computer-aided analysis of electronic circuits: algorithms & computational techniques

Leon O. Chua | Pen-Min Lin

蔡少棠 | 林本銘

COMPUTER-AIDED ANALYSIS OF ELECTRONIC CIRCUITS

PRENTICE-HALL SERIES IN
ELECTRICAL AND COMPUTER ENGINEERING

LEON O. CHUA, *editor*

Chua and Lin *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*

Preface

Traditionally, the design of most electronic circuits starts with paper-and-pencil work by an engineer who, besides his basic training, is armed with a wealth of design charts, tables, and monograms. He relies very heavily on his intuition, past experience, and knowledge to make reasonable approximations. Then comes the "breadboarding stage," where the result of the preliminary design is confirmed, and perhaps improved, by adjusting circuit element values in a trial-and-error fashion.

The advent of integrated circuits, however, has greatly changed the picture. Not only are the circuits much larger but the specifications are also much tighter. The paper-and-pencil method is no longer adequate when we consider the required accuracy of the results and the time to complete a design. Breadboarding is also of little help because it is impossible to duplicate an integrated circuit with discrete components. Actual production of a mask for an integrated circuit is very costly. Aside from the cost consideration, neither method permits a tolerance or worst-case study. It is in such an environment that the digital computer emerges as an important design tool. Instead of simulating a circuit via breadboarding, a computer program is developed to simulate and analyze the circuit. Such computer-aided circuit analysis is the first step toward an automated circuit design. The other important ingredient is an efficient optimization technique. Today, circuit simulation programs are generally recognized as indispensable tools in any sophisticated circuit design.

This book is devoted to *computer-aided circuit analysis, with emphasis on computational algorithms and techniques*. It is intended as a *textbook* for senior and first-year graduate students in electrical engineering. We hope that the student, after finishing the book, will have not only a thorough understanding of the basic principles and algorithms used in many existing computer simulation programs, but also the capability to develop a small, special-purpose program for his own use. The *prerequisites* for the study of this book are (1) elementary circuit analysis (sinusoidal steady-state analysis, Laplace transform method) and (2) elementary matrix algebra (multiplication, inverse, partitioning). Such background is common among all senior students of electrical engineering.

The preparation of the present text was motivated by an unfulfilled need. At the time of the writing of this book, which started about four years ago, there were already about a dozen texts on the market dealing with various aspects of computer-aided circuit analysis. We examined these and found that they all fell under one of the following categories:

1. *Very narrow coverage.* Some books discuss in great detail how to use certain specific programs, with little or no emphasis on the basic theory. Although useful for training designers whose company happens to have these programs, such texts obviously are unsuitable for college students.
2. *Very elementary.* Such books touch on many important topics superficially, never deep enough to be useful. A case in point is the discussion of the formulation of state equations which is restricted to linear *RLC* networks without controlled sources. Such texts cannot satisfy the demand of serious students.
3. *Very advanced.* These are the books written for those who are already fairly knowledgeable in the field.
4. *Edited volume.* This category seems most prevalent. Because the efforts of a group of experts are pooled, it is possible to cover topics that, at the time, represent the state of the art. On the negative side, such texts usually suffer from notational difficulties and poor coordination among the chapters. These books are good references for researchers, but rarely good textbooks for students.

It was under such circumstances that the authors started four years ago to prepare the present book. It is the outgrowth of class notes, portions of which have been used at the University of California at Berkeley and at Purdue University. The book leads the students step by step from very elementary to fairly advanced topics. Where the frontier of knowledge is reached, and the level is beyond that assumed by the book, we give references to literature. We emphasize algorithms rather than actual programming details, since the former possess "universality," while the latter vary with the programming languages used (FORTRAN, APL, etc.). We do not advocate any *one* approach, to the extent of ignoring the others. The selection of topics is balanced, as evidenced from the table of contents and the following brief synopsis of the chapters.

In the first three sections of *Chapter 1*, we define the problems to be solved with computer simulation programs, describe the main ingredients of a computer simulation program, and give some actual examples of the use of these programs. In the remaining sections of Chapter 1, we give extremely elementary explanations of some

此为试读,需要完整PDF请访问: www.ertongbook.com

basic techniques, idiosyncrasies, and numerical problems associated with many computer simulation programs, with the hope of motivating the students to an in-depth study of the subsequent chapters.

In *Chapter 2* we discuss the principles of modeling electrical components and devices commonly used in electronic circuits. Models for junction diodes and bipolar transistors are described. Models for other semiconductor devices are merely referenced, since a detailed study of this topic is clearly beyond the scope of the present book. A potentially powerful unified *black-box approach* for synthesizing dc circuit models for three-terminal devices is presented with more detail, since it is not available elsewhere. Among other things, a solid understanding of this approach will permit a circuit designer to transform many circuit elements previously not allowed by a particular computer simulation program into an acceptable equivalent circuit. Also included in this chapter is a black-box "macro" circuit model of an *operational amplifier* that is capable of simulating not only the frequency and phase characteristics but also various important nonlinear effects, such as the op amp's slew-rate limitation. The principle used for deriving this model is presented in some detail, not only because it demonstrates the usefulness of the black-box modeling approach but also because the same approach could conceivably be applied to the modeling of other *IC* modules.

In *Chapter 3* we present the fundamentals of graph theory which are applicable to circuit analysis. The results of this chapter are used extensively in the subsequent chapters: in hybrid matrix formulation (Chapter 6), state equation formulation (Chapters 8 and 10), symbolic network analysis (Chapter 14), and adjoint-network sensitivity analysis (Chapter 15). Particular attention is paid to computer generation of various topological matrices.

In *Chapter 4* we present a detailed study of the nodal analysis of linear networks, with emphasis on the use of digital computational techniques. The content of this chapter is practically self-contained. A skeleton program called NODAL is included in the appendix to this chapter so that it may be used as the starting point of a student project to expand it into a full-scale program. Our experience shows that such a project is extremely beneficial for learning the material in this book.

In *Chapter 5* we extend the nodal-analysis method to nonlinear resistive networks. The *fixed-point algorithm* is introduced here as a unifying concept from which several latter algorithms—Newton–Raphson (Section 5-4), predictor-corrector (Section 11-5-2), and periodic solution (Section 17-5-1)—can be considered as special cases. The most commonly used Newton–Raphson method for solving nonlinear functional equations is discussed in detail. The solution of nonlinear resistive networks is actually the cornerstone of most computer simulation programs, and the solution process is usually referred to as *dc analysis*. The use of "discretized *linear* resistive circuit models" for implementing the Newton–Raphson method for *nonlinear* resistive networks is given a detailed treatment because this important technique is usually presented elsewhere without rigorous justification.

In *Chapter 6* we describe the computer formulation of *hybrid matrices* for linear resistive n-ports. Like Chapter 3, the material from this chapter is used in several subsequent chapters (Chapters 7, 8, 10, and 15). The general analysis technique that depends on this material is usually referred to as *hybrid analysis*. Although not as widely known as nodal analysis, there are many circuits for which it is computationally advantageous, if not necessary, to resort to hybrid analysis. For example, this approach is particularly suited for analyzing circuits containing nonmonotonic voltage-controlled and current-controlled nonlinear resistors. It is also quite useful for analyzing nonlinear circuits containing many *linear resistors* because the number of nonlinear equations that need to be solved repeatedly would then be equal to the number of nonlinear resistors in the circuit.

In *Chapter 7* we apply the hybrid-matrix method of Chapter 6 to the analysis of nonlinear resistive networks. Several recent algorithms for solving piecewise-linear networks are described. Although computationally less efficient than the Newton–Raphson algorithm of Chapter 5, the piecewise-linear approach has at least two advantageous features. First, for a large class of nonlinear networks, the piecewise-linear approach presented in Section 7-3 is guaranteed to converge, regardless of the initial guess. Second, it is still the only approach capable of finding all solutions of a nonlinear resistive network.

The formulation of state equations is thoroughly discussed in *Chapter 8* for dynamic linear networks, and in *Chapter 10* for dynamic nonlinear networks. The concept of hybrid matrices from Chapter 6 plays a fundamental role in these two chapters. Although it is true that from a programming and computational point of view, the *state-equation approach* is not as appealing as the *tableau approach* (Section 17-2), in so far as developing *large* simulation programs is concerned, the concept and formulation of state equations are of basic importance in many other respects. For example, any qualitative analysis concerning stability, transient decay, bifurcation behavior, etc., requires the formulation of the network's state equations.

Numerical integration techniques for state equations and the associated stability and time-constant problems are given a thorough treatment in *Chapters 11, 12, and 13*. These chapters form a self-contained package containing an *up-to-date* and *in-depth* study of the numerical integration of *ordinary differential equations* and could be virtually lifted out as supplementary text material for a course on *system analysis, modeling*, or *simulation*. Although these chapters are much more mathematical than the rest, all results are derived by elementary methods. A unifying approach is used to derive the three important families of integration methods: the *explicit Adams–Bashforth algorithm*, the *implicit Adams–Moulton algorithm*, and the *implicit Gear's algorithm*. Using a novel approach, a simple yet *generalized formula* which gives the *local truncation error* for all multi-step algorithms is derived. This analysis leads to the formulation of an efficient *variable order, variable step-size* predictor–corrector algorithm. (The explicit Adams–Bashforth algorithm serves admirably here as the predictor.) Instead of using the well-known z-transform approach to perform the *stability analysis*, a *difference-equations approach* is adopted here, since it is a more direct approach and it provides additional insights. In particular, it shows vividly how the *parasitic terms* could destroy the stability of an otherwise accurate numerical integration algorithm.

Several special numerical techniques that are applicable only to linear networks are discussed in *Chapter 9*. Also included in this chapter is the computer determination of the transfer function $H(s)$ of a linear circuit. This subject is also encountered in Chapter 14, albeit from a different point of view. Finally, the highly efficient ***QR* algorithm** for computing *eigenvalues* of the associated *A* matrix is presented with many illustrative examples.

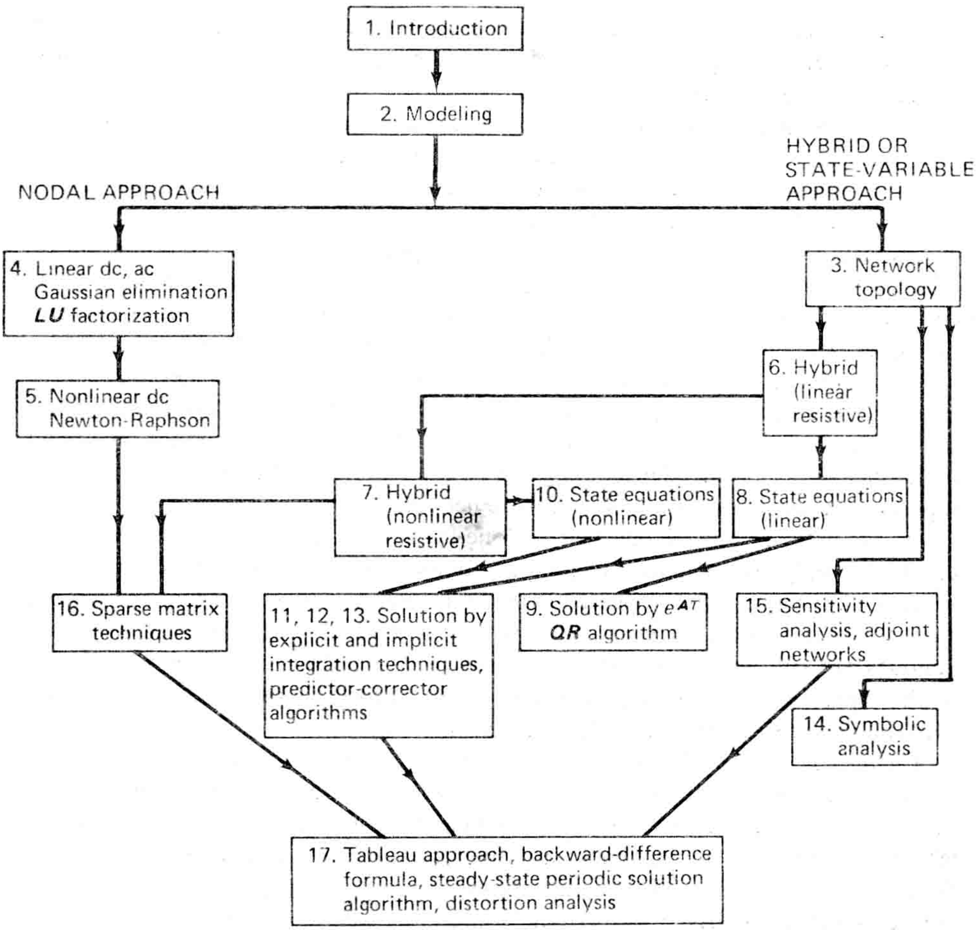
The remaining four chapters of the book cover somewhat specialized, and yet very important, topics in computer-aided circuit analysis. As is well known, analysis is the first step to design. The automated design of electronic circuits, a subject of extensive research at present, requires a good analysis program and a good optimization strategy. Most optimization techniques require the efficient computation of partial derivatives of network functions, and these are often obtained from a computer-aided sensitivity analysis. In *Chapter 15* we describe several methods of sensitivity analysis. The adjoint network method receives full attention here because of its generality and efficiency. However, the potential of the symbolic method is also pointed out as a natural sequel to *Chapter 14*, in which a unified treatment of the computer generation of symbolic network function is presented. Although symbolic analysis has some serious shortcomings, it is nevertheless useful for carrying out a comprehensive sensitivity study of small-size networks.

As the network size becomes larger (one hundred nodes or more), it becomes a necessity to apply sparse-matrix techniques to solve the large systems of equations. Although many papers have appeared in the literature, it is difficult to find one written for the novice in this area. *Chapter 16* is written to fill this need. Since sparse-matrix techniques are still an active field of research, new algorithms much more efficient than those presented in this chapter will no doubt be forthcoming, and the interested reader should consult the recent literature on this subject.

Chapter 17, the final chapter, is devoted to a number of recent results from computer-aided circuit analysis. Among other things, the recent *tableau approach* for analyzing large-scale networks and a new efficient algorithm for solving differential-algebraic systems using implicit backward differentiation formulas are presented. Also included in this chapter are efficient algorithms for computing the steady-state periodic response of networks driven by periodic sources, as well as efficient algorithms for computing the periodic solutions of nonlinear oscillators. This final chapter ends with the presentation of the latest algorithm for carrying out a spectrum and distortion analysis of nonlinear communication circuits.

A flowchart depicting the relationships among the chapters is given here, and immediately following the Preface is a complete listing of all algorithms and computational techniques presented in this book. We believe that most of the important results and methods relevant to computer-aided circuit analysis are covered in this book. We have decided not to include any material on *optimization techniques* and *tolerance analysis* because to do justice to these important subjects would have increased the length of this book considerably. Moreover, these subjects are really more relevant to computer-aided *design* and are therefore outside the scope of this book.

The book contains enough material for a one-year course at senior level. As a rule, the sections in each chapter are organized in the order of increasing level of sophistication. In fact, the last two sections in each chapter often contain advanced materials that can be omitted without loss of continuity. Such sections are coded with a star. With judicious selection of topics (not necessarily in the order of the chapters),



Flowchart of relationships among chapters

it is possible to use the book for a one-semester, a one-quarter, a two-semester, or a two-quarter course. The table on page xxiii illustrates a possible organization of topics for six typical courses in computer-aided circuit analysis. In each course, we recommend that only the first three sections of Chapter 1 be covered initially in class for motivational purposes. The remaining sections may be assigned for self-reading in order to give the students a bird's eye view of the many important ideas which will be covered in depth later. Alternately, each of the remaining sections may be used as a lead-in *synopsis* for a subsequent chapter at the appropriate point in time. Although mathematical proofs are given for most propositions and theorems in this book, they are mainly intended for research-oriented students and may be omitted in any introductory course where the emphasis is on the interpretation and application of these results.

Problems are included at the end of each chapter. Most of these problems have been *class-tested* to ensure that their level of difficulty and degree of complexity are proper for the students. When the book is used for undergraduate classes, those prob-

COURSE	LENGTH OF COURSE AND LEVEL	CHAPTERS AND SECTIONS COVERED	REMARKS
I	One semester, 15 weeks, 3 hours/week, senior	Chaps. 1–6 (omit Sections 4-4 and 6-6); Sections 7-1, 7-2, 8-1, 8-2, 8-4, 9-5; Chaps. 10–13; and selected topics from Chaps. 15 and 16	Covers dc, ac, and transient analysis using both nodal and state-variable methods; emphasizes implicit integration techniques
II	Two semesters, 30 weeks, 3 hours/week, senior	Practically the whole book except Chap. 17	Special-purpose computer simulation programs could be assigned as student group projects
III	One semester, 15 weeks, 3 hours/week, first-year graduate	Quick review: Chaps. 1, 2, 3, 4, 9; Lectures: Chaps. 5, 6, 8, 10, 11, 12, 13, 15, 16; and selected topics from Chap. 17	Graduate students having prior undergraduate backgrounds in computer-aided circuit analysis could skip Chaps. 1, 2, 3, and 4
IV	One quarter, 10 weeks, 3 hours/week, senior	Chaps. 1, 2, 4, 5; Chaps. 11–13; and selected topics from Chaps. 15 and 16	Confined to dc, ac, and transient analysis using the nodal method; emphasizes implicit integration techniques
V	Two quarters, 20 weeks, 3 hours/week, senior	Plan IV, plus Chaps. 3, 6, 7; Sections 8-1, 8-2, 8-4, 9-5; Chap. 10; and selected topics from Chaps. 15 and 16	Second quarter covers hybrid and state variable methods, adjoint-network, and sparse-matrix techniques
VI	One quarter, 10 weeks, 3 hours/week, first-year graduate	Quick review: Chaps. 1, 2, 3, 4, 9; Lectures: Chaps. 5, 6, 8, 10, 11, 12, 13; and selected topics from Chaps. 15, 16, and 17	The star sections dealing with programming details may be assigned as term projects

lems requiring more than elementary matrix algebra should be omitted. This judgment is left to the instructor. To avoid assigning the same problems over successive years, many problems contain several similar parts that differ only in numerical parameters or other trivial details. In these cases, the instructor should assign only the parts that do not duplicate each other.

Although the emphasis of the book is on algorithms rather than programming details, we have included three FORTRAN programs for good reasons. The NODAL program in the appendix of Chapter 4 makes use of the formulation and solution techniques described in that chapter. The program is small enough so that each student may be issued a deck of the source program. He may then use it as a starting point to expand into a full-fledged dc, ac, and transient-analysis program. This avoids much of the frustration he will encounter if he writes the program from scratch.

The HYBRID program in the appendix of Chapter 6 makes use of the concepts and techniques discussed in Chapter 3 and Section 6-6. It is an excellent example of putting theory into practice. Moreover, since no program for obtaining hybrid matrices in the most general case (all four types of controlled sources are allowed) has yet been reported in the literature, the inclusion of this program should be of value to both students and researchers in computer-aided design.

The program SPARSE in the appendix of Chapter 16 is included for the same pedagogical reasons. It is very difficult to find in any textbook a FORTRAN program that is simple enough, and yet illustrates clearly the essence of the sparse-matrix technique.

The authors would like to thank Dr. Y-F Lam and Dr. S. M. Kang of the University of California, Berkeley, and Professor L. P. Huelsman of the University of Arizona for offering many useful suggestions for improving the preliminary version of this book. The authors would also like to acknowledge the contributions of the following colleagues who have read portions of the final manuscript: Professor A. Dervisoglu of the Technical University of Istanbul, Turkey; Dr. T. Roska of the Institute of Telecommunication in Budapest, Hungary; Professor A. Ushida of the Technical College of Tokushima University, Japan; and Professors B. J. Leon and S. C. Bass of Purdue University, Indiana. Two of the computer programs used in Chapter 1 were made available by the following colleagues: SPICE by Professor D. O. Pederson and Dr. L. W. Nagel of the University of California, Berkeley, and CORNAP by Professor C. Pottle of Cornell University. A number of the authors' students have also contributed in various aspects during the evolution of this book. They are G. E. Alderson, L. K. Chen, E. Cohen, W. J. Kim, D. J. Reinagel, T. E. Richardson, C. Stewart, J. Stockman, N. N. Wang, and B. C. K. Wong. They would also like to acknowledge the support provided by the National Science Foundation under Grant 32236. The second author wishes to thank Professor C. L. Coates and J. C. Hancock of Purdue University for their support and encouragement. Finally, the authors would like to thank Mary Ann Ratch, Beth Harris, Linda Stovall, June Harner, and Helen Hancock for their expert typing of various portions of this book.

LEON O. CHUA
Berkeley, California

PEN-MIN LIN
Lafayette, Indiana

此为试读，需要完整PDF请访问：www.ertongbook.com

List of Algorithms and Chapters Where Found

Contents

8 Computer Formulation of State Equations for Dynamic Linear Networks 328

9 Numerical Solution of State Equations for Dynamic Linear Networks 364

10 Computer Formulation of State Equations for Dynamic Nonlinear Networks 400

此为试读，需要完整PDF请访问：www.ertongbook.com