a special session was devoted to the Network Analysis and Design by

Digital Computers.

R. M. Golden, "Digital Computer Simulation of Communication Systems Using the Block Diagram Computer: BLØDIB"
J. Katzenelson and L. H. Seitelman, "An Iterative Method for Solution of Nonlinear Resistor Networks"
M. L. Liou, "A Numerical Solution of Linear Time-Invariant System"
C. Pottle, "On the Partial Fraction Expansion of a Rational Function with Multiple Poles by Digital Computer"
H. C. So, "Analysis and Design of Linear Networks with Variable Parameters Using On-Line Simulation"
A. D. Waren and L. S. Lasdon, "Practical Filter Design Using Mathematical Optimization"

In the following Bibliography the entries are arranged in the alphabetic order of the last name of the first author of each paper. A subject index and a chronological index are appended.

References

[1] H. S. Scheffler, ed., "Reliability Analysis of Electronic Circuits," Autonetics Division of North American Aviation Inc., 1964.

[2] University of Michigan Engineering Summer Conference on "Application of Computers to Automated Design," 1964 and 1965.

[3] D. T. Ross and J. E. Rodriguez, "Theoretical Foundations for the Computer-Aided Design System," Proc. Spring Joint Computer Conference, AFIPS, vol. 23, pp. 305-322, 1963.

[4] R. F. Clippinger, B. Dimsdale, and J. H. Levin, "Automatic Digital Computers in Industrial Research," Jour. SIAM,
pt. I, vol. 1, pp. 1-15 Sept. 1953
pt. II, vol. 1, pp. 91-110 Dec. 1953
pt. III, vol. 2, pp. 36-56 Mar. 1954
pt. IV, vol. 2, pp. 113-131 June 1954
pt. V, vol. 2, pp. 184-200 Sept. 1954
pt. VI, vol. 3, pp. 80-89 June, 1955

A BIBLIOGRAPHY ON DIGITAL COMPUTER-AIDED CIRCUIT ANALYSIS AND DESIGN

(revised May, 1966)

(1) M.W. Aarons and M.J. Goldberg, "Computer Methods for Integrated Circuit Design", Electro-Technology, vol. 75, no. 5, pp. 77-81; May, 1965.

(2) M.R. Aaron and J.F. Kaiser, "On the Calculation of Transient Response", Proc. IEEE, vol. 53, p. 1269; September, 1965.

(3) J.R. Abrahams, "Amplifier Design with a Digital Computer", Electronic Engrg., vol. 37, pp. 740-745; November, 1965.

(4) G.E. Adams and J.E. Gerngross, "IBM 650 as a Tool for Analysis of Transmission and Distribution System Problems", Power Apparatus and Systems, no. 40, (AIEE Trans. vol. 77, pt. 3) pp. 1236-1244; February, 1959.

(5) R.K. Adams, "Digital Computer Analysis of Closed-Loop Systems Using the Number Series Approach", Applications and Industry, no. 67, (AIEE Trans. vol. 82, pt. 3) pp. 238-240; July, 1963.

(6) F.A. Applegate, "Statistical Circuit Analysis Based on Part Test Data", Electro-Technology, vol. 71, no. 5, pp. 140-145; May, 1963.

(7) K.G. Ashar, et al. "Transient Analysis and Device Characterization of ACP Circuits", IBM J. Res. and Dev., vol. 7, pp. 207-223; July, 1963.

(8) W.D. Ashcraft and W. Hochwald, "Design by Worst-Case Analysis: a Systematic Method to Approach Specified Reliability Requirements", IRE Trans. on Reliability and Quality Control, vol. RQC-10, no. 3, pp. 15-21; November, 1961.

(9) A.G. Atwood and L.C. Drew, "Computer Analysis of an Integrated Circuit Amplifier", RCA Engineer, vol. 10, no. 3, pp. 27-29; October/November, 1964.

(10) W. Austin and K. Angel, "Computer Solutions Aid Rectifier Filter Design", Electronic Industries, vol. 23, no. 12, pp. 54-57; December, 1964.

(11) H.S. Balaban, "Selected Bibliography on Reliability", IRE Trans. on Reliability and Quality Control, vol. RQC-11, pp. 86-103; July, 1962.

(12) T.C. Bartee, "Computer Design of Multiple-Output Logical Networks", IRE Trans. Electronic Computers, vol. EC-10, pp. 21-30; March, 1961.

(13) T.R. Bashkow and C.A. Desoer, "Digital Computers and Network Theory", WESCON Conv. Record, vol. 1, pt. 2, pp. 133-136; 1957.

(14) __________, "Network Analysis", Mathematical Methods for Digital Computers (book) Chapter 26, pp. 280-290; John Wiley, 1960.

(15) W.F. Baur, D.L. Gerlough and J.W. Granholm, "Advanced Computer Application", IRE Proc. vol. 49, pp. 296-304; January, 1961.

(16) J.H. Beaudette and P.A. Honkanen, "PECANS Circuit Analysis of Nonlinear Systems", IBM 7090 Program Writeup, IBM Development Lab., Poughkeepsie, N.Y. 1961.

(17) F. Beck, "Harmonic Analysis Using a Digital Computer", The Computer J. vol. 1, p. 117; October, 1958.

(18) S.D. Bedrosian, "Finding the Maximum Complete Subgraph in Coding Models for Random Access Communications", IEEE Internat'l. Conv. Record, vol. 14, pp. 250-253; 1966.

(19) D.T. Bell, "Digital Computers as Tools in Designing Transmission Networks", WESCON Conv. Record, vol 1, pt. 2, pp. 145-153: 1957.

(20) A.H. Benner and B. Meredith, "Designing Reliability into Electronic Circuits", Proc. Nat'l. Electronics Conf., vol. 10, pp. 137-145; 1954.

(21) C.L. Bertin, "Transmission-Line Response Using Frequency Techniques", IBM J. Res. and Dev. vol. 8, pp. 52-63; January, 1964.

(22) H.D. Bickley, et al. "Digital Techniques for Voltage Regulation Studies", Automatic Control, vol. 15, no. 6, pp. 26-30; December, 1961.

(23) J.A.C. Bingham, "A New Method of Solving the Accuracy Problem in Filter Design", IEEE Trans. on Circuit Theory, vol. CT-11, pp. 327-341; September, 1964.

(24) G.T. Bird, "On the Basic Concepts of Reliability Prediction", Proc. 7th Nat'l. Symp. on Reliability and Quality Control, pp. 51-54; January, 1961.

(25) B. Birtwistle and B.M. Dent, "Digital Computer as an Aid to the Electrical Design Engineer", The Engineer, vol. 201, pp. 440-442; May 4, 1956.

(26) W. Bongenaar and N.C. de Troye, "Worst-Case Considerations in Designing Logical Circuits", IEEE Trans. on Electronic Computers, vol. EC-14, pp. 590-599; August, 1965.

(27) A. Brameller and J.K. Denmead, "Some Improved Methods for Digital Network Analysis", Proceedings of I.E.E. vol. 109, Part A, pp. 109-116; 1962.

(28) F.H. Branin, Jr., "D-C Analysis Portion of PETAP - A Program for Analyzing Transistor Switching Circuits", IBM Tech. Rept. 00.701, IBM Development Lab. Poughkeepsie, N.Y. 1959.

(29) __________, "D-C and Transient Analysis of Networks Using a Digital Computer", IRE Conv. Record vol. 10, pt. 2, pp. 236-256; 1962.

(30) __________, "Machine Analysis of Networks and Its Application", IBM Tech. Rept. TR 00.855; March 30, 1962.

(31) F.H. Branin, Jr., "A New Method for Steady-State A-C Analysis of RLC Networks", IEEE Internat'l. Conv. Record vol. 14, pt. 7, pp. 218-223; 1966.

(32) N.G. Brooks and H.S. Long, "A Program for Computing the Transient Response of Transistor Switching Circuits-PETAP", IBM Tech. Rept. 00.700, IBM Development Lab. Poughkeepsie, N.Y. 1959.

(33) A. Brown, et al, "Mathematical Circuit Analysis and Design", Remington Rand Univac Div. Scientific Rpt. no. 2. AFCRL-191; March, 1961 ASTIA AD 259786.

(34) H.E. Brown, L.K. Kirchmayer, C.E. Person and G.W. Stagg, "Digital Calculation of 3-phase Short Circuits by Matrix Method", Power Apparatus and Systems no. 52, pp. 1277-82 (AIEE Trans. vol. 79 pt. 3); February, 1961.

(35) R.R. Brown, "A Generalized Computer Procedure for the Design of Optimum Systems", AIEE Trans. Comm. and Electronics vol. 78, no. 1, pp. 285-293; July, 1959.

(36) J.D. Brule and B.P. Sah, "Time Response Characteristics of Linear Networks and Transformation Methods in Network Synthesis", Syracuse Univ. Report; August, 1961; AD 257 822.

(37) K.J. Butler and J.N. Warfield, "A Digital Computer Program for Reducing Logical States to a Minimal Form", Proc. Nat'l. Electronics Conf. vol. 15, pp. 456-466; 1959.

(38) R.T. Byerly, R.W. Long and C.W. King, "Logic for Applying Topological Methods to Electric Networks", Commun. and Electronics, no. 39, (AIEE Trans. vol. 77, pt. 1) pp. 657-667; November, 1958.

(39) D.A. Calahan, "Modern Network Analysis" (book), vol. 1, Chapter 5, Use of the Computer in Approximation. Hayden Book Co. New York, N.Y. 1964.

(40) __________, "Computer Generation of Equivalent Networks", IEEE Conv. Record, vol. 12, pt. 1, pp. 330-337; 1964.

(41) __________, "Computer Solution of the Network Realization Problem", Proc. 2nd Allerton Conf. on Circuit and System Theory, pp. 175-195; 1964.

(42) __________, "Computer Design of Linear Frequency Selective Networks", Proc. IRE vol. 53, pp. 1701-1706; November, 1965.

(43) J.T. Carleton, N. Chackan and T.W. Martin, "The Use of Automatic Programming Techniques for Solving Engineering Problems", Commun. and Electronics no. 45, (AIEE Trans. vol. 78, pt. 1) pp. 596-601; November 1959.

(44) E.V. Carter, "Thermal Analysis of Integrated Circuits", Proceedings 2nd Design Aids Symposium, Autonetics, Anheim, Calif., Pub. no. 558-A-14, paper no. II; September, 1963.

(45) L.C. Casady and H.T. Breen, "Transistor and Diode State Finding Routine", Proceedings 2nd Design Aids Symposium, Autonetics, Anheim, Calif., Pub. no. 558-A-14, paper no. V; September, 1963.

(46) P.W. Case, et al. "Solid Logic Design Automation", IBM J. Res. and Dev. vol. 8, pp. 127-140; April, 1964.

(47) Y.N. Chang and O.M. George, "Use of High-Speed Digital Computors to Study Performance of Complex Switching Networks Incorporating Time Delays", Commun. and Electronics no. 46, (AIEE Trans. vol. 78, pt. 1), pp. 982-987, 1959.

(48) L-H Shang Cheo, "Arbitrary Delay Equalization Utilizing Digital Computer IBM 704", Univ. of California Report; July, 1961. PB 150 512.

(49) D.H. Chung and J.A. Palmieri, "Design of ACP Resistor-Coupled Switching Circuits", IBM J. Res. and Dev. vol. 7, pp. 190-198; July, 1963.

(50) O.P. Clark, "Design of Transistor Feedback Amplifiers and Automatic Control Circuits with the Aid of a Digital Computer", IRE Convention Record, vol. 10, pt. 2, pp. 228-235; 1962.

(51) C. Clunies-Ross and S.S. Husson, "Statistical Techniques in Circuit Optimization", Proc. Nat'l Electronics Conf. vol. 18, pp. 325-334; 1962.

(52) G.H. Cohen and D. Platnick, "The Design of Transistor IF Amplifiers Using an IBM 650 Digital Computer", IRE Trans. Circuit Theory, vol. CT-8 pp. 237-243; September, 1961.

(53) E.U. Cohler, "Statistics Applied to Computer Circuit Design", The Sylvania Technologist, vol. 12, pp. 134-139; October 1959.

(54) D. Coleman, F. Watts and R.B. Shipley, "Digital Calculation of Overhead-Transmission-Line Constants", Power Apparatus and Systems, no. 40, (AIEE Trans. vol. 77, pt. 3), pp. 1266-1268; February, 1959.

(55) D.F. Cooper, "An Integrated Circuit Design for a High Speed Commercial Computer", Solid State Design, vol. 6, no. 6, pp. 21-25; October, 1965.

(56) M.S. Corrington, "Simplified Calculation of Transient Response", Proceedings of IEEE, vol. 53, pp. 287-292; March, 1965.

(57) E.L. Cox, "A Computer-Programmed Component Tolerance Analysis", Diamond Ord. Fuze Labs Report; March 20, 1962, ASTIA AD 269 212.

(58) M.D. Creech, "Finding the Natural Frequencies of an Undamped Linear System with a Digital Computer", American Society of Naval Engineering, pp. 129-132; February, 1962.

(59) D.R. Crosby and H.R. Kaupp, "Calculated Waveforms for Tunnel Diode Locked-Pair Circuit", Proc. Eastern Joint Computer Conf. vol. 18, pp. 233-239; December, 1960.

(60) J.B. Dennis, "Mathematical Programming and Electrical Networks", (book), John Wiley and Sons 1959.

(61) __________, R.F. Nease and R.M. Saunders, "System Synthesis with Aid of Digital Computers", Commun. and Electronics no. 45, (AIEE Trans. vol. 78, pt. 1), pp. 512-515; November, 1959.

(62) __________, "Distributed Solution of Network Programming Problems", Proc. First Allerton Conf. on Circuit and System Theory, pp. 367-384; 1963.

(63) C.A. Desoer and S.K. Mitra, "Design of Lossy Ladder Filters by Digital Computer", IRE Trans. Circuit Theory, vol. CT-8, pp. 192-201; September, 1961.

(64) A.J. DeVilbiss, D.J. Spencer and G.R. Hogsett, "A Worst-Case Circuit Design Technique Utilizing a Small Digital Computer", Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California; October, 1964.

(65) G. Diaz, "Computer Methods for Analyzing Test Data", Electro-Technology vol. 71, no. 5, pp. 152-155; May, 1963.

(66) R.J. Domenico, "Simulation of Transistor Switching Circuits on IBM 704", IRE Trans. on Electronic Computers, vol. EC-6, pp. 242-247; December, 1957.

(67) R.H. Doxtator and F. Arnold, "PRESS: An IBM 704 Program for Performance and Reliability Evaluation by Synthetic Sampling", Tech. Rept. 01.01.112.602, IBM, Endicott, N.Y.; December, 1959.

(68) L.C. Drew and A.G. Atwood, "Using the Computer for Integrated Circuit Analysis", Electronic Industries vol. 23, no. 7, pp. 52-57; July, 1964.

(69) T.E. Duby, "Linear Circuit Analysis Using the SPADE Program", Proceedings 2nd Design Aids Symposium, Autonetics, Anaheim, Calif. Pub. no. 558-A-14, paper no. I: September, 1963.

(70) W.J. Dunnet and Y.C. Ho, "Statistical Analysis of Transistor-Resistor Logic Networks", IRE Conv. Record, vol. 8, pt. 2, pp. 11-40; 1960.

(71) A.H. El-Abiad, "Digital Calculation of Line-to-Ground Short Circuits by Matrix Method", Power Apparatus and Systems no. 48, (AIEE Trans. vol. 79, pt. 3) pp. 323-332; June, 1960.

(72) __________, R. Guidone and G.W. Stagg, "Calculation of Short Circuits Using the High-Speed Digital Computer", Power App. and Systems no. 57, (AIEE Trans. vol. 80, pt. 3), pp. 702-708; December, 1961.

(73) __________, "Digital Computer Analysis of Large Linear Systems", Proc. First Allerton Conf. on Circuit and System Theory, pp. 205-220, 1963.

(74) Electro-Technology Staff Report, "The Computer as a Design Tool - The M.I.T. Approach", Electro-Technology, vol. 72, no. 5, pp. 112-115, pp. 119; November, 1963.

(75) L.R. Fairbrother and H.G. Bassett, "Computer Program for Analysing Networks containing Three-terminal Active Devices characterized by their Two-part Parameters", The Radio and Electronic Engineer, vol. 29, pp. 85-92; February, 1965.

(76) H. Falk, "The Computer as a Design Tool", Electro-Technology, vol. 72, no. 7, pp. 48-50; July, 1963.

(77) __________, "Computer Programs for Circuit Design", Electro-Technology, vol. 73, no. 3, pp. 101-104, pp. 162,164; March, 1964.

(78) J.V. Fall, "A Digital Computer Program for the Design of Phase Correctors", IRE Trans. Circuit Theory, vol. CT-8, pp. 223-236; September, 1961.

(79) __________, "Digital Computer Analysis of Three-Terminal Two-Port Networks", Proc. Inst. Radio Electronics Engrs., Australia; vol. 26, pp. 167-73; May, 1965.

(80) R. Ferguson, "Application of Digital Computer Techniques to Power System Analysis", Proc. First Midwest Symp. on Circuit Analysis, pp. 14-1 to 14-9; 1955.

(81) R.W. Ferguson, R.W. Long and L.J. Rindt, "Digital Calculation of Network Functions Used in Loss Formula Studies", Commun. and Electronics, no. 39, (AIEE Trans. vol. 77, pt. 1), pp. 647-652, November, 1958.

(82) D.C. Fiedler, "A Combinatorial-Digital Computation of a Network Parameter", IRE Trans. Circuit Theory, vol. CT-8, pp. 202-209; September, 1961.

(83) T. Fleetwood, "Automatic Solution of Network Frequency Response", Electronic Engrg. vol. 37, pp. 612-614; September, 1965.

(84) D.A. Franks, "A New Automatic Method for the Design of Low Voltage Transformers on IBM 704", IRE Conv. Record, vol. 8, pt. 6, pp. 193-204; 1960.

(85) W.D. Fryer and W.C. Schultz, "Digital Simulation of Transfer Functions", Proc. Nat'l Electronics Conf. vol. 17, pp. 419-420; 1961.

(86) T. Fujisawa, "Optimization of Low-Pass Attenuation Characteristics by a Digital Computer", Proc. 6th Midwest Symp. on Circuit Theory, pp. P1-P13, 1963.

(87) P.R. Geffe, "Predistorted Filter Design with a Digital Computer", WESCON Record, vol. 2, pt. 2, pp. 10-22; 1958.

(88) ________, "Computer Program Reviews", IRE Trans. on Circuit Theory, CT-9 p. 307; September, 1962
CT-11 p. 512; December, 1964
CT-12 p. 301; June, 1965.

(89) M. Goldberg, "Network Analysis by Computer", Instruments and Control Systems, vol. 38, pp. 175-178; September, 1965.

(90) R.M. Golden, "Digital Computer Simulation of Communication Systems Using the Block Diagram Computer: BLØDIB", Proc. of 3rd Allerton Conf. on Circuit System Theory, pp. 690-707; October, 1965.

(91) G.H. Goldstick and D.G. Mackie, "Design of Computer Circuits Using Linear Programming Techniques", IRE Conv. Record, vol. 9, pt. 2, pp. 224-240; 1961; also IRE Trans. on Electronic Computers, vol. EC-11, pp. 518-530; August, 1962.

(92) R.G. Goodman and R.L. Cummins, "Computer Determination of Isomorphisms in Linear Graphs", Proc. 8th Midwest Symp. on Circuit Theory pp. 12-1 to 12-12; June, 1965.

(93) B.J. Grinnell, "Analysis Testing for Improved Circuit Reliability", Proc. 8th Nat'l. Symp. on Reliability and Quality Control, pp. 103-108; 1962.

(94) P.P. Gupta and M.W.H. Davies, "Digital Computers in Power System Analysis", Proc. Inst. Elec. Engrs. (London) vol. 108, part A, pp. 383-404; October, 1961.

(95) T.J.B. Hannom and G. Kaskey, "Digital Computer as a Design Tool", IRE Conv. Record. vol. 13, pt. 6, pp. 27-38; 1965.

(96) J.N. Hatfield, "A Linear Circuit Analysis Program for the IBM 1620/1311 20k Data Processing System (CIRCS)", Jet Propulsion Lab. Pasadena, Calif.; May, 1964.

(97) M.K. Haynes, "Transient Analysis of Cryotron Networks by Computer Simulation", Proc. IRE, vol. 49, pp. 245-257; January, 1961.

(98) C.L. Hegedus, "TRL Circuit Design Implemented by Computer", Electronic Design vol. 12, pp. 40-46; March 16, 1964.

(99) L. Hellerman, "Monte Carlo Analysis and Design Program", IBM Technical Note, TN00.11000.54.

(100) ________ and M.P. Macite, "Reliability Techniques for Electronic Circuit Design", IRE Trans. on Reliability and Quality Control vol. RQC-14, pp. 9-16; September, 1958.

(101) ________, "A Computer Application to Reliable Circuit Design", IRE Trans. Reliability and Quality Control vol. RQC-11, pp. 9-18; May, 1962.

(102) ________ and E.J. Skiko, "Methods of Analysis of Circuit Transient Performance", IBM J. Res. and Dev. vol. 5, pp. 33-43; January, 1961.

(103) S. Hellerstein, "Synthesis of All-Pass Delay Equilizers", IRE Trans. on Circuit Theory, vol. CT-8, pp. 215-222; September, 1961.

(104) R.H. Hindricks, "A Statistical Method for Analyzing the Performance Variation of Electronic Circuits", Convair Report no. ZX-7-009; October, 1953.

(105) ________, "A Second Statistical Method for Analyzing the Performance Variation of Electronic Circuits", Convair Report No. AZ-7-0100; February, 1956.

(106) W. Hochwald and W.D. Ashcraft, "Design by Worst-Case Analysis", IRE Trans. on Reliability and Quality Control; November, 1961.

(107) J.L. Hogin, "An Active Low-Pass RC Filter Configuration Utilizing the Voltage Follower", Proc. 7th Midwest Symp. on Circuit Theory, pp. 179-191; 1964.

(108) K.H. Hosking, B.M.J. Kavanaugh and M. Sadler, "The Application of 'Deuce' to the Analysis of Linear Passive Networks", Marconi Rev. vol. 25, no. 2, pp. 139-156; 1962.

(109) T. Iedoroko, T. Tsuchiya and H. Watanabe, "A New Calculation Method for the Design of Filters by Digital Computer with the Special Consideration of the Accuracy Problem", IEEE Conv. Record, vol. 11, pt. 2, pp. 100-113; 1963.

(110) International Business Machines Corp., "1620 Electronic Circuit Analysis Program (ECAP 1620-EE-02X)", IBM, Tech. Publ. Dept., White Plains, N.Y.; 1965.

(111) ________., "ASAP, An Automated Statistical Analysis Program", Tech. Rept. prepared for Goddard Space Flight Center, Greenbelt, Md., Contract no. NAS 5-3373.

(112) M.N. John, "A General Method of Digital Network Analysis Particularly Suitable for Use with Low-Speed Computers", Proc. Inst. Elec. Engrs. (London) vol. 108, part A, pp. 369-382; October, 1961.

(113) H.J. Joyal, "Power-Supply Circuit Design by Digital Computer Method", Electrical Mfg. vol. 65, pp. 171-177; May, 1960.

(114) G. Kaskey, N.S. Prywes and H. Lukoff, "Applications of Computers to Circuit Design for UNIVAC LARC", Proc. WJCC, Los Angeles, Calif. vol. 19, pp. 185-205; May, 1961.

(115) J. Katzenelson and L.H. Seitelman, "An Iterative Method for Solution of Nonlinear Resistor Networks", Proc. of 3rd Allerton Conf. on Circuit and System Theory; pp. 647-658; October, 1965.

(116) T.A. Keenan, G.H. Cohn and D. Platnick, "Circuit Study Using Computer Techniques", Rochester University, Final Rept. 74 p.; July 15, 1959. ASTIA AD229842 OTS PB157-593.

(117) J.B. Kida, T.E. Edgerton and C.F. Chen, "Transfer Function Synthesis in the Time Domain--An Extension of Levy's Method", IEEE Trans. on Education, vol. E-8, pp. 62-67; June-September, 1965.

(118) W.H. Kin, et al., "On Interative Factorization in Network Analysis by Digital Computer", Proc. EJCC, vol. 18, pp. 241-253; December, 1960.

(119) E.G. Kimme and F.F. Kuo, "Synthesis of Optimal Filters for a Feedback Quantization System", IEEE Trans. on Circuit Theory, vol. CT-10, pp. 405-413; September, 1963.

(120) S. Klapp, "Empirical Parameter Variation Analysis for Electronic Circuits", IEEE Trans. on Reliability and Quality Control, vol. RQC-13, pp. 34-40; March, 1964.

(121) T.B. Knapp, "An Application of Nonlinear Programming to Mismatched Filters", IEEE Trans. on Circuit Theory, vol. CT-12, pp. 185-193; June, 1965.

(122) F.F. Kuo, "Network Analysis by Digital Computer", Proc. IEEE, vol. 55; June, 1966.

(123) __________, "Computer Techniques in Circuit Analysis", Network Analysis and Synthesis (book), 2nd edition, pp. 438-460; John Wiley, 1966.

(124) __________ and J.F. Kaiser, "System Analysis by Digital Computer", (book) John Wiley, 1966.

(125) C. Kurth, "Analysis of Diode Modulators Having Frequency-Selective Terminations Using Computers", Electrical Communication, vol. 39, no. 3, pp. 369-378; 1964.

(126) D.M. Larson and B.J. Grinell, "A Comparison of Methods of Drift Reliability Determination", Proc. 7th Nat'l. Symp. on Reliability and Quality Control, pp. 448-454; 1961.

(127) G.L. Lasher and J.C. Morgan, "A General Method of Predicting the Transient Response of a Nonlinear Circuit", IBM Research Rept. RC-7; 1957.

(128) A.P. Lechler, D.C. Mark and H.S. Scheffler, "Applying Statistical Techniques to the Analysis of Electronic Networks", Proc. 1962 Nat'l Aerospace Electronics Conf. pp. 168-172.

(129) G.H. Leichner, "Network Synthesis Using a Digital Computer", Proc. N.E.C. vol. 12, pp. 830-838; 1956.

(130) __________, "Designing Computer Circuits with a Computer", J. Assoc. Computing Mach., vol. 4, pp. 143-147; April, 1957.

(131) B.J. Leon and C.A. Bean, "Analysis and Design of Parametric Amplifiers with the Aid of a 709 Computer", IRE Trans. on Circuit Theory, vol. CT-8, pp. 210-215; September, 1961.

(132) V.S. Levadi, "Simplified Method of Determining Transient Response from Frequency Response of Linear Networks and Systems", IRE Conv. Record, vol. 7, pt. 4, pp. 57-68; 1959.

(133) J.I. Levin, "Failure Prevention Through Design Optimization Utilizing Monte Carlo Simulation Techniques", Solid State Design, vol. 6, no. 10, pp. 26-29; October, 1965.

(134) R.W. Levinge, "A Transfer Function Computer", Electronic Engrg., vol. 37, pp. 218-224; April, 1965.

(135) I.C. Liggett, "Examples of Engineering Applications on IBM Digital Computers", Electrical Engrg., vol. 74, pp. 233-236; March, 1955.

(136) M.L. Liou, "Numerical Techniques of Fourier Transforms with Applications", Proc. 2nd Allerton Conf. on Circuit and System Theory, pp. 114-134; 1964.

(137) __________, "A Numerical Solution of Linear Time-Invariant System", Proc. of 3rd Allerton Conf. on Circuit and System Theory, pp. 669-676; October, 1965.

(138) ________, "A Novel Method of Evaluating Transient Response", Proc. IEEE vol. 54, pp. 20-23; January, 1966.

(139) A.J. McElroy and R.M. Porter, "Digital Computer Calculation of Transients in Electrical Networks", Power Apparatus and Systems, no. 59, (AIEE Trans. vol. 82, pt. 3), pp. 88-96; April, 1963.

(140) F.J. McWilliams, "Topological Network Analysis as a Computer Program", IRE Trans. on Circuit Theory, vol. CT-6, pp. 135-136; March, 1959.

(141) A.F. Malmberg, F.L. Cornwell and F.N. Hofer, "NET-1 Network Analysis Program, 7090/94 Version", Report no. LA-3119, Los Alamos Scientific Laboratory, August, 1964.

(142) R.A. Mammano, K.D. Pope and E.P. Schneider, "Computer-Assisted Circuit Analysis", EDN vol. 10, no. 14, pp. 132-139; November, 1965.

(143) J. Marini and R.T. Williams, "The Evaluation and Prediction of Circuit Performance by Statistical Techniques", Proc. Joint Military-Industry Symp. on Guided Missile Reliability; 1957.

(144) D.G. Mark and L.H. Stember, "Variability Analysis", Electro-Technology, vol. 76, no. 1, pp. 37-48; July, 1965.

(145) W. Mayeda and M.E. Van Valkenburg, "Network Analysis and Synthesis by Digital Computers", WESCON Conv. Record, vol. 1, pt. 2, pp. 137-144; 1957.

(146) ________, "Analysis of Nonreciprocal Networks by Digital Computer", IRE Conv. Record, vol. 6, no. 2, pp. 70-75; 1958.

(147) C.S. Meyer, "A Digital Computer Representation of the Linear Constant-Parameter Electronic Network", M.I.T., Cambridge, Mass., Tech. Memo 8436-TM-3, p. 110; August, 1960, ASTIA AD 248437. (OTS PB 155408).

(148) R.S. Miles, "Transient Response using Matrizant Procedures", Proceedings 2nd Design Aids Symposium, Autonetics, Anheim, Calif., Pub. no. 558-A-14, paper no. IV; September, 1963.

(149) M.I.T. Res. Lab. Electronics, "Circuit Simulation on a Digital Computer", Quarterly Progress Report #61; December, 1960 - February, 1961.

(150) E.F. Morris and T.E. Wohen, "Automatic Implementation of Computer Logic", Commun. ACM, vol. 1, no. 5, pp. 14-30; May, 1958.

(151) F. Moskowitz, "The Analysis of Redundancy Networks", Commun. and Electronics, no. 39, (AIEE Trans. vol. 77, pt. 1), pp. 627-632; November, 1958.

(152) W.F. Niesen, "System Analysis Using Digital Computers", Electronic Industries, vol. 19, no. 8, pp. 212-13; August, 1960.

(153) E. Nussbaum, E.A. Irland and C.E. Young, "Statistical Analysis of Logic Circuit Performance in Digital Systems", Proc. IRE vol. 49, pp. 236-244; January, 1961.

(154) E.A. Pacello, "The Use of 'Deuce' for Network Analysis", The Marconi Review, vol. 24, no. 142, Third Quarter; 1961.

(155) K.C. Parton and D.A. Newey, "Power-System Design - New Techniques", J. Science and Technology, vol. 29, no. 1, pp. 9-19; 1962.

(156) C.M. Pierce, "The Design and Analysis of Electrical, Electronic Systems by Means of Digital Computers, An Annotated Bibliography", Lockheed Missiles and Space; September, 1963, p. 34, AD 439 440.

(157) S.C. Plumb, "A Program for Statistical Reliability Evaluation by Synthetic Sampling (STRESS)", Tech. Publ. no. TR 00.834, IBM Poughkeepsie, N.Y.; January, 1962.

(158) C. Pottle, "On the Partial Fraction Expansion of a Rational Function with Multiple Poles by Digital Computer", IEEE Trans. on Circuit Theory, vol. CT-11, pp. 161-162; March, 1964.

(159) __________, "Comprehensive Active Network Analysis by Digital Computer - A State-Space Approach", Proc. of 3rd Allerton Conf. on Circuit and System Theory, pp. 659-676; October, 1965.

(160) M.B. Reed and others, "A Digital Approach to Power-System Engineering", (in 4 parts), Power Apparatus and Systems, no. 48 (AIEE Trans. vol. 79, pt. 3) pp. 198-225; June, 1961.

(161) Remington Rand Univac Division, "Mathematical Circuit Analysis and Design", Remington Rand; February 29, 1960, 40 pp., OTS PB 148222.

(162) A.L. Riemenschneider and C.W. Cox, "Digital Computer Analysis of the Tunnel Diode Relaxation Oscillator", Electronic Engineering, vol. 36, pp. 382-385; 1964.

(163) D. Rigney, L. Kraus and H. Malamud, "Synthesis of Current Waveforms by Type C. Networks", Republic Aviation Report; August, 1961, PB 155 213.

(164) L.G. Roberts, "Circuit Simulation on a Digital Computer", M.I.T. Cambridge, Mass., Quarterly Progress Report, no. 61, pp. 239-251; April, 1961.

(165) L.P.A. Robichaud, M. Boisvert and J. Robert, "Signal Flow Graphs and Applications", (book), Chapter 7 "Algebraic Reduction of Flow Graphs Using a Digital Computer", pp. 181-195; Prentice-Hall, 1962.

(166) C.W. Rosenthal and H. Simon, "Computers as an Aid in the Development of Transmission Circuits", Proc. 2nd Allerton Conf. on Circuit and System Theory, pp. 337-365; 1964.

(167) D.T. Ross and J.E. Rodriguez, "Theoretical Foundations for the Computer-Aided Design System", Proc. 1963 Spring Joint Computer Conf. pp. 305-322.

(168) N.A. Routledge, "Logic on Electronic Computers: A Practical Method for Reducing Expressions to Conjunctive Normal Form", Proc. Camb. Phil. Soc., vol. 52, pp. 161-173; April, 1956.

(169) H. Ruston and J. Bordogna, "Electric Networks: Functions, Filters, Analysis", (book), Section 6.4 "Analysis with a Digital Computer", pp. 435-445; McGraw-Hill Book Co., 1966.

(170) N. Sato, "Digital Calculation of Network Inverse and Mesh Transformation Matrices", Power Apparatus and Systems, no. 50 (AIEE Trans. vol. 79, pt. 3), pp. 719-726; October, 1960.

(171) H.S. Scheffler and F.R. Terry, "Description and Comparison of Five Computer Methods of Circuit Analysis", Proc. 6th Symposium on Ballistic Missiles and Aerospace Technology, Los Angeles, 1961, Ed. by C.T. Morrow, L.D. Ely and M.R. Smith, Academic Press, 1961.

(172) __________, J.J. Duffy and B.C. Spradlin, "MANDEX-A Worst-Case Circuit Analysis Computer Program", Proc. 1962 Nat'l Aerospace Electronics Conf. pp. 38-53.

(173) J.D. Schoeffler, "The Synthesis of Minimum Sensitivity Networks", IEEE Trans. on Circuit Theory, vol. CT-11, pp. 271-276; June, 1964.

(174) H. Schorr, "Computer-Aided Digital System Design and Analysis Using a Register Transfer Language", IEEE Trans. on Electronic Computers, vol. EC-13, pp. 730-737; December, 1964.

(175) L.S. Schwartz, "Statistical Methods in the Design and Development of Electronic Systems", Proc. IRE, vol. 36, pp. 664-670; May, 1948.

(176) C.L. Semmelman, "Experience with a Steepest Decent Computer Program for Designing Delay Networks", IRE Conv. Record, vol. 10, pt. 2, pp. 206-210; 1962.

(177) L. Shalla, "Automatic Analysis of Electronic Digital Circuits Using List Processing", Communications ACM, vol. 9, pp. 372-380; May, 1966.

(178) Y. Shiger, T. Takebe, T. Shinozaki, K. Kimura and S. Watanabe, "Network Design by the Taylor Series Method", Rev. Elect. Commun. Lab. (Japan) vol. 10, no. 9-10, pp. 483-500; September - October, 1962.

(179) R.B. Shipley, D. Coleman and C.F. Watts, "Transformer Circuits for Digital Studies", Power Apparatus and Systems, no. 64 (AIEE Trans. vol. 81, pt. 3) pp. 1028-1031; February, 1963.

(180) J.K. Skwirzynski, "The Use of Digital Computers for Network Analysis", Marconi Rev., vol. 28, no. 158, pp. 195-210; 1965.

(181) B.R. Smith and G.C. Temes, "An Iterative Approximation Procedure for Automatic Filter Synthesis", IEEE Trans. on Circuit Theory, vol. CT-12, pp. 107-112; March, 1965.

(182) H.C. So, "Analysis and Design of Linear Networks with Variable Parameters Using On-Line Simulation", Proc. of 3rd Allerton Conf. on Circuit and System Theory; pp. 634-646; October, 1965.

(183) L.H. Stember, H.S. Scheffler and J.J. Duffy, "Circuit Analysis Techniques Utilizing Digital Computers", Proc. 7th Nat'l Symp. on Reliability and Quality Control, pp. 361-374; 1961.

(184) C.W. Stempin, "Application of Matrizant Operators for General Network Response", Proceedings 2nd Design Aids Symposium, Autonetics, Anheim, Calif., no. 556-A-14, paper no. III; September, 1963.

(185) R.W. Stineman, "Digital Time-Domain Analysis of Systems with Widely Separated Poles", J. of Assoc. Computing Machinery, vol. 12, pp. 286-293; April, 1965.

(186) G. Szentirmai, "Theoretical Basis of a Digital Computer Program Package for Filter Synthesis", Proc. 1st Allerton Conf. on Circuit and System Theory, pp. 37-49; 1963.

(187) N.H. Taylor, "Designing for Reliability", Proc. IRE vol. 45, pp. 811-822; June, 1957.

(188) G.C. Temes, "Filter Synthesis Using a Digital Computer", IRE Conv. Record, vol. 10, pt. 2, pp. 211-227; 1962.

(189) A.O. Thomas, "Calculation of Transmission-Line Impedance by Digital Computer", Power Apparatus and Systems, no. 40 (AIEE Trans. vol. 77, pt. 3) pp. 1270-1274; February, 1959.

(190) J.B. Tommerdahl, A.C. Nelson and K.K. Mazuy, "Mathematical Models for Predicting Pulse Characteristics in Digital Logic Systems", IEEE Trans. on Electronic Computers, vol. EC-13, pp. 705-710; December, 1964.

(191) __________ and A.C. Nelson, "Switching Circuit Transient Performance Prediction Using Empirical Mathematical Modeling Techniques", IEEE Trans. on Reliability and Quality Control, vol. RQC-14, pp. 5-14; March, 1965.

(192) J.G. Truxal, "Numerical Analysis for Network Design", IRE Trans. on Circuit Theory, vol. CT-1, no. 3, pp. 49-60; September, 1954.

(193) C.G. Veinott, "Electric Machinery Design by Digital Computer - After Nine Years", Elect. Engrg. vol. 82, pp. 275-280; April, 1963.

(194) S.Y. Wong and M. Kochen, "Automatic Network Analysis with a Digital Computation System", Comm. and Electronics, no. 24 (AIEE Trans. vol. 75, pt. 1), pp. 172-175; May, 1956.

(195) A.D. Waren and L.S. Lasdon, "Practical Filter Design Using Mathematical Optimization", Proc. of 3rd Allerton Conf. on Circuit and System Theory; pp. 677-689; October, 1965.

(196) H. Watanabe, "A Computational Method for Network Topology", IRE Trans. on Circuit Theory, vol. CT-7, pp. 296-302; September, 1960.

(197) W.J. West and H.S. Scheffler, "Design Considerations for Reliable Electronic Equipment", Pub. no. 558-A-2, Autonetics; October, 1962.

(198) D. Wildfeuer, "The Use of Automatic Digital Computation Machine in Design of Miniature Pulse Transformers and Power Transformers", Proc. Nat'l Electronics Conf. vol. 16, pp. 631-651; 1960.

(199) S.B. Williams, P.A. Abetti and E.F. Magnusson, "How Digital Computers Aid Transformer Designers", General Electric Rev., vol. 58, pp. 24-25; May, 1955.

(200) __________ and H.J. Mason, "Complete Design of Power Transformers with a Large Size Digital Computer", Power Apparatus and Systems, no. 34, (AIEE Trans. vol. 77, pt. 3), pp. 1282-1291; February, 1958.

(201) H.M. Wall, "Using a Computer for Circuit Analysis", Electronics, vol. 2, pp. 56-61; November 2, 1964.

(202) B.A.M. Willcox and R.C.V. Macario, "Iterative Procedure for Analyzing Certain Network by Digital Computer", Proc. Inst. Elect. Engrs. (London) vol. 112, pp. 2243-53; December, 1965.

(203) J.L. Wirth, "The Design and Analysis of Electronic Circuits by Digital Computers", Report no. SC-R-64-1355, Sandia Corp., October, 1964.

(204) K. Yamanoto, K. Fujimoto and H. Watanabe, "Programming the Minimum Inductance Transformation", IRE Trans. Circuit Theory, vol. CT-3 pp. 184-191; September, 1961.

(205) G.W. Zorbist and G.V. Lago, "Digital Computer Analysis of Passive Networks Using Topological Formulas", Proc. 2nd Allerton Conf. on Circuit and System Theory, pp. 573-595; 1964.

SUBJECT INDEX

Ladder Network	63		154							
Logical Network	12		26	37	46	47	70	150	168	
Nonlinear Network	16		115	127						
Numerical Method	5		39	136	137	192				
Power Network	4		54	71	72	80	94	155	160	
	189									
Reliability	11		20	24	67	93	100	101	120	
	126		144	157	187	197				
Simulation	66		85	90	97	149	164	182		
Statistical Technique	6		51	53	70	99	104	105	111	
	128		133	143	144	153	157	175		
Survey	13		15	25	43	76	95	122	135	142
	144		171	193						
Topological Method	18		38	82	92	140	165	196	205	
Transfer Function	85		117	134	158					
Transformer	84		179	198	199	200	204			
Transients	2		7	29	32	36	56	97	102	
	127		132	138	139	148	191			
Transistor	28	45	49	52	59	98	162			
Worst-Case	8		26	64	106	172				

CHRONOLOGICAL INDEX

<u>1964</u>	9	10	21	23	39	40	41	46
	64	68	77	88	96	98	107	120
	125	136	141	158	162	166	173	174
	190	201	203	205				
<u>1965</u>	1	2	3	26	42	55	56	75
	79	83	88	89	90	92	95	110
	115	117	121	133	134	137	142	144
	159	180	181	182	185	191	195	202
<u>1966</u>	18	31	122	123	124	138	169	177

Section II. Study of the Available Programs -- ASAP and ECAP

Three existing digital computer programs written expressly for circuit analysis and evaluation were reviewed and examined, namely, the Automated Statistical Analysis Program (ASAP) [1], the Circuit Analysis System (CIRCS) [2], and the Electronic Circuit Analysis Program (ECAP) [3]; the first and the third being developed by the International Business Machines Corporation, and the second at the Jet Propulsion Laboratory. They may be regarded as the offspring of the same lineage, because they share the same philosophy of attacking the problem and they possess strong similarities in modeling and formating. All three programs have the capability of accepting a topological description of the circuit in simple language, writing the circuit equations according to Kirchhoff's current law, and carrying out the analysis requested.

The ASAP is primarily designed to perform a Monte Carlo statistical analysis on the d-c currents and voltages of circuits containing transistors and diodes. It computes two types of sensitivities. The first type is a qualitative analysis where the measure of the spread of each parameter about the mean value is taken into consideration. The second type is based on a one per cent deviation of each component parameter from its nominal value. The CIRCS program provides options of d-c, a-c, and transient analysis, and also the Mandex worst case and sensitivity calculations. The ECAP, which has recently been released to general public, has the additional feature of including mutual inductance as a circuit element without finding its equivalent tee or pi. The ASAP works on the IBM-7090/94 computer while the other two operate on IBM-1620 with a 1311 disk file system.

CIRCS requires a 20k core storage unit while ECAP requires a 40k core storage. Because ECAP is inclusive of the features of CIRCS, discussions and observations will be made in the this section of the report with regard to ASAP and ECAP only.

One of the justifications in using the digital computer for circuit analysis and design is to obtain information concerning the circuit operation and performance which would otherwise be unobtainable by other means, either for physical reasons or for time considerations. For instance, in the reliability study of a circuit comprising many component parts, it is practically impossible to find out systematically all the effects on the output of varying each component to a different extent on a lab bench. However, a well conceived computer program will have the fortet of carrying out the simulation faithfully and exhaustively. It is with this objective that the ASAP uses Monte Carlo method to produce various statistics of the circuit voltages and currents for any assigned range of tolerance and any shape of statistical distribution curve for each circuit element.

Diodes and transistors in the circuit are to be specified by piecewise linear v-i curves for the diodes, by I_b-V_{be} and I_c-V_{ce} curves for the transistor. There may be 2 to 10 values for each curve. The program determines the equivalent circuit for the diode or transistor and an iterative procedure is followed in locating the operating point. The automatic computation requires a large amount of input data and computer time. Moreover, the convergence of the process in arriving at a satisfactory operating point may be difficult to realize.

ASAP, in writing the nodal equations from the topological description in the data input, uses a pattern recognition subroutine to produce a trace table and establishes the algebraic equations satisfying Kirchhoff's current law. It is significant that the equations are solved algebraically in symbolic form by the Gauss reduction method without back-substitution. The back-substitution occurs numerically during the execution phase. It is quite probable that, during the solution process, some intermediate equation may become longer than the alloted storage space. This may arise as a result of the complexity of the circuit or of the particular sequence of solving one unknown after another. It would be an important factor which could severely limit the actual size of the circuit which can be handled by the program. The official statement concerning the capability limits of the ASAP lists 50 dependent nodes (a dependent node is defined as any node other than ground or those connected to a voltage source) and 40 diodes plus transistors. If these figures truly represent the upper limit of the program, it seems that ASAP will be found useful in quite large population of circuit configurations in practice.

The ASAP program requires a relatively large machine configuration to operate, which may not be readily available in some circumstances. Designers are hoping to be able to make use of digital computers as compactly as a cathode-ray oscilloscope, if not demanding the comparable size and elementary simplicity in use as a slide rule. Technology will advance and meet the challenge in time. At the present time, however, efforts are made in developing programs adaptable for small size computer operation. The ECAP is such an undertaking. The complete ECAP program can be obtained through the IBM 1620 Users' Group.

The ECAP is a card input program designed for operation on IBM 1620 with 1311 disk storage drive. It has the features of automatic equation writing, three options of analysis, d-c, a-c, and transient, computation of partial derivatives and sensitivity coefficients of voltages, and automatic logarithmic modification of frequency in the a-c analysis portion.

Transistors and diodes are represented by their equivalent circuits in the analysis. In the transient calculations the parameter values of the diode and transistor can be made to vary as a function of circuit voltages and currents. To accomplish this, the complement of the circuit elements which are recognized by ECAP contains a "switch" element, which presents the pertinent equivalent circuit for a particular operating region. Thus the three commonly referred to regions of operation of a transistor, cutoff, active, and saturation, can be handled adequately; in a similar manner the diodes can be conducting with different forward resistance or nonconducting.

In the ECAP program the sensitivity coefficients are defined and calculated only for node voltages as their change for a 1% change in the branch parameter. In the worst case analysis both worst-case maximum and worst-case minimum are computed. In the former calculation, positive partial derivatives are multiplied by positive tolerances and negative partial derivatives by negative tolerances. In the latter, positive partial derivatives are multiplied by negative tolerances, etc. The basic assumption is that the circuit output variables are linearly related to the parameter values. This approximation is valid when the parameter tolerances are small. When the tolerance exceeds 10% of the nominal value, the manual recommends the parameter substitution method. First, the partial derivatives of the node voltages are calculated. Then the nominal value of each parameter in the circuit is replaced with its maximum or minimum value, in accordance with the sign of the corresponding partial derivative, and the result is treated as a new ECAP job.

In the d-c analysis program, the d-c parameter modification solutions for a given circuit are obtained by correcting the nodal impedance matrix or the equivalent current vector associated with the circuit. This imposes the condition that the tolerance has to be limited in range in the automatic determination of worst cases. However, the a-c modification solutions, on the other hand, are completely new. Consequently it allows any extent of parameter change in the calculation.

A maximum of five coupled inductances can be included in the circuit that is to be analyzed. This is a feature not often found in other programs.

The transient response of node voltages and element currents are produced by ECAP at the start of a transient solution and at uniform intervals of time thereafter, until the end of the solution is reached. In addition, these output variables are also produced immediately before and immediately after each switch actuation, if any. The time of the switch actuation is also given.

The system of integro-differential equations which arises in the transient analysis is solved in ECAP by an implicit numerical integration technique. It involves two main tasks: the replacement of the system of integro-differential equations by an equivalent set of algebraic difference equations, and the repetitive numerical solution of the algebraic equations. In solving the equations at the end of each series of discrete intervals of time, each new solution is dependent upon the results of the previous solution. That is, the values of certain of the terms in the set of algebraic equations are always computed from the results of the previous solution. For the first solution (at the end of the first time step) these terms are evaluated from the circuit initial conditions. Therefore, the results of each solution become the initial conditions of the succeeding solutions.

References

[1] International Business Machine Corp., "ASAP, An Automated Statistical Analysis Program," Tech. Rept. prepared for NASA Goddard Space Flight Center, Greenbelt, Md., Contract No. NAS 5-3373.

[2] J. N. Hatfield, "A Linear Circuit Analysis Program for the IBM 1620/1311 20k Data Processing System: CIRCS," Jet Propulsion Lab., Pasadena, Calif., May, 1964.

[3] International Business Machines Corp., "1620 Electronic Circuit Analysis Program: ECAP 1620-EE-02X," IBM Tech. Publ. Dept., White Plains, N.Y., 1965.

Section III Adaption of Current Techniques of Computer-Aided Circuit Analysis to Moderate Size Computer

One of the areas of interest to the project for investigation is the possible use of the moderate size computer for circuit analysis. Since the IBM 1620 digital computer is a relatively small machine and is available on campus, it was decided to study programming methods that could be performed using this computer. One method that seemed particularly suitable for programming by the IBM 1620 is the scheme used on the British general purpose computer called Deuce [1]. This method will give the solution of driving point and transfer functions of cascaded networks as a function of sinusoidal frequencies.

The Deuce method of programming was selected for the following reasons: (1) many practical circuits consist of cascaded stages with simple network geometry, although the circuits are composed of many components; (2) it permits and encourages the circuit designer to analyze his design with a minimum of programming experience in a span of time commensurate with bread-boarding a circuit; (3) the program can easily be modified to cope with configurations of various complexities and (4) the program can be run by the designer on a small computer.

1. The Analysis Procedure

The Deuce type program consists essentially of determining the steady state behavior of linear networks consisting of a number of three or four-terminal networks connected in cascade. The technique is designed primarily for identical sections in series. The sections may be one of the following structures: shunt and series branch (ladder network), bridged-T, or lattice. If the structure varies from one section to another, the most complicated segment is taken as the parent structure of the configuration. Other sections are then regarded as special cases of the parent structure by assigning proper values (either short circuit or open circuit) at proper places.

In the simple case of an ordinary ladder network, each section is an L with two branches (Fig. 1a), one shunt and one series. A program written to handle the ladder network is included in this report and will be discussed in detail later. Other programs may be written to handle cascaded networks having bridged-T or lattice sections as the parent structure (Fig. 1b, c, d).

As an illustration of determining the basic structure of a given network, consider the network of Fig. 2. Since one section of the network is the bridged-T, the network is considered a cascade connection of three bridged-T sections, two of which have branches missing. Once the basic structure is decided, the equation for driving point and transfer functions are derived. A table of these functions for all common network configurations can be made and used as needed in the programs.

The program analysis proceeds step by step beginning with the output terminals of the networks and working toward the input terminals as shown in Fig. 3. (It could also be developed by proceeding from the input terminals to the output terminals).

Each section is analyzed knowing the output voltage and output admittance. As a starting point, the output voltage V_o is assigned the reference value of 1.0 volt at an angle of 0^o, and the output admittance Y_o is assigned the value of zero mhos at an angle of 0^o. The input voltage V_1 and input admittance Y_1 of the section are calculated using appropriate equations which have been prepared by the designer and stored in the program. Thus, in general, with V_i and Y_i known, V_{i+1} and Y_{i+1} of the next section are calculated. This procedure is continued until the input voltage V_n and the input admittance Y_n are determined.

Note that although V_o is assumed equal to one volt, the actual value of V_n will ultimately determine the true value of V_o. Similarly, Y_o may be other than zero but this is simply specified at the start of the program, before $Y_1, Y_2, \ldots, Y_n$ are computed.

2. Transforming a Network Diagram to Computer Input

In transforming a given circuit diagram to computer input, the basic component is taken as the series combination of one resistor, one inductor, and one capacitor. Let this RLC series combination be called a "twig". In Fig. 4a is shown several possible forms of a twig. Note that two elements of the same kind, e.g. two resistors, in series, form two twigs. The parallel connection of two or more twigs is a "nest". A "branch" may be formed by a twig, a nest, or a combination of the two. See Fig. 4b and 4c. In the particular case of a ladder network, the twigs, nests, and branches may appear either in the series arm or in the shunt arm. As shown in Fig. 5, the series arm is specified in terms of its impedance and the shunt arm in terms of its admittance.

The key idea of writing the circuit into the computer input is to assign a proper code to each and every twig. The code is interpreted by the machine and thus determines the location of the twig with respect to others in the basic configuration of the network. A twig may be found in several locations in the ladder network. For example it may (a) stand alone in series or shunt arm, (b) be in parallel with other twigs forming a nest, or, (c) be part of a branch composed of a twig and a nest. This is illustrated in Fig. 6.

It often happens that the network structure requires "dummy" twigs be introduced. The program is written in such a way that each series branch and shunt branch must end in a single twig. This twig serves the program control that causes the total impedance or admittance to be calculated. When the given structure does not contain the twig, the dummy twig is inserted. The dummy twig has zero values of inductance, susceptance and resistance and does not affect calculations other than its use as a program control. The use of the dummy twig will be included in the ladder example to be worked out in the following paragraph.

Input data for the circuit to be analyzed are punched on standard 80 column IBM cards. Each twig of the circuit is represented on one IBM card. In general, each IBM card is divided into a number of fields as illustrated in Fig. 7. Two fields F (I) and G (I) are used to designate the position of the twigs in the structure of the cascaded section. Three other fields are used to indicate the value of the L, C and R components. Note that the type of component is designated by giving its value in a specific position of the fields. In the program, the symbol S (where $S = 1/C$) is used instead of C, since values of infinite C cannot be processed by the computer. The symbol H instead of L was used since L represents a number without a decimal in Fortran programming. If H, S or R are short circuits, the value of zero is entered into the respective fields.

3. Analysis of a Ladder Network

Consider the ladder network shown in Fig. 5 where it is desired to find V_n and Y_n, the input voltage and input admittance respectively, from some assumed V_o and Y_o at the output end.

First the equation for the solution of the voltage transfer function V_{r+1}/V_r and the driving-point admittance Y_r per section of the ladder network are derived by the circuit designer.

$$\frac{V_{r+1}}{V_r} = 1 + z_2 \ (Y_r + y_1)$$

$$Y_{r+1} = \frac{V_r}{V_{r+1}} \ (Y_r + y_1)$$

where z and y denote branch impedance (series) and admittance (shunt) respectively, of each ladder section, and Y_r is the input admittance to the section.

Next, it is necessary to decide on a coding scheme in the first two fields, F (I) and G (I), on the data cards for entering the detailed structure of series and shunt branches. In the following example of coding, nine different combinations are possible in stating the location of one twig with respect to others.

F (I)	G (I)	
1	0	Indicates a twig that is part of a nest.
1	1	Indicates a current source G, of value H (I) and angle S (I).
1	2	Indicates a voltage source E, of value H (I) and angle S (I).
-1	-1	Indicates a twig that is in series with a nest, both of which are in the impedance part of the structure.
-1	-1	Indicates a twig that is in series with a nest, both of which are in the admittance part of the structure.
-1	-2	Indicates only one twig exists in an admittance part of the structure.
0	0	Indicates only one twig exists in an impedance part of the structure.
0	-1	not presently used.
0	1	not presently used.

Let us take the specific ladder network of Fig. 8 into consideration. Note that two branches are made up of single nests. At locations designated by (A) and (B) dummy twigs must be inserted. Note that these dummy twigs are located at the high potential ends of the Z and Y branches. The dummy twigs will be the last elements of the branches examined and will, therefore, terminate the branches.

In order to use the ladder network program, there are three types of input cards that must be inserted with the program deck. These cards are:

(1) The input control card. This card sets the limits of the four program loops. Only three of the loops are actually satisfied when solving a problem. In particular, the control card specifies J, L, M and N where

a) J = the number of twigs in the circuit. The network of Fig. 8 shows twelve twigs identified by circled numbers. The twigs numbered 4 and 7 are dummy twigs. This loop must be satisfied since J is equal to the number of input data cards. In the example of Fig. 8, the value of j is 12.

b) L = the number of frequencies at which analysis is desired. The attached program is written to read in five values of frequency-in radians/sec but could easily be extended. The program calls for 5 values of frequency on the read statement and, hence, at least 5 values must be available on the frequency input card. The value of L determines how many of the 5 frequencies will be used in making analysis computations. Therefore, L must be 1, 2, 3, 4, or 5. This loop must be satisfied.

c) M = the number of sections in the complete network. This number tells the machine when the input terminals have been reached. In the example of Fig. 8, there are 3 sections. This loop must be satisfied.

d) N = the number of twigs per section. This number varies from one section to the next. The value of N may be greater than or equal to the maximum number of twigs per section. This loop need not be satisfied. In the example of Fig. 8, there are 6 twigs in one section, hence, the value of N is set to 6 or more.

(2) Input data cards. As input "J" cards are inserted, each card identifies one twig and the order in which the cards are entered is of prime importance. The first input card must identify an admittance branch. If this branch possesses a nest, the first card must identify one of the twigs of the nest. Successive cards identify remaining twigs of the nest and then the terminating series twig, or if none is available, a dummy twig. After the admittance branch is terminated, the nest of the impedance branch, if one exists, is encountered. The last twig of the impedance branch must be a series twig or a dummy twig. In the network of Fig. 8,. the twigs are numbered 1 through 12 in the order in which the data cards should be inserted with the program. Of course, cards 2 and 3, cards 5 and 6, as well as cards 8 and 9 may be interchanged but the order of the other cards may not be changed. As input data cards for the network of Fig. 10, the following cards would be inserted:

Card #	F(I)	G(I)	H(I)	S(I)	R(I)
1	-1	-2	0	0	7
2	1	0	0	0	6
3	1	0	0	10^4	0
4	-1	-1	0	0	0
5	1	0	$3x10^{-3}$	0	0
6	1	0	0	0	5
7	-1	1	0	0	0
8	1	0	$2x10^{-3}$	0	0
9	1	0	0	$5x10^3$	4
10	-1	-1	0	0	3
11	-1	-2	0	$2x10^3$	2
12	0	0	10^{-3}	10^3	1

Note: Columns F (I) and G (I) indicate code while other columns H (I), S (I) and R (I) indicate magnitude of paramenters.

(3) As additional input data, the several values of frequency for which the analysis is desired are specified.

It should be noted that a given problem may be coded in more than one way. Consider the network structure given by Fig. 9. This network may be coded as a single twig impedance in series with a single nest and dummy twig admittance. The input data will be in the following form:

Card #	F(I)	G(I)	H(I)	S(I)	R(I)
1	1	0	L_2	0	R_3
2	1	0	L_1	1/C	0
3	1	0	0	0	R_2
4	-1	1	0	0	0
5	0	0	0	0	R_1

Alternatively, the network may be drawn as shown in Fig. 10, and coded as single y twigs and single z dummy twigs.

Card #	F(I)	G(I)	H(I)	S(I)	R(I)
1	-1	-2	L_2	0	R_3
2	0	0	0	0	0
3	-1	-2	L_1	1/C	0
4	0	0	0	0	0
5	-1	-2	0	0	R_2
6	0	0	0	0	R_1

For the example of Fig. 8, calculations will proceed in this manner: First the impedance value is calculated for the first twig which is R_7. The value of the total admittance $y_t = Y_o + (1/R_7)$ is then obtained. Next the impedance of the R_6 twig is calculated. Since this is a twig of a nest, the value $y_a = 1/R_6$ is calculated. Then the admittance of the C_4 twig is calculated, and then the total admittance $y_a = (1/R_6) + C_4$. The dummy card is read in as the twig of zero value terminating the branch.

In a similar fashion the total impedance is obtained for the series branch. As a consequence, we have

$$V_1 = V_o \; [\, 1 + (\text{total admittance})\,(\text{total impedance})]$$

$$Y_1 = (\text{total admittance})\,(V_o/V_1)$$

The program then moves on to the section 2.

In section 2, the twig containing L_3 is encountered. The impedance is calculated as L_3 and the admittance becomes $1/\ L_3$. Next the twig R_5 is read. The impedance is calculated and the admittance becomes $(1/R_5) + (1/\ L_3)$. Finally the dummy card is read and the total admittance $y_t = (1/R_5) + (1/\ L_3) + Y_1$ is obtained.

The process is repeated until the input terminals are reached.

The complete write-up of the computer program for analyzing the ladder structure of Fig. 5 is contained in Appendix B of this report. It follows the flow chart Fig. 11 and involves four iterative loops of operation. They can be explained as follows.

Block 1

The parameters read here refer to: the number of twigs in the network; the number of frequencies at which analysis is desired; the number of sections in the total network; the maximum number of twigs in a section.

Block 2

The code and value of each twig is read and stored in the memory.

Block 3

The number of values of frequency at which analysis is desired are read and stored in the memory.

Blocks 4, 5 and 6

Initial values of output voltage and output admittance are specified. Note that some of these conditions are within loops and, hence, are executed more or less times than others.

Block 7

The impedance of a twig is calculated by separating the real and imaginary parts and then obtaining the magnitude of the impedance and the associated phase angle.

Block 8

The code of the twig being operated upon is identified. One of six subroutines is chosen.

Blocks 9, 10, 11, 12, 13, 14

Each twig is identified as having a form which must be handled by one of these subroutines. Only one of these subroutines is used for any one twig.

Block 15

At this point, a decision is made. If the loop has been repeated N times, then there are no more twigs in the section and the program precedes to Block 16. If not, the program begins operating on the next twig.

Block 16

The input voltage to the section just operated upon is calculated along with the appropriate phase angle.

Block 17

The input admittance to the section just operated upon is calculated along with the appropriate phase angle. This completes the calculations for this section.

Block 18

The calculated values of input voltage and input admittance of the section are assigned as the output voltages and output admittance for calculations of the next section.

Block 19

At this point, a decision is made. If the section just handled was the final section, then the values of input voltage and associated angle as well as input admittance and angle are punched as output data. If the section was not the final section, loop M is followed which then causes calculations of the next section to begin.

Block 20

It is at this point that output data is punched. The program is written to punch output data at the end of each section and at the end of the last stage. If only the input voltage and admittance are needed, the extra punch statement may be deleted. (Extra punch statement not shown in flow diagram. The statement would occur between Blocks 18 and 19.)

Block 21

At this point, the final decision is made. The complete network analysis has been performed at one frequency. If analysis is desired at additional frequencies, the loop L is entered; if not, the program is complete.

4. Two Numerical Examples Using Ladder Analysis Program

Example 1.

The circuit in Fig. 8 with the following given element values is analyzed.

$R_1 = 1$ ohm	$L_1 = 1$ mh	$C_1 = 10^{-3}$ f
$R_2 = 2$ ohms	$L_2 = 2$ mh	$C_2 = 0.5 \times 10^{-3}$ f
$R_3 = 3$ ohms	$L_3 = 3$ mh	$C_3 = 0.2 \times 10^{-3}$ f
$R_4 = 4$ ohms		$C_4 = 2 \times 10^{-3}$ f
$R_5 = 5$ ohms		
$R_6 = 6$ ohms		
$R_7 = 7$ ohms		

In order to made use of the program, the following steps must be taken.

(i) Determine the number of dummy twigs to be added.

(ii) Count the total number of twigs including the dummies.

(iii) Assign values to J, L, M and N. See the section under "The Input Control Card."

(iv) Code each twig.

(v) Determine the value of frequencies at which the analysis is made.

As a result, the input data cards as printed out in Table 1 is obtained. The output is printed in Table 2.

Example 2.

A two-stage RC coupled transistor amplifier as given in Fig. 12a is to be analyzed. Using the short-circuit admittance model of the transistor in Fig. 12c, the given circuit is replaced by its equivalent Fig. 12b.

Input data are printed out in Table 3 and output is printed out in Table 4.

5. Concluding Remarks

The program described in this section when used to determine the voltage gain and input admittance of a two state RC coupled amplifier occupied approximately thirty thousand positions in the IBM 1620 memory and required approximately two and one-half minutes to process, including the compiling and loading time. Some of the conclusions which may be drawn from the numerical examples shown above may be stated as follows: (1) Fortran language circuit-analysis programs can be generated by circuit designers with some assistance of experienced programmers. (2) Advantage of analyzing cascaded type networks by a ladder method rather than a matrix method is the ability to analyze networks of many stages for only a small increase of memory space. (3) Complex numbers are easily manipulated by separating real and imaginary components. (4) The circuit parameter identification and data are easily entered on a punched card. (5) The program can easily be modified to accomodate many types of cascaded networks.

The inherent limitation of a Deuce type program is the network geometry restriction to cascaded networks. This problem can be resolved by using a matrix program as given in Section IV, but it should be noted that the size of the network will be severely limited due to the large memory space required for the matrix manipulations.

References

[1] E. A. Pacello, "The Use of Deuce for Network Analysis." Marconi Review, vol. 24, pp. 101-114; 1961.

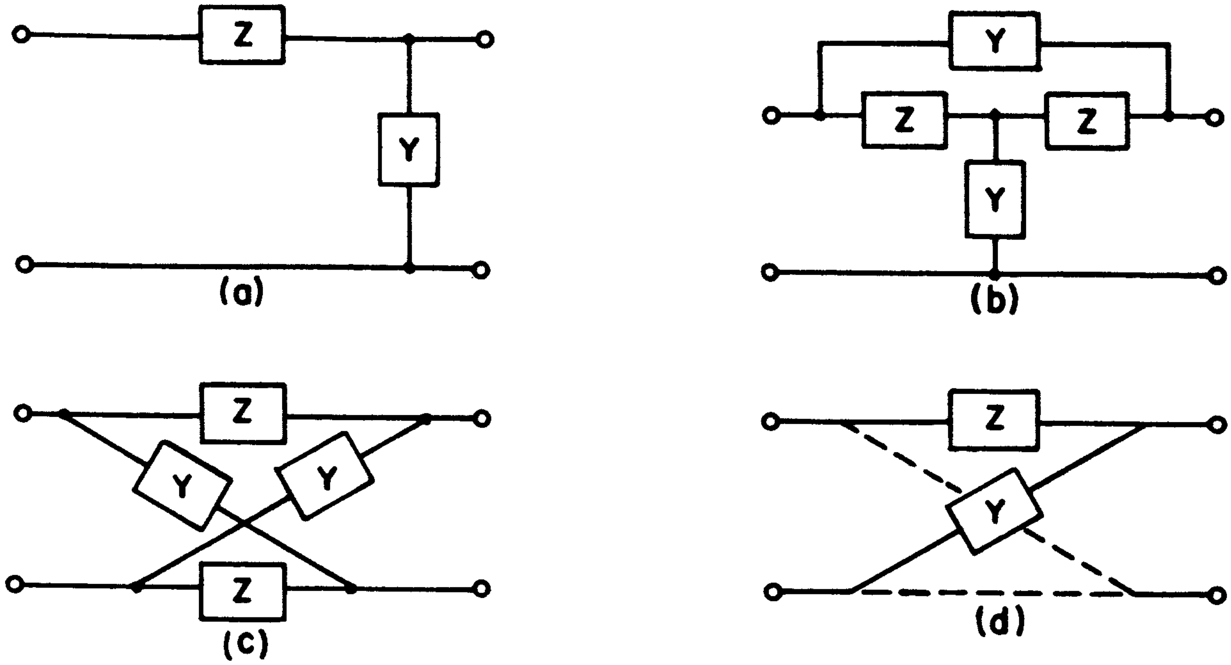


FIG. 1 FOUR BASIC FOUR-TERMINAL NETWORKS.

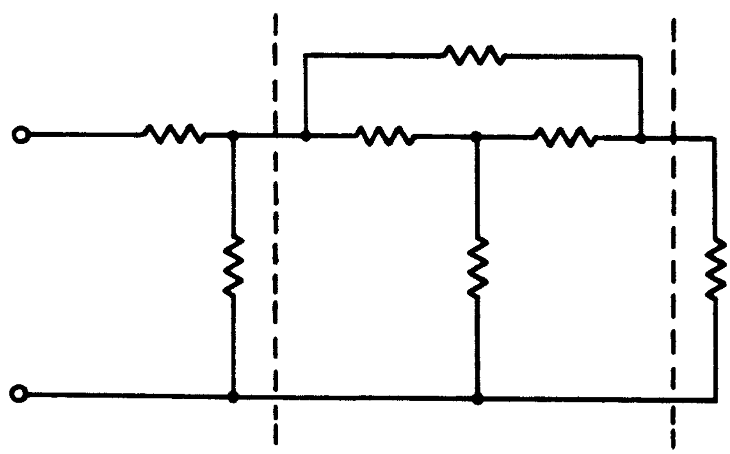


FIG. 2 A CASCADED BRIDGED-T WITH DEGENERATE SECTIONS.

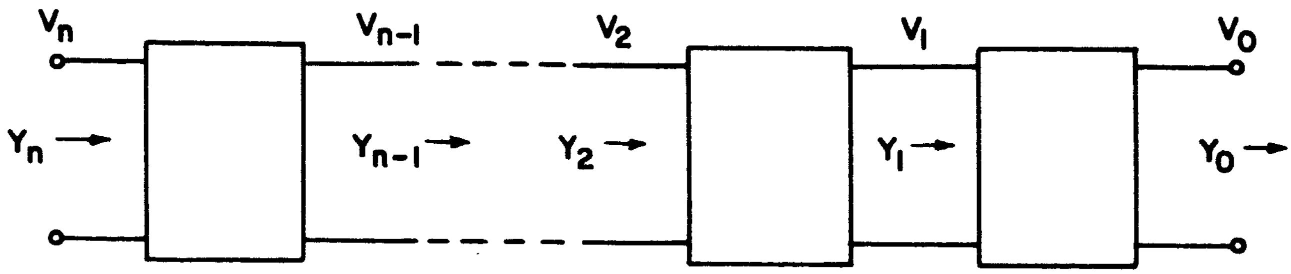


FIG. 3 CASCADED NETWORK CONFIGURATION.

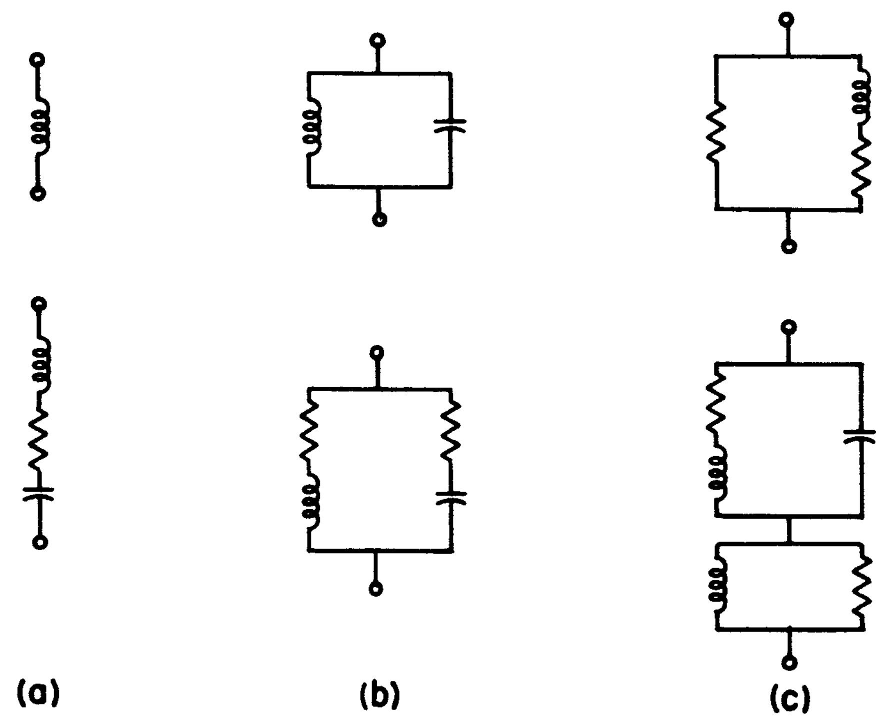


FIG. 4 SEVERAL VARIATIONS OF A TWIG (a), A NEST (b), AND A BRANCH (c).

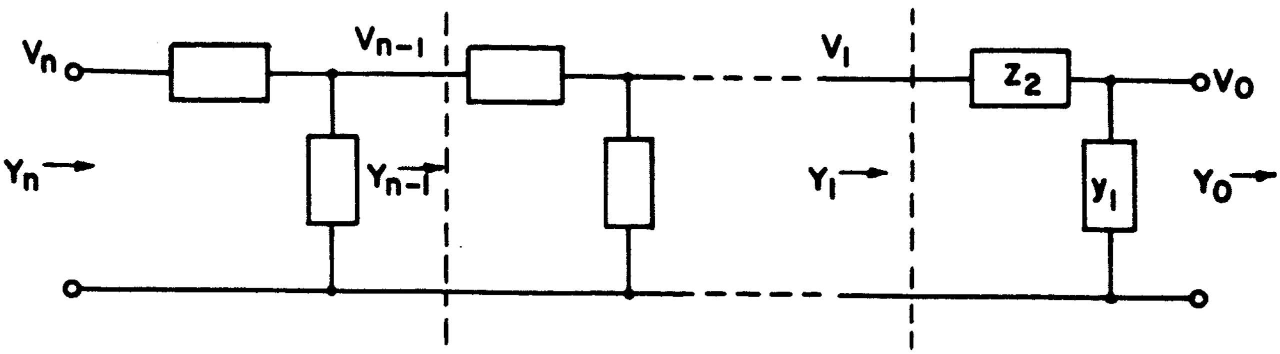


FIG. 5 A LADDER NETWORK.

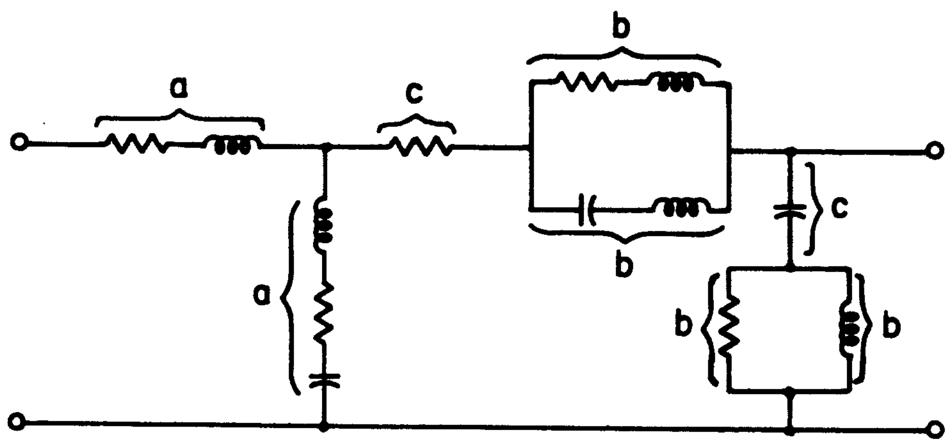


FIG. 6 (a) TWIGS STANDING ALONE; (b) TWIGS IN A NEST; (c) TWIGS IN BRANCHES COMPOSED OF OTHER NESTS.

COLUMN	1-4	5-8	9-23	24-38	39-53	54-80
	F(I)	G(I)	H(I)	S(I)	R(I)	UNUSED

FIG. 7 FIELDS OF AN IBM CARD

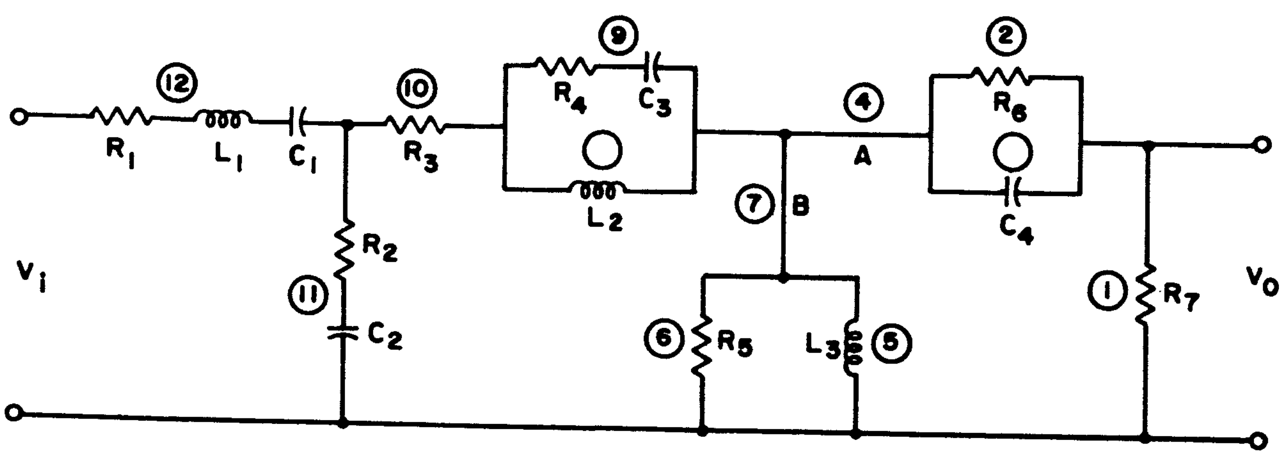


FIG. 8 AN RLC LADDER

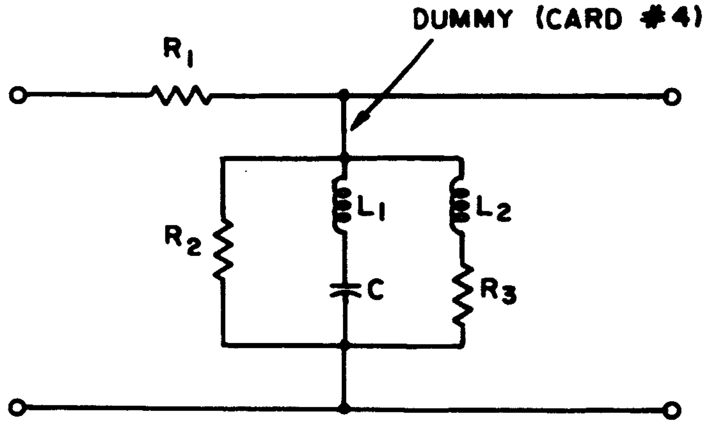


FIG. 9 AN INVERTED L SECTION.

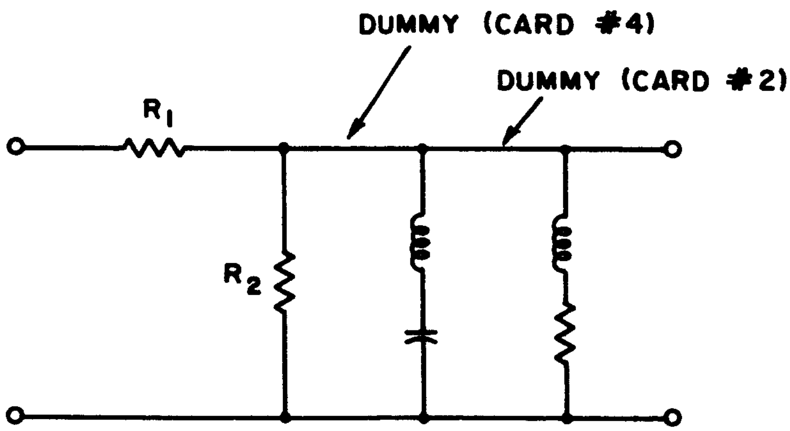


FIG. 10 AN ALTERNATE FORM OF INVERTED L SECTION.

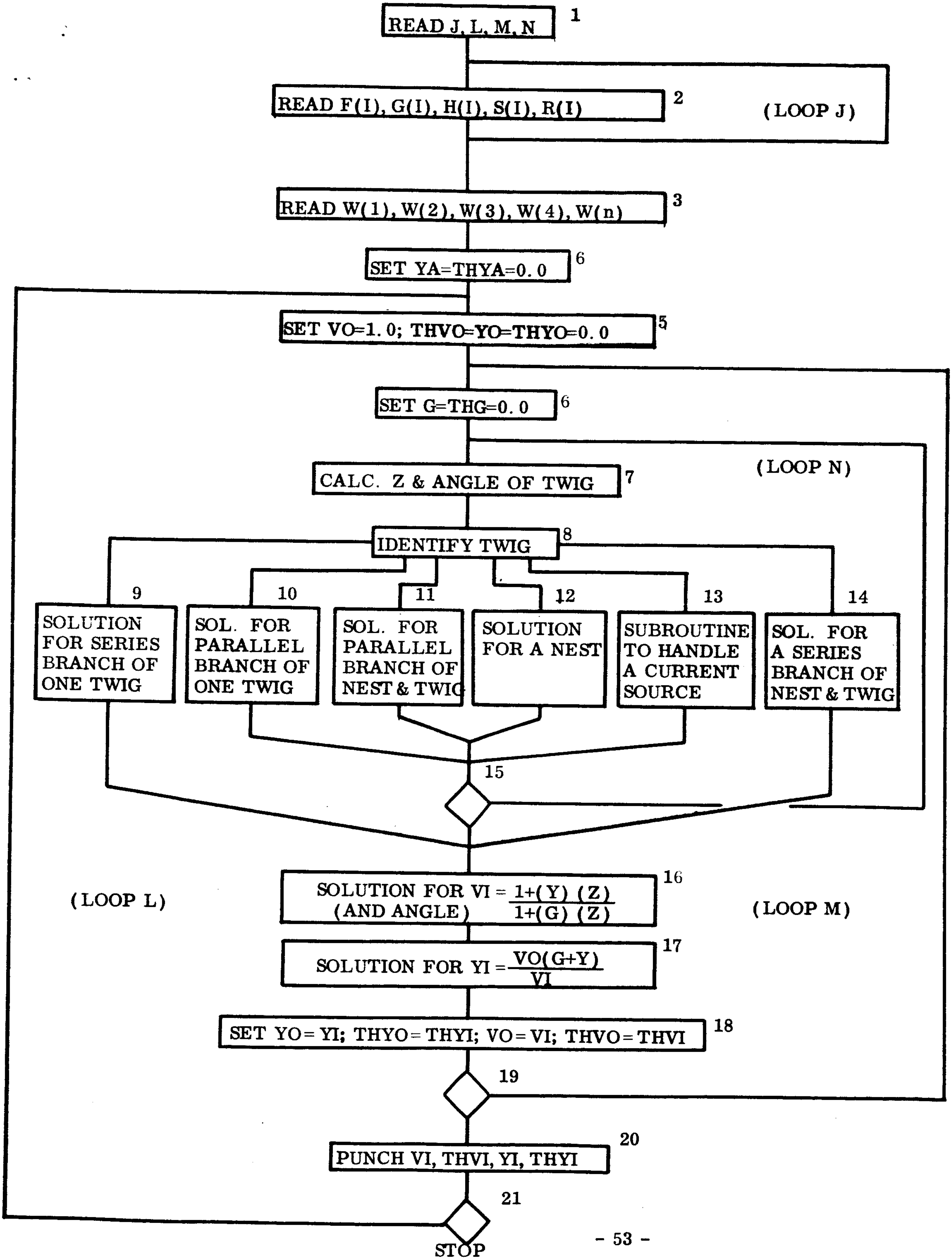


FIG. 11 FLOW CHART OF THE LADDER ANALYSIS PROGRAM

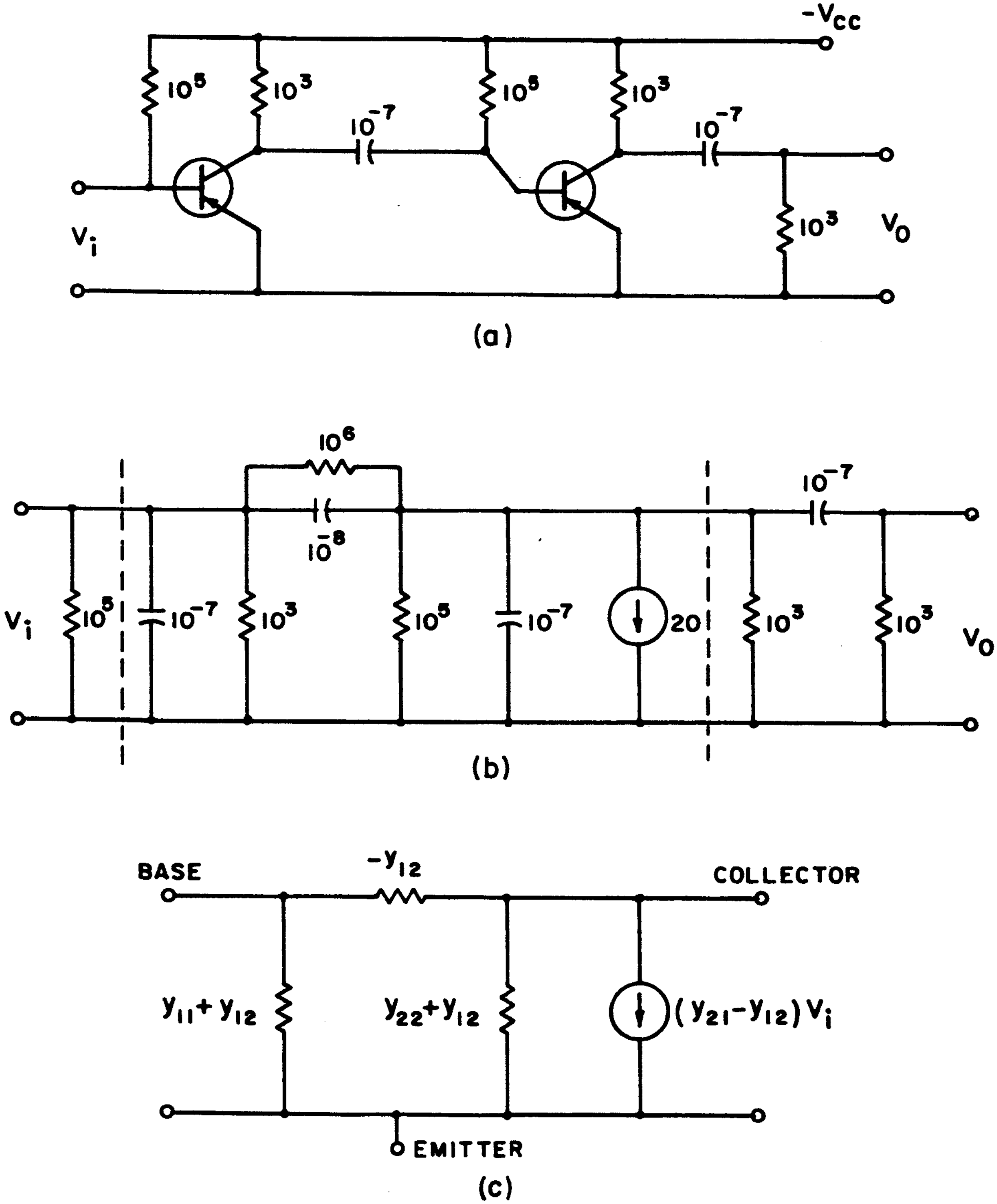


FIG. 12 A TWO-STAGE RC COUPLED AMPLIFIER

```
  12   3    3    6
-1.  -2.          0.0E00          0.0E00          7.0E00
1.   0.           0.0E00          0.0E00          6.0E00
1.   0.           0.0E00          1.0E04          0.0E00
-1.  -1.          0.0E00          0.0E00          0.0E00
1.   0.           3.0E-3          0.0E00          0.0E00
1.   0.           0.0E00          0.0E00          5.0E00
-1.  1.           0.0E00          0.0E00          0.0E00
1.   0.           2.0E-3          0.0E00          0.0E00
1.   0.           0.0E00          5.0E03          4.0E00
-1.  -1.          0.0E00          0.0E00          3.0E00
-1.  -2.          0.0E00          2.0E03          2.0E00
0.   0.           1.0E-3          1.0E03          1.0E00
     5.0E02     1.0E03     2.0E03     0.0E00     0.0E00
```

TABLE 2

```
C··C
 16.66E-02    .00E-99
 17.40E-02 29.14E-02
 18.01E-01-13.13E-02 79.28E-03 13.13E-02 50.00E+01

 66.66E-02 15.70E-01
 69.60E-02 12.79E-01
 10.00E-01 15.70E-01
 10.86E-01 15.39E-01
 54.96E-01 49.76E-02 24.00E-02 55.14E-02 50.00E+01

 98.81E-01 42.48E-02 24.79E-02 89.20E-02 50.00E+01

  9.8816889E+00  4.2481755E-01  2.4799813E-01  8.9207560E-01  5.0000000E+02
 16.66E-02    .00E-99
 19.43E-02 54.04E-02
 16.73E-01-22.79E-02 85.36E-03 22.79E-02 10.00E+02

 33.33E-02 15.70E-01
 38.87E-02 10.30E-01
 50.00E-02 15.70E-01
 62.95E-02 14.15E-01
 43.00E-01 48.08E-03 17.59E-02 61.82E-02 10.00E+02

 61.80E-01 29.55E-02 36.73E-02 48.24E-02 10.00E+02

  6.1807711E+00  2.9552675E-01  3.6731263E-01  4.8245680E-01  1.0000000E+03
 16.66E-02    .00E-99
 26.03E-02 87.60E-02
 14.15E-01-30.23E-02 10.09E-02 30.23E-02 20.00E+02

 16.66E-02 15.70E-01
 26.03E-02 69.47E-02
 25.00E-02 15.70E-01
 40.45E-02 11.10E-01
 37.57E-01-24.58E-02 13.39E-02 52.94E-02 20.00E+02

 57.29E-01 50.57E-02 38.10E-02-27.27E-02 20.00E+02
```

TABLE 3

27	5	5	8	
-1.	-2.	0.0E00	0.0E00	1.0E03
0.	0.	0.0E00	1.0E07	0.0E00
1.	0.	0.0E00	0.0E00	1.0E03
1.	1.	2.0E01	0.0E00	0.0E00
1.	0.	0.0E00	0.0E00	1.0E05
1.	0.	0.0E00	1.0E07	0.0E00
-1.	1.	0.0E00	0.0E00	0.0E00
1.	0.	0.0E00	0.0E00	1.0E06
1.	0.	0.0E00	1.0E08	0.0E00
-1.	-1.	0.0E00	0.0E00	0.0E00
1.	0.	0.0E00	0.0E00	1.0E03
1.	0.	0.0E00	1.0E07	0.0E00
1.	0.	0.0E00	0.0E00	1.0E03
-1.	1.	0.0E00	0.0E00	0.0E00
0.	0.	0.0E00	1.0E07	0.0E00
1.	0.	0.0E00	0.0E00	1.0E03
1.	1.	2.0E01	0.0E00	0.0E00
1.	0.	0.0E00	0.0E00	1.0E05
1.	0.	0.0E00	1.0E07	0.0E00
-1.	1.	0.0E00	0.0E00	0.E00
1.	0.	0.0E00	0.0E00	1.0E06
1.	0.	0.0E00	1.0E08	0.0E00
-1.	-1.	0.0E00	0.0E00	0.0E00
1.	0.	0.0E00	1.0E07	0.0E00
1.	0.	0.0E00	0.0E00	1.0E03
-1.0	1.0	0.0E00	0.0E00	0.0E00
0.	0.	0.0E00	0.0E00	0.0E00
1.0E00	1.0E03	1.0E04	1.0E05	1.0E06

TABLE 4

```
10.00E+03-15.70E-01 10.00E-08 15.70E-01 10.00E-01

10.00E-04    .00E-99
10.10E-04    .00E-99
10.10E-04 99.00E-06
10.00E-07    .00E-99
10.00E-07 99.99E-04
50.55E-02-47.12E-01 19.78E-03 98.54E-04 10.00E-01

10.00E-04    .00E-99
10.00E-04 99.99E-06
20.00E-04 49.99E-06
11.01E+04-62.73E-01 10.00E-08 15.70E-01 10.00E-01

10.00E-04    .00E-99
10.10E-04    .00E-99
10.10E-04 99.00E-06
10.00E-07    .00E-99
10.00E-07 99.99E-04
55.66E-01-94.15E-01 19.78E-03 98.54E-04 10.00E-01

10.00E-08 15.70E-01
10.00E-04 99.99E-06
55.66E-01-94.15E-01 20.78E-03 93.84E-04 10.00E-01

 5.5666943E+00 -9.4153038E+00  2.0784913E-02  9.3848923E-03  1.0000000E+00
10.04E+00-14.71E-01 99.50E-06 14.71E-01 10.00E+02

10.00E-04    .00E-99
10.10E-04    .00E-99
10.14E-04 98.68E-03
10.00E-07    .00E-99
10.04E-06 14.71E-01
52.36E-05-44.10E-01 19.28E-02 12.69E-01 10.00E+02

10.00E-04    .00E-99
10.04E-04 99.66E-03
20.02E-04 49.95E-03
10.14E-01-47.21E-01 99.95E-06 15.70E-01 10.00E+02

10.00E-04    .00E-99
10.10E-04    .00E-99
10.14E-04 98.68E-03
10.00E-07    .00E-99
10.04E-06 14.71E-01
52.36E-06-76.58E-01 19.46E-02 12.66E-01 10.00E+02

10.00E-05 15.70E-01
10.04E-04 99.66E-03
52.36E-06-76.58E-01 19.50E-02 12.61E-01 10.00E+02

 5.2363348E-05 -7.6587626E+00  1.9506404E-01  1.2616574E+00  1.0000000E+03
14.14E-01-78.53E-02 70.71E-05 78.53E-02 10.00E+03

10.00E-04    .00E-99
```

```
10.10E-04   .00E-99
14.21E-04 78.04E-02
10.00E-07   .00E-99
10.00E-05 15.60E-01
15.56E-05-31.12E-01 90.89E-02 74.68E-02 10.00E+03

10.00E-04   .00E-99
14.14E-04 78.53E-02
22.36E-04 46.36E-02
14.18E-02-39.36E-01 99.92E-05 15.69E-01 10.00E+03

10.00E-04   .00E-99
10.10E-04   .00E-99
14.21E-04 78.04E-02
10.00E-07   .00E-99
10.00E-05 15.60E-01
16.53E-06-59.56E-01 85.83E-02 43.92E-02 10.00E+03

10.00E-04 15.70E-01
14.14E-04 78.53E-02
16.53E-06-59.56E-01 85.96E-02 43.97E-02 10.00E+03

 1.6531502E-05 -5.9567135E+00  8.5965164E-01  4.3979027E-01  1.0000000E+04
10.04E-01-99.66E-03 99.50E-05 99.66E-03 10.00E+04

10.00E-04   .00E-99
10.10E-04   .00E-99
10.05E-03 14.70E-01
10.00E-07   .00E-99
10.00E-04 15.69E-01
56.67E-05-18.48E-01 17.73E-01 17.78E-02 10.00E+04

10.00E-04   .00E-99
10.04E-03 14.71E-01
10.19E-03 13.73E-01
10.08E-02-32.30E-01 99.89E-04 15.65E-01 10.00E+04

10.00E-04   .00E-99
10.10E-04   .00E-99
10.05E-03 14.70E-01
10.00E-07   .00E-99
10.00E-04 15.69E-01
10.5 E-05-48.52E-01 95.16E-02 50.75E-03 10.00E+04

10.00E-03 15.70E-01
10.04E-03 14.71E-01
10.59E-05-48.52E-01 95.32E-02 61.17E-03 10.00E+04

 1.0595160E-04 -4.8523403E+00  9.5324231E-01  6.1179165E-02  1.0000000E+05
10.00E-01-99.99E-04 99.99E-05 99.99E-04 10.00E+05

10.00E-04   .00E-99
10.10E-04   .00E-99
10.00E-02 15.60E-01
10.00E-07   .00E-99
```

```
10.00E-03 15.70E-01
55.01E-04-15.98E-01 18.17E-01 23.17E-03 10.00E+05

10.00E-04    .00E-99
10.00E-02 15.60E-01
10.00E-02 15.50E-01
10.09E-02-30.37E-01 99.42E-03 15.16E-01 10.00E+05

10.00E-04    .00E-99
10.10E-04    .00E-99
10.00E-02 15.60E-01
10.00E-07    .00E-99
10.00E-03 15.70E-01
10.57E-04-46.37E-01 95.55E-02 40.48E-03 10.00E+05

10.00E-02 15.70E-01
10.00E-02 15.60E-01
10.57E-04-46.37E-01 96.57E-02 14.40E-02 10.00E+05

 1.0571774E-03 -4.6379896E+00  9.6578132E-01  1.4408966E-01  1.0000000E+06
```

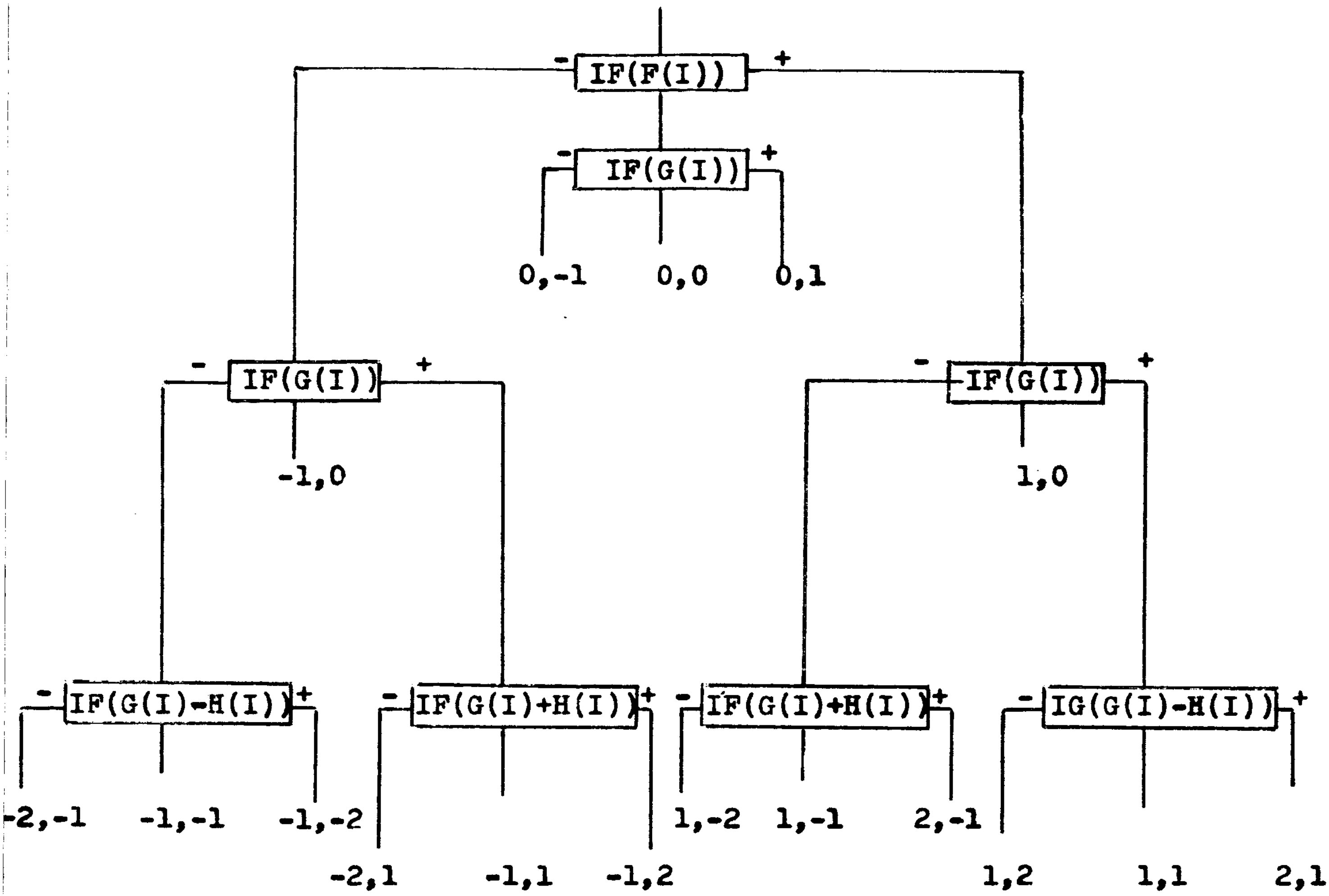
Appendix A

The Ladder Network Program was written in such a way that nine possible codes could be used. To identify twigs, current sources and voltage sources only seven codes were used. Nine codes were, therefore, more than enough to enter appropriate subroutines in the case of the simple ladder network program. When writing programs to handle more complex network structures, it is obvious that a greater number of subroutines will be used and, hence, a greater number of code combinations will be needed. The following sequence of IF statements can be used to enter any one of seventeen subroutines:

```
   IF (F(I)) 1, 2, 3
1  IF (G(I)) 4, 10, 5
2  IF (G(I)) 11, 12, 13
3  IF (G(I)) 6, 14, 7
4  IF (F(I) - G(I)) 15, 16, 17
5  IF (F(I) + G(I)) 18, 19, 20
6  IF (F(I) + G(I)) 21, 22, 23
7  IF (F(I) - G(I)) 24, 25, 26
```

This enables the programmer to enter the following subprogram statement numbers corresponding to the given code.

Statement #	Code F(I)	G(I)
10	-1	0
11	0	-1
12	0	0
13	0	1
14	1	0
15	-2	-1
16	-1	-1
17	-1	-2
18	-2	1
19	-1	1
20	-1	2
21	1	-2
22	1	-1
23	2	-1
24	1	2
25	1	1
26	2	1


IF(F(I))
IF(G(I))
0,-1
0,0
0,1
IF(G(I))
-1,0
IF(G(I))
1,0
IF(G(I)-H(I))
-2,-1
-1,-1
-1,-2
IF(G(I)+H(I))
-2,1
-1,1
-1,2
IF(G(I)+H(I))
1,-2
1,-1
2,-1
IG(G(I)-H(I))
1,2
1,1
2,1

APPENDIX B

```
C · C   RLC ACTIVE-PASSIVE LADDER NETWORK ANALYSIS J HICKS 9/8/65
        READ 22, J,L,M,N
        DIMENSION W(10)
        DIMENSION F(50),G(50),H(50),S(50),R(50)
        DO 1 I=1,J
      1 READ 20, F(I),G(I),H(I),S(I),R(I)
        READ 21, W(1), W(2), W(3), W(4), W(5)
        YA=0.0
        THYA=0.0
        DO 19 J=1,L
        I=0
        VO=1.0
        THVO=0.0
        YO=0.0
        THYO=0.0
        DO 18 K=1,M
        CEYI=0.0
        THCEY=0.0
        DO 17 INDEX=1,N
        I=I+1
        X=W(J)*H(I) - S(I)/W(J)
        Z=SQRT(R(I)**2 + X**2)
        IF(Z)2,3,2
      2 IF(R(I))4,5,4
      3 THZ=0.0
        GO TO 8
      4 THZ=ATAN(X/R(I))
        GO TO 8
      5 IF(X)6,3,7
      6 THZ=-1.57079632
        GO TO 8
      7 THZ=1.57079632
      8 IF(F(I))9,10,11
      9 IF(F(I)-G(I))12,13,14
C       SOLUTION FOR SERIES BRANCH OF ONLY TWIG
     10 ZT=Z
        THZT=THZ
        GO TO 23
     11 IF(F(I)-G(I))15,15,16
C       SOL FOR PARALLEL BRANCH OF A SERIES AND PARALLEL CKT
     12 A=Z*COS(THZ) + 1.0/YA*COS(-1.0*THYA)
        B=Z*SIN(THZ) + 1.0/YA*SIN(-1.0*THYA)
        ZTP=SQRT(A**2 + B**2)
        YTP=1.0/ZTP
        THYTP=-ATAN(B/A)
C       SOL FOR YO + YTP
        A=YTP*COS(THYTP) + YO *COS(THYO)
        B=YTP*SIN(THYTP) + YO*SIN(THYO)
        YT=SQRT(A**2 + B**2)
        THYT=ATAN(B/A)
        YA=0.0
        THYA=0.0
        GO TO 17
C       SOL FOR SERIES BRANCH OF MORE THAN ONE TWIG ZT=Z+1/YA
     13 A=Z*COS(THZ) + 1.0/YA*COS(-1.0*THYA)
```

```
      B=Z*SIN(THZ) + 1.0/YA*SIN(-1.0*THYA)
      ZT=SQRT(A**2 + B**2)
      THZT=ATAN(B/A)
      YA=0.0
      THYA=0.0
      GO TO 23
C     SOLUTION FOR PARALLEL BRANCH WITH ONLY ONE TWIG YP=1/Z+YO
   14 A=1.0/Z*COS(-1.0*THZ) + YO*COS(THYO)
      B=1.0/Z*SIN(-1.0*THZ) + YO*SIN(THYO)
      YT=SQRT(A**2 + B**2)
      THYT=ATAN(B/A)
      GO TO 17
   15 CEYI=H(I)
      THCEY=S(I)
      GO TO 17
C     SOLUTION FOR NEST YA=1/Z+YA
   16 A=1.0/Z*COS(-1.0*THZ) + YA*COS(THYA)
      B=1.0/Z*SIN(-1.0*THZ) + YA*SIN(THYA)
      YA=SQRT(A**2 + B**2)
      THYA=ATAN(B/A)
      PUNCH 24, YA, THYA
   17 CONTINUE
C     SOLUTION FOR VI=VO(1.0+YT*ZT)/(1.0-CEYI*ZT)
   23 YTZ=YT*ZT
      THYTZ=THYT + THZT
C     SOLVE FOR C=1.0 + YTZ
      A=1.0 + YTZ*COS(THYTZ)
      B=YTZ*SIN(THYTZ)
      C=SQRT(A**2+B**2)
      THC=ATAN(B/A)
      CZT=CEYI*ZT
      THCZT=THCEY+THZT
      A=1.0-CZT*COS(THCZT)
      B=-CZT*SIN(THCZT)
      E=SQRT(A**2 + B**2)
      IF(A)30,31,31
   30 THE=ATAN(B/A) + 3.14159264
      GO TO 32
   31 THE=ATAN(B/A)
   32 VI=VO*C/E
      THVI=THVO+THC-THE
C     YI=CEYI + YT*VO/VI
      D=YT*VO/VI
      THD=THYT+THVO-THVI
      A=CEYI*COS(THCEY)+D*COS(THD)
      B=CEYI*SIN(THCEY) + D*SIN(THD)
      YI=SQRT(A**2+B**2)
      THYI=ATAN(B/A)
      PUNCH 24, VI, THVI, YI, THYI, W(J)
      YO=YI
      THYO=THYI
      VO=VI
   18 THVO=THVI
      PUNCH 21, VI, THVI, YI, THYI, W(J)
   19 CONTINUE
```

```
  .
 ..20 FORMAT(2F4.0, 3E15.8)
   21 FO-MAT(5E15.7)
   22 FORMAT(4I4)
   24 FORMAT (5E10.2/)
      STOP
      END
```

Section IV On-Line Experience in the Time-Sharing Computing System

In celebration of M.I.T.'s Centennial Year, the School of Industrial Management of the Massachusetts Institute of Technology sponsored a series of evening lectures on the theme, "Management and the Computer of the Future", in March 1961. During one of the sessions Professor John McCarthy discussed the time-sharing computer systems [1] and introduced the notion of a community utility capable of supplying computer power to each "customer" where, when and in the amount needed. Such a utility would in some way be similar to an electrical power distribution system. There is a large, very large computer complex in some place. Computing services may be obtained at different locations by "inserting a plug into the wall". The time-sharing computer system interacts with many simultaneous users through a number of remote consoles. Such a system will look to each user like a large private computer. This idea goes quite a while back [2], [3], but only recently has it caught wide attention and keen interest in the computing profession. Its experimentation at M.I.T. bears the name of the research project MAC [4]. Other large time-sharing computer systems known in operation include those at System Development Corporation and Carnegie Institute of Technology.

QUIKTRAN [5], developed by the International Business Machines Corporation and Desk Side Computer System [6], developed by the General Electric Company are offered on a commercial basis.

The present computation facilities for academic activities at Villanova University consist of the IBM 1620 Data Processing System. On a first-come-first-served basis, the Computing Center has seen so many instances of overcrowding of many jobs to be processed in the rush hour, and of the inconvenience and frustration of the waiting period before one can get on the computer again in order to fix a misplaced comma in the program. In taking advantage of the time-sharing computing serivce of the General Electric Company, a direct tie line has been established between Villanova University and the General Electric Computer Center at Valley Forge, Pennsylvania, since September 1965.

Villanova University is one of the 85 users that time share the General Electric Computer Complex at Valley Forge. The main frame is the GE 235 Computer with a 20-bit word length and 6 microsecond core memory. The terminal teletypewriter at the user's end does not reach the central processing unit directly; it is first connected to an intermediate computer called Datanet 30 which is analogous to a telephone operator between the main switchboard and the telephone subscribers. Presently there are fifteen lines associated with the CPU through Datanet 30.

The teletype console accepts keyboard input and/or paper tape input. The tie line is rented from the local Bell Telephone Company. The user is allowed to store 32 programs in the computer, each of which is limited to 6,000 characters. He can exercise the option of either using a stored program or submitting a new program in operation. Associated with the Datanet 30 is a mass storage system disk of 20 million bits in BCD form. The computer spends 10 seconds with the user at each round. The Datanet 30, however, is asynchronous in serving the users. There are four Datanet 30 system units for the pool of 15 lines. The computer records the elapsed time in hundredths of a second and prints it out at the end of the task if requested.

The response of the Villanova engineering students to this facility can be judged by the average monthly use in excess of one hundred hours of on-line time. The reason for the immediate and enthusiastic use of this computer facility is the conversational mode of operation where the diagonastic language incorporated for debugging the programs is the main attraction. Also worthy of note is the degree of freedom in using G.E. Fortran as well as a library of mathematical subroutines useful in the solution of engineering problems.

One of the problems which had been worked out on the G.E. facilities is the a-c solution of electric networks using an approach different from that described in Section III. A method for the solution of network responses due to sinusoidal driving forces developed by T. Fleetwood [7] was studied. This approach to nodal circuit analysis is unique in that it does not require the analyst to develop Kirchoff's current equations for the network under study. The method requires only the information of the number of nodes of the network and the elements between the nodes. Between any two nodes, only one element may appear, but this restriction is simplified by reducing series and parallel circuits before the calculation of the responses.

Complex Numbers

One of the difficult problems of using the digital computer for circuit analysis is the processing of complex numbers. This difficulty can be bypassed by replacing each complex quantity by a group of real numbers as shown by the following example:

Given the equations:

$$Y_{11}V_1 + Y_{12}V_2 = 0 \qquad (1a)$$

$$Y_{21}V_1 + Y_{22}V_2 = 0 \qquad (1b)$$

If equation (1a) is separated into real and imaginary parts, one obtains:

$$(a_{11}+jb_{11})(V_{1R}+ jV_{1I}) + (a_{12}+jb_{12})(V_{2R}+jV_{2I}) = 0$$

Then multiplying and simplifying:

$$a_{11}V_{1R}-b_{11}V_{1I}+a_{12}V_{2R}-b_{12}V_{2I}+j(a_{11}V_{1I}+b_{11}V_{1R}+a_{12}V_{2I}+b_{12}V_{2R}) = 0$$

For the above equation to be true both the real and the imaginary parts must be equal to zero giving the following equations:

$$a_{11}V_{1I}+b_{11}V_{1R}+a_{12}V_{2I}+b_{12}V_{2R} = 0$$
$$-b_{11}V_{1I}+a_{11}V_{1R}-b_{12}V_{2I}+a_{12}V_{2R} = 0$$

equation (1b) could be broken up similarly, giving:

$$a_{21}V_{1I}+b_{21}V_{1R}+a_{22}V_{2I}+b_{22}V_{2R} = 0$$
$$-b_{21}V_{1I}+a_{21}V_{1R}-b_{22}V_{2I}+a_{22}V_{2R} = 0$$

All four equations now contain only real numbers and can be solved by conventional methods. Writing these in matrix notation, the equations would be:

$$\begin{vmatrix} a_{11} & b_{11} & a_{12} & b_{12} \\ -b_{11} & a_{11} & -b_{12} & a_{12} \\ a_{21} & b_{21} & a_{22} & b_{22} \\ -b_{21} & a_{21} & -b_{22} & a_{22} \end{vmatrix} \times \begin{vmatrix} V_{1I} \\ V_{1R} \\ V_{2I} \\ V_{2R} \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

From this one can see that each admittance, Y_{ij}, is replaced by the real group $\begin{vmatrix} a_{ij} & b_{ij} \\ -b_{ij} & a_{ij} \end{vmatrix}$ and if this is extended to the general case, the system of equations becomes:

$$
\begin{vmatrix}
a_{11} & b_{11} & a_{12} & b_{12} & \cdots & a_{1n} & b_{1n} \\
-b_{11} & a_{11} & -b_{12} & a_{12} & \cdots & -b_{1n} & a_{1n} \\
a_{21} & b_{21} & a_{22} & b_{22} & \cdots & a_{2n} & b_{2n} \\
-b_{21} & a_{21} & -b_{22} & a_{22} & \cdots & -b_{2n} & a_{2n} \\
\cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\
a_{n1} & b_{n1} & a_{n2} & b_{n2} & \cdots & a_{nn} & b_{nn} \\
-b_{n1} & a_{n1} & -b_{n2} & a_{n2} & \cdots & -b_{nn} & a_{nn}
\end{vmatrix}
\times
\begin{vmatrix}
V_{1I} \\ V_{1R} \\ V_{2I} \\ V_{2R} \\ \cdot \\ \cdot \\ \cdot \\ V_{nI} \\ V_{nR}
\end{vmatrix}
=
\begin{vmatrix}
0 \\ 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 0
\end{vmatrix}
$$

To solve this system of equations, two node voltages must be known, and for ease of operation node one has been used as the input and set equal to one volt; node two is used as the reference or ground node.

Nodal Equations

Once the equations have been written, there are standard routines available for their solution. However, the writing of the equations can often become tedious and drawn out. It is into the removal of this work, that efforts were put.

It is known that any admittance, Y, connected between two nodes, m and n, will be represented in the equation for each node as a self admittance and as a mutual admittance. Since the value of the self admittance is Y and that of the mutual admittance is -Y, the following chart shows where and how to enter Y in the matrix:

Col. / Row	m	n
m	Y	-Y
n	-Y	Y

(3)

Referring to the real number group, the real and imaginary parts will be entered in matrix (4) in the sixteen positions given below:

Col. / Row	2m-1	2m	2n-1	2n
2m-1	a	b	a	b
2m	-b	a	-b	a
2n-1	a	b	a	b
2n	-b	a	-b	a

(4)

Voltage Controlled Current Sources

The effect of a voltage controlled current source upon the matrix can be shown by the following example, see Fig. 1 .

In writing the equations for the five nodes, the effect of the source upon the equations can be observed.

node a $\quad Y_1V_a - 0V_b - Y_1V_c + 0V_d + 0V_e + I_{in} = 0$

node b $\quad 0V_a + (Y_2+Y_4)V_b - Y_2V_c - Y_4V_d - 0V_e - GM(V_d - V_b) = 0$

node c $\quad -Y_1V_a - Y_2V_b + (Y_1+Y_2+Y_3)V_c - Y_3V_d - 0V_e = 0$

node d $\quad 0V_a - Y_4V_b - Y_3V_c + (Y_3+Y_4+Y_5)V_d - Y_5V_e + GM(V_c - V_b) = 0$

node e $\quad 0V_a - 0V_b - 0V_c - Y_5V_d + Y_5V_e - I_0 = 0$

Combining like terms and writing in matrix form, these equations become:

$$\begin{vmatrix} Y_1 & 0 & -Y_1 & 0 & 0 \\ 0 & (Y_2+Y_4)+GM & -Y_2-GM & -Y_4 & 0 \\ -Y_1 & -Y_2 & Y_1+Y_2+Y_3 & -Y_3 & 0 \\ 0 & -Y_4-GM & -Y_3+GM & Y_3+Y_4+Y_5 & -Y_5 \\ 0 & 0 & 0 & -Y_5 & Y_5 \end{vmatrix} \times \begin{vmatrix} V_a \\ V_b \\ V_c \\ V_d \\ V_e \end{vmatrix} = \begin{vmatrix} -I_{in} \\ 0 \\ 0 \\ 0 \\ I_0 \end{vmatrix}$$

Examining these equations, it can be seen that plus or minus GM is added to certain elements in the matrix. The elements to which it is added are given by the following table:

Col. / Row	c	b
b	-GM	+GM
d	+GM	-GM

(5)

where this means that to Y_{bb}, +GM is added and to Y_{bc}, -GM is added, etc.

If the notation GM_{mnpq} is adopted to indicate that a voltage from node m to node n causes a current to flow from node p to node q, the above source would be given by GM_{cbbd}. From this and the preceding table, the table can be written in a general form.

Col. / Row	m	n
p	-GM	+GM
q	+GM	-GM

(6)

Since GM is a real number, when the change is made to the real number form (see eq. 4), only the real parts are affected. The additions to the matrix then become:

Col. / Row	2m-1	2m	2n-1	2n
2p-1	-GM	0	GM	0
2p	0	-GM	0	GM
2q-1	GM	0	-GM	0
2q	0	GM	0	-GM

Using these methods, the nodal equations can be written, and then solved by real number matrix techniques.

Example Problem

The single stage amplifier shown in Fig. 2 was chosen for an example, because of its relative simplicity and the ease with which the results could be checked.

The unusual grid resistor circuitry was chosed only to show how to use the condensation commands to reduce the number of nodes.

The AC equivalent circuits are shown in Figs. 3 and 4. The cathode bias was not simplified to show that the paralleling could be done while writing the equations.

To run this problem on the computer, the input data would be as follows:

5	(total number of nodes)
7	(number of resistors)
0	(number of inductors)
5	(number of capacitors)
10	(number of frequencies for which problem is to be run)

(frequencies)
20
40
80
100
200
400
800
1,000
2Kc
4Kc
8Kc
10Kc
20Kc
40Kc
80Kc
100Kc

2	(number of condensations)
10	(number of passive elements after condensations are made)
1	(number of active sources)

(inductances)	(If there were any inductors, the order in which the values were read would indicate the position in storage, i.e. the first value is stored as 2,1; the second as 2,2, etc.)
(capacitances) 1E-06 3.8E-12 2.8E-12 1E-12 1E-06	(The first value is stored as 3,1; the second as 1,2; the third as 1,3, etc.)
(resistances) 1E06 1E06 5E05 1E04 1E03 6.6E03 1E04	(The first value is stored as 1,1; the second as 1,2; the third as 1,3, etc.)
(condensations)	(The first number signifies the operation; 1 for series combination, 2 for parallel combination. The next four numbers are the locations in storage of the two elements to be combined. After the combination they will be stored in the same place as the first element, i.e., the first condensation says to parallel the elements 1,1 and 1,2 and put the results in 1,1)
(passive topology) 1 3 3 1 2 3 1 1 3 4 3 2 3 5 3 3 2 4 1 4 2 4 1 5 2 4 1 6 4 5 3 4 4 5 1 7 2 5 1 8	(The first two numbers are the nodes between which the element is connected, and the remaining two are the location of the element in storage, i.e., the first group says to put element 3,1 between nodes 1 and 3.)

(active topology) 3.08E-03 3 4 5 4 (The first entry is the transconductance of the source and the next two numbers are the controlling nodes and the last two numbers are the nodes between which the current flows i.e., the voltage from node 3 to node 4 causes a current to flow from node 5 to node 4.)

The preceding data was then put on paper tape and run on the computer. The results as printed out by the computer were:

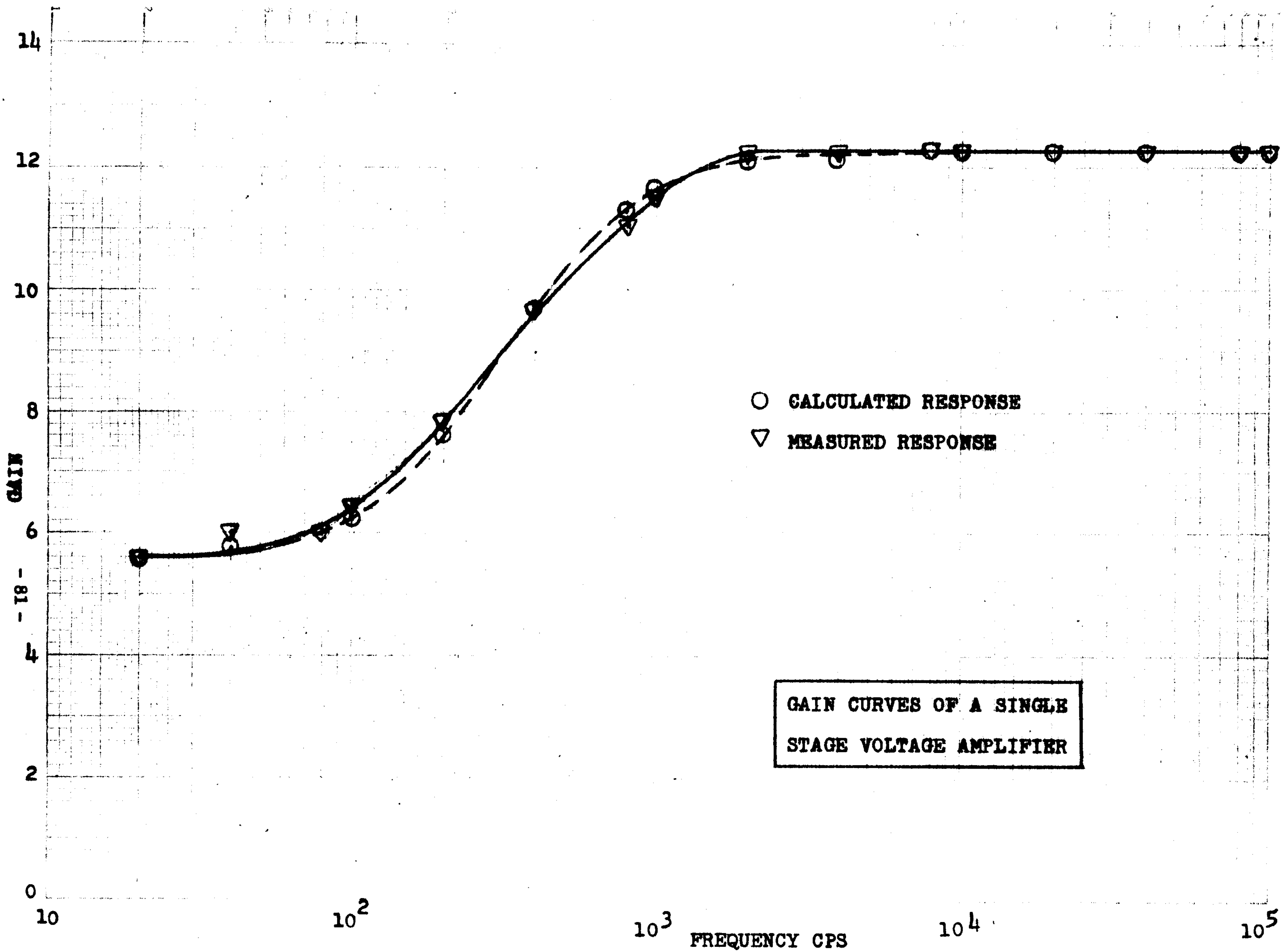
GAIN	DB GAIN	PHASE	FREQUENCY IN CPS
0.56769436E+01	0.15082292E+02	183.957	0.20000000E+02
0.57618576E+01	0.15211250E+02	187.081	0.40000000E+02
0.60765022E+01	0.15673073E+02	192.769	0.80000000E+02
0.62900701E+01	0.15973110E+02	195.065	0.10000000E+03
0.75865646E+01	0.17600903E+02	201.060	0.20000000E+03
0.96962458E+01	0.19732072E+02	199.879	0.40000000E+03
0.11324897E+02	0.21080685E+02	193.044	0.80000000E+03
0.11622281E+02	0.21305827E+02	190.857	0.10000000E+04
0.12076287E+02	0.21638669E+02	185.736	0.20000000E+04
0.12201907E+02	0.21728554E+02	182.893	0.40000000E+04
0.12234158E+02	0.21751481E+02	181.418	0.80000000E+04
0.12238049E+02	0.21754244E+02	181.114	0.10000000E+05
0.12243233E+02	0.21757922E+02	180.471	0.20000000E+05
0.12244468E+02	0.21758798E+02	180.062	0.40000000E+05
0.12244530E+02	0.21758842E+02	179.685	0.80000000E+05
0.12244371E+02	0.21758730E+02	179.541	0.10000000E+06

As a check, the preceding circuit was set up in the lab and the gain was checked for the same frequencies. The results are summarized in the following table:

FREQUENCY IN CPS	GAIN	FREQUENCY IN CPS	GAIN
20	5.6	2,000	12.25
40	6.0	4,000	12.25
80	6.0	8,000	12.25
100	6.5	10,000	12.25
200	7.85	20,000	12.25
400	9.75	40,000	12.25
800	11.05	80,000	12.25
1,000	11.5	100,000	12.25

From the curves plotted, it can be seen that the response as calculated by the computer is in agreement with the experimental data.

The input and computing time for this problem was fifteen minutes, while it took close to an hour to set up the circuit and make the required measurements. The saving in time is even greater than it seems because the computer also calculated phase response while the laboratory procedure did not.


GAIN CURVES OF A SINGLE STAGE VOLTAGE AMPLIFIER
CALCULATED RESPONSE
MEASURED RESPONSE
GAIN
FREQUENCY CPS
0
2
4
6
8
10
12
14
10
10^2
10^3
10^4
10^5

Appendix

Actual Program

Implementing the preceding principles a program has been written for the G.E. Desk Side Computer System (DSCS). The program can be applied to any network made up of admittances and voltage controlled current sources with up to seven nodes. The size limitation is only a factor of memory space and with a larger memory available could easily be extended to twenty or more nodes. Two limitations that have been imposed on the system is that node one be connected only to node three and that the output node have the highest number.

The first thirty-three statements in the program deal with putting in the required data. Statement thirty-four repeats everything that is to follow for each value of frequency in question.

The next sixteen statements calculate the impedance and the admittance for all inductors and capacitors for the frequency in question. If any condensations must be made, the next twenty-two statements will do the required calculations. Statements fifty-seven to sixty-three will combine two elements in series and statements sixty-four to seventy will combine two elements in parallel.

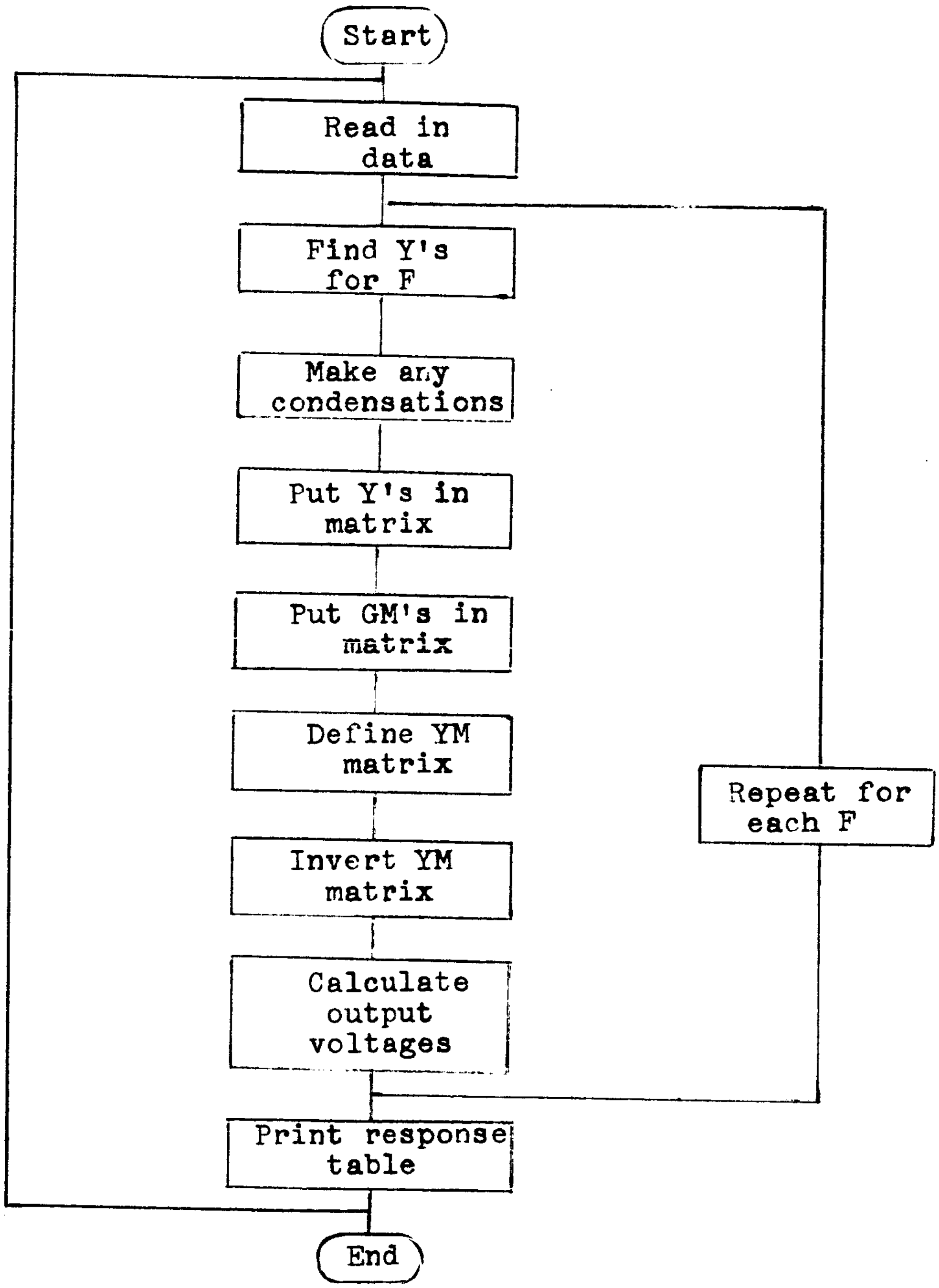
The next twenty-five statements put the admittances into their proper place in the Y matrix, following equation (4). Also included in these statements is the ability to parallel elements by adding the new admittance to any value that was previously entered in the same position.

Then following equation (7), the next eighteen statements add the controlled sources, if any, to the proper places in the matrix.

Since V_{1I} and V_2 were defined as zero, all elements from the first, third and fourth columns and rows disappear. Also since V_{1R} was defined as one volt, all elements of column two are constants and can be moved to the other side of the equal sign. The remaining matrix must then be inverted and to do this it was necessary to define a new matrix YM which does not contain the values from the first four rows and columns.

The next sixty-three statements write the YM matrix and then invert it using the Gauss-Jordan Method. [8] After the matrix has been inverted, it is multiplied by the constant vector to obtain the output. This is simplified since it was stipulated that node one must be connected only to node three and by doing this the constant vector has only one non-zero member and this has the value of the admittance between nodes one and two.

After obtaining the magnitude and phase of the output voltage, the process is repeated for each frequency and then the entire frequency response is printed out.


Start
Read in data
Find Y's for F
Make any condensations
Put Y's in matrix
Put GM's in matrix
Define YM matrix
Invert YM matrix
Calculate output voltages
Print response table
End
Repeat for each F

References

(1) J. McCarthy, "Time-Sharing Computer Systems", collected in Management and the Computer of the Future, ed. M. Greenberger, pp. 219-248, M.I.T. Press, 1962.

(2) R. Fano, "The MAC System: the Computer Utility Approach", IEEE Spectrum, vol. 2, pp. 56-64, January, 1965.

(3) C. Strachey, "Time Sharing in Large, Fast Computers", Proc. International Conf. on Information Processing, UNESCO, Paris, 1960, pp. 336-341.

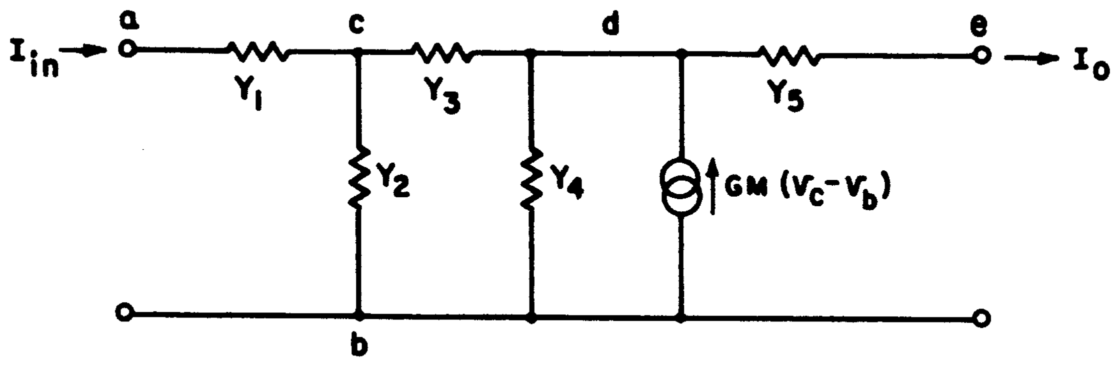
(4) J.C.R. Licklider, "Man-Machine Symbiosis", IRE Trans. on Human Factors in Electronics, vol. HFE-1, pp. 4-11, March, 1960.

(5) International Business Machines Corp., "IBM 7040/7044 QUIKTRAN System Programmer's Guide", File No. 7070-25, September 1965.

(6) General Electric Co., "Desk Side Computer System Reference Manual", April, 1966.

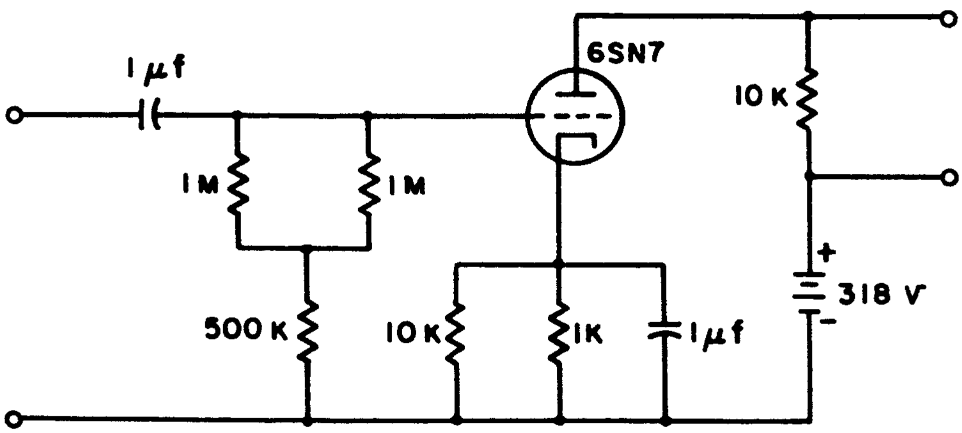
(7) T. Fleetwood, "Automatic Solution of Network Frequency Response", Electronic Engineering, September, 1965.

(8) J.M. McCormick and M.G. Salvadori, "Numerical Methods in Fortran", New Jersey: Prentice-Hall, 1965.



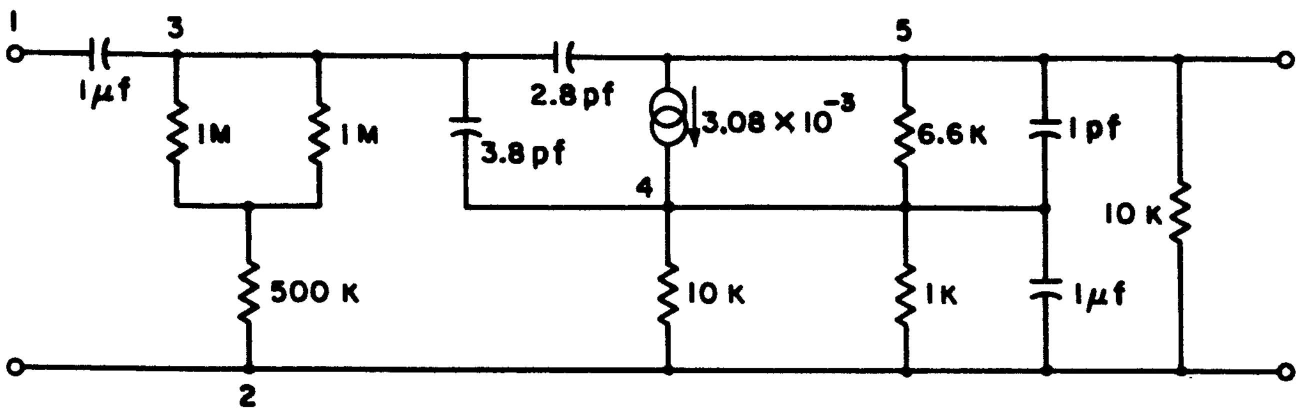
I in
a
c
d
e
I o
Y1
Y3
Y5
Y2
Y4
GM (Vc-Vb)
b

FIG. 1



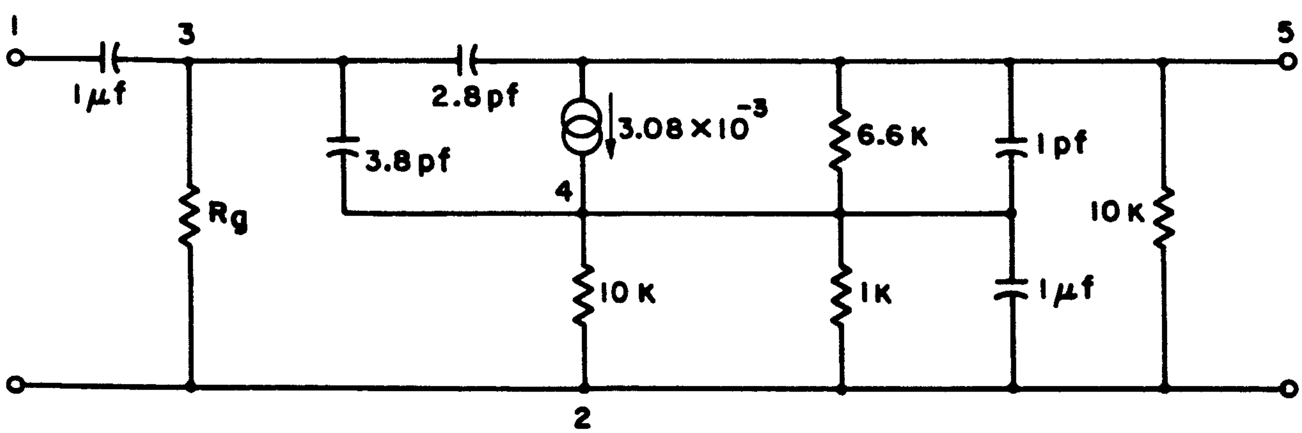
1 μf
6SN7
10 K
1 M
1 M
+
318 V
-
500 K
10 K
1K
1 μf

FIG. 2



1
3
5
1μf
2.8pf
1M
1M
3.8pf
3.08×10^{-3}
6.6K
1pf
4
10K
500K
10K
1K
1μf
2

FIG. 3



1
3
5
1μf
2.8pf
3.08×10^{-3}
6.6K
1pf
3.8pf
Rg
4
10K
10K
1K
1μf
2

FIG. 4

```
C     SOLUTION OF AN ELECTRIC NETWORK
       COMMON Z(3,21),AL(20),C(20),ZA(3,21),ZR(3,21),Y(14,21)
      1ZI(3,21),YA(3,21),YI(3,21),YR(3,21),YB(3,21),YM(14,14)
      3AGM(5),ATL(5),ATM(5),ATN(5),ATK(5),CONO(10),COK(10)
      4COJ(5),COL(5),COM(5),PTI(20),PTJ(20)
      5INDEX(10,3),F(20),V(20),GA(20),PH(20),PTL(20),PTM(20)
 7     PRINT 1003
       READ:NAM,NR,NL,NC,NF
       READ:(F(N),N=1,NF)
       READ:NCON,NTOP,NATO
       NUM=NAM*2
       PI=3.1415927
       IF (NL) 15,15,9
 9     READ:(AL(I),I=1,NL)
 15    IF(NC) 20,20,21
 21    READ:(C(K),K=1,NC)
 20    READ:(Z(1,K),K=1,NR)
       DO 25 K=1,NR
       YB(1,K)=1.0/Z(1,K)
 25    ZA(1,K)=YA(1,K)=0.0
       IF(NCON) 23,23,24
 24    PRINT 1003
       READ:(CONO(I),COJ(I),COK(I),COL(I),COM(I),I=1,NCON)
 23    PRINT 1003
       READ:(PTI(I),PTJ(I),PTL(I),PTM(I),I=1,NTOP)
       IF(NATO) 61,61,62
 62    PRINT 1003
       READ:(AGM(I),ATL(I),ATM(I),ATN(I),ATK(I),I=1,NATO)
 61    CONTINUE
       DO 792 MAK=1,NF
       IF(NL) 28,28,29
 29    DO 27 K=1,NL
       Z(2,K)=2.0*PI*AL(K)*F(MAK)
       YB(2,K)=-1.0/Z(2,K)
 27    ZA(2,K)=YA(2,K)=PI/2.
 28    IF(NC) 34,34,31
 31    DO 33 K=1,NC
 32    YB(3,K)=2.0*PI*C(K)*F(MAK)
       Z(3,K)=1.0/YB(3,K)
 33    ZA(3,K)=YA(3,K)=PI/2.0
 34    IF(NCON) 60,60,42
 42    DO 40 JO=1,NCON
 45    NO=CONO(JO)
       J=COJ(JO)
       K=COK(JO)
       L=COL(JO)
       M=COM(JO)
 47    GO TO (48,58),NO
 48    ZR(J,K)=Z(J,K)*COSF(ZA(J,K))+Z(L,M)*COSF(ZA(L,M))
       ZI(J,K)=Z(J,K)*SINF(ZA(J,K))+Z(L,M)*SINF(ZA(L,M))
       Z(J,K)=SQRTF(ZR(J,K)**2+ZI(J,K)**2)
       ZA(J,K)=ATANF(ZI(J,K)/ZR(J,K))
```

```
      YB(J,K)=1.0/Z(J,K)
      YA(J,K)=-ZA(J,K)
      GO TO 40
58    YR(J,K)=COSF(-ZA(J,K))/Z(J,K)+COSF(-ZA(L,M))/Z(L,M)
      YI(J,K)=SINF(-ZA(J,K))/Z(J,K)+SINF(-ZA(L,M))/Z(L,M)
      YB(J,K)=SQRTF(YR(J,K)**2+YI(J,K)**2)
      Z(J,K)=1.0/YB(J,K)
      YA(J,K)=ATANF(YI(J,K)/YR(J,K))
      ZA(J,K)=-YA(J,K)
40    CONTINUE
60    DO 70 J=1,NUM
      DO 70 K=1,NUM
70    Y(J,K)=0.0
      DO 99 LJ=1,NTOP
      I=PTI(LJ)
      J=PTJ(LJ)
      L=PTL(LJ)
      M=PTM(LJ)
      Y(2*I,2*J)=-YB(L,M)*COSF(YA(L,M))+Y(2*I,2*J)
      Y(2*J,2*I)=-YB(L,M)*COSF(YA(L,M))+Y(2*J,2*I)
      Y(2*I-1,2*J-1)=Y(2*I,2*J)
      Y(2*J-1,2*I-1)=Y(2*J,2*I)
      Y(2*I,2*I)=YB(L,M)*COSF(YA(L,M))+Y(2*I,2*I)
      Y(2*J,2*J)=YB(L,M)*COSF(YA(L,M))+Y(2*J,2*J)
      Y(2*I-1,2*I-1)=Y(2*I,2*I)
      Y(2*J-1,2*J-1)=Y(2*J,2*J)
      Y(2*I-1,2*J)=-YB(L,M)*SINF(YA(L,M))+Y(2*I-1,2*J)
      Y(2*J-1,2*I)=-YB(L,M)*SINF(YA(L,M))+Y(2*J-1,2*I)
      Y(2*I,2*J-1)=-Y(2*I-1,2*J)
      Y(2*J,2*I-1)=-Y(2*J-1,2*I)
      Y(2*I-1,2*I)=YB(L,M)*SINF(YA(L,M))+Y(2*I-1,2*I)
      Y(2*J-1,2*J)=YB(L,M)*SINF(YA(L,M))+Y(2*J-1,2*J)
      Y(2*I,2*I-1)=-Y(2*I-1,2*I)
      Y(2*J,2*J-1)=-Y(2*J-1,2*J)
99    CONTINUE
      IF(NATO) 115,115,100
100   DO 106 MM=1,NATO
      GM=AGM(MM)
      L=ATL(MM)
      M=ATM(MM)
      N=ATN(MM)
      K=ATK(MM)
110   Y(2*N,2*L)=Y(2*N,2*L)+GM
      Y(2*N-1,2*L-1)=Y(2*N-1,2*L-1)+GM
      Y(2*K-1,2*L-1)=Y(2*K-1,2*L-1)-GM
      Y(2*K,2*L)=Y(2*K,2*L)-GM
      Y(2*N-1,2*M-1)=Y(2*N-1,2*M-1)-GM
      Y(2*N,2*M)=Y(2*N,2*M)-GM
      Y(2*K-1,2*M-1)=Y(2*K-1,2*M-1)+GM
      Y(2*K,2*M)=Y(2*K,2*M)+GM
106   CONTINUE
115   DO 120 J=5,NUM
```

```
          DO 120 K=5,NUM
    120   YM(J-4,K-4)=Y(J,K)
```

```
C     SOLUTION OF AN ELECTRIC NETWORK(PART TWO)
          N=NUM-4
    125   DETERM=1.0
    135   DO 145 J=1,N
    145   INDEX(J,3)=0.0
    155   DO 550 I=1,N
    165   AMAX=0.0
    175   DO 180 J=1,N
          IF(INDEX(J,3)-1) 185,180,715
    185   DO 205 K=1,N
          IF(INDEX(K,3)-1)195,205,715
    195   IF(AMAX-ABSF(YM(J,K))) 215,205,205
    215   IROW=J
    225   ICOLUM=K
          AMAX=ABSF(YM(J,K))
    205   CONTINUE
    180   CONTINUE
          INDEX(ICOLUM,3)=INDEX(ICOLUM,3)+1
    260   INDEX(I,1)=IROW
    270   INDEX(I,2)=ICOLUM
    130   IF(IROW-ICOLUM) 140,310,140
    140   DETERM=-DETERM
    150   DO 200 L=1,N
    160   SWAP=YM(IROW,L)
    170   YM(IROW,L)=YM(ICOLUM,L)
    200   YM(ICOLUM,L)=SWAP
    310   PIVOT=YM(ICOLUM,ICOLUM)
          DETERM=DETERM*PIVOT
    330   YM(ICOLUM,ICOLUM)=1.0
    340   DO 350 L=1,N
    350   YM(ICOLUM,L)=YM(ICOLUM,L)/PIVOT
    380   DO 550 L1=1,N
    390   IF(L1-ICOLUM)400,550,400
    400   T=YM(L1,ICOLUM)
    420   YM(L1,ICOLUM)=0.0
    430   DO 450 L=1,N
    450   YM(L1,L)=YM(L1,L)-YM(ICOLUM,L)*T
    550   CONTINUE
    600   DO 710 I=1,N
    610   L=N+1-I
    620   IF(INDEX(L,1)-INDEX(L,2)) 630,710,630
    630   JROW=INDEX(L,1)
    640   JCOLUM=INDEX(L,2)
    650   DO 705 K=1,N
    660   SWAP=YM(K,JROW)
    670   YM(K,JROW)=YM(K,JCOLUM)
    700   YM(K,JCOLUM)=SWAP
```

```
 705  CONTINUE
 710  CONTINUE
      DO 730 K=1,N
      IF(INDEX(K,3)-1) 715,720,715
 715  ID=2
      GO TO 740
 720  CONTINUE
 730   CONTINUE
      ID=1
 740  GO TO (750,760),ID
 760  LIN=0
      PRINT:LIN
        GO TO 850
 750  CONTINUE
      DO 780 I=1,N
      YM(I,1)=YM(I,1)*Y(1,2)
      YM(I,2)=YM(I,2)*Y(2,2)
 780  YM(I,2)=YM(I,2)+YM(I,1)
      V(MAK)=SQRTF(YM(N,2)**2+YM(N-1,2)**2)
      GA(MAK)=20.0/LOGF(10.0)*LOGF(V(MAK))
      PH(MAK)=180.0/PI*ATANF(YM(N-1,2)/YM(N,2))
      IF(YM(N,2)) 775,775,791
 775  PH(MAK)=180.0+PH(MAK)
 791  PRINT:(YM(I,2),I=1,N)
 792  CONTINUE
      PRINT 9000
      PRINT 899,(V(MAK),GA(MAK),PH(MAK),F(MAK),MAK=1,NF)
 850  GO TO 7
 899  FORMAT(E16.8,E16.8,3X,F10.3,3X,E16.8)
 1003 FORMAT("DATA "/)
 9000 FORMAT(/8X,"GAIN",10X,"DB GAIN",10X,"PHASE",6X,
     1 "FREQUENCY IN CPS")
      STOP
      END
```

Section V. On the Accuracy of Monte Carlo Method

It is generally recognized that in order to get meaningful answers from Monte Carlo simulation it is necessary to run the "experiment" a great number of times, varying from run to run only the particular random numbers in generating the combination of parameter values, to provide a large sample typical of the system.

The Automated Statistical Analysis Program [1] developed by IBM for circuit analysis, for example, uses 10,000 as the standard setting for the number of cases to be tested. In ascertaining the adequacy of this number of runs there are two extremes to be kept in mind. Cursory computation on one hand imparts little significance to the results to be useful in assessing the true performance of the circuit under investigation. On the other hand, exhaustive testing, even if it would not exceed the capability of the computer facilities available, defeats the purpose of random sampling which attempts at conclusive results from random selection of sample points. It is to be noted, however, that the accuracy of the statistical method actually bears a nonlinear relationship with the number of runs in the test. The intuitive idea that the more times the computation is carried through, the more meaningful the result will be, is often a vague and sometimes misleading notion.

The accuracy of the Monte Carlo method mainly depends upon two factors: the number of runs and the randomness of the random numbers to be used in the tests. Take the instance of finding the area of an irregular geometric figure by the Monte Carlo method. First, draw the figure on a piece of paper of known dimension and therefore known area, put your finger down at random. Possible outcomes will be (a) the finger will land inside the irregular figure, a "success"; (b) it will be outside the figure, a "failure"; (c) it will come down on the boundary of the area or it may miss the paper entirely. After a large number of trials and ignoring the outcome of (c), the unknown area can be estimated by multiplying the total area of the paper divided by the sum of the number of successes and failures. The accuracy of the answer depends upon two factors. First, the number of trials must be large; second, the finger must be put down in a random manner each time.

Pursued by hand, the Monte Carlo method will only lead to bruised thumbs and poor estimates of the area. Mechanical means can be used to provide random numbers which tell the machine how to "put its finger down". But the wear of mechanical parts will develop a bias in favor of a particular number. With the advent of electronic digital computers, this situation is relieved; and we shall be able to approach randomness as nearly as allowed by the scheme we can devise.

In this section, the accuracy of the Monte Carlo method and some of its main characteristics will be discussed. An exposition on the generation of random numbers will be presented in Section VI.

Bernoulli's Theorem

In the theory of probability one of the most important and beautiful theorems was discovered by Bernoulli (1654-1705) and published with a proof remarkably rigorous in his admirable posthumous book " Ars Conjectandi" (1713). If, in n trials, an event E occurs m times, the number m is called the " frequency" of E in n trials, and the ratio m/n receives the name of " relative frequency" . Bernoulli's Theorem reveals an important porbability relation between the relative frequency of E and its probability p. It may be stated as follows: with the probability approaching 1 or certainty as near as we please, we may expect that the relative frequency (m/n) of an event E in a series of independent trials with constant probability p will differ from that probability by less than any given number $\delta > 0$, provided the number of trials is taken sufficiently large.

In other words, given two positive numbers δ and α, the probability P of the inequality

$$\left| \frac{m}{n} - p \right| > \delta \qquad (1)$$

will be greater than $1 - \alpha$ if the number of trials is above a certain limit depending upon δ and α.

To illustrate Bernoulli's Theorem, Uspensky [2] has given the example that, if $p = 1/2$, $\delta = .01$, $\alpha = .001$, the formula

$$n \geq \frac{1 + \delta}{\delta^2} \ln \frac{1}{\alpha} + \frac{1}{\delta} = 69,869 \tag{2}$$

shows that in 69,869 trials or more there are at least 999 chances against 1 that the relative frequency will differ from 1/2 by less than 1/100. The number 69,869 found as a lower limit of the number of trials is much too large. A much smaller number of trials would suffice to fulfill all the requirements. From a practical standpoint, it is important to find as low a limit as possible for the necessary number of trials (given δ and α).

Since p is the required quantity while m/n is the approximate value obtained by the Monte Carlo method, it follows that the difference $\frac{m}{n} - p$ is the error of the Monte Carlo method. It is clear from the above that this error may be estimated probabilistically with a degree of reliability $1 - \alpha$.

The Limit Theorem in the Bernoulli's Case

The concept of Bernoulli trials, which deals with experiments having only two possible outcomes, is extremely useful because we are often interested only whether a certain result occurs among many possible outcomes or not. For example, although the output voltage of an electric circuit may assume a range of possible values, we are concerned only with whether it exceeds a specified value or not. By Bernoulli's Theorem it is justified to use the ratio of the number of successes m to the total number of trials n, m/n, as an estimate of the binomial probability of success, p. The number of successes changes from one binomial experiment of size n to another. It is thus a random variable, which will be designated as M, with possible values m = 0, 1, 2 . . . n. Since M is a random variable, so is $\hat{p} = \frac{m}{n}$, with possible values 0, 1/n, 2/n, . . . (n-1)/n, 1.

The statistical averages of the random variables M and $\hat{p}$ are:

$$E(M) = np \tag{3}$$

$$Var(M) = npq \qquad \sigma_M = \sqrt{npq} \tag{4}$$

$$E(\hat{p}) = E(\frac{M}{n}) = \frac{1}{n} E(M) = p \tag{5}$$

$$Var(\hat{p}) = Var\ (\frac{M}{n}) = \frac{1}{n^2} Var(M) = \frac{pq}{n} \quad \sigma_{\hat{p}} = \sqrt{pq/n} \tag{6}$$

where q = 1-p. A comparison of equations (4) and (6) reveals the interesting fact that σ_M increases as n increases for fixed p, while $\sigma_{\hat{p}}$ decreases as n increases. If the variance is small, then the value of the random variable tends to be close to its mean, which in this case (so called " unbiased estimate") means close to the true value of the parameter in question.

There are two approaches to find more precisely the relationship between the size of the sample, n, and the error of $\hat{p}$ in the estimation of p, $|\hat{p} - p|$.

1. Conservative Chebyshev Approach

The well-known inequality bearing the name of the Russian mathematician Chebyshev (1821-1894) gives the upper (or lower) bound of such probabilities $P\left[|X - E(X)| \leq C\right]$ when $E(X) = \mu$ and $Var(X) = \sigma^2$ are given. It may be stated as follows: for any positive number C,

$$P\left[|X - \mu| \geq h\sigma\right] \leq \frac{1}{h^2} \tag{7}$$

This means that the probability assigned to values of X outside the interval $\mu - h\sigma$ to $\mu + h\sigma$ is at most $1/h^2$. In other words, at least the fraction $1 - (1/h^2)$ of the total probability of a random variable lies within h standard deviation of the mean.

In applying the Chebyshev inequality with $\mu = p$ and $\sigma = \sqrt{pq/n}$ in the case at hand, we find the probability that p is within $h\sqrt{pq/n}$ of p is at least $1 - (1/h^2)$. One difficulty is that σ is dependent upon the exact value of p which is to be estimated by $\hat{p}$. However, we can find the value of p that maximizes $\sigma^2 = pq/n$. Since the graph of $pq = p(1 - p)$ is a parabola that is symmetrical about the line of $p = 1/2$, the maximum value of pq is attained when $p = q = 1/2$. Therefore, the maximum value of pq is $1/2 \cdot 1/2 = 1/4$, and

$$\max \sigma = \sqrt{pq/n} = 1/\sqrt{4n}$$

Therefore we can say conservatively that the probability is at least $1 - (1/h^2)$ with the distance

$$|\hat{p} - p| \leq \frac{h}{\sqrt{4n}} \tag{8}$$

For example, if $n = 1,000$ and if we choose $h = 2$, the probability is at least 0.75 that

$$|\hat{p} - p| \leq \frac{2}{\sqrt{4 \times 1,000}} = .032$$

or, in words, at least 75% of the probability distribution of $\hat{p}$ is within .032 of p. For $n = 1,000$ and $h = 5$, at least 96% of the probability distribution of the error is less than $5/\sqrt{4,000} \cong .065$.

It is clear from equation (8) that the error in the approximate solution of a problem by the Monte Carlo method can be reduced by increasing the number of trials n, i.e. by increasing the computational time. For example, the time necessary to complete the solution must be increased by a factor of 100 if the accuracy is to be improved by one order of magnitude.

2. Conservative Normal Approach

From DeMoiore-Laplace Theorem [3] in the theory of probability, it is known that when the mean value μ is " far" from 0 and n, the extreme values of the binomial random variable X, (at least 3σ from both 0 and n), it is justified to use the stronger normal distribution theory instead of the Chebyshev Theorem. In our case then, if np is at least $3\sqrt{npq}$ from both 0 and n, we know that the new random variable $Z = (X - np)/\sqrt{npq}$ is approximately normally distributed. Recalling that $\hat{p} = \frac{X}{n}$, we have

$$Z = \frac{X - np}{\sqrt{npq}} = \frac{\frac{X}{n} - p}{\sqrt{pq/n}} = \frac{\hat{p} - p}{\sqrt{pq/n}}$$

Now, since Z is approximately distributed according to the standard normal distribution, we can say that the probability is approximately 0.95 that

$$-2 \leq Z \leq 2 \quad \text{or} \quad -2 \leq \frac{\hat{p} - p}{\sqrt{pq/n}} \leq 2. \tag{9}$$

where the Z's represent 2 standard deviations, to approximate the more precise value 1.96 from the normal table [4].

We now multiply all terms of the right-hand expression of the inequality (9) by $\sqrt{pq/n}$, and get

$$-2\sqrt{pq/n} \leq \hat{p} - p \leq 2\sqrt{pq/n}$$

$$\text{or} \qquad |\hat{p} - p| \leq 2\sqrt{pq/n}$$

Maximizing pq as before at pq = $\frac{1}{4}$, we find from the normal distribution that the probability is approximately 0.95 and

$$|\hat{p} - p| \leq \frac{2}{\sqrt{4n}} = \frac{1}{\sqrt{n}}$$

If we choose h standard deviations instead of 2, the appropriate probability should be obtained from the normal table.

In general, the number of runs (n) required in the Monte Carlo method can thus be determined on the basis of normal distribution approach by the simple relation

$$n = \frac{C}{4E^2}$$

where E is the tolerable error range in per cent and C is the square of probability value for a given confidence limit.

For example, for 90 per cent confidence limit, C has the value of (1.64) = 2.69; for 95 confidence limit, C = (1.96) = 3.84; for 99 per cent confidence limit, C = (2.57) =6.61. If we want the simulation result to be within ± .05 error range, the number of runs corresponding to the three confidence limits would be 269, 384 and 661 respectively. Returning to the figure given in the ASAP operating manual, a 10,000-run computation will guarantee the result to be within ± .013 error range with 99 per cent confidence limit, or, alternatively, ± .02 error range with 99.99 per cent confidence limit.

References

(1) International Business Machines Corp., "ASAP, An Automated Statistical Analysis Program", Tech. Rept. prepared for NASA Goddard Space Flight Center, Greenbelt, Md., Contract No. NAS 5-3373.

(2) J.V. Uspensky, "Introduction to Mathematical Probability", McGraw-Hill Book Co., 1937, p. 101.

(3) E. Parzen, "Modern Probability Theory and Its Applications", p. 239; Wiley and Sons, 1960.

(4) M. Abramowitz and I.A. Stegun, "Handbook of Mathematical Functions", Bational Bureau of Standards, Applied Mathematics Series AMS-55; U.S. Government Printing Office, Washington, D.C., p. 968.

Section VI. Generation of Random Numbers on IBM 1620 Computer*

*This is part of a thesis submitted by J.J. Perkowski in partial fulfillment of the requirements for the M.S. Degree to the Electrical Engineering Faculty of Villanova University, June, 1966.

CHAPTER I

INTRODUCTION

A group of n numbers are random if each number in the group has the same probability of occurring. An important property of random numbers is that knowing some of the numbers we cannot predict any other number in the sequence. In addition, the sequence of true random numbers whould not be limited to a finite length. Thus (1) total unpredictability, (2) equal likelihood of the outcomes and (3) infinite length of the sequence form the three basic properties of random numbers.

When the random digits are generated on a digital computer by means of some repetitive arithmatical process they are called pseudo-random digits. Pseudo is defined as deceptively resembling a specified thing, and the deception encountered here is that a pseudo-random process cannot generate an infinitely long random sequence. Eventually the process will either end up in a string of zeroes or will start repeating itself. Thus pseudo-random numbers violate the third property of random numbers.

Nevertheless pseudo-random numbers are best suited for computer applications as long as they pass predetermined statistical tests which will be used to test randomness in this paper.

Let us consider some of the methods available for generating pseudo-random numbers:

A. Von Neumann's Center Squaring Method 6, 12

Running through the actual procedure of this method gives a hint of what can be expected in these random processes. Proceed as follows:

1) Start with some large number a_o containing 2k digits; any number will do.

2) Square a_o to get a_o containing 4k digits.

3) Take the middle 2k digits of a_o and call this a_1, the next random number.

4) a_1 is then squared and the process continues.

The assumption in this method is that any digit is as likely to occur as any other so the numbers will be random. Let us see if this is true with some examples.

Example 1

1) Let $a_o = 1234$, number of digits = 2k = 4

2) $a_o = 01522756$, 4k = 8

3) The middle 4 digits are 5227 so $a_1 = 5227$

This seems perfectly legitimate but certain numbers do not work so well.

Example 2

Let $a_o = 64$	then	$a_o = 4096$	$a_1 = 09$	$a_1 = 0081$
$a_2 = 09$		$a_1 = 0081$	$a_2 = 08$	$a_2 = 0064$
$a_3 = 06$		$a_3 = 0036$	$a_4 = 03$	$a_4 = 0009$
$a_5 = 00$				

A. Von Neumann's Center Squaring Method [6, 12]

Running through the actual procedure of this method gives a hint of what can be expected in these random processes.

Proceed as follows:

1) Start with some large number a_o containing 2k digits; any number will do.
2) Square a_o to get a_o^2 containing 4k digits.
3) Take the middle 2k digits of a_o^2 and call this a_1, the next random number.
4) a_1 is then squared and the process continues.

The assumption in this method is that any digit is as likely to occur as any other so the numbers will be random. Let us see if this is true with some examples.

Example 1

1) Let $a_o = 1234$, number of digits = 2k = 4
2) $a_o^2 = 01522756$, 4k = 8
3) The middle 4 digits are 5227 so $a_1 = 5227$

This seems perfectly legitimate but certain numbers do not work so well.

Example 2

Let $a_o = 64$	then	$a_o^2 = 4096$	this gives
$a_1 = 09$	"	$a_1^2 = 0081$	" "
$a_2 = 08$	"	$a_2^2 = 0064$	" "
$a_3 = 06$	"	$a_3^2 = 0036$	" "
$a_4 = 03$	"	$a_4^2 = 0009$	" "
$a_5 = 00$			

This process degenerates into a string of zeroes for $a_0 = 64$ and for many other values. In addition this method often degenerates into short cycles of two or three numbers. Obviously this is not a good method and experience has shown unsatisfactory results if a_0 has less than eight digits. The National Bureau of Standards tried this method [6] and produced sixteen programs ranging in length from 11 to 104 numbers of four digits each with an average length of 52. This is not very ideal for practical applications.

B. Modified Von Neumann Method [6]

Considerable better results are obtained by a modified version of Von Neumann's method, in which a pair of numbers, a_0 and a_1, are multiplied together and the central digits of the product are used for the number a_2. The process is repeated for a_1 and a_2 to give a_3. So if $a_0 \times a_1 = (1234) \times (5678) = 07006652$ then $a_2 = 0066$. This type of process gives pseudo-random numbers that are more random and with a larger period than the mid-square method. In the tests run by NBS, ten sequences were computed, all of which degenerated into a string of zeroes. The lengths of the sequences ranged from 19 to 1253 with an average length of 591.

This method will be used later in the computer to generate data.

To summarize:

1) Select a_o and a_1; any 2k digit number will do.

2) Take the product of a_o and a_1; 4k digits.

3) Take the middle 2k digits of this product and call this a_2

4) Take the product of a_1 and a_2 to get a_3 etc.

C. IBM Method

This method was taken from the IBM reference manual [1] and will be used later on the computer. The basic formula for this process is:

$$u_{n+1} = \text{last } d \text{ digits of } xu_n \tag{1.1}$$

This will produce $5 \cdot 10^{d-2}$ terms before repeating (for d greater than 3). An outline of this method as dictated in the IBM reference manual follows:

1) Choose for a starting value any integer u_o not divisible by 2 or 5; u_o is d digits long.

2) Choose t for equation 1.2 as any integer

3) Choose r for equation 1.2 as any of the values 3, 11, 13, 19, 21, 27, 29, 37, 53, 59, 61, 67, 69, 77, 83, and 91.

4) Take the values from 2) and 3) and choose as a constant multiplier an integer x of the form:

$$x = 200t \pm r \tag{1.2}$$

(The plus-minus sign is used because x must be odd and odd numbers have the form $2n \pm 1$, $2n \pm 3$, etc.; the plus-minus sign simplifies selecting a value close to $10^{d/2}$ as a choice for x.)

5) Compute xu_o, a product 2d digits long

6) Discard the high order d digits leaving u_1 consisting of the last d digit of the product.

7) The process is repeated.

As an example let $d = 4$ and $u_0 = 2357$. Since $10^{d/2} = 100$ a good choice for x is 109. So $xu_0 = (0109)\ (2357) = 00256913$. Then $u_1 = 6913$, $xu_1 = (0109)\ (6913) = 00753517$ so $u_2 = 3517$. This method will be studied in much further detail in later discussions.

D. Lehmer Method

D. H. Lehmer is an important name in random numbers and a very simple method [7] which he developed calls for successive multiplications by a constant number (he chooses 23):

1) Choose an eight digit number u_0; any number will do.
2) Multiply u_0 by 23 to get a nine or ten digit u_0'.
3) The first and second digits on the left are removed and subtracted from what remains of u_0' giving u_1
4) Continue the process with $23u_1$

Example

1) $u_0 = 12345678$
2) $23u_0 = 0283950594$
3) $u_1 = 83950594-02 = 83950592$
4) $23u_1$ = etc.

This method supposedly does not repeat until 5,882,352 sequences have been computed which contains about 47 million random digits. And so this paper will contain a method similar to this that produces sequences of six digits each. The method was modified slightly to better fit the Fortran computer language.

E. Residue Method

In these four methods discussed so far, instructions state to choose any initial value for u_o or a_o etc. But when looking at the results of these methods, it will be seen that only certain initial values give good long programs; the others give short deteriorating programs. Just what the proper initial value is, though, can only be determined by trying many different values and selecting the best by observing the results. This, of course, entails a lot of guess work and a good bit of computer hours. And so a method is needed in which one does not have to pick a special initial value in order to get long sequences of usuable numbers. Such a method is the power residue method [8] which is extensively used today by anyone wishing to generate random numbers. The IBM manual spells out the procedure for this method. The method is based on the equation:

$$u_{n+1} = xu_n \pmod{10^d} \tag{1.3}$$

The procedure is:

1) 10^d represents the word size of the machine and this will produce $5 \cdot 10^{d-2}$ terms before repeating. So in order to have at least 5,000 terms let $d = 5$.
2) The value of x is arrived at from the congruence

$x \equiv \pm (3, 11, 13, 19, 21, 27, 29, 37, 53, 59, 61, 67, 69, 77, 83, 91) \pmod{200}$

3) Choose u_0 as any integer not divisible by 2 or 5.

4) Compute $xu_0 \pmod{10^d}$ using fixed point integer arithmetic

5) Continue process for u_1 etc.

Example

1) $d = 5$

2) $x = 3379$

3) $u_0 = 389$

4) $xu_0 \pmod{100{,}000}$ is simply this: $xu_0 = 1{,}314{,}431$

$\frac{xu_0}{100{,}000} = 13$ plus a remainder of 14431

It is this remainder that is u_1

$u_1 = 14431$ etc.

Later on in Chapter III when this method is discussed emphasizing computer techniques, very interesting manipulations must be made to adapt this program to the computer.

But before the computer programs are discussed, the statistical tests to be used must first be listed.

CHAPTER II

STATISTICAL TESTS

INTRODUCTION

Before beginning the observations of the computer programs, it is necessary to explain the tests that were performed on the random numbers. By studying these tests in great detail now, we eliminate the possibility of their interfering with the flow of thought from one program to the next in the following chapter.*

CHI-SQUARED TEST

The major problem that will be encountered when testing random numbers is which ones to keep as random and which ones to discard. The chi-squared (x^2) test of goodness of fit will be used to tell whether or not a set of numbers is satisfactory.

Whenever an experiment is performed (throwing dice for example), certain expected outcomes can be calculated using the formulas of probability theory. Then when the experiment is performed, the results may be compared with the theoretical calculations. Often these calculated values are put in the form of a probability distribution as in figure 2.1 where

* References for this chapter: see 9 to 11 in Bibliography

f(n) is the probability of the number n appearing on the dice. We will refer to this as a parent distribution since it is the norm which we are trying to match in the experiment.

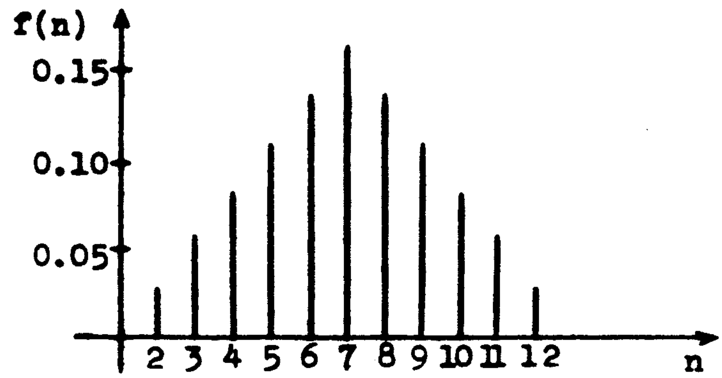


fig. 2.1

The chi-squared test is used to tell just how much disagreement between the parent distribution and the experimental values (call this the sample distribution) can be reasonably expected or in other words how great the disagreement must be in order to justify that the dice do not obey the parent distribution.

These distributions are expressed most naturally as frequencies of events where the frequency of an event is the total number of times this event occurs among all the trials. Let f_o be the frequency of occurrence of event n for a sample that will consist of N trials. If the parent distribution is f(n) then the frequency predicted by the parent distribution is Nf(n) written as f_c. These frequencies are related in the following way to get the chi-squared goodness to fit:

$$x^2 = \frac{\sum^{n}(f_o - f_c)^2}{f_c} \qquad (2.1)$$

For a sample of n events, n-1 events are independent leaving one dependent event. As an example, suppose we are running a test of the frequency of each digit, zero to nine, in a sample. If there are 1000 digits in the sample and there are 910 digits from one to nine, then the total number of zeroes is already determined and is dependent on the other values. So we say that this sample has nine degrees of freedom (v) or independent digits. In general $v = n-1$.

How are these results then interpreted? Clearly if the observed and calculated values agree exactly then $x^2 = 0$. The greater the difference between the sample and parent distribution, the greater will be the value of x^2 so generally speaking the larger x^2, the worse the fit. The x curve is plotted as follows:

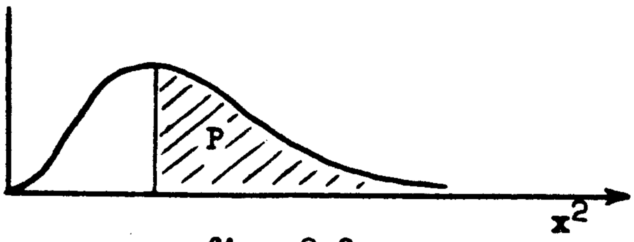


fig. 2.2

Chi-squared tables are found in most statistics books. So as an example, if the number of degrees of freedom is 10 and x^2 is calculated as 3.94 then the tables say that the probability that $x^2 \geq 3.94$ is 0.95. That is the probability of obtaining by chance a value of x^2 at least as bad as the observed fit is 0.95. So 95 times out of a hundred a worse fit will occur so we deduce that $x^2 = 3.94$ is a good fit. But suppose we calculated $x^2 = 23.2$ for 10 degrees of freedom. The table gives $P = 0.01$, so only one time out of a hundred will we get a worse fit; 99 out of 100 times a better fit occurs so we easily see that $x^2 = 23.2$ for 10 d.f. is not a good fit.

In most of our measurements we will use the 10 per cent points as our confidence limits:

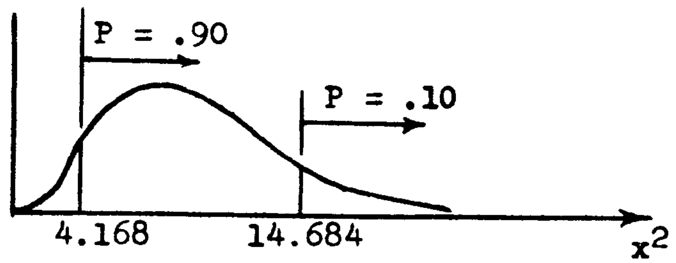


fig. 2.3

So for 9 degrees of freedom, we will generally only accept values of x^2 that fall in the range 4.168 to 14.685. These are very tight limits. If we wish to get more lax, we will reduce the limits to the 5 per cent points.

As a short example, take the count of the odd number digits of a group of 500 random digits. Using the decimal system the probability of each digit is one-tenth. So the expected frequency (f_c) of each is Nf(n) or (500)(1/10) = 50 digits. This set of random numbers contains 40 ones, 43 threes, 47 fives, 54 sevens, and 59 nines. Calculate x^2 to see if these numbers are random.

The following table is set up:

Table 2

n	f_o	f_c	$f_o - f_c$	$(f_o - f_c)^2$
1	50	40	10	100
3	50	43	7	49
5	50	47	3	9
7	50	54	−4	16
9	50	59	−9	81
				255

$$x^2 = \sum \frac{(f_o - f_c)^2}{f_o} = \frac{255}{50} = 5.1 \quad \text{from equation 2.1}$$

$x = 5.1$ for 4 d.f.* $\quad P \cong 0.27$

This is within the 10 per cent confidence limits so this is a good set of random numbers.

STANDARD DEVIATION

The standard deviation will be used in conjunction with the mean or average to gain certain knowledge about the random digits.

It is defined as follows:

$$\sigma = \sqrt{\frac{1}{n}\sum_{n}(f_o - f_c)^2}$$

where: σ = standard deviation

n = number of trials, etc.

In general the probability for a measurement to occur in an interval within $T\sigma$ of the median is

* d.f. = degrees of freedom

$$P(T) = \frac{1}{\sqrt{2\pi}} \int_{-T}^{T} e^{-\frac{t^2}{2}} dt$$

The probability (see fig. 2.4) for a few values of T is:

$P(1) = 0.683$	$1-P(1) = 0.317$
$P(2) = 0.954$	$1-P(2) = 0.046$
$P(3) = 0.997$	$1-P(3) = 0.003$

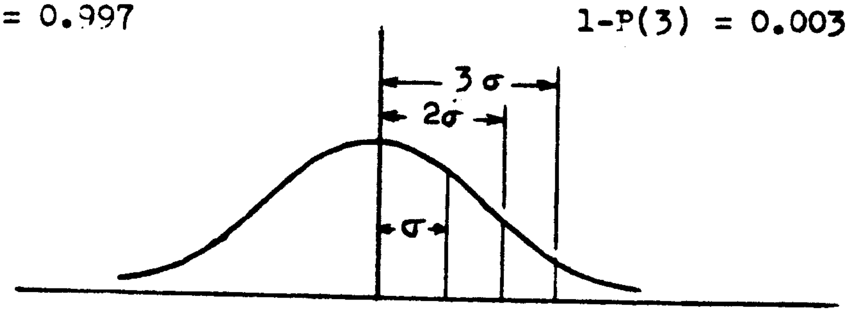


fig. 2.4

This means that the probability for a measurement to fall within one standard deviation of the mean is about 68 per cent, the probability of being farther away than 2σ is 4.6 per cent and farther away than 3σ is 0.3 per cent. So normally we should expect about 30 per cent of the data to fall outside the first standard deviation.

As an example, let us again take the odd numbered digits. The last column of Table II is also the $(f_o - f_c)^2$ term in the formula for standard deviation (equation 2.2). So then:

$$\sigma = \sqrt{\frac{1}{n}\sum(f_o - f_c)^2} = \sqrt{\frac{1}{5}(255)} = \sqrt{51}$$

$$\sigma = 7.14$$

This gives the following results:

	Range	# Readings Within Range
σ	42.86 to 57.14	3
2σ	35.72 to 64.28	5
3σ	28.58 to 71.42	5

These results are very favorable. Three-fifths or 60 per cent fall within one σ compared with 68 per cent theoretically, and none fall further than 2σ away.

FREQUENCY TEST

This test is basically the comparison of the frequency of occurrence of each digit 0 to 9 with the expected value of the digit, i.e. one-tenth the number of digits in the group. Chi-squared test and standard deviation are used to see how close the digits are to the expected.

Remember where this expected or parent distribution comes from. We have mentioned that one of the properties of random numbers is that each digit is equally probable and even though we are generating pseudo-random numbers, this property still holds true. So we are justified in saying that the expected value for the frequency of occurrence of a digit is 1/10 the number of digits in a decimal system.

Variations of the frequency test would be running tests on every other digit, every third digit,......., every tenth digit, and also the frequency of odd digits to even digits is often compared as well as frequency of numbers below the mean (0, 1, 2, 3, 4) to numbers above the mean (5, 6, 7, 8, 9).

SERIAL TESTS

This test involves counting the frequencies of all pairs of numbers (00-99) and comparing them with the normal using x^2 or σ. This gives a good indication of whether certain digits tend to follow certain other digits, i.e. a given digit being dependent on the digit preceding it.

RUNS TESTS

Three different types of runs tests will be performed:

1) Run test above and below the median
2) Run test of individual digits
3) Run test up and down

(1) The run test above and below the median consists of dividing the numbers letting 0, 1, 2, 3, 4 equal a and 5, 6, 7, 8, 9 equal b. So a series of digits 2728910447 would give ababbaaaab, which contains four runs of one, a run of two b's, and a run of four a's. The total number of runs and runs of one, two, etc. are then compared with expected values which are calculated as follows:

$$\text{expected total number of runs} = \frac{N+1}{2} \qquad (2.4)$$

$$\text{expected number or runs of length } k = (N - k+3)2^{-k-1} \qquad (2.5)$$

where N = number of digits being tested

Confidence limits for expected total number of runs are found from table 47, page 203 in [9].

A small sample of this table follows:

Table 3

number of runs expected (m)	90 per cent limits lower limits	upper limits
100	88	114
200	178	224
300	268	334
400	358	444
500	448	554

N.B. For $m>10$, the number of runs is approximately normally distributed with mean $m+1$ and variance (σ^2) equal to $m(m-1)/(2m-1)$.

(2) The run test up and down consists in determining if the differences between successive digits is positive or negative. So for N points ($u_1, u_2, \ldots\ldots, u_n$) we write a binary sequence whose nth term is "u" if $u_n<u_{n+1}$ and is "d" if $u_n \geq u_{n+1}$.

So again for the sequence 2728910447 we get uduuddu-u. Letting the dash be a "u" this contains two runs of one, two runs of two, and one run of three. The results are, of course, then compared by x^2 with expected values that are calculated as follows:

$$\text{expected total number of runs} = \frac{(2N-1)}{3} \qquad (2.6)$$

$$\text{expected number of runs of length 1} = \frac{5N+1}{12} \qquad (2.7)$$

$$\text{expected number of runs of length 2} = \frac{11N-14}{60} \qquad (2.8)$$

$$\text{expected number of runs of length } k = \frac{2\,(k^2+3k+1)N-(k^3+3k^2-k-4)}{(k+3)!} \qquad (2.9)$$

where N = number of digits being tested.

As noted in the example above, often a dash will occur in the case where $u_n = u_{n+1}$. A good way to overcome this is to take the u's and d's from the start of the sequence and use them in the place of each dash that turns up.

CHAPTER III

ANALYSIS OF PROGRAMS GENERATED ON THE COMPUTER

INTRODUCTION

The background of random numbers noted and the tests to be used understood, the discussion of the random numbers that I have generated on the computer can begin.

These programs will begin at the simplest level and proceed toward the complex, but useful, methods. Each method will generally be an improvement over the one preceding it, and these improvements will be emphasized a good deal. Consideration will also be given to variation of inputs and the effects on the results. Chapter I discussed these methods purely from the mathematical viewpoint, the theoretical side, but this chapter considers the problems of getting the programs to work on a computer. So computer techniques will be emphasized but will be tied in closely with the discussions of Chapter I.

As an aid to understanding this chapter, the actual Fortran language computer programs can be found in Appendix A while most of the actual numbers generated will be found in Appendix B.

METHOD I - IBM METHOD

This method has been discussed previously in Chapter I taken from the IBM reference manual. Repeating the general formula for the method we get:

$$u_{n+1} = \text{last } d \text{ digits of } xu_n \tag{3.1}$$

The selection of initial values (u_o), the input values for the computer is the most difficult task for this method.

The constant d was first selected ($d = 4$) so the number of terms before repeating is $5 \cdot 10^{d-2}$ which gives 500 terms.

The multiplier u_o is chosen as any number not divisible by 2 or 5. Let $u_o = 2357$.

The x is then chosen by the formula $x = 200t \pm r$ where t is any integer and r is any of the values listed in Chapter I which gives a value of x close to $10^{d/2}$ (100 in this case). Then t is chosen as one and r as 91 then

$$x = (200)\ (1) - (91)$$

$$x = 109$$

So the initial values are in summary:

$$d = 4$$

$$u_o = 2357$$

$$x = 109$$

Computing xu_o these values will produce a product 8 digits long; but the high order 4 digits are discarded and the 4 low order digits are the value of u_1. (See Appendix A.)

The problem remains of programming this on the computer.

The program was written entirely in fixed-point mode. To show the effect of fixed point, suppose a certain product is <u>76</u>7215.<u>72</u>. Operation in this mode will discard the underlined digits leaving only +7215. So fixed-point mode rejects all decimals and digits to the left of the four low order digits. When the 8 digit product xu_0 is calculated, only the four low order digits are printed. This is exactly the u_1 that is required. This program consisted of numbers of four digits in length and contained 500 terms before repeating (see Appendix B). But looking at columns of numbers, it is noticed that the period of each column is not 500. The period for each column is

units column	T = 2
tens column	T = 10
hundreds column	T = 50
thousands column	T = 500

So the low order digits of the numbers are far from random. The periodicity of the digits increases as the order of the digit position increases.

The units column consists simply of the alternating digits 3 and 7. This column can be discarded as not random.

The tens column is composed of the 10 digit series 1157933975. Each digit appears twice, but they are only odd numbered digits. No even digits occur in this column so it certainly is not random.

The hundreds column contains 50 digits before repeating.

Twenty are even and thirty are odd. The probability is only 16 per cent that there could be 10 more odd digits than even digits at these values,

$$(x_2 = \frac{5^2+5^2}{25} = 2 \quad P = 0.16 \text{ for 1 d.f.})$$

so the hundreds column is rejected.

The thousands column consists of 500 digits distributed as follows:

51 zeroes	50 threes	50 sixes
50 ones	50 fours	50 sevens
50 twos	49 fives	50 eights
		50 nines

Calculating x^2 gives:

$$x^2 = \frac{1 + 8(0) + 1}{50} = 0.04 \text{ for 10 degrees of freedom.}$$

From the x^2 table a $x^2 = 0.04$ gives a probability P = 0.999999.... for 9 d.f. This means that only one chance in 10,000..... will give a better fit. So it seems logical that this is a good set of random numbers. But statisticians caution about numbers that are too close to the norm. When numbers get too close to what is expected, they cease to be random. Hence we have mentioned before that limits of x^2 for acceptable results are the range 4.168 to 14.684. This lower limit is chosen to avoid these numbers that follow the norm too closely and are as a result not random. For this reason the numbers in the thousands column must be rejected.

The entire method I is rejected then for the various reasons cited.

METHOD II - IMPROVED IBM METHOD

It was the purpose of this method to attempt to make improvements on Method I so that every column in Method II would be random instead of just one column.

In Method I it was the last digit which was least random so in this method the last digit is eliminated. After the product xu_0 has been computed and the first four digits are dropped, the remaining digits (formerly u_1) are now divided by the constant 10. So if u_1 was equal to 4487, dividing by 10 gives 448.7. But in the computer language (Fortran language) this number is in fixed-point mode so only the digits 448 are retained as the new u_1.

Two statements are taken from Appendix A to show the difference in computer language

7 I(J) = N*K............Method I

7 I(J) = N*K/N..........Method II

where

7 = statement number

I(J) = u(n+1)

N = X

K = u_n

Let us now see if this improvement has helped generate numbers that are more random. Ninety-seven numbers of three digits each were generated before they started repeating. (See Appendix B.) Already an improvement can be seen. The hundreds column of Method I had a period of 50 while in this program this column has a period of 97. The other columns also have the same period, and hence it is increased many times over Method I.

a) Frequency Tests: There are 291 digits so there should be statistically speaking 29.1 of each digit. The frequency test on these digits gave the following table which is similar to table 2 in Chapter II:

Table 4

n	f_c	f_o	$f_o - f_c$	$(f_o - f_c)^2$
0	29.1	29	0.1	0.01
1	29.1	27	2.1	4.40
2	29.1	31	1.9	3.60
3	29.1	23	6.1	37.30
4	29.1	29	0.1	0.01
5	29.1	32	2.9	8.40
6	29.1	34	4.9	24.00
7	29.1	27	2.1	4.40
8	29.1	30	0.9	0.81
9	29.1	29	0.1	0.01
	291.0	291.		82.94

$x^2 = \frac{82.94}{29.1} = 2.84$ for 9 d.f. P = .965

remembering that

u = digit being tested

f_c = expected number of each digit = $Nf(n)$

f_o = observed number of each digit = $F(n)$

At first glance these do not seem to agree with the present confidence limits so let us look at this with odd and even numbers separated. There are $\left\{\begin{matrix} \text{145 odd digits} \\ \text{138 even digits} \end{matrix}\right\}$

and x^2 for this information gives

$$x^2 = \frac{(3.5)^2 + (3.5)^2}{41.5} = 0.173 \text{ for 1 d.f.}$$

which gives P = 0.65. This means the probability of having 7 more odd numbers than even in this particular case is 0.65. This is a good result. Also x^2 for odd number digits is 1.87 or P = 0.75 for 4 d.f. and for even digits 0.977 or P = 0.91 for 4 d.f. These deviations do not appear significant for rejection.

The standard deviation (σ) of this set is

$$\sigma = \sqrt{\Sigma\frac{(f_o - f_c)^2}{N}} = \sqrt{\frac{82.94}{10}} = \sqrt{8.3} = 2.88$$

$m = 29.1$

So for each standard deviation:

	Range	Observed	Expected
$m \pm \sigma$	31.98 26.12	7	6.8
$m \pm 2\sigma$	24.86 23.24	8	9.5

	Range	Observed	Expected
$m \pm 3\sigma$	37.74 20.36	10	9.9

There are only two readings past 3σ. All the others fall within range.

b) Runs Tests: A run test above and below the mean was performed with the following results:

number of runs counted------------------155

number of runs expected-----------------146

range permitted as 90 per cent limits-------------------------134 - 160

These results were good.

A run test up and down was also performed. There were 194 runs expected and 206 observed. For 90 per cent limits the range allowed is from 173 to 217. The observed value falls within this limit.

According to these tests there is little evidence of any divergence from the normal expectations. Only in the frequency test of these numbers is the result questionable. So we can conclude that these numbers are random, but there is one glaring fault with these random numbers. There is not enough of them. There are only 97 terms in the series; far from enough to apply this method to a Monte Carlo method.

METHOD III - CENTER SQUARING METHOD

So far the methods that have been investigated have consisted of multiplying various numbers with a definite constant over and over. A better way for generation would be to have two new multipliers for each number generated.

This method (Von Neumann's Center Squaring Method) has been discussed in great detail in Chapter I. Short cycles have been obtained by some people that have used this method.

Three sets of random numbers were generated on an adding machine using three different initial values of a_o:

$a_o = 1111$	gave	54	terms
$a_o = 1234$	"	82	"
$a_o = 6043$	"	66	"

These give an average period of 67 numbers of four digits each. (See Appendix B).

For $a_o = 1234$ (82 terms) the frequency test gives:

0 - 9	3 - 10	6 - 6
1 - 13	4 - 9	7 - 5
2 - 13	5 - 8	8 - 8
		9 - 3

This has $x^2 = 11.75$ for 9 d.f. or a P = 0.23 which is good. But notice the digits divided in this manner:

$$\left.\begin{array}{l}(1, 2, 3, 4, 5) = 53 \text{ digits} \\ (6, 7, 8, 9, 0) = 31 \quad " \end{array}\right\} \text{av.} = 42$$

The probability of this occurring is calculated:

$$x^2 = \frac{11^2 + 11^2}{42} = \frac{242}{42} = 5.76 \text{ for 1 d.f.}$$

$P = 0.018$

There is only about one chance in 50 of this occurring so this series is definitely biased toward the lower five digits.

The mid-square method is then out of consideration due to its short period and bias to certain digits.

METHOD IV - MODIFIED VON NEUMANN

Center squaring does not work satisfactorily so logically Method IV will be tried.

In programming this method a_o and a_1 were multiplied together giving an eight digit number (C = 07006652). C is then divided by a factor D = 0.01 giving the product 070066.52. But this product is printed out in fixed-point mode so only the digits 0066 are printed; this is called a_2 or in Fortran language, I(2). (See Appendix A.)

Two different inputs picked at random were fed into the computer. They were as follows (with length of period included):

Input		Period
$a_o = 1111$	$a_1 = 1111$	$T = 61$
$a_o = 1234$	$a_1 = 5678$	$T = 1137$

Both sequences ended in a string of zeroes. The period for our runs averages out to T = 599 where the NBS tests gave T = 591 for ten sequences.

It was virtually impossible to run any tests on the program resulting from the first input. However, some indication is given that this might be a good method by looking at the frequencies of the digits:

16 zeroes	23 threes	18 sixes
20 ones	21 fours	14 sevens
16 twos	17 fives	16 eights
		18 nines

This gives an $x^2 = 3.72$ for 9 d.f. or $P = 0.92$. So nine times out of ten a worse fit will occur.

The frequency of digits for the second input were as follows:

212 zeroes	174 threes	213 sixes
216 ones	192 fours	191 sevens
203 twos	205 fives	206 eights
		188 nines

These were from a test of the first 250 numbers of four digits each. So for two thousand digits we expect two hundred of each number. Table 5 contains the frequency test.

These two tests show very good results concerning the randomness of these numbers. The probabilities for the frequency and odd versus even test were well within the confidence limits which we set. (See Table 5.)

Table 5

n	f_c	f_o	$f_o - f_c$	$(f_o - f_c)^2$
0	200	212	12	144
1	200	216	16	256
2	200	203	3	9
3	200	174	-26	676
4	200	192	- 8	64
5	200	205	5	25
6	200	213	13	169
7	200	191	- 9	81
8	200	206	6	36
9	200	188	-12	144
				1604

$$x^2 = \frac{1604}{200} = 8.02$$

P = 0.52 for 9 d.f.

For odd versus even we get

n	f_o	f_c	$f_o - f_c$	$(f_o - f_c)^2$
odd	974	1000	-26	676
even	1026	1000	+26	676

$$x^2 = \frac{1352}{1000} = 1.352 \text{ for 1 d.f.}$$

$P \cong 0.25$

A runs test above and below the median was taken with the following results:

Table 6

Length of Run	Observed	Expected
1	479	500.5
2	268	250.1
3	135	125.0
4	59	62.4
5	31	31.3
6	13	15.7
7	10	7.8
8	4	3.9
9	1	2.0
Total	1000	998.7

The results of this runs test are very good; and along with the frequency test, these give very good indication that the numbers generated in this method are random.

METHOD V - LEHMER'S METHOD

This is the method devised by D. H. Lehmer as was discussed in Chapter I, Section D.

Lehmer's formula is summarized:

$$u_n = 8\text{RHDO}\left[23u_{n-1} - 2\text{LHDO}\,(23\ u_{n-1})\right] \tag{3.2}$$

where RHDO = right hand digits of

LHDO = left hand digits of

In terms of congruences this is written (according to Lehmer) as

$$u_n = u_o 23^n (\text{mod } 10^8 + 1) \tag{3.3}$$

which gives 5,882, 352 eight digit numbers.

The IBM 1620 was better adapted to produce a six digit number as a result. So the initial value u_n is an eight digit number, but this formula is used:

$$u_n = 6\text{RHDO}\left[23u_{n-1} - 2\text{LHDO}(23u_{n-1})\right] \qquad (3.4)$$

as opposed to equation (3.3)

In the actual generation of the numbers(see Appendix A and B) certain problems arose with exponents exceeding the computers E$\pm$99 limit. So three IF statements were used in the program to limit these exponents. This particular program prints out 801 six digit numbers; 4806 random digits total for each input applied. The program can be continued by using its last number as the input to the continued program.

Testing will now begin to determine whether these numbers are acceptable for use.

FREQUENCY TESTS

With 4806 random digits we expect 480.6 of each digit. The results of counting were:

511 zeroes	470 threes	459 sixes
490 ones	505 fours	473 sevens
475 twos	450 fives	483 eights
		490 nines

To calculate x^2, a chi-squared table is set up.

Table 7

n	f_o	f_c	$(f_o - f_c)$	$(f_o - f_c)^2$
0	511	480.6	30.4	924.16
1	490	480.6	9.4	88.36
2	475	480.6	- 5.6	31.36
3	470	480.6	- 10.6	112.36
4	505	480.6	24.4	595.36
5	450	480.6	- 30.6	936.36
6	459	480.6	- 21.6	466.56
7	473	480.6	- 7.6	57.76
8	483	480.6	2.4	5.76
9	490	480.6	9.4	88.36
				3306.40

$$x^2 = \frac{\Sigma(f_o - f_c)^2}{f_c} = \frac{3306.4}{480.6} = 6.879 \text{ for 9 d.f.}$$

This gives P = 0.65

For odd digits

$$x^2 = \frac{1283.2}{480.6} = 2.67 \text{ for 4 d.f. } (P = 0.60)$$

For even digits

$$x^2 = \frac{2023.2}{480.6} = 4.20 \text{ for 4 d.f. } (P = 0.38)$$

So the x^2 value for 9 degrees of freedom is 6.879 and the probability of exceeding this value is approximately 0.65.

The total number of even digits is 2433 as against 2373 odd digits. Assuming that an even digit is as likely to occur as an odd, the probability of a departure from normal as high as this (2433-2373) is approximately 0.40; in other words, a difference greater than this might occur two times in five so the deviation does not appear significant. Calculation of this follows:

n	f_o	f_c	$f_o - f_c$	$(f_o - f_c)^2$
odd	2373	2403	-30	900
even	2433	2403	30	900

$$x^2 = \frac{1800}{2403} = 0.749 \text{ for 1 d.f.} \quad (P \cong 0.40)$$

With all these calculations considered, we conclude that there is no indication of any discrepancy in the behavior of odd versus even digits.

An inspection of the frequencies of occurrence shows that the digit that appeared most frequently (zero) was associated with a probability of 0.106 (P = 511/4806) and the least frequent digit (five) had a probability of 0.093 (P = 450/4806).

The standard duration of the frequencies is

$$\sigma = \sqrt{\frac{1}{N}\Sigma(f_o - f_c)^2} = \sqrt{\frac{3306.4}{10}} = \sqrt{330.64} = 18.2$$

Since the mean is 480.6, the range of values for one standard deviation is 462 to 499. Six of the ten values of n fall in this range. This compares favorably with the 68 per cent expected.

These frequency tests give no indication of any abnormalcy compared with normal distribution.

SERIAL TEST

The frequency of the occurrence of all possible pairs of digits is given in Table 8. It was formed by entering the pair of digits ij into the ith row and the jth column.

Table 8

Serial Test

Frequency of 1st Digit ↓

Frequency of 2nd Digit →

	0	1	2	3	4	5	6	7	8	9	Total
0	52	51	55	51	64	42	47	51	51	47	511
1	55	47	60	52	44	52	39	48	39	54	490
2	56	49	37	43	53	37	51	51	60	37	474
3	51	51	41	41	60	41	41	44	41	58	469
4	50	54	44	56	41	55	41	54	53	51	505
5	51	49	44	47	46	38	38	43	52	41	448
6	43	49	46	47	44	46	54	47	40	43	458
7	46	54	53	55	49	41	43	47	36	48	472
8	63	41	46	45	49	49	48	50	46	46	483
9	44	45	48	32	55	48	51	37	65	63	490
Tot	511	490	474	469	505	448	458	472	483	490	4800

Only 4800 of the 4806 digits were used in this test; the last 6 pairs were ignored.

This test is performed to show that the table is a random sample from a sequence in which one pair of digits is as likely to occur as another.

The chi-squared test compares the frequency (f_o) in each position of the table with the expected frequency ($f_c = 48$). The number of degrees of freedom, v, is 90 due to the constraint that totals of corresponding rows and columns are the same.

We find:

$$x^2 = \frac{4161}{48} = 86.69 \text{ for } 90 \text{ d.f.}$$

$P \approx 0.40$

This is within our confidence limits so by the serial test this sequence of numbers seems to be random.

RUNS TESTS

This will be the most severe test performed on the numbers. Having passed the frequency and serial tests, these digits will certainly be random if they can get by the runs tests. A set of non-pseudo random numbers may get past one or two tests but will certainly not get past all three tests.

Three runs tests were performed: runs test above and below the median, runs test up and down, and runs test of individual numbers as are explained in Chapter II.

(1) Above and below the median:- All the generated numbers were used in this test and 5, 6, 7, 8, and 9 were considered above the median while 0, 1, 2, 3, and 4 were below the median. Table 9 shows the results of this test.

Table 9

Length of Runs	Expected	Observed	% Error
1	1202.00	1143	- 4.9%
2	600.90	600	- .1%
3	300.40	267	- 11.1%
4	150.20	158	+ 5.2%
5	75.06	72	- 4.1%
6	37.50	32	- 14.7%
7	18.76	19	+ 1.3%
8	9.39	7	- 25.5%
9	4.68	6	+ 28.2%
10	2.34	3	+ 28.6%
11	1.17	2	+ 71.0%
12	0.59	2	+230.0%
	2401.82	2311	- 3.8%

Range for the Total (90% limits): 2258-2534

The expected values were calculated using the formulas given in Chapter II. Per cent error was calculated for each length of run. For 4806 digits, 2402 runs are expected but with a 90% leeway allow, the expected range is 2258 to 2534. The Lehmer method produced 2311 runs which is within the 90 per cent confidence limits.

The observed number of runs for the smaller lengths (1 to 6) fall below the expected amount on the average while lengths seven to twelve fall above the expected amount. But the excess of higher order runs does not effect the total picture too much since there are only thirteen runs from lengths nine to twelve anyway. The difference between observed and expected runs of lengths one and three (59 and 33 digits respectively or 92 digits as compared with the 91 digit differential of the total) actually cause the deficiency, for the most part, in the total number of runs observed with respect to the number expected; but as was shown, this total is well within the 90% limits. This test then gives no indication of these numbers not being pseudo-random.

(2) Up and down:- This test was performed on 914 of the 4806 digits. Again expected values were calculated for Chapter II expressions, and the results given in Table 10.

Table 10

Length of Runs	Expected	Observed
1	382.40	365
2	168.10	158
3	48.20	45
4	10.50	13
5	1.87	2
Totals	611.07	583

Note that 611 runs are expected and 583 are observed. The observed total is 4.6% in error of the calculated value which is a rather good result. The counts for individual lengths of runs also give no indication of the level of these numbers varying too slowly or too quickly.

(3) Individual numbers:- This test was performed on all

the numbers. The expected values were estimated from a test of similar nature by the Rand Corporation [11]. The results follow:

Table 11

Length of Run	Expected	Observed
1	3898.0	3958
2	389.8	369
3	38.9	39
4	3.9	3
5	0.4	0
Totals	4331.0	4369

The total number of runs counted are off from the expected value by only 0.88%. This is an excellent result and more or less confirms the decision that these numbers are pseudo-random.

CONCLUSION.

Ample evidence for the pseudo-randomness has thus been given. The first property of total unpredictability has been upheld by the serial test and runs tests. The serial test showed that no two digits depended on each other overall while the runs tests proved that the digits were not dependent on their preceding or following digit. They were, in fact, unpredictable. Secondly, it was shown that the digits were equally probable by the results of the frequency test on the ten different digits involved.

We thus conclude that none of these tests contradict the assumption that the numbers generated by the Lehmer method are pseudo-random.

METHOD VI - RESIDUE METHOD

This is the method recommended very highly by the IBM computer manual. It entails obtaining products using the power residue method. This can be adapted to the computer using, once again, fixed-point mode of operation.

Repeating equation 1.3 we get

$$u_{n+1} \equiv xu_n(\text{mod } 10^d) \tag{3.5}$$

This process is separated into three distinct steps:

1) multiplying x by u_n
2) obtaining the residue of modulus 10^d is done by dividing xu_n by 10^d, dropping off the decimals of this result and multiplying this whole number by 10^d.
3) now take the result of 2) and subtract it from the result of 1). This gives u_{n+1}.

Example

1) $xu_n = 1{,}314{,}431$

2) $\frac{xu_n}{10^d} = \frac{1{,}314{,}431}{10{,}000} = 13.14431$

dropping decimals gives 13

13 times 10^d gives 1,300,000

3) subtract: 1,314,431 - 1,300,000 = 14,431

$u_{n+1} = 14{,}431$

Let us now put these series of events into a simple equation which requires only the basic arithmetic operations (addition, subtraction, multiplication, and division).

$$u_{n+1} = \underbrace{\frac{xu_n}{1}}_{1} - \underbrace{\frac{xu_n}{10^d}}_{2} 10^d \qquad (3.6)$$

where operation 2 is that special division which ignores remainders (drops off decimals).

This equation can now be applied very nicely to the computer as follows:

1) let Y = X•U(N) (same as operation 1 in equation 3.6)

2) let J = Y/P where P = 10^d and J is a fixed-point variable. Fixed point is ideal for operation 2 because it drops off decimals and retains only whole numbers.

3) let Z = J thus putting this whole number into floating-point mode so it matches up with other variables in the equation.

4) the final computer equation is

   U(N+1) = Y-Z*P (3.7)

   (See Appendix A.)

Equation 1.3, 3.6, and 3.7 then are all the same, but the last two grew from the need of the simple operations that are required on the computer.

The numbers generated by this method (see Appendix B) were tested for randomness by the usual statistical tests.

The list consisted of five digit numbers with only zeroes appearing in the units column. The periods for each power of ten was as follows:

units	T = 1
tens	T = 2
hundreds	T = 10
thousands	T = 50
ten thousands	T = 500

The low order digits then are far from random and will be excluded from the analysis. Looking then at the frequency of the digits in the high order column we get:

50 zeroes	50 threes	50 sixes
51 ones	50 fours	50 sevens
49 twos	50 fives	50 eights
		50 nines

Statistically speaking this results in a

$x^2 = 0.04$ for 9 d.f.

$P >> 0.99$.

This is what is called a fit that is "too good" and usually a sample giving these results is discarded. This sequence is, then, of no use but the reason for this is that in our program d (the word length) was equal to 4. In order for true randomness to occur, the IBM manual states cases where d, being equal to 10 and 35, gives excellent results. The IBM 704, 709, and 7090 with a 35-bit word length makes it possible to generate a sequence of over 8.5 billion numbers. The ten-digit word length of the 650 and 7070 allows for a sequence of 500 million terms.

So using these other computers, this becomes probably the best method available today for generating random numbers. But the numbers produced by the 1620 must be discarded.

CONCLUSION

As a result of these tests then, it is rather apparent that this sample distribution of numbers generated by the residue method is inadequate. The most evident failing is that the length of the period of these numbers is too short ($T = 500$) for use in any large scale Monte Carlo problems. The reason for this is that the IBM 1620 limits us to a five digit output using fixed-point arithmetic on this computer. This can be overcome on the other computers recommended that have a longer word length. Using a $d = 8$ or higher on another machine will give a much longer cycle.........$T = 5 \cdot 10^{d-2}$ so the period will be five million terms or higher; surely enough for anyone's desires.

Taking these things into consideration along with the results of all six methods and the type of computer that was available, it has been decided that method five (D. H. Lehmer's method that was modified to fit the IBM 1620) is the one which best fits the properties of a pseudo-random number. Observe the following: The Improved IBM method passed all the statistical tests (but so did the Lehmer method), the Modified Von Neumann method had a long cycle (but so did the Lehmer method) and Lehmer's is a quick source of numbers obtainable directly from a computer.

From this statement, it is clear that only the Lehmer method is 1) truely pseudo-random, 2) is of long cycle, and 3) is easily obtainable from the computer that was made available to us.

This method then will be used in Chapter IV in the Monte Carlo application. Good approximations from the Monte Carlo problem will be further evidence that the Lehmer method is a good source for pseudo-random numbers.

This next method is considered merely from a curiosity point of view to see just how good or bad the numbers from a roulette wheel really are with respect to randomness.

METHOD VII - ROULETTE METHOD

Leaving now the arithmetic processes behind, we turn to a physical process which is manually controlled; that is, the spinning of a roulette wheel. Though this process cannot be seriously considered as a prime source of random digits (the method is much too slow); nevertheless, it will be interesting to see how this physical process compares with the fast arithmetic processes for randomness.

A small roulette wheel was used for this experiment and the following procedure was used:

IF THESE NUMBERS CAME UP ON THE ROULETTE WHEEL		THESE DIGITS WERE USED AS RANDOM NUMBERS
0 to 9 10 to 19 11 to 29	....	the last digits only were used so 17 became 7 and 30 became 0 etc.
30 to 36 double zero	....	these numbers were not used since they would have unbalanced the system

Using this method 2,200 digits were produced. The results of the tests follow.

FREQUENCY TEST:

The frequency test produced the following data:

Table 12

n	f_o	f_c	$f_o - f_c$	$(f_o - f_c)^2$
0	205	220	−15	225
1	184	220	−36	1296
2	212	220	− 8	64
3	208	220	−12	144
4	225	220	5	25
5	194	220	−26	676
6	265	220	45	2025
7	187	220	−33	1089
8	262	220	42	1764
9	258	220	38	1444
	2200	2200		8752

$x^2 = \frac{8752}{220} = 39.8$ for 9 d.f. $(P <<< 0.01)$

Needless to say this is not within the 90% confidence limits for 9 d.f. (4.168 to 14.68). There is less than one chance in a hundred that there will be a worse fit than this, which is pretty bad.

For some reason there were too many sixes, eights, and nines. Their probability of occurrence was 0.12, 0.119, and 0.117 respectively compared with the expected 0.1. The number which showed up least was one with a probability of 0.0836, compared with 0.1. This great deviation is not typical for a good set of random numbers.

In comparing odd with even digits x^2 becomes:

n	f_o	f_c	$f_o - f_c$	$(f_o - f_c)^2$
odd	1031	1100	69	4761
even	1169	1100	69	4761

$x^2 = \frac{9522}{1100} = 8.65$ for 1 d.f. ($P \ll 0.01$)

Another very poor fit and once again there is less than one chance in a hundred of a worse fit.

Apparently there is some unevenness in the physical structure of the wheel because the conditions effecting the experiment were maintained at a constant level.

RUNS TEST UP AND DOWN:

Table 13 gives the results that were found in the runs test up and down. These results are pretty good so these numbers were distributed fairly well about the median.

Nevertheless these numbers obtained on the roulette wheel must be declared non-random on the basis that the frequency test showed uneven distribution among the digits that are theoretically supposed to be equally likely.

Table 13

Length of Runs	Observed	Expected
1	533	550.5
2	285	275.1
3	128	137.5
4	72	68.7
5	27	34.4
6	23	17.3
7	11	8.6
8	6	4.6
9	3	2.2
	1088	1098.6

Property number one of pseudo-random numbers is violated and these numbers are rejected.

APPENDIX A

FORTRAN LANGUAGE PROGRAMS

This is a complete list of the Fortran language programs used to generate the random numbers in this paper.

METHOD I

Initial quantities: $x = N = 109$, $u_o = K = 2357$

```
  DIMENSION I(1000)
  PRINT 2
  PRINT 4
  J=1
  N=109
  READ 1,K                ....input card:  2357
7 I(J)=N*K
  TYPE 3,I(J)
  IF (J-1000)5,5,6
5 K=I(J)
  J=J+1
  GO TO 7
6 STOP
1 FORMAT (I4)
2 FORMAT (36RHANDOM NUMBERS GENERATED BY IBM 1620//)
3 FORMAT (I6)
4 FORMAT (13HM=109 K=2357//)
  END
```

METHOD II

Initial quantities: $x = N = 91$, $u_o = K = 2357$, $L = 10$

```
  DIMENSION I(1000)
  PRINT 2
  PRINT 4
  J=1
  K=2357
  L=10
```

(program continued on next page)

```
  READ 1,N                   ....input card:  0091
7 I(J)=N*K/N
  TYPE 3,I(J)
  IF (J-1000)5,5,6
5 N=I(J)
  J=J+1
  GO TO 7
6 STOP
1 FORMAT (I4)
2 FORMAT (36HRANDOM NUMBERS GENERATED BY IBM 1620//)
3 FORMAT (I6)
4 FORMAT (12HN=91 K=2357//)
  END
```

METHOD III

The random numbers for this method were not generated on the IBM computer; an adding machine was used.

METHOD IV

Initial quantities: 1) a_o = A = 1111, a_1 = B = 1111, D = 0.01

2) a_o = A = 1234, a_1 = B = 5678 D = 0.01

```
  DIMENSION I(1300)
  J=3
  PRINT 2
  READ 1,A,B,D               ....input card:  1111. 1111. 0.01
7 C=A*B                                       1234. 5678. 0.01
  I(J)=C*D
  F=I(J)
  PUNCH 3,F
  IF (J-1300)5,5,6
5 A=B
  B=F
  J=J+1
  GO TO 7
6 STOP
1 FORMAT (2F6.0,F10.6)
2 FORMAT (13HA=1111 B;1111,//)
3 FORMAT (F8.0)
  END
```

METHOD V

Initial quantities:

$1)u_0 = A = 12345678,\ B = 23,\ D = 0.000001$
$2)u_0 = A = 68470236,\ B = 23,\ D = 0.000001$

```
      DIMENSION G(2000)
      PRINT 2
      J=1
      READ 1,A,B,D              ....input card:  12345678.  23.  .000001
    7 C=A*B                                      68470236.  23.  .000001
      I=C*D
      F=I
      G(J)=C-F
      PRINT 3,G(J)
      IF (J-175)5,5,6
    5 P=0.1
      A=G(J)*P
      J=J+1
      GO TO 7
    6 IF (J-410)8,8,9
    8 Q=0.01
      A=G(J)*Q
      J=J+1
      GO TO 7
    9 IF (J-800)10,10,11
   10 R=0.1
      A=G(J)*R
      J=J+1
      GO TO 7
   11 STOP
    1 FORMAT (F11.0,F4.0,F10.8)
    2 FORMAT (35HLEHMER METHOD IGNORE FIRST 2 DIGITS//)
    3 FORMAT (E14.8)
      END
```

This method prints out data in the following manner:

```
.28395031E+09 .65308506E+09 .15020941E+10..34548130E+10 .79
460620E+10 .18275935E+1 .42034649E+11 .96679687E+11 .222363
28E+12 .51143554E+12 .11763017E+13 .27054939E+13 .62226360E
+13 .14312063E+14 .32917745E+14 .75710814E+14 .17413487E+15
.40051020E+15 .92117346E+15 .21186990E+16 .48730077E+16 .11
207918E+17 .25778211E+17 .59289885E+17 .13636674E+18 .31364
350E+18 .72138005E+18 .16591741E+19 .38161004E+19 .87770309
```

METHOD VI

Initial quantities: $x = 3379$, $u_0 = U(0) = 389$, $P = 100000$

```
  DIMENSION U(2000)
  PRINT 2
  N=0
  READ 1,X,U(N),P        ....input card:  3379.  389.  100000.
7 Y=X*U(N)
  J=Y/P
  Z=J
  U(N+1)=Y-Z*P
  PUNCH 3,U(N+1)
  IF (N-2000)5,5,6
5 N=N+1
  GO TO 7
6 STOP
1 FORMAT (F8.0,F6.0,F8.0)
2 FORMAT (33HRESIDUE METHOD IGNORE LAST 2 NOS.//)
3 FORMAT (F9.0)
  END
```

APPENDIX B

RANDOM NUMBERS

METHOD I (280 of 500 numbers)

6913	3837	7913	2837	8913	1837	9913	0837
3517	8233	2517	9333	1517	0233	0517	1233
3353	7397	4353	6397	5353	5397	6353	4397
5477	6273	4477	7273	3477	8273	2477	9273
6993	3757	7993	2757	8993	1757	9993	0757
2237	9513	1237	0513	0237	1513	9237	2513
3833	6917	4833	5917	5833	4917	6833	3717
7797	3953	6797	4653	5797	5953	4797	6953
9873	0877	0873	9877	1873	8877	2873	7877
6157	5593	5157	6593	4157	7593	3157	8593
1113	6937	2113	8637	3133	7637	4113	6637
1317	0433	0317	1433	9317	2433	8317	3433
3553	7197	4553	6197	5553	5197	6553	4197
7277	4473	6277	5473	5277	6473	4277	7473
3193	7557	4193	6557	5193	5557	6193	4557
8037	3713	7037	4713	6037	5713	5037	6713
6033	4717	7033	3717	8033	2717	9033	1717
7597	4153	6597	5153	5597	6153	4597	7153
8073	2677	9073	1677	0073	0677	1073	9677
9957	1793	8957	2793	7957	3793	6957	4793
5313	5437	6313	4437	7313	3437	8313	2437
9117	2633	8117	3633	7117	4633	6117	5633
3753	6997	4753	5997	5753	4997	6753	3997
9077	2673	8077	3673	7077	4673	6077	5673
9393	1357	0393	0357	1393	9357	2393	8357
9517	7553	5077	9593	1437	8833	2597	1273
7353	3277	3393	5637	6633	2797	3073	8757
1477	7193	9837	4433	2997	4873	4957	4513
0993	4037	2233	3197	6673	1157	0313	1917
8237	0033	3397	8473	7357	6113	4117	8953
7833	3597	0273	3557	1913	6317	8753	5877
3797	2073	9757	7713	8517	8553	4077	0593
3873	5957	3513	0717	8353	2277	4393	4637
2157	9313	2917	8153	0477	8193	8837	5433
5113	5117	7953	8677	1993	3037	3233	2197

METHOD II (All numbers included)

998	285	744	753	033	587	463	035	910	204
228	174	360	482	778	355	129	249	487	082
739	011	852	607	374	673	408	689	785	327
182	592	816	069	151	626	458	397	024	073
897	534	331	263	590	548	950	572	656	206
422	863	016	989	063	163	915	820	619	554
465	409	771	107	849	419	665	274	898	577
600	401	724	219	109	758	740	581	658	
420	515	646	618	691	660	418	941	090	
994	385	262	662	868	562	552	793	213	

METHOD III (three inputs)

1) a_o = 1111

1111	0228	6756	9606	0342	7758	5980	4996
2343	0519	6435	2752	1169	1865	7604	9600
4896	2693	4092	5735	3665	4782	8208	1600
9708	2522	7444	8902	4322	8675	3712	5600
2452	3604	4131	2456	6796	2556	7789	3600
0123	9888	0651	0319	1856	5331	6685	9600
0151	7725	4238	1017	4447	4195	6892	1600

2) a_o = 1234

1234	1684	8579	7491	1406	6915	6863	2900
5227	8358	5992	1150	9768	8172	1007	4100
3215	8561	9040	3225	4138	7815	0140	8100
3362	2907	7216	4006	1230	0742	0196	6100
3030	4506	0706	0480	5129	5505	0384	2100
1809	3040	4984	2304	6306	3050	1474	4100
2724	2416	8402	3084	7656	3025	1726	8100
4201	8370	5936	5110	6143	1506	9790	
6484	0569	2360	1121	7394	2680	8441	
0422	3237	5696	2566	2281	1824	2504	
1780	4781	4444	5843	2166	3269	2700	

3) a_o = 6043

6043	9163	3558	8601	7156	3025	0384	8100
5178	9605	6593	9772	2083	1506	1474	6100
8116	2560	4676	4919	2488	2680	1726	2100
8694	5536	8649	1965	4785	1824	9790	4100
5856	6472	8052	1812	8962	3269	8441	8100
2927	8867	8347	1665	3174	6863	2504	
5673	6236	6724	7722	0742	1007	2700	
1829	8876	2121	6292	5505	0140	2900	
3452	2833	4986	5892	3050	0196	4100	

METHOD IV (2 inputs)

1) $a_0 = 1111$, $a_1 = 1111$

1111	1778	1475	9659	4122	3956	0091	0005
1111	5034	7436	1953	7938	9144	0098	0000
2343	9504	9618	3867	7204	0549	0089	0000
6030	8431	9879	1601	1853	0035	0087	0000
1282	1282	6385	1910	3490	0012	0077	
7304	8085	0774	0579	4669	0067	0066	
3637	3649	9419	1058	2948	0128	0050	
5646	5021	2903	6125	7642	0085	0033	
5345	3216	3433	4802	5286	0108	0016	

2) $a_0 = 1234$, $a_1 = 5678$

1234	0365	1965	1921	9065	5243	8736	2317
5678	5190	7432	8165	0883	5666	7744	3420
0066	8943	6038	6849	0043	7069	6515	9241
3747	4141	8744	9220	0379	0472	4521	6042
2473	0329	7962	1477	0162	3360	4543	8341
2663	3623	6197	6179	0613	5859	5389	3963
5855	1919	3405	1263	0993	6862	4822	0552
5918	9525	1007	8040	6087	2044	9857	1915
6498	2784	4228	1545	0443	0259	5304	0598
4551	5176	3180	4218	6965	5293	2815	1279
5723	4099	6358	5168	0854	3708	9307	7533
0453	2164	2184	7986	9481	6263	1992	6347
5925	8702	8858	2716	0967	2269	5395	8199
6840	8311	3458	6899	1681	2130	7468	5312
5270	3223	6309	7376	6255	8329	2898	1281
0468	7863	8165	7870	5146	7407	6422	8046
4663	3424	5129	4251	1882	6929	6109	3069
1822	9229	8782	7063	6847	3231	2319	6931
4959	6000	0428	0248	8860	3875	1667	2712
0352	3740	7586	7516	6644	5201	8657	7968
7455	4400	2468	8639	8658	1538	4312	6092
6241	4560	7222	8307	5237	9991	3289	5410
5266	0640	8238	4031	3019	3661	1821	9577
8651	9184	4948	5165	9053	5770	9892	8155
5561	8777	7616	8201	9522	1239	0133	7173
1082	6079	6839	3581	2026	1490	3156	2088
0170	3553	0858	3667	2915	8461	4197	9772
1839	5986	8678	1673	9057	6068	2457	4039
3126	2682	4457	1516	4011	3413	3120	4691
7487	0544	6778	6352	3276	7100	6658	9469
4043	4590	2095	1287	1400	2323	7729	4190
2699	4969	1909	9008	5864	4933	4596	6751
9120	8077	1879	5932	2096	4593	5224	2866
6148	1346	7561	4354	2909	6572	0095	3483
0697	8716	2071	8279	0972	1851	4962	9822
2851	7317	6588	0467	8275	1647	4713	2100
9871	7749	6437	8662	0433	0485	3859	6262
1422	6994	4069	0451	5830	7987	1874	1502

METHOD IV (cont.)

4055	6909	5192	3767	6959	1209	5812	6086
0906	7367	5125	0141	3023	7264	9774	3681
6738	8986	6090	5311	0370	7812	6321	4025
1046	1998	2112	7488	1185	8117	5918	8160
0479	9540	8620	7687	4384	4830	4076	8440
5010	0609	2054	5602	1950	2051	1217	8704
3997	8098	7054	0625	5488	9063	9605	4617
0249	9316	4889	5012	7016	5882	6880	1863
9952	4409	4870	1325	5038	3085	0755	6014
4780	0742	8094	6409	3466	1459	1944	2040
5705	2714	4177	4919	4617	5010	4677	2685
2699	0137	8086	5258	0025	3095	0920	4774
3787	3718	7752	8641	1154	5059	3028	8181
7339	5093	6826	4242	0288	6576	7857	0560
1872	8357	9151	5278	3323	2679	7909	5813
7387	6552	4647	9223	9570	6171	1410	2552
8265	3070	5246	6789	8011	5321	1516	8347
0452	1146	3781	6149	6652	8358	1375	3015
7357	5182	8351	7455	2891	4729	0845	1662
3253	9385	5751	8407	2309	5249	1618	0109
9232	6330	0226	6741	6753	8225	3672	1811
3277	4070	5297	6175	5926	1730	9412	1973
5514	7631	4090	2658	0182	2292	5608	5731
0693	0581	6647	8484	0785	9651	7824	
8212	4336	1862	5504	1428	1200	8769	

METHOD V (all numbers: 4806)

839501	506272	990397	238698	248515	192777	327344
530856	464430	837790	749003	171593	743393	455285
502091	968199	226913	242271	189468	309806	347166
454810	832681	721914	157222	735773	451251	698473
946060	215177	236043	186169	292270	337887	770645
827595	694901	142897	728171	447220	677134	072499
203469	229829	182863	274792	328619	765741	366730
667967	128602	720592	443200	655807	061208	154343
223638	179570	257355	319366	760832	340776	955004
114354	713037	439192	634536	049923	148374	139658
176307	239979	310141	755949	314826	941273	221198
705499	435198	613328	038670	142404	136492	028756
222630	300947	751062	288943	927546	613931	386611
431203	592171	027450	136450	133336	012048	189211
291775	746203	263130	913851	606666	382771	335182
571084	016267	130521	130188	995431	180372	687091
741347	237396	900195	599423	378922	314858	880318
005100	124609	127047	978688	171539	682414	924711
211736	886584	592201	375090	294533	869557	052689
118690	123914	962077	162721	677740	899987	721173
873007	585001	371275	274261	858806	046990	085873
120798	945501	153930	673085	875252	708098	497507
577821	367466	254058	848082	041307	082869	744259
928985	145172	668431	850590	695010	490581	321174
363665	233894	837398	035631	079853	728332	038713
136430	663796	826007	681963	484663	317519	989037
213805	826732	029982	076850	712417	030289	607478
659171	801481	668950	476756	313853	299656	697199
816104	024343	073861	696541	021865	603016	503555
777039	655980	469872	310208	950299	686941	955812
018711	070876	680719	013473	598560	479951	498389
643043	463012	306566	930986	676707	950382	034628
067903	664933	005109	594124	456430	484889	379659
456170	302936	911731	666490	944979	031754	473180
649199	996759	589694	432939	473455	373038	258836
299315	892524	656303	939578	028898	457974	895317
988427	585288	409505	461020	366457	255333	659212
873371	646141	934180	026031	442841	887269	531624
580872	386146	448628	359887	251851	640715	522729
636013	928813	023184	427726	879267	527363	102279
362845	436263	353321	248379	622314	512942	863525
923453	020347	412644	871269	523131	079765	286108
423946	346789	244904	603919	503208	858344	858031
017508	397606	863282	518902	057367	274194	267348
340266	241447	585564	493475	853198	830641	214899
382619	855333	514680	034982	262340	261048	199420
238005	567269	483760	848049	803404	200412	758681
847402	510478	012652	250506	254781	196098	344968
549028	474081	842912	776169	185998	751010	459348

METHOD V (cont.)

356484	095212	996531	096212	437936	074945	107226
719923	418996	149206	402134	710720	087234	546616
775584	166369	643160	224899	934665	800643	857206
083835	982659	079288	417268	049733	104141	347157
392839	146007	398238	705771	081431	539549	098454
160351	635823	215940	923738	787309	840947	126449
968805	062393	396660	024593	101083	343412	639087
142827	394350	701233	075653	532487	089869	769889
628491	207000	912833	774003	824721	106684	670718
045549	376115	999519	098021	339686	634530	994267
390470	696506	069883	525458	081279	759431	586800
198098	901969	760744	808538	086939	646707	054965
355610	974517	094974	335965	629995	988742	426410
691791	064134	518432	072711	748987	574108	580769
891128	747513	792397	067240	622673	052041	283570
894581	091922	332257	625468	983215	419704	952225
058404	511432	064179	728576	561397	565312	790118
734328	776320	047615	598713	049120	280020	561727
088894	328553	620956	977704	412973	940056	591961
504454	055667	728184	548724	549851	771314	261529
760256	028031	574827	046202	876465	557408	900156
324850	616442	972216	406273	935877	582024	370347
047177	771826	536084	534439	752506	238667	005182
008501	551008	043294	272911	553076	894890	311915
611959	966733	399586	927719	572079	358256	317404
707496	523480	519058	733740	215770	002399	223008
527248	040409	269384	548767	889621	305512	812904
961268	392921	919575	562146	346143	302680	469687
510916	503728	715039	192941	996123	219616	488025
037511	265858	544456	884371	299103	805126	422463
386272	911463	552255	334063	287953	451772	871666
488448	696371	170188	968358	216228	483908	810481
262335	540167	879146	292726	797322	412992	164111
903376	542384	322027	273257	433853	849875	577455
677778	147488	940651	212847	479786	805478	202818
535883	873928	286356	789553	404509	152581	066470
532548	310018	258607	415979	828069	550947	165280
124848	913048	209480	475672	800452	196715	680167
868711	279994	278184	394057	141040	052453	164378
298044	243996	398141	806316	524401	162066	417801
885495	206117	471573	795458	190619	672743	260953
273664	774073	384629	129544	038410	147315	500195
229421	280372	784621	497942	158838	413888	725049
202765	467488	790465	184524	665320	251934	967606
766366	375213	118064	024410	130230	479439	125499
362645	762998	471553	155610	409953	720272	098864
463405	784490	178452	657915	242899	956628	827388
365836	106620	010459	113204	458653	100237	110297
741431	445241	152404	406038	715494	093050	553680
780522	172408	650351	233886	945632	814022	873481

Method V (cont.)

350904	382524	421750	612392	496334	854328
107079	007986	270041	060857	441559	816490
146261	318354	521095	439956	915588	567932
643645	332214	729857	611902	820582	906253
780372	226410	978659	290730	187340	358433
694854	820740	150915	968691	630898	124401
999814	487709	104716	827997	215100	186136
599580	492170	840834	570437	094748	652819
057904	431999	113399	612013	171791	801464
433178	893592	560804	307620	695115	743370
596309	815529	889842	910715	198779	010973
287150	175719	354669	394732	425712	625245
960443	604134	115720	010789	279159	063805
809028	208965	166171	324814	542044	446750
566076	080587	648216	347074	734674	627535
601972	168534	790901	229829	989742	294334
284543	687634	719087	828609	176417	796968
905444	181552	005388	505784	110377	847026

METHOD VI

14431	89670	24870	72070	51270	82470	85670
62349	94930	35730	24530	41330	66130	78930
77270	68470	31670	86870	54070	53270	04470
95330	60130	12930	33730	02530	99330	04130
20070	79270	90470	73670	48870	36070	55270
16530	53330	98130	30930	31730	80530	57330
54870	02070	81270	12470	15670	10870	18070
05730	94530	11330	36130	48930	29730	58530
61670	16870	84070	83270	34470	57670	72870
82930	03730	72530	69330	74130	66930	27730
20470	03670	78870	66070	85300	56470	99670
68130	00830	01730	50530	27330	12130	84930
11270	42470	45670	40870	48070	87270	78470
81330	06130	18930	99730	28530	85330	50130
14070	13270	64470	87670	02870	30070	89270
42530	39330	44130	36930	87730	06530	43330
08870	96070	15270	86470	29670	64870	12070
71730	20530	97330	82130	54930	95730	84530
75670	70870	78070	17270	08470	71670	26870
88930	69730	98530	55330	20130	72930	93730
94470	17670	32870	60070	19270	30470	13670
14130	06930	67730	76530	13330	58130	90930
45270	16470	56970	94870	42070	21270	52470
67330	52130	24930	65730	54530	71330	96130
08070	47270	38470	01670	56870	24070	23270
68530	25330	90130	42930	63730	32530	29330
62870	90070	49270	60470	43670	18870	06070
37730	46530	83330	28130	67930	61730	10530

METHOD VII

34952	60471	61711	93776	26624	05995	08842	24603
35127	25844	26638	34063	96199	84343	07349	58047
18016	39578	49763	18036	28019	37442	75582	90675
20707	72736	27380	06462	36834	59647	21510	33894
02089	56703	89359	49088	99370	24983	62205	67621
06864	28916	98981	75856	62690	39800	44419	93936
82984	72156	82689	89950	11630	98604	16257	60845
83668	07181	61425	59376	91491	62225	19270	81669
76785	28060	69981	16896	46217	76946	65427	21750
59422	27285	34793	57593	62853	10911	21228	32098
26050	82179	45790	80923	42656	44760	42024	56031
08805	66646	51533	68718	13898	85901	54703	01506
00341	79158	49524	30935	09349	97364	05891	12182
03779	60954	64978	03279	89866	44127	67397	64898
43996	88943	11640	69406	01549	68492	99045	42686
50707	69868	77889	63652	98524	42354	25199	86868
15378	18483	82622	15821	33607	68938	73186	04481
45577	82011	24735	43720	88090	65992	59149	14620
86497	70045	39931	50992	63667	13093	10123	49386
47674	30983	86279	71382	44059	13199	91438	18701
07572	52689	21911	99951	74464	81789	60268	12648
46437	33942	13521	09244	48477	51802	50105	46485
46580	59189	46571	40438	94822	82979	41586	42706
46992	96902	88312	98913	62863	27146	08965	61608
87054	42439	66246	14436	28318	30996	77184	54742
58458	56247	03734	09224	94862	98692	21676	93469
54028	66194	28020	67434	18798	15244	89967	69783
08387	61683	12188	95846	00536	95992	53751	52015
36492	96935	62390	23598	13176	46336	39380	42675
09897	09807	10385	58159	59470	57850	24945	96763
84962	53012	36515	29384	73549	53395	58747	56676
89203	78581	29858	86035	68103	08333	50530	86544
84331	27445	83799	81486	75460	32533	78894	60190
54309	02866	68831	49243	79307	42354	73615	67624
70288	47016	94808	28093	81090	78037	57785	89380
79842	28977	06480	37873	34771	37888	27864	95566
05969	96973	27580	85842	72470	63838	44363	10675
45668	07042	02841	24588	53867	08848	73209	80950
97560	48078	77699	19682	32202	11285	45632	60864
59181	65625	66044	18909	42762	47119	23340	03183
83357	16857	93649	58496	43700	39779	61368	41034
26978	60659	09626	85501	10882	22340	71188	68142
70823	20730	77864	49555	16734	14092	21872	15861
97345	08270	82141	55513	73557	57662	24221	61525
65781	38934	24132	24799	86009	06036	77462	20592
94937	43273	99552	14369	66691	91421	38162	93203
93163	14424	55580	93089	08689	92105	42666	31069
51039	86966	65533	13884	51632	68689	45918	27916
83860	16508	22444	29585	09108	56911	06088	63083

BIBLIOGRAPHY

1. IBM Reference Manual, Random Number Generation and Testing, International Business Machine Corporation, 1959.

2. The RAND Corporation, A Million Random Digits with 100,000 Normal Deviates, The Free Press, Glencoe, Illinois, 1955.

3. M. G. KENDALL and B. Babington SMITH, Randomness and Random Sampling Numbers, Journal of the Royal Statistical Society, Vol. 101, Part 1 (1938) pp.147 - 166.

4. F. MOSTELLER, R. ROURKE, and G. THOMAS, Jr., Probability with Statistical Applications, Addison Wesley Publishing Co., 1961.

5. S. H. HOLLINGDALE, High Speed Computing, The Mac Millan Co., New York, 1959.

6. George E. FORSYTHE, Generation and Testing of Random Digits at the National Bureau of Standards, The Monte Carlo Method, The National Bureau of Standards Applied Mathematics Series, Los Angeles, No. 4, 1951, pp. 34 - 35.

7. D. H. LEHMER, Mathematical Methods in Large-Scale Computing Units, Proceedings of the 2nd Symposium on Large Scale Digital Calculating Machinery, Harvard University Press, 1951, pp. 141 - 146.

8. H. GRIFFIN, Elementary Theory of Numbers, Mc Graw-Hill Book Co. Inc., New York, 1954, pp. 53 - 65.

9. J. F. KENNEY and E. S. KEEPING, Mathematics of Statistics Part I, D. Van Nostrand Company Inc., Princeton, New Jersey, 1957.

10. Hugh D. YOUNG, Statistical Treatment of Experimental Data, Mc Graw-Hill Book Company, Inc., New York, 1962.

11. Bernice BROWN, Some Tests of the Randomness of a Million Random Digits, RAOP-44, The RAND Corporation, Santa Monica, California, 1948.

These sources also were used:

12. Bernard A. GALLER, The Language of Computers, The Mc Graw-Hill Book Co. Inc., 1959.

13. G. R. STIBITZ and J. A. LARRIVEE, Mathematics and Computers, Mc Graw-Hill Book Company Inc., 1957.

14. Howard W. ALEXANDER, Elements of Mathematical Statistics, John Wiley and Sons Inc., 1961.

15. J. M. CAMERON, Results on Some Tests of Randomness on Pseudo-Random Numbers, Annals of Mathematical Statistics 23, 1952, pp. 138.

16. W. J. COCHRAN, The x^2 Test of Goodness of Fit, Annals of Mathematical Statistics 23, 1952, pp. 315 - 345.

17. B. LINDGREN and G. MC ELRATH, Introduction to Probability and Statistics, The Mac Millan Company, 1959.

18. Herbert A. MEYER (Ed), Symposium on Monte Carlo Methods, John Wiley and Sons, Inc., 1956, Bibliography.

19. D. D. MC CRACKEN, The Monte Carlo Method, Scientific America, Vol. 192, May 1955, pp. 90 - 96.

Section VII Future Plans

1. Judging from the number of requests for reprints of the Bibliography on Computer-Aided Circuit Analysis and Design compiled earlier this year, it seems there is an existing demand for such a reference in various quarters. Continuous attention will be given to newly published literature so that the Bibliography may be updated by bringing in additional entries from time to time.

2. The time-sharing computer service provided by General Electric Company on the Villanova University campus has been used by graduate and undergraduate students to solve various types of problems ranging from the relatively short ones connected with simple laboratory experiments through more complicated and lengthy thesis problems. This research project is particularly interested in the use of a circuit analysis program developed by the General Electric staff called STANPAK (Statistical Tolerance Analysis Package). It is basically a reliability prediction and tolerance analysis and adjustment program using the statistical approach. It handles the steady states only and has not been extended to transient computations. Because of incomplete knowledge about the program and the peculiar limitations of the Desk Size Computer at the input terminal, the program has not been managed to smooth operation yet. Further effort will be made in the study and evaluation of the STANPAK in the coming months.

3. In the process of circuit design there is a stage of parameter optimization after the circuit geometry has been chosen. Instead of numerical values available for analysis, the circuit components and other parameters such as frequency may be represented by symbols. Nonnumerical manipulation by the digital computer is a vast

area to be explored. The practical programs written for symbolic manipulation are far sparse than theoretical dissertation published on the subject. Techniques will be attempted for efficient manipulation of mathematical symbols. Again the interest will be centered around the computer size of the same order as IBM 1620 with a disk file.

4. In the manual solution of the transient response of electric circuits, the frequency domain approach by Laplace transform is always a favorable one. Several recent research papers were published in the new methods of finding the inverse of the Lapalce transforms. An attempt will be made, using the digital computer as an aid, to evaluate and compare various approaches for accuracy, time and storage requirements for computation, and the ease with which these methods may be applied to various types of electronic circuits.