计算机辅助电路分析与设计

上册

电工教研室
余庆健 编

华东工程学院
一九八二年二月

随着电子数字计算机的迅速发展及其使用的日益广泛，随着电子网络的规模日益巨大和电子线路的集成，使用计算机完成对电路的分析与设计已成为电路理论和技术的一个崭新的课题。这个课题自六十年代中间世以来，在短短的十多年时间内得到了迅速的发展，无论在理论和实际应用方面都达到了相当成熟的程度。这样，计算机辅助电路分析与设计已成为电路理论与技术的一个重要的分支。

目前，在我国高等院校中正普遍地从事该课题的研究，许多高等院校已将该课题列为研究生的课程或高年级学生的选修课，在科研单位和工厂，也正在从事其研究工作和推广其使用。编者编写本书的目的是较为系统地向读者介绍计算机辅助电路分析和设计的理论和方法，以便使读者能够迅速掌握这门先进的技术，为我国的四个现代化的目标服务。

计算机辅助电路分析与设计是以电路理论、计算数学和计算机程序设计这三门学科作为其基础的，或者可以说，它是这三门学科的有机结合与综合运用。在编写本书时，考虑到读者应熟悉经典的电路理论，因而在本书中只着重介绍了与计算机辅助电路分析有关的网络拓扑方面的知识。在数学基础方面，线性代数是贯穿全书的基本数学工具。考虑到线性代数已有多许专门书籍加以介绍，因而本书没有写入，缺乏这方面知识的读者在阅读本书前请先熟悉一下关于向量和矩阵的基本知识。计算数学，特别是线性代数的数值计算方法、常微分方程的数值计算方法和和最优化方法是本书的又一重要数学基础。考虑到目前国内在这方面的教材多以计算数学专业的学生为对象，以工程技术人员为对象的教材较少；另一方面有些计算机辅助电路分析与设计中特别有用的一些计算方法在许多计算数学的教材中尚未得到足够的重视和充分的反映，加之目前中年工程技术人员多缺乏计算数学方面的理论基础，因而本书对于所用的计算数学的方法作了比较详细的介绍。当然由于本书的目的不是研究数学，因而不可能涉及到计算数学的广泛的内容和严格的数学推理和证明，有兴趣的读者可参看专门的书籍。

计算机辅助电路分析与设计，顾名思义，是要通过编写和使用计算程序来完成电路分析与设计的任务的。纵观目前国外的有关教材，在处理电路分析与设计程序方面有两种不同的做法。一种做法是以介绍算法为主，很少介绍程序设计及程序本身；另一种做法是在介绍算法的同时详细地介绍程序。这两种做法各有优缺点：前者可以使学者集中注意力于这门课程的中心内容，而不致于在编写和校正程序上花掉许多时间；后者则能使学者不仅能掌握课程的内容和方法，而且能够付诸实践。作者根据自己的亲身体会比较倾向于后一种做法，因为编写计算程序不仅需要掌握基本的理论和方法，而且要对一切可能出现的情况和细节作全面深入的考虑，它有助于加深对于理论的理解和熟悉其运用。另一方面，电路分析程序既含有一般程序的共同点但又有其特殊性，这一点只有通过编写程序才能体会和掌握。因而本书在介绍理论和方法的同时，也以一定篇幅介绍了计算程序。本书的程序是以 *FORTRAN* 语言编写的，编者希望读者具有 *FORTRAN* 语言程序设计的基本知识。考虑到一个完整的程序往往

篇幅很长在阅读时要花掉很多时间，因而本书中的许多程序以子程序的形式出现，以介绍实现某一种电路分析方法所需的核心子程序，读者可以在此基础上编写出完整的电路分析程序。本书的程序以 *DJS*—6 机的 *FORTRAN* 语言编写和试算通过，在写入本书时，将其输入输出语句改用标准 *FORTRAN*（*FORTRAN*—Ⅳ）书写，以适用于其他类型的计算机。在本书的附录中介绍了美国的 *ASTAP* 程序，以便于读者了解综合性电路分析程序。

计算机辅助电路分析与设计，作为一门课程，国外已有许多教材、近年来国内也编出了一些教材，本书是在参考国内外有关教材的基础上编写的。此外，它作为电路理论和技术的一个分支，目前正在不断地发展。为了使读者了解这方面的情况，本书尽编者可能接触到的资料加入了七十年代以来出现的一些新的技术和方法。本书可以作为大学研究生或高年级学生的教材或参考书，也可供工程技术人员自学使用。由于编者在本课程的教学上尚未积累许多经验，在编写本书时没有从教学的角度（如教学大纲、时数、习题等）作很多考虑，因而它作为一本教材来使用定有许多不足之处，有待于在今后的教学中不断完善。

本书是在南京华东工程学院电工教研室的领导和同志们的支持下编成的。傅冲、曾向秋、俞天锡同志为本书进行了审阅。教研室的许多同志和院绘图室的同志参加了本书的出版工作。本书的上、下册分别由南京邮电学院和华东工程学院印刷厂印制。本书在拟定提纲时曾承南京邮电学院郭祥沄同志提出宝贵意见。编者对上述同志和单位表示感谢。

由于编者的理论水平和实际经验有限，本书在选材、组织及内容阐述等方面定有许多缺点和错误，敬请读者提出宝贵意见。

编　者　识

一九八一年二月

计算机辅助电路分析与设计

上 册 目 录

第一章 概 述……1

§1—1 电路的计算机辅助设计的内容和意义……1

§1—2 电路的计算机辅助分析的过程……3

§1—3 对电路的计算机辅助分析的基本要求……5

第二章 网络图论的基本概念和方法……15

§2—1 网络图的基本概念……15

§2—2 图的矩阵……19

§2—3 电网络方程……24

§2—4 描述支路特性的矩阵和向量……32

§2—5 常用的电路分析方法……41

第三章 线性代数方程组的数值解法……53

§3—1 Gauss 消去法……53

§3—2 主元素 Gauss 消去法……59

§3—3 Gauss—Jordan消去法……70

§3—4 解线性方程组的 LU 分解法……75

§3—5 对称正定系数矩阵方程组的解法……91

第四章 线性直流与正弦交流电路分析……99

§4—1 节点电压法……99

§4—2 直流电路节点电压法分析程序——拓扑法……106

§4—3 直流电路节点电压法分析程序——直接法……118

§4—4 正弦交流电路的节点电压法分析程序……124

§4—5 一些特殊问题的处理方法……136

§4—6 电路的混合型方程组……144

§4—7 正弦交流电路混合型方程组形成子程序……155

第五章 非线性电阻电路分析……163

§5—1 非线性电阻元件……163

§5—2 非线性电阻电路的方程组……169

§5—3 非线性代数方程组的解……177

§5—4 友网络分析法……………………………………………………………………………… 186
§5—5 分析非线性电阻网络的折线法………………………………………………………… 201
§5—6 非线性电阻网络多值解的求法………………………………………………………… 208

第六章 网络的瞬态分析………………………………………………………………………… 213

§6—1 常微分方程的数值解法初步…………………………………………………………… 213
§6—2 数值解的误差……………………………………………………………………………… 221
§6—3 数值解法的稳定性………………………………………………………………………… 226
§6—4 分析线性网络瞬态的混合型方程组…………………………………………………… 229
§6—5 分析线性网络瞬态的友网络方法……………………………………………………… 234
§6—6 一个简单的线性网络瞬态分析程序…………………………………………………… 238
§6—7 时变电源的处理…………………………………………………………………………… 255
§6—8 非线性网络的瞬态分析………………………………………………………………… 257

第七章 网络的状态方程组……………………………………………………………………… 266

§7—1 状态变量法………………………………………………………………………………… 266
§7—2 线性 *RLCM* 网络状态方程组的形成方法……………………………………………… 271
§7—3 特定树及基本割集矩阵的形成方法…………………………………………………… 293
§7—4 含受控源的线性网络的状态方程组…………………………………………………… 306
§7—5 用迭加法形成线性网络的状态方程组………………………………………………… 317
§7—6 非线性网络的状态方程组……………………………………………………………… 336

第八章 线性网络状态方程组的数值解法……………………………………………………… 349

§8—1 线性网络状态方程组的解……………………………………………………………… 349
§8—2 计算 e^{At} 的矩阵标准化法……………………………………………………………… 353
§8—3 计算 e^{At} 的矩阵多项式法……………………………………………………………… 369
§8—4 递推公式…………………………………………………………………………………… 381
§8—5 计算矩阵特征值的乘幂法………………………………………………………………… 389
§8—6 计算矩阵特征值的 Mueller 方法………………………………………………………… 396
§8—7 计算矩阵全部特征值的 *QR* 方法……………………………………………………… 397
§8—8 计算矩阵特征向量的一般方法………………………………………………………… 412
§8—9 状态方程组的频域解……………………………………………………………………… 413
§8—10 *STIFF* 方程组及其解法………………………………………………………………… 420

第一章　概　述

§1—1　电路的计算机辅助设计的内容和意义

电子数字计算机是二十世纪的重大发明之一，是科学技术上的一项卓越成就。电子数字计算机具有运算速度快、精确度高、能存贮信息、能进行逻辑判断、以及能按程序进行自动运算和处理的特点，因而使它在工农业生产、科学研究、军事技术及社会生活各个领域中都得到了广泛的应用。目前，电子数字计算机已经成为现代科学技术的一个必不可少的工具，它的应用程度已成为一个国家科学技术水平高低的一个重要标志。

计算机辅助设计（简称*CAD*—Computer-Aided Design）是电子计算机在工程技术上的一项重要应用，它大大改变了设计方式，加快了设计进程，提高了设计质量。

就电路设计而言，它的典型过程如图1—1—1所示。在设计要求确定以后，设计人员按照自己的经验并参考有关资料，选定一个预期能达到设计要求的电路结构，并确定电路中元件的参量，这一过程称为电路的综合。下一步是对综合好的电路进行分析。如果分析结果达不到原定的技术指标，就需要进行电路结构或元件参量的修改，否则就可以制作实验线路进行测量以验证分析计算是否正确，这是实验分析阶段。如果实验分析通不过，则又得修改原设计，重复上述过程，如果能通过，样品的设计即告完成。从样品设计到正式投入生产，还要经过小批量产品的试生产阶段，在这一阶段中考验设计是否合理和可靠。如果在这一阶段发现产品质量有问题，或是产品合格率低等毛病，则又需重新修改设计，一直到试生产阶段获得通过，一个电路的设计阶段才告基本结束。由此可知，电路的设计过程是一个反复分析、反复测量、反复修改使电路满足设计要求的过程。

传统的电路设计是以典型电路的分析为基础的。设计者根据已有的公式、图表等进行计算，这种方法使得设计者难以跳出固有框框的限制。由于主要依靠手算和简单的计算工具，为了避免分析计算过于复杂而难于实现，一般要对元件的等效电路或模型作大量的近似和简化，并忽略许多寄生参量的影响，从而使分析计算的结果与实际特性往往不一致，需要长时间的实验调试和反复凑试来修改设计。由于手算难以对复杂的电路作统计分析，使得设计者难以规定元件容差的合理数值，结果往往造成电路中元件的互换性差、产品的合格率低等问题，使得试生产阶段难以通过。上述方法还不便于模拟电路中的某些故障，从而不能确保产品的可靠性。这些矛盾随着电子线路的越来越复杂、对于电路的技术要求越来越严、对于设计周期要求越来越短而日益尖锐。对于大规模集成电路的设计，传统的方法已无能为力。

电子计算机的应用大大地改变了电路设计方式。由于计算机的计算速度快存贮容量大，因此可以采用较精确的元件的等效电路或模型，使分析的结果尽量接近于实际特性。用计算机直接模拟电路的功能，使设计者易于摆脱固定的框框而实现电路的最优化设计。用计算机进行设计，便于进行电路的统计分析和故障分析，从而大大提高产品的合格率和可靠性、降低产品的成本。这样，运用电子数字计算机进行电路设计，就可以从原来传统设计主要通过试验电路进行反复凑试的方式转变为主要通过计算机进行分析和最优化的方式。它不仅可以

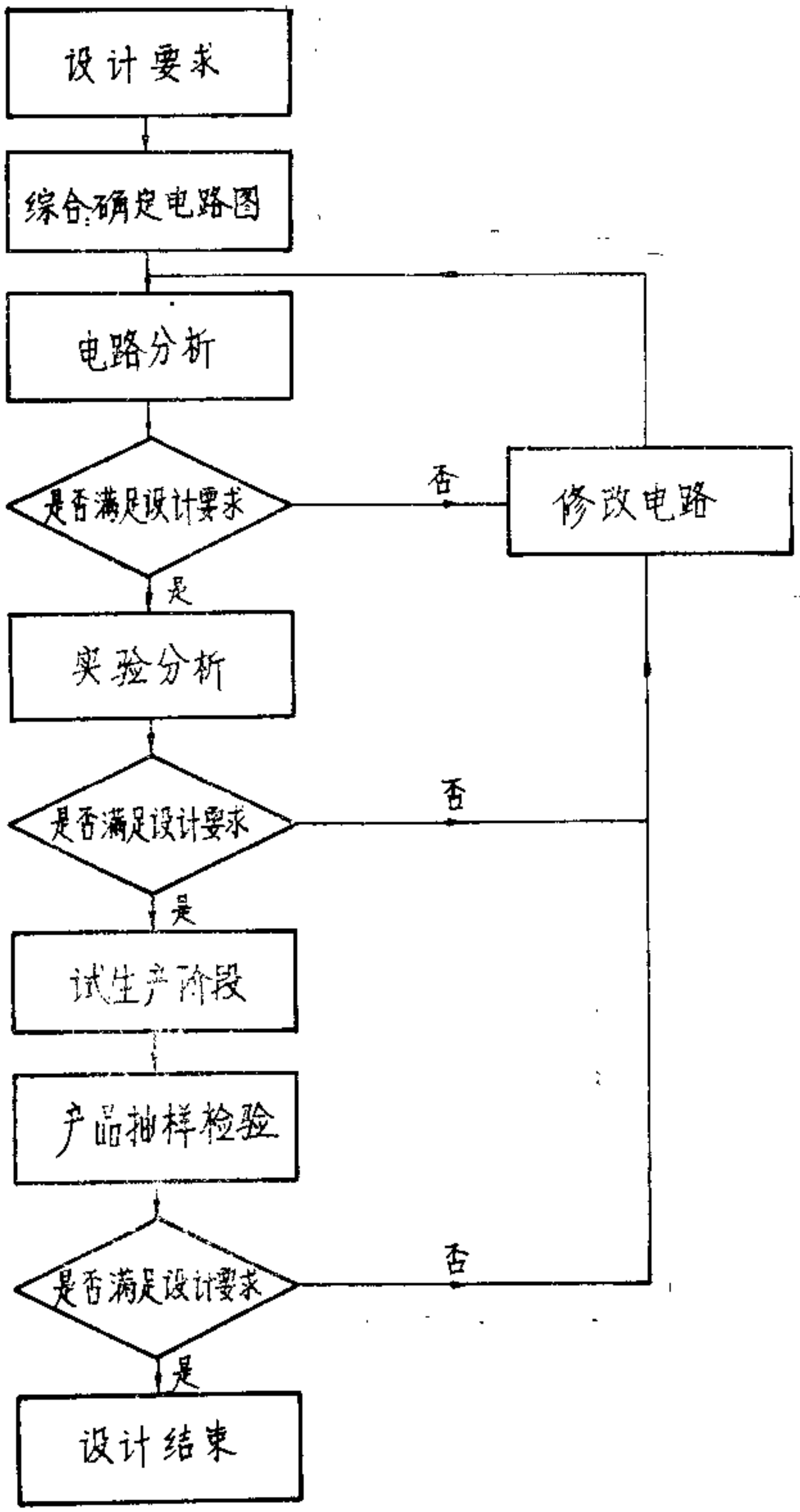
设计要求
综合确定电路图
电路分析
是否满足设计要求
否
修改电路
是
实验分析
是否满足设计要求
否
是
试生产阶段
产品抽样检验
是否满足设计要求
否
是
设计结束

图 1—1—1

大大缩短设计的综合、实验分析和试生产阶段，降低设计费用和加快设计进程，而且可以更加全面、仔细、深入地分析整个设计，从而获得传统方法所无法达到的高质量的设计。

目前，计算机已应用于电路设计的许多阶段。在原理线路设计中，可用它对原理线路作分析计算使之最优化。在原理线路设计成功后，计算机可进行印刷线路板和集成电路的制版和布线设计。在试验测量阶段，计算机可完成对测量数据的处理和分析。但是由于电路种类千变万化，目前计算机还不可能实现对电路的完全自动化设计。一般说来，设计者还必须提供原始的设计方案，然后用计算机来完成上述各项工作。因此，人们把上述过程称为计算机辅助设计。

由上所述，电路的计算机辅助设计的内容是十分丰富的，本书不可能涉及到它的所有方面，仅能就计算机辅助电路分析和最优化设计向读者作一介绍。

§1—2　电路的计算机辅助分析的过程

用计算机进行电路分析的过程如图 1—2—1 所示，它大致经历三个步骤。

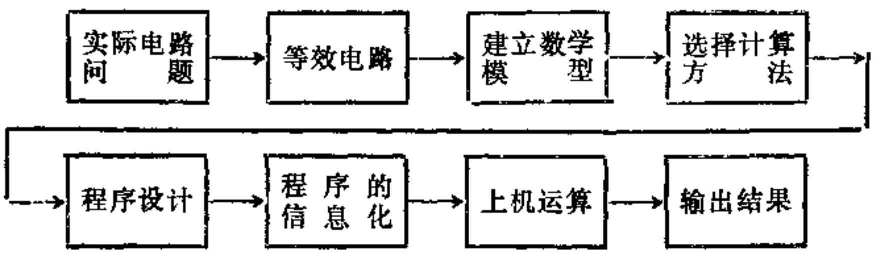

图 1—2—1

一、程序设计的准备阶段

在这一阶段中，完成将实际给定的电路转化为相应的等效电路，建立其数学模型和选择相应的计算方法等项工作。

众所周知，一般的电路分析方法都是以理想元件构成的电路为对象的，因而对于实际电路中的各种元件和器件，首先要根据它们的特性和使用条件，作出其由理想元件构成的等效电路，然后建立描述它们特性的数学关系式，即它们的数学模型。

电路中各种常用的典型元件，如电阻、电感、电容、半导体二极管、双极晶体管、场效应晶体管等，它们的等效电路和数学模型已有许多书籍加以叙述，我们不准备详细讨论这个问题。这里要指出的一点是，为了适应计算机辅助分析的需要，各种元件、器件的等效电路和数学模型和手算分析时所用的会有所不同。例如对于电感、电容这样一类动态元件，在手算分析其瞬态过程时其数学模型是由微分方程来描述的，面用计算机辅助分析时，有时则用相应的差分方程来描述。我们将在今后有关章节中详细讨论这个问题。

在电子线路的分析中，电子元件(如晶体管)的等效电路是很复杂的。如果每次都必须由人来完成从给定电路到等效电路的转换，无疑要耗费大量时间。另一方面，元件的等效电路往往随条件变化（如工作点的变化，温度的变化，非性线元件的电压电流的变化等）而变化，需要在计算时作自动的更换。因而在近年来的电路的计算机辅助设计中，往往把这一工作归为分析程序的一部分，即电路分析程序具有对于典型的电路元件自动转换成它们的等效电路

的能力，人们只需要指明这些元件所有的特性参数即可。不仅如此，许多典型的电路分析程序还附有专门的数据库，贮存各种类型常用元件的典型特性参数。如果所分析的电路中的元件符合该数据库规定的典型参数，那么人们只要指明该元件的类别、型号及少量参数（如温度等），分析程序即可从数据库中调出有关该元件的全部数据，并通过适当的运算建立起相应的等效电路。这无疑会大大地减轻分析的准备工作，给分析者带来极大的方便。

计算方法的选择，是机助分析的关键。这个问题包括两个方面：一是选择适当的电路分析方法，二是选择适当的数值计算方法。随着机助分析的应用和发展，电路分析方法也相应地在发展变化。如早已形成的网络图论的一些方法长期以来在手算分析中很少采用，现在却成为机助分析的主要方法。而在手算分析中常用的一些方法（如等效变换法，Laplace 变换法等）在机助分析中很少采用。早期的电路分析程序，由于某些数值计算方法比较落后，使得它的计算结果（特别是瞬态分析的结果）误差很大，分析者由于不能确信计算结果的可靠性而往往不敢使用这些程序。近年来由于计算数学的发展，使这一问题得到了较好的解决。由上述可知，电路理论和计算数学已成为计算机辅助电路分析的两门基础学科。

二、程序设计

在确定了计算方法后，下一步的工作就是编出执行计算过程的分析程序。

一般的电路分析程序大致应包括如下几个部分：

1. 输入原始数据和对它们进行相应的处理；
2. 建立电路方程组；
3. 解电路方程组；
4. 输出计算结果的数据、曲线等。

设计电路分析程序，通常分为两步进行：第一步是将整个分析计算分解为若干较小的过程，作出其程序框图；第二步是在程序框图的基础上编出源程序。对于较简单的分析程序，源程序可仅由一个程序段即主程序构成；对于较复杂的分析程序，则宜先按照框图编出若干过程（如子程序），然后再编出包含各个过程段的完整的主程序。

目前电路分析程序多是用算法语言写成的。国内比较流行的算法语言有 *FORTRAN*，*ALGOL* 和 *BASIC* 等，究竟使用哪一种语言应根据所使用的计算机来选择，本书中运用 *FORTRAN* 语言编写程序。

应当指出，完全按照*FORTRAN*（或*ALGOL*等）语言规定的格式写成的电路分析程序，在使用中往往感到不很方便，特别是在输入数据阶段更是如此。因而目前比较好的一些综合性电路分析程序，其数据输入和处理程序段往往规定用一种不同于 *FORTRAN* 而便于电路分析设计的格式，即所谓用自由格式输入。采用这种格式输入之后，程序再按照*FORTRAN* 语言的要求进行相应的处理。关于这个问题，读者可以在附录一、二中了解到它的概貌。

除此之外，电路分析程序的编写还有赖于计算机所使用的外部设备。国内目前多数计算机采用纸带输入方式，本书中的程序就是以此为基点编写的。当使用卡片输入方式时，程序应当有所不同（见第四章）。如果计算机备有光笔图形显示器，那就需要有专门的将图形转换成数字信息的程序，关于这个问题的讨论已超出了本书的范围，它是研究计算机软件的工作者的课题。

程序的编制是比较复杂费时的，对于初学计算机辅助分析和设计的人来说往往感到它比

掌握原理要困难得多。这固然与程序设计技术的掌握程度有很大的关系，但经验表明，程序的失败往往是由于对于整个计算过程缺乏周密仔细的分析和考虑所造成的。只有对于所采用的电路分析方法和计算方法熟悉精通，对于分析计算的全过程作精细的考察，一个不漏地考虑到计算过程中可能出现的情况并采取相应的措施才，能编出一个正确的程序。当然一个复杂的程序也往往要需经过多次反复的运算和修改才能获得成功。在这里，细致和耐心是必需的。

三、上机运算

为了使用编写好的程序分析计算具体问题，要需将程序及有关数据信息化后输入计算机，即按照一定的编码系统将源程序及数据编成相应的代码，通过穿孔纸带、穿孔卡片、键盘或光笔输入计算机，计算机内的编译系统将上述源程序翻译成用机器语言表示的目标程序，最后计算机执行目标程序，输出分析结果。

上述计算机辅助电路分析的步骤是就分析的全过程而言的。实际上一旦程序已经编好并证明是有效的，那么其他人就可以使用这个程序进行分析。对于使用程序的人来说，他所需要了解的只是程序的功能及使用的方法，他可以完全不了解程序的原理、结构和采用的方法。这样，对于大多数程序的使用者来说，进行机助分析实在是比进行手算分析还要容易得多的事，何况源程序通常是贮存在计算机的外部设备（如磁带、磁盘）之中，使用者仅需准备有关的数据纸带、卡片等等即可。

§1—3 对电路的计算机辅助分析的基本要求

为了正确地选择计算方法和设计出良好的分析程序，必须明确对电路机助分析的基本要求。本节介绍这些基本要求和达到这些要求的一些一般性的方法，具体的方法将在今后各章中予以讨论。

一、程序的通用性

读者对于常用的电路分析方法一定是很熟悉的。我们从事手算分析时，第一步就是要求对于给定的电路及求解的问题作详细的分析，找出它的特点并选择最简便的计算方法。在这时，问题的特殊性成为主要的注意点，而运用电路的等效变换来简化所分析的电路成为常用的手段。

当我们使用计算机来分析电路时，情况就有所不同。一方面计算机辅助分析的对象一般是较为复杂和大型的电路，不如简单电路易于实现等效变换；另一方面，在计算机辅助分析中，编制程序所花的时间往往比上机运算的时间多得多。因此，与其针对每一个具体电路编写一套分析程序，不如编写一套有广泛适用性的通用程序，使之能解决一般的电路分析问题，从而大大提高机助分析的效率。因而，一般的电路分析程序应具有广泛的通用性。

目前国外已经投入使用的一些综合性的电路分析程序，其功能是很强的。它既能分析直流电路和正弦交流电路，又能作电路的瞬态分析；既能分析线性电路，又能分析非性线电路；既能从事典型参数的分析计算，又能从事统计分析。这样的电路分析程序就具有很强的适应性，能基本上满足各种电路分析的需要。当然就每一个具体的单位和和个人来说，如果

经常遇到的是某一类电路分析问题，那么编制通用于这一类电路分析的程序也是合适的。这种程序往往篇幅较少，易于编写，使用时占用的计算机存贮容量较少，易于在中小型计算机上使用。

二、计算的准确性

计算的准确性是对任何计算的基本要求，对于机助分析自然也不例外，决不可认为计算机本身有准确性高这一特点就可以高枕无忧，事实上必须对于这个问题给予充分的注意。

计算机辅助电路分析和任何工程计算一样，都是有误差的。误差的主要来源如下。

1． 元件模型和参量带来的误差

在计算机辅助分析中，电路元件都要用它的等效电路模型和相应的参量来表示，电路模型的近似程度和参量的准确程度是关系到计算结果准确性的一个原始因素。应当特别注意的是，非线性元件如果作线性化近似处理，则随着工作点的不同等效电路的参量有很大的差异。如果处理不当，计算结果可能与实际情况相差甚远。元件的寄生参量在有些情况下可以忽略，在另一些情况下却变得非常重要。如果在后一种情况下仍然忽略它们的影响，将会造成计算结果的错误。

2． 计算机本身固有的误差

在计算机中，每一个数的有效数字的位数是有限的，它取决于计算机的字长。一般说来，如果计算机中数的尾数部分有 t 位，那么这个数可能具有 2^{-t} 的相对误差。当将数输入计算机和计算机进行运算时都会由于字长有限而产生舍入误差。通常我们输入计算机的是十进制数，而计算机中进行运算的是二进制数。在许多情况下，一个有限位数的十进制小数不能用有限位数的二进制小数准确地表示。这样当机器完成数的十进制至二进制的转换时就不可避免地会带来舍入误差。同样在进行四则运算时也会有相应的问题发生。因此，当使用分析程序进行计算时，必须事先了解计算机的字长。如 DJS—6 计算机字长 48 位，它的尾数的准确度可达 10^{-12}，这对于一般的电路分析计算是可以满意的，而有些微型机，字长只有16位，它的尾数的准确度只有 10^{-8} 左右，这对于计算准确性要求较高的问题显然是不合适的。

考虑到舍入误差的存在，我们在编写程序时对于计算的步骤和方法应当谨慎小心。下面举一个例子说明如果忽视这个问题，计算机的舍入误差可能造成计算出错。

设某计算机的尾数的准确度可达 10^{-5}。现在用幂级数的展开式

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \cdots\cdots$$

来计算 $e^{-5.5}$。这时它的计算过程如下。

$$\begin{aligned} e^{-5.5} = &\quad 1.0000 \\ &-5.5000 \\ &+15.125 \\ &-27.730 \\ &+38.129 \\ &-41.942 \\ &+38.446 \\ &-30.208 \end{aligned}$$

$$
\begin{array}{rr}
 & +20.768 \\
 & -12.692 \\
 & +\ \ 6.9803 \\
 & -\ \ 3.4902 \\
 & +\ \ 1.5997 \\
 & -\ \ \cdots\cdots\cdots \\
= & +\ \ 0.0026363
\end{array}
$$

这样，经过25项求和得到结果为0.0026363，再往下计算结果已经没有变化（到第5位有效数字），但实际上，$e^{-5.5}=0.00408677$，可见计算结果是错误的。

这里计算出错主要是由于机器的舍入误差引起的。由上述过程可见在计算中有8个大于10的数出现，这些数的可能误差是小数点后第三位，它与计算结果的第一位有效数字的数位相同，这样的计算自然无法保证结果的准确性。此外整个计算是在一系列加、减交替的过程中进行的，由很大的数相减得到很小的数又将本来各数已存在的舍入误差放大（指相对误差）。

为了避免上述情况的出现，本例应改用如下方法计算：

$$e^{-5.5}=\frac{1}{e^{5.5}}$$

$$=\frac{1}{1+5.5+15.125+\cdots}$$

$$=0.0040865$$

在使用同一机器的条件下，结果的准确度可达0.007%。

一般说来，用计算机计算时舍入误差的影响和手算时是相似的，所以我们在手算进行数据处理时常用的一些避免由舍入误差引起结果出错的方法在编制程序时都应适当注意采用。

3. 计算方法带来的误差

一般说来，计算方法带来的误差取决于以下三个方面。

1) 方法本身的近似程度。由于用数字计算机进行计算时，一切运算都要化为四则运算，这常常会带来一定的误差。如果方法本身的近似程度好，计算结果的准确性就高；反之，则计算结果的准确性就低。

2) 方法的稳定性

在数值计算中，经常使用迭代公式，这时方法的稳定性对于计算结果的准确性往往有决定作用。为了说明这个问题，我们举一个例子。

设要求计算积分

$$E_n=\int_0^1 x^n e^{x-1}dx \qquad n=1,2,\cdots\cdots$$

用分部积分法

$$\int_0^1 x^n e^{x-1}dx=x^n e^{x-1}\Big|_0^1-\int_0^1 nx^{n-1}e^{x-1}dx$$

或 $$E_n=1-nE_{n-1} \qquad n=2,3,\cdots\cdots$$

已知 $E_1=1/e$，如果计算机的字长可使尾数具有10^{-6}的准确度，则可算得

$$E_1 = 0.367879 \qquad E_2 = 0.264242$$
$$E_3 = 0.207274 \qquad E_4 = 0.170907$$
$$E_5 = 0.145480 \qquad E_6 = 0.127120$$
$$E_7 = 0.110160 \qquad E_8 = 0.118720$$
$$E_9 = -0.0684800$$

但是 $E_9 = \int_0^1 x^9 e^{x-1} dx$ 显然应当是正数（因为被积函数在积分区间内为正），所以现在算出的 E_9 已经出错，再继续迭代下去得到的 E_{10}、E_{11}……自然全是错的。

问题出在什么地方呢？由迭代公式 $E_n = 1 - nE_{n-1}$ 可见，当 $E_1 = 1/e$ 有 10^{-6} 的相对误差，即有约 4×10^{-7} 的绝对误差时，E_2 就将该误差放大 1 倍，而 E_3 再放大 2 倍，一直到 E_9 其误差放大为 $4\times10^{-7}\times9!$ 就成为一个相当大的数字（相对 E_9 本身来说）。这种随着迭代的进行，误差（绝对值）不断被放大的算法称为是不稳定的，因而是不能采用的。

为了消除上述问题，我们将迭代公式改为

$$E_{n-1} = \frac{1 - E_n}{n}$$

这个算法就是稳定的。为了选一个迭代初值，由

$$E_n = \int_0^1 x^n e^{x-1} dx < \int_0^1 x^n dx = \frac{1}{n+1}$$

知当 $n \to \infty$ 时 $E_n \to 0$。例如我们选 $E_{20} = 0$，它最多有 $\frac{1}{21}$ 的误差，由 E_{20} 算得 E_{19} 时，该误差乘了1/20，变为 $\frac{1}{21}\times\frac{1}{20} = 0.0024$。到了 E_{15}，该误差被缩小到小于 4×10^{-8}，已经可以忽略不计了。

3) 所计算的问题对于数据的灵敏度。

某些计算问题的解对于原始数据的灵敏度很高，也就是说原始数据的微小变化会带来解的巨大变化。如果所选用的计算方法要解这一类问题，那么计算结果可能出错。

这种解对于原始数据十分灵敏的一类典型问题是多项式求根。为了说明这个问题，Wilkinson 举了一个例子。

设多项式为

$$P(x) = (x-1)(x-2)\cdots\cdots(x-19)(x-20)$$
$$= x^{20} - 210x^{19} + \cdots\cdots$$

该多项式根为 1，2，……，20。如果现在 x^{19} 的系数 -210 产生了 2^{-23} 的绝对误差变为 $-210+2^{-23}$（它的相对误差小于10^{-9}），那么求出的解将是

1.000000000
2.000000000
3.000000000
4.000000000
4.999999928
6.000006944

6.999697234

8.007267603

8.917250249

20.846908101

$10.095266145 \pm j0.643500904$

$11.793633881 \pm j1.652329728$

$13.992358137 \pm j2.518830070$

$16.730737466 \pm j2.812624894$

$19.502439400 \pm j1.940330347$

由此可见多项式系数的极小变化引起了解的多么巨大的变化。为了说明上述现象的原因，我们来求它的根对系数的灵敏度。为此，将原多项式$P(x)$改写为

$$P(x,\alpha) = x^{20} - \alpha x^{19} + \cdots\cdots$$

该多项式的根满足

$$P(x,\alpha) = 0$$

上式两端对α求导，得

$$\frac{\partial P}{\partial x}\frac{\partial x}{\partial \alpha} + \frac{\partial P}{\partial \alpha} = 0$$

故

$$\frac{\partial x}{\partial \alpha} = -\frac{\dfrac{\partial P}{\partial \alpha}}{\dfrac{\partial P}{\partial x}}$$

$$= \frac{x^{19}}{\sum\limits_{i=1}^{20}\prod\limits_{\substack{j=1\\j\neq i}}^{20}(x-j)}$$

对于不同的根，根对系数α的灵敏度如下表所示。

根	$\frac{\partial x}{\partial \alpha}$	根	$\frac{\partial x}{\partial \alpha}$
1	-8.2×10^{-18}	11	-4.6×10^{7}
2	8.2×10^{-11}	12	2.0×10^{8}
3	-1.6×10^{-6}	13	-6.1×10^{8}
4	2.2×10^{-3}	14	1.3×10^{9}
5	-6.1×10^{-1}	15	-2.1×10^{9}
6	5.8×10	16	2.4×10^{9}
7	-2.5×10^{3}	17	-1.9×10^{9}
8	6.0×10^{4}	18	1.0×10^{9}
9	-8.3×10^{5}	19	-3.1×10^{8}
10	7.6×10^{6}	20	4.3×10^{7}

由上表可见，对于$x>3$，$\left|\frac{\partial x}{\partial \alpha}\right|$很大，因而$\alpha$的微小偏差会使根产生巨大的误差。

正是由于上述原因，所以在程序中应尽量避免将问题转化为先求多项式的系数再求多项式的根。例如早年的一些矩阵求特征值的问题（这个问题对电路分析是很重要的）曾采用过先求矩阵的特征多项式的系数再求其根的方法。由上述分析可知这种算法是不好的，因而本书不作介绍。

三、计算的快速性

对于计算速度的要求，是编制计算机分析程序、选择计算方法的基本出发点之一。虽然电子数字计算机具有高速运行的特点，但是一台通用的高速电子计算机承担的任务十分繁重，单位机时的费用也是十分昂贵的。如果在解每一个题时不注意提高运算速度，那么累计起来在时间上的浪费可能是十分惊人的，在费用上的消耗也将是十分可观的。特别是对于大型电路的分析，运算时间的耗费与节省更不是无关紧要的小事。实际上目前虽然有了每秒钟计算速度达上千万次至上亿次的大型高速数字计算机，但是要完成满足一定瞬态特性大型电路的最优设计仍需几小时至几十小时甚至更多的时间和耗费大量的计算费用，因而提高计算速度仍然是目前正在深入研究的一个课题。

提高计算速度的途径之一是注意计算机中数字运算的特点。在计算机中作四则运算时，加减最快，乘法次之，除法最慢。因而能用加减完成的运算尽量不要变成乘法运算，在某些场合，除法运算也尽量化为乘法运算。

例 1　如果我们要将某一具有 n 个元素 a_1，a_2，……，a_n 的数组中各元素均除以某数 x，那么宜先求出 $t=\frac{1}{x}$，再将 t 与 a_i（$i=1, 2, \cdots\cdots, n$）相乘。这样用 n 次乘法和 1 次除法运算代替 n 次除法运算，当 n 较大时是会节省运算时间的。

例 2　具有如下形式的 n 阶矩阵 B 称为 Pascal 矩阵

$$B=\begin{bmatrix} 1 & 1 & 1 & \cdots & \cdots & \cdots & 1 \\ & 1 & 2 & \cdots & \cdots & \cdots & n-1 \\ & & 1 & \cdots & \cdots & \cdots & \frac{(n-1)(n-2)}{2} \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

其中

$$b_{ij}=\begin{cases} 0 & i>j \\ C_{j-1}^{i-1} & i\leqslant j \quad (C_{j-1}^{0}=1) \end{cases}$$

如一个四阶的 Pascal 矩阵为

$$B=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

设 $\mathbf{x}$ 为 n 维向量，现在要计算向量

$$\mathbf{y} = B\mathbf{x}$$

如果我们按

$$y_{ii} = \sum_{k=i}^{n} b_{ik} x_k$$

计算，则共需作 $\frac{n(n+1)}{2}$ 次乘法运算。

现在改变一下算法。仍以四阶 Pascal 矩阵为例，由

$$B\mathbf{x} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

得

$$\begin{aligned} y_4 &= x_4 \\ y_3 &= x_3 + 3x_4 \\ y_2 &= x_2 + 2x_3 + 3x_4 \\ y_1 &= x_1 + x_2 + x_3 + x_4 \end{aligned}$$

我们用如下 3 步（对于 n 阶矩阵，是 $n-1$ 步）循环来完成上述运算。

先置 $x_i \Rightarrow y_i$ ($i = 1, 2, 3, 4$)，作

1） $y_4 + y_3 \Rightarrow y_3$ （使 $y_3 = x_3 + x_4$）

$y_3 + y_2 \Rightarrow y_2$ （使 $y_2 = x_2 + x_3 + x_4$）

$y_2 + y_1 \Rightarrow y_1$ （使 $y_1 = x_1 + x_2 + x_3 + x_4$）

2） $y_4 + y_3 \Rightarrow y_3$ （使 $y_3 = x_3 + 2x_4$）

$y_3 + y_2 \Rightarrow y_2$ （使 $y_2 = x_2 + 2x_3 + 3x_4$）

3） $y_4 + y_3 \Rightarrow y_3$ （使 $y_3 = x_3 + 3x_4$）

这样，对于 n 阶矩阵是 $n-1$ 步循环，其中各作 $n-1$，$n-2$，……，1 次加法运算，总共作 $\frac{n(n-1)}{2}$ 次加法运算。这种算法显然比前面直接用乘法计算要快得多。

提高计算速度的关键是采用适当的算法。为了衡量一个算法的运算速度，通常采用时间复杂度的概念。对于一个规模为 n 的问题（一个问题的规模是反映该问题大小的一个整数，例如解线性方程组 $A\mathbf{x} = \mathbf{b}$，则 A 矩阵的阶数 n 可以看作是这个问题的规模），解该问题的算法所需用的时间是 n 的函数 $f(n)$。我们称 $f(n)$ 为该算法的时间复杂度。当 n 增长时，$f(n)$ 的极限特性称为该算法的渐近时间复杂度。算法的渐近时间复杂度是衡量一个算法在解规模较大的问题时有效程度的一个性能指标。

显然，对于同一个问题，采用时间复杂度较小的算法可以得到较高的计算速度。实际上，这往往比使用一台更高速的计算机更为有效。例如某算法的时间复杂度为 n^3，如果将计算机的运算速度增大为 10 倍，则在相同时间内能解的问题的规模增大为原来的 2.15（$10^{\frac{1}{3}}$）倍。如果采用一个时间复杂度为 n^2 的算法，那么在 n^3 的时间内，能解问题的规模扩大为原来的 $\sqrt{n}$ 倍，在 n^6 的时间内，能解问题的规模扩大为原来的 n 倍。

有些在手算时行之有效的算法，由于其渐近时间复杂度太大（在手算时问题规模很小因而这个问题不突出）在计算机辅助分析中舍弃不用了。例如手算解线性方程组 $A\mathbf{x}=\mathbf{b}$ 时，经常用 Cramer 法则：

$$x_i=\frac{\Delta_i}{\Delta}\qquad i=1,2,\cdots\cdots,n$$

其中 Δ 为 A 的行列式，Δ_i是用列向量 $\mathbf{b}$ 取代 A 中第 i 列后所得之矩阵的行列式。这样解一个含 n 个未知量的线性方程组，要计算 $n+1$ 个行列式。如果采用手算时常用的 Laplace 展开式来求行列式的值，则当 n 较大时所需作的乘法运算次数约为 $e\cdot(n+1)!$。对于一个含16个未知量的方程组，即使用每秒运算1亿次的计算机来算，1年也算不完，这显然是不合实用的。

降低算法的时间复杂度有许多途径，其中一种简便易行的方法是分割一组合法。有名的快速傅里叶变换（Fast Fourier Transform 简称 FFT）就是这种方法的一个应用（见第十章)。这个方法的基本思想是将一个规模较大的问题分解为若干个（通常是2个）规模较小的问题，求出较小问题的解再通过适当运算组合成原问题的解。这种分割可以反复进行到问题规模不能再小为止。这样处理的结果往往可以得出有效的算法。

举例来说，设有含 $n=2^m$ 个数的数组

$$\{A\}=\{a_1,\ a_2,\ \cdots\cdots,\ a_n\}$$

今欲选出$\{A\}$中之最大元素 A_{max} 和最小元素 A_{min}。

对于这个问题，可采用如下一般算法：

1. 置 $AMAX\Leftarrow a_1$；

2. 自 $i=2$ 至 $i=n$ 将 a_i 与 $AMAX$ 比较，若$AMAX<a_i$，置 $AMAX\Leftarrow a_i$。

这样经过 $n-1$ 次比较后可以选出 A_{max}，然后对于除去该元素后剩下的$\{A\}$中的 $n-1$ 个元素作类似的比较则可选出 A_{min}。这样的算法共需作 $2n-3$ 次比较。

现在我们改用如下算法。将 A 中每两个元素作为一组（如 a_1 与 a_2，a_3 与 a_4 等)，对这 2^{m-1} 个组每组作一次比较即可决定该组的最大元素 A_{imax} 和最小元素 A_{imin}($i=1$, 2, $\cdots\cdots$, 2^{m-1})。设

$$\{A^1_{max}\}=\{A_{1max},\ A_{2max},\ \cdots\cdots,\ A_{kmax}\}$$

$$\{A^1_{min}\}=\{A_{1min},\ A_{2min},\ \cdots\cdots,\ A_{kmin}\}$$

式中 $k=2^{m-1}$，则显然 $A_{max}\in\{A^1_{max}\}$，$A_{min}\in\{A^1_{min}\}$。对于 $\{A^1_{max}\}$ 用同样的方法分割为 2^{m-2} 组元素，经过 2^{m-2} 次比较得到一个含2^{m-2}个元素的数组。这种方法继续下去到最后得到一个只含2个元素的数组，经1次比较选出A_{max}。这样从 $\{A^1_{max}\}$ 到 A_{max} 共需作的比较次数为

$$2^{m-2}+2^{m-1}+\cdots\cdots+1=2^{m-1}-1$$

同样从$\{A^1_{min}\}$用相同的方法找到 A_{min} 也需作相同次数的比较。这样找到 A_{max} 和 A_{min} 的总的比较次数为

$$2^{m-1}+2(2^{m-1}-1)=\frac{n}{2}+2\left(\frac{n}{2}-1\right)=\frac{3}{2}n-2$$