教学用
电路分析程序

杨永萱　编著

四川科学技术出版社

教学用电路分析程序　杨永萱　编著

教学用电路分析程序

杨永萱　编著

四川科学技术出版社

一九八五年·成都

责任编辑：田　霞
封面设计：申　融

教学用电路分析程序

杨永萱　编著

出版：四川科学技术出版社
印刷：内江新华印刷厂
发行：四川省新华书店
开本：787×1092毫米 1/32
印张：6.75　插页：1
字数：143千
印数：1—13,900
版次：1985年9月第一版
印次：1985年9月第一次印刷
书号：15298·102
定价：1.45元

前　言

当今，计算机应用科学迅速发展，电子计算机已广泛应用于科学研究、工程分析和设计、生产、管理的各个领域。在电子工程方面，由于器件、电路、系统和算法之间的关系已非常紧密，界线也逐渐消失，器件、电路和系统的分析与设计已离不开计算机，因此人们越来越重视大专院校中CAD（Computer Aided Design）教学。然而电路的CAD技术又是以电路的CAA（Computer Aided Analysis）为基础的，因此，电路的CAA教学具有重要意义。

电路的CAA、CAD技术是计算机的工程应用中发展得较早、较完善和较成熟的部分。这方面系统而全面的书在国内已出版了好几种。然而，由于电路的CAA技术涉及广泛的知识，诸如电路基本理论、矩阵理论和数值方法、有源电路元件模型、容差和优化理论等，因此，已出版的这些书只宜于作为专设的电路CAA或CAD技术放到高年级或研究生课程中开设。这种情况与微型机已日趋普及，与现代社会已向院校提出较高的计算机教育要求的形势是不相适应的。作者认为，除了专设的计算机基础方面的课程外，在大专院校的各类课程和各个教学环节中，让学生尽早地接触计算机的工程应用以及不断得到现代分析与设计方法的训练，也是加强

计算机应用能力教育的重要方面。本书就是基于这种认识，为了在通常是大学二年级开设的《电路分析基础》、《电工原理》这类课程中引入计算机辅助分析教学所作的尝试工作的总结。

本书假定读者具备带有BASIC语言软件的微型机上机条件；中学或大学一年级学习过BASIC语言；正在学习李瀚荪编《电路分析基础》或类似的电工原理课程。所以，分析方法统一采用节点电位法，电路方程的解法统一采用高斯—约旦消去法（微分方程和非线性方程的解法在简要介绍了数值积分算法、牛顿—拉夫森算法后引导向迭代的直流分析），元件模型只包括独立电源、受控源、线性R、L、C和互感以及非线性电阻等。各通用程序均简述了算法原理和编程方法，提供了完整的程序，并举例说明程序的使用方法，便于自学，便于配合电路分析主教材作CAA教学用。考虑到教学特点，瞬态分析和正弦稳态分析程序着重设计了打印响应曲线，便于一目了然地概观分析结果。

将该书用于教学时可以这样考虑：将通用程序存入计算机外存，供上机调用。配合主教材各部分的教学，将本书相应章节当成实验指导书指定学生自学。第一次上机前教师用较少时间结合所使用的计算机机型介绍一些实用的操作和编辑方法。让学生自选题目并准备好输入数据，上机前教师作检查，下机时要求交部分结果摘抄或部分结果打印件，并引导学生对分析结果进行讨论，以求促进基本理论的教学。这样做的出发点是，除了开辟10学时左右的上机时间外（相当于另辟一类实验课），争取不增加或只少量增加教学学时，然而能够提高学生学习基本理论的兴趣，开发现代分析方法的

基本训练，而且为学习CAD技术打下基础。

本书提供的五个基本通用程序均在S/09微型计算机上调试成功。各程序所占的计算机内存容量分别为：直流分析程序DCAP约4K；固定步长瞬态分析程序TSAP约为6K；自动选步长瞬态分析程序ZDAP约为8K；交流分析程序ACAP约为12K；非线性电阻电路分析程序FXAP约为7K。

在研制程序和成书过程中，作者曾得到桂林电子工业学院裘明信老师，我院关本康、张其劭、虞厥邦等老师的支持和鼓励。承成都科技大学肖可达、史济民老师审阅了文稿并提供了宝贵意见。袁秀琼、张昌兵、唐东、郑永超和万晓东等同志为调试部分程序付出了许多劳务，还得到王治伦、张子康、李爱芳等同志的许多帮助，谨一并致以衷心的谢意。

由于作者水平有限，编写时间仓促，错误和不妥之处一定不少，诚挚地希望读者赐教。

作　者

1984年11月于成都电讯工程学院

目　　录

第一章　线性代数方程的求解

§1—1　概　述

线性时不变的电阻性网络，不论用节点分析、网孔分析、回路分析还是割集分析，均可写出一组线性代数方程来。而且，用数值方法求解非代数方程的电路方程时，往往也使问题最终地归结为求解一个线性代数方程组。因此，我们首先简单介绍怎样用计算机来求线性代数方程组的解。

线性代数方程组的解法可大致分为直接法和迭代法两大类。所谓直接法是指只借助于有限步初等算术运算就可以给出问题的解的方法，并以此区别于迭代法。近年来，人们往往更倾向于使用直接解法，而且直接解法在解具有大量零元的大型稀疏矩阵方面有了新的发展。这里主要从便于配合电路分析课程的教学出发，仅介绍一种最简单而又最常用的一种直接法，即消去法。这是一种古老的方法，然而用在计算机上依然十分有效。

为了一般起见，让我们讨论n阶线性方程组：

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases} \tag{1—1}$$

此方程可以写成矩阵形式：

$$[A][x]=[b] \tag{1—2}$$

其中

$$[A]=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

$$[x]=[x_1 \ \ x_2 \ \cdots \ x_n]^T$$

$$[b]=[b_1 \ \ b_2 \ \cdots \ b_n]^T$$

如果方程（1—1）是描写网络的方程，则$[A]$、$[x]$、$[b]$可分别称为网络的系数矩阵、响应列向量和激励列向量。若矩阵$[A]$非奇异，则方程（1—1）有唯一解。

以Δ表示系数行列式$\det[A]$，以Δ_j表示把Δ中第j列换成$[b]$后所得的n阶行列式，根据Cramer法则，只要$\Delta \neq 0$，可以得到满足方程（1—1）的唯一解：

$$x_j=\frac{\Delta_j}{\Delta}, \quad j=1,\ 2,\ \cdots n \tag{1—3}$$

但是用Cramer法则求解线性方程组，要算$(n+1)$个n阶行列式的值，总共需要$n!(n-1)(n+1)$次乘法，计算量很大。例如，求解20阶左右不算太大的方程组，即使用每秒一千万次的计算机也要工作三百多万年才行。这是一种不现实的算法。因此需要寻求其它求解方法。

§1—2　高斯消去法

一种有效的求解向量$[x]$的方法是大家熟知的Gauss消去法。它实际上就是代数课本里讲的加减消去法。这个方法

是以保持方程（1—1）中每个方程的等价性的初等变换为基础，即用一个非零的数乘一方程以及用一个数乘一方程加到另一方程上，并不影响方程的解。

为了回顾这一方法，给出一个数值例：

$$\begin{cases} 3x_1+2x_2+x_3=6 \\ 2x_1+2x_2+2x_3=4 \\ 4x_1-2x_2-2x_3=2 \end{cases} \qquad (1—4)$$

写成矩阵的形式是：

$$\begin{array}{l} ①行 \\ ②行 \\ ③行 \end{array} \begin{pmatrix} 3 & 2 & 1 \\ 2 & 2 & 2 \\ 4 & -2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \\ 2 \end{pmatrix} \qquad (1—5)$$

对它施行第一次变换，使a_{11}归一并使a_{21}、a_{31}消去。为了便于阅读，将变换手续注在式子左边。第一次变换的结果是

$$\begin{array}{l} \frac{1}{3}\times①\to①' \\[2ex] ②-2\times①'\to②' \\[2ex] ③-4\times①'\to③' \end{array} \begin{pmatrix} 1 & \frac{2}{3} & \frac{1}{3} \\[1ex] 0 & \frac{2}{3} & \frac{4}{5} \\[1ex] 0 & -\frac{14}{3} & -\frac{10}{3} \end{pmatrix} \begin{pmatrix} x_1 \\[1ex] x_2 \\[1ex] x_3 \end{pmatrix} = \begin{pmatrix} 2 \\[1ex] 0 \\[1ex] 16 \end{pmatrix} \qquad (1—6)$$

第二次，将式(1—6)中的$a_{22}=2/3$归一并消去$a_{32}=-14/3$，即变换成

$$\begin{array}{r} ①' \\[2ex] \frac{3}{2}\times②'\to②'' \\[2ex] ③'+\frac{14}{3}\times②''\to③'' \end{array} \begin{pmatrix} 1 & \frac{2}{3} & \frac{1}{3} \\[1ex] 0 & 1 & 2 \\[1ex] 0 & 0 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\[1ex] x_2 \\[1ex] x_3 \end{pmatrix} = \begin{pmatrix} 2 \\[1ex] 0 \\[1ex] -6 \end{pmatrix} \qquad (1—7)$$

类似，再进行一次变换得：

$$
\begin{array}{r}
①' \\ ②'' \\ \frac{1}{6}\times③''\to③'''
\end{array}
\begin{pmatrix} 1 & \frac{2}{3} & \frac{1}{3} \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}
=
\begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix}
\qquad (1-8)
$$

当主对角线的元素均已归一时，消去过程结束，接着便回代，即

$$x_3 = -1$$

$$x_2 = 0 - 2\times(-1) = 2$$

$$x_1 = 2 - \frac{2}{3}\times 2 - \frac{1}{3}\times(-1) = 1$$

到此，所有未知数均已求得。

在进行上述过程时，可能遇到的唯一困难是用作除数的主对角线元素太小或等于零。这时把适当的二行交换一下可避免这一点。如果主对角线和它下面的数都为零，这方程就不能解，其原始方程不独立，不存在唯一的解。

Gauss 消去法是求解线性方程组的最简单的方法，方程阶数n很大时，所需的乘法和减法运算的次数大约是$n^3/3$，除法次数是$n^2/2$。

§1—3 高斯—约旦消去法

在Gauss消去法中，容易想到，在第二次消元时，除把第二列主对角线元素下面的元素消为零外，同时也

可把同一列上面的元素消为零，即把前述式(1—7)改造成为

$$
\begin{array}{r}
①' - \dfrac{2}{3}\times ②'' \to ①'' \\
\dfrac{3}{2}\times ②' \to ②'' \\
③' - \left(-\dfrac{14}{3}\right)\times ②'' \to ③''
\end{array}
\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \\ 0 & 0 & 6 \end{pmatrix}
\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}
=
\begin{pmatrix} 2 \\ 0 \\ -6 \end{pmatrix}
\quad (1—9)
$$

同样，在第三次消元时，把第三列主对角线元素上面和下面的元素（此数值例中无下面的元素）均消为零，于是式(1—8)改造成为：

$$
\begin{array}{r}
①'' - (-1)\times ③'' \to ①''' \\
②'' - 2\times ③'' \to ②''' \\
\dfrac{1}{6}\times ③'' \to ③'''
\end{array}
\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}
=
\begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}
\quad (1—10)
$$

这种消元方法就是Gauss—Jordan消去法，它可以直接把未知量计算出来而不必利用回代。

我们再考察一下n阶线性代数方程的矩阵通式(1—2)。如果把方程右边的向量〔b〕写到系数矩阵〔A〕的第n＋1列上，写成

$$
[A_N] = [A \vdots b] = \left(\begin{array}{cccc:c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \cdots & \cdots & \cdots & \cdots & \cdot \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right) \quad (1—11)
$$

〔A_N〕称之为〔A〕的增广矩阵。可以看出，上面讨论的两种消元法，实际上只要对增广矩阵〔A_N〕实施n次归一和消元

的步骤，变换成

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & \vdots & b_1^{(n)} \\ 0 & 1 & 0 & \cdots & 0 & \vdots & b_2^{(n)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \vdots & \cdot \\ 0 & 0 & 0 & \cdots & 1 & \vdots & b_n^{(n)} \end{pmatrix} \tag{1—12}$$

的形式，则第$n+1$列上的$b_1^{(n)}$、$b_2^{(n)}$、$b_n^{(n)}$即为方程(1—2)的解。

最后，需要说明一个问题。在实施第 i 个归一步骤时，总是利用主对角线的元素a_{ii}做除法。如果a_{ii}的绝对值太小或为零，计算过程就会引入很大的舍入误差或不能进行①。为了改善计算结果，可以选取绝对值最大的元素作为所谓的主元。设第 i 次归一消元时主元在第k个方程（行），即

$$|a_{ki}^{(i-1)}| = \max_{i\leqslant l\leqslant n} |a_{li}^{(i-1)}|,\quad k\geqslant i$$

上标（$i-1$）表示第（$i-1$）次消元的结果。现在我们把第k行与第 i 行互换位置，然后再着手消元，这一手续称为选主元。通常的做法是按列选主元，此时消去法就称为列主元消去法。

例　设$[A_N]$为

$$\begin{pmatrix} 2 & -1 & 3 & 1 \\ 4 & 2 & 5 & 4 \\ 1 & 2 & 0 & 7 \end{pmatrix}$$

对这一增广矩阵施行Gauss—Jordan消元手续。

① 舍入误差问题请参阅参考文献〔3〕第2.7节。

显然，第一个主元素为第 2 行的 4 。先将第 1 行和第 2 行对调位置

$$1、2行易位\rightarrow\begin{pmatrix}4 & 2 & 5 & 4\\ 2 & -1 & 3 & 1\\ 1 & 2 & 0 & 7\end{pmatrix}$$

$$第 1 步消元\rightarrow\begin{pmatrix}1 & 0.5 & 1.25 & 1\\ 0 & \boxed{-2} & 0.5 & -1\\ 0 & 1.5 & -1.25 & 6\end{pmatrix}第 2 个主元为-2$$

$$第 2 步消元\rightarrow\begin{pmatrix}1 & 0 & 0.375 & 0.75\\ 0 & 1 & -0.25 & 0.5\\ 0 & 0 & \boxed{-0.875} & 5.25\end{pmatrix}第3个主元为-0.875$$

$$第三步消元\rightarrow\begin{pmatrix}1 & 0 & 0 & 9\\ 0 & 1 & 0 & -\\ 0 & 0 & 1 & -6\end{pmatrix}$$

如果此增广矩阵各行对应的未知数为x_1、x_2和x_3，则解为

$$x_1=9,\ x_2=-1,\ x_3=-6$$

§1—4　高斯—约旦法程序

Gauss—Jordan列主元消去法的程序框图见图1—1。用BASIC语言编写的子程序（本书的所有程序均用BASIC语言编写，后面不再一一说明）见程序1—1。

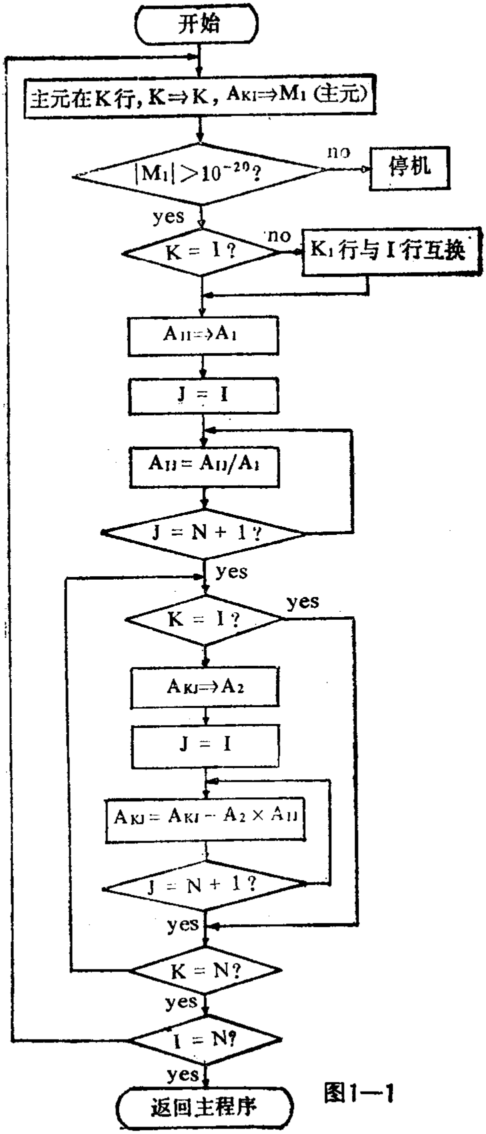
开始
主元在K行，K⇒K，AKI⇒MI（主元）
|MI|>10^-20?
no
停机
yes
K = I?
no
KI行与I行互换
AII⇒AI
J = I
AIJ = AIJ/AI
J = N + 1?
yes
K = I?
yes
AKI⇒A2
J = I
AKJ = AKJ - A2 × AIJ
J = N + 1?
yes
K = N?
yes
I = N?
yes
返回主程序

图1—1

程序1—1　Gauss—Jordan法子程序①

```
6500 for I=1 to N
6525 M1=0
6530 for K=I to N
6535 if abs(A(K, I))<=abs(M1) then 6550
6540 M1=A(K, I)
6545 K1=K
6550 next K
6555 if abs(M1)> 1E-20 then 6570
6560 print 'No unique solution'
6565 stop
6570 if K1=I then 6600
6575 for J=I to N1
6580 A1=A(I, J)
6585 A(I, J)=A(K1, J)
6590 A(K1, J)=A1
6595 next J
6600 A1=A(I, I)
6605 for J=I to N1
6610 A(I, J)=A(I, J)/A1
6615 next J
6620 for K=1 to N
6625 if K=I then 6650
6630 A2=A(K, I)
6635 for J=I to N1
```

① 程序均在S/09微型机上调试。该机的BASIC版本赋值语句中的"LET"可以省略不写，例如LET　$M_1=0$可以写成$M_1=0$。程序中$N_1=N+1$。