



高等学校教材

# 计算结构力学

——杆件结构

钟万勰 丁殿明 程耿东



高等教育出版社

# 前 言

杆件计算结构力学是计算结构力学的重要组成部分。众所周知，结构力学始终是与计算密切相关的，但是自从有了电子计算机后，繁重的计算工作可用计算机进行，这促使了结构力学的面貌发生重大的变化。本门教材就是把传统的结构力学理论和高速数字计算机的计算方法紧密结合起来，为工程结构计算提供有力的工具。

本书的特点是把经典杆件结构力学的基本概念、矩阵位移法的现代理论以及在微型计算机上结构化程序设计思想和数据结构知识融合在一起。作者也非常注意编制程序的技巧，其目的不仅为了分析复杂结构，而且也是为了用小型机，特别是微型机来解决大问题。书中附有从简单的平面桁架程序到复杂的空间刚架等一系列杆件结构程序。这些程序，结构清楚。可读性好、功能相当完备，并具有强有力的前后处理功能。例如：它可将结构图形的三视图在屏幕上显示或在普通的打印机上绘制，并自动标注结点和杆件号码。整个程序是按模块化方式组成的。读者既可使用它们去解决大量的工程实际问题，也可容易地修改它们，以适应特殊问题的需要，如计算机绘图和辅助设计等。

本书编著者从七十年代以来，致力于计算结构力学软件的研究工作，编制了许多著名的程序。如大型结构分析程序系统——JIGFEX，多单元多工况结构分析程序系统——DDJ—W，结构优化程序系统——DDDU，钢筋混凝土平面框架计算机辅助设计程序系统——FCAD等。这些程序不但在国内解决了一批国民经济、国防建设中重大生产实际问题，而且也远销国外。本教材反应了编制这些程序系统的基本思路，是一本计算结构力学的基本教材，也是作者多年科研成果的部分总结。

本书所介绍的程序是用当今世界上使用的MS—PASCAL语言所编写，它是一种在IBM-PC及其兼容机(如我国的长城0520系列机)等各类微机上均能实现的程序设计语言，其特点是功能强、使用方便等。

本书编著者为钟万颢、丁殿明、程耿东。

本书由寿楠椿、卓家寿、匡文起三位审阅，他们对本书提出了宝贵意见；钱令希教授给予了热情指导；顾元宪、史奇二位曾协助调试部分程序，在此，我们表示感谢。

大连理工大学

钟万颢 丁殿明 程耿东

## 内 容 提 要

本书以结构力学基本原理为主体,将经典结构力学的概念,矩阵位移法的近代理论,结构化程序设计的思想方法,数据结构的处理和工程有关知识等融合一起,结合我国情况采用当今世界上使用的 PASCAL 高级语言,在 IBM-PC 及其兼容微型计算机上介绍编制结构分析程序的原理、方法和技巧。在书中给出具有前后处理功能较强的杆件结构系列程序,这些程序结构清晰、简明易懂,实用性强,计算方便。

本书共分十章。为了教学需要和自学方便,书中附有大量习题和部分答案,并附有附录线性代数方程组的计算机算法。

全书采用国际单位制(SI 单位)。

本书可作为土建、力学等专业的教材,也可作为水工结构、造船、起重机械等专业大学生、研究生、教师及有关工程技术人员的参考用书。

责任编辑 余美茵

高等学校教材  
**计算结构力学**  
——杆件结构

钟万勰 丁殿明 程耿东

高等教育出版社出版  
新华书店北京发行所发行  
人民教育出版社印刷厂印装

开本 787×1092 1/16 印张 24.75 字数 560 000

1989年3月第1版·1990年3月第1次印刷

印数 0001—2,010

ISBN7-04-002520-5/TB·149

定价 5.50 元

# 目 录

概述.....	1	§ 5-2 矩阵位移法算例.....	94
<b>第一章 计算结构力学与结构化程序设计</b> .....	<b>2</b>	§ 5-3 最小总势能原理的应用.....	98
<b>设计</b> .....	<b>2</b>	§ 5-4 矩阵位移法的求解步骤.....	99
§ 1-1 杆件结构有限元的发展概述.....	2	§ 5-5 结构计算简图的数据结构.....	99
§ 1-2 计算结构力学.....	3	§ 5-6 位移未知数的确定.....	102
§ 1-3 结构分析程序设计的原则.....	4	§ 5-7 单根杆件的分析.....	103
§ 1-4 结构化程序设计语言——PASCAL语言.....	13	§ 5-8 结构总刚度矩阵的形成.....	106
§ 1-5 程序质量的衡量.....	22	§ 5-9 杆件内力的计算.....	111
<b>第二章 杆件结构基本概念</b> .....	<b>24</b>	§ 5-10 能量原理和矩阵位移法.....	112
§ 2-1 引言.....	24	§ 5-11 习题.....	114
§ 2-2 结构力学的研究对象和基本任务.....	24	<b>第六章 平面桁架计算程序的编制</b> .....	<b>116</b>
§ 2-3 结构的计算力学模型.....	25	§ 6-1 概述.....	116
§ 2-4 杆件结构的分类.....	28	§ 6-2 程序元语言表示.....	116
<b>第三章 杆件结构的机动分析及计算机算法</b> .....	<b>30</b>	§ 6-3 总体数据结构设计.....	118
§ 3-1 机动分析的目的.....	30	§ 6-4 自由格式数据输入.....	120
§ 3-2 平面体系的自由度.....	31	§ 6-5 SKAIN 的功能设计与模块划分.....	121
§ 3-3 几何不变体系的组成基本法则.....	34	§ 6-6 杆件分刚度矩阵的计算.....	125
§ 3-4 结构的机动分析示例.....	37	§ 6-7 结构总刚度矩阵的组装.....	127
§ 3-5 应用计算机进行机动分析.....	39	§ 6-8 结构正则方程式的求解.....	140
§ 3-6 习题.....	44	§ 6-9 杆件内力的计算过程.....	143
<b>第四章 杆件结构基本理论</b> .....	<b>46</b>	§ 6-10 平面桁架计算源程序——TRUSS·PAS.....	145
§ 4-1 静定结构与超静定结构.....	46	§ 6-11 习题.....	148
§ 4-2 叠加原理.....	49	<b>第七章 平面桁架计算程序的改进</b> .....	<b>149</b>
§ 4-3 结构分析的力法和位移法.....	50	§ 7-1 概述.....	149
§ 4-4 最小总势能原理.....	53	§ 7-2 结点位移规格数的概念.....	150
* § 4-5 最小总势能原理应用举例.....	57	§ 7-3 结构独立位移总数的确定.....	151
* § 4-6 瑞利-里兹法.....	61	§ 7-4 结点对号信息的形成与提取.....	153
§ 4-7 最小总余能原理.....	65	§ 7-5 “对号入座”过程的形成.....	154
§ 4-8 位移计算.....	71	§ 7-6 用“紧缩法”输入数据.....	154
§ 4-9 互等性定理.....	84	§ 7-7 “紧缩法”输入数据方式的程序实现.....	160
§ 4-10 影响线的概念及其定理.....	87	§ 7-8 总外力向量的形成.....	161
§ 4-11 习题.....	90	§ 7-9 改进后的平面桁架源程序.....	162
<b>第五章 平面桁架程序的计算原理</b> .....	<b>93</b>	§ 7-10 程序的编辑、编译、连接与运行.....	169
§ 5-1 矩阵位移法.....	93	§ 7-11 程序的分块独立编译.....	170
		§ 7-12 微机操作及计算结果输出.....	174
		§ 7-13 空间桁架程序的编制.....	179

§ 7-14	空间桁架计算源程序	182	§ 9-6	杆件两端刚性区域的影响	262
§ 7-15	桁架程序的灵活应用	187	§ 9-7	改进的平面刚架源程序—— FRAME2·PAS	275
§ 7-16	空间桁架程序使用说明与例题	191	§ 9-8	平面刚架程序的数据文件	283
§ 7-17	习题	203	§ 9-9	程序的灵活应用	287
<b>第八章</b>	<b>平面刚架计算程序的编制</b>	<b>206</b>	§ 9-10	刚架结构对称性的利用	290
§ 8-1	概述	206	§ 9-11	具有铰接点的平面刚架	291
§ 8-2	计算方法和程序功能的元语言表示	206	§ 9-12	平面刚架算例	298
§ 8-3	总体数据结构的设计	209	§ 9-13	平面板架的计算原理	310
§ 8-4	描述刚架的数据格式	211	§ 9-14	平面板架计算源程序	312
§ 8-5	单根杆件转角位移方程式的建立	215	§ 9-15	平面板架程序的数据文件	320
§ 8-6	刚架轴向变形的影响	220	§ 9-16	平面板架算例	321
§ 8-7	关于单根杆件刚度矩阵讨论	222	§ 9-17	习题	323
§ 8-8	单根杆件在全局坐标系中的刚度 矩阵	224	<b>第十章</b>	<b>空间刚架的计算</b>	<b>327</b>
§ 8-9	计算单根杆件刚度矩阵的过程	227	§ 10-1	概述	327
§ 8-10	结构的总刚度矩阵及其组装	229	§ 10-2	描述空间刚架结构的本体数据	328
§ 8-11	刚架总外力向量的形成	234	§ 10-3	空间杆件弹性区域的单元刚度 矩阵	330
§ 8-12	求解刚架的结点位移	240	§ 10-4	空间杆单元的坐标变换	332
§ 8-13	计算杆件各截面的内力	240	§ 10-5	弹性区域刚度矩阵向单元刚度矩阵 的转换	341
§ 8-14	平面刚架源程序——FRAME·PAS	244	§ 10-6	总刚度矩阵的组装及其三角化	342
§ 8-15	算例	248	§ 10-7	外力的描述	343
§ 8-16	计算结果的校核	250	§ 10-8	总外力向量的形成及位移求解	343
* § 8-17	杆件结构影响线的一个统一算法	251	§ 10-9	杆件截面内力的计算	345
§ 8-18	习题	253	§ 10-10	空间刚架计算源程序	346
<b>第九章</b>	<b>平面刚架程序的改进及平面板架 (交叉梁系)的计算</b>	<b>255</b>	§ 10-11	空间刚架程序数据文件的内容	358
§ 9-1	概述	255	§ 10-12	空间刚架算例	361
§ 9-2	结点位移规格数及对号信息	255	§ 10-13	习题	368
§ 9-3	杆件单元表示方法	256	<b>部分答案</b>		<b>370</b>
§ 9-4	剪切变形的影响	257	<b>参考文献</b>		<b>371</b>
§ 9-5	单元刚度矩阵过程	260	<b>附录</b>	<b>线性代数方程组的计算机算法</b>	<b>372</b>

## 概 述

将经典的力学理论与近代的矩阵位移法紧密结合,在微型计算机上如何编制杆件结构计算机程序,是本教材着重介绍的内容。

本书第一章中概括介绍杆件计算结构力学的有关内容,结构化程序设计的基本原则及 PASCAL 语言特点等。第二章至第四章中介绍结构分析中的基本概念和理论,这些内容是学习本书必须首先掌握的。第五章中,通过对简单平面桁架的分析,介绍了矩阵位移法的基本概念。这些概念对希望掌握矩阵位移法,有限元法的读者是十分重要的。第六章,仍以平面桁架为例,具体地讨论了如何在微型计算机上实现矩阵位移法,介绍了程序有关数据结构,数据输入的自由格式,总刚度矩阵的组装及存储,位移求解,内力计算,并给出一个最基本的平面桁架程序,牢固地掌握这些内容是非常重要的。第七章是第六章内容的继续,着重介绍如何改进、补充、提高、完善基本程序,并结合 IBM—PC 微型机特点介绍程序的分块编译。精读本章改进后的平面桁架计算程序,对理解以后各章中较为复杂的程序是十分有益的,也是学习本书内容的第一次飞跃。第八章介绍平面刚架程序的编制原理和方法,它与第五章的内容一样,都是最基本和最重要的。作为第八章内容的继续和引深,第九章着重介绍程序设计的一些技巧和处理不同结构的方法,是学习本书内容的第二次飞跃。第十章介绍了空间刚架的计算程序的编制原理和方法。为了考虑偏心杆件的影响,我们引入了四个坐标系统,使问题显得稍为复杂。但是只要掌握了第九章的内容,学习本章是不会困难的。

本书给出了五个实用程序(平面桁架、空间桁架、平板架、平面刚架和空间刚架)及其使用说明和例题,可以供读者直接用来解决实际问题。这些程序虽然只适用于 IBM—PC 微型计算机及其兼容系统所属各类微机,但其原理具有普遍意义。对其它不同类型的微机系统,稍加改造也是适用的。为了帮助读者顺利阅读程序,在程序中都加上了适当的注释。利用微机还可以将结构计算简图、变形图和内力图形象地显示在屏幕上或用绘图机画在图纸上。限于篇幅本书对此不作介绍。

附录中结合本书内容,着重介绍求解线性代数方程组的主要解法——带状阵的 LDLT 法和变带宽的 LDLT 法,并给出用 ALGOL—60 语言和 PASCAL 语言编写的相应计算程序,以便初学者对比这两种语言的差异。

关于学习各章节的学时分配,教师可按学生的专业性质灵活掌握,书中带“\*”号内容可供不同专业的需要选用。根据我们多年的经验,对初学结构力学的学生总学时可定为 90 学时左右,而对学习过经典结构力学的学生,总学时可为 60 学时左右。

最后强调指出,读者自己修改和编写计算程序,并在计算机上操作是牢固掌握本书内容极其重要的环节。

# 第一章 计算结构力学与结构化程序设计

## § 1-1 杆件结构有限元的发展概述

古代的工程结构主要是人们根据建筑累积的经验和粗略的估计来建造的。实际上,结构力学一直到十九世纪中叶才从力学中划分出来,而成为一门独立的学科。十九世纪后半期,钢结构被广泛应用。结构计算成为结构设计的必要步骤,结构计算的理论得到了很大的进展,结构变形和超静定结构的一般理论也建立起来了。英国学者麦克斯韦(Maxwell, J. C.)在1864年的论文中发表了超静定结构的力法方程。意大利学者卡斯蒂里亚诺(Castigliano, A.)在1879年的著作中论述了利用变形势能求结构位移和计算超静定结构的理论。约在同期,德国学者莫尔(Mohr, O.)发展了利用虚位移原理求位移的一般理论。他们的工作为杆件结构分析奠定了基础。

然而,有关结构矩阵分析方法论的概念,则在此后八十年左右才形成,它们成为有限元法的前奏。这一段的发展在今天看来可以说是很慢。其原因在很大程度上是由于在实际计算中,人们只能求解很有限的若干个未知数的代数联立方程。因此,工程师们的注意力花费在平面桁架与框架的计算上,而且一般都是用力法进行的。直到1932年Cross, H. 发展了刚架分析的力矩分配法,使结构分析的数值方法有了一个飞跃,才能够计算较为复杂一些的结构。力矩分配法统治了20多年,在我国也有很大的发展<sup>[1]</sup>。

电子计算机在40年代后期已经出现。开始阶段,在五十年代中期,在飞机结构的分析中将一个连续部件看成是很多离散的元素组合,然后用杆件结构力学中位移法(变位法)的概念分析这些元素连接点的内力,杆件本身就是一种天然的有限元组合模型。把连续体离散化成人工有限元组合,这个思想并不新鲜,但是,在杆系力学的理论与电子计算技术相结合后,立刻显示了有限元法的威力,获得很大的成功。

从五十年代初开始,西德的Argyris, J. H. 采用能量原理、矩阵分析方法分析组合结构,1956年Turner, M. J. 及Clough, R. W. 等人第一次正式提出了有限元的名词,并给出了最基本的平面单元,成为有限元法开始确立的标志。因此人们往往就认为有限元法由此开始。但如果认为有限元法的主要概念都从这时开始也不是很妥当的。一些关键性的要点在1943年已由Courant, R. <sup>[2]</sup>表达了。参考文献[2]中所述的方法并不单纯限于结构分析的内容,因而它也是特别重要的。

五十年代中叶之后,有限元分析大发展中有几个重要的阶段可以从Zienkiewicz, O. C. <sup>[3]</sup>的总结性文章中看到。人们首先在杆件及平面应力(应变)问题中有了发展,接着很快建立了块元、板弯曲元、薄壳元、厚壳元以及其它结构形式的单元。当这些单元在线弹性静力学问题中确立之后,又发展到了动力反应,稳定性,材料或几何非线性问题的各方面。显然,仅有有限元理论及



其公式表示是不行的,而且还要具体将课题计算出来为工程师或其它方面所用。因此,随着有限元分析的发展,也推进了计算机有限元通用程序的发展。我们绝不能低估发展有限元程序工作的重要性,当我们一方面致力于开发新单元时,另一方面就应关注程序的实现。发达国家通常花费并一再投入数以百万计的资金用于有限元通用程序的开发。虽然耗资巨大,但是它对实际工作产生巨大的效益,因此是完全值得的。

## § 1-2 计算结构力学

在工程力学中,结构力学是个极重要的组成部分。任何工程都要设计和建造工程结构,都有结构力学的问题需要解决。现代化的工程中,结构日益复杂,对力学的要求也愈来愈高。近一、二十年来,结构力学发生了深刻的变化,导致这些变化的诸因素中,电子计算机应用是最主要的。现在所称的“计算结构力学”,其实质就是计算机化的结构力学。力学与电子技术紧密结合,促使结构力学的对象、任务、理论、计算模型、数学工具都发生了全面的深刻变化。

从结构力学的对象来看,过去这门课程研究的对象是杆件系统结构,至于板和壳的问题以及其它连续体力学,虽然它们有很多共性,但仍须列为各种不同的课程,这些课程各有自己的研究对象,这些对象的处理必须自成体系。即使是杆件结构,研究者为了计算各种类型的杆件结构,也必须寻找各种不同的特殊方法,这都是为了怎样才能克服计算上的困难以便算出具体的结果。现在,计算机化了的结构力学,它可以通过同一个途径来计算杆系、板、壳和连续体,它的对象可以是一个各种构件的组合物。结构本来是组合的,一个建筑物、一条船、一架飞机,都是杆、板、壳和连续体的组合。过去是一个一个地计算,现在则可以较好地按结构本来的面貌来研究<sup>[4]</sup>。

结构力学的任务也有了很大变化,过去结构力学的任务限于结构的分析,只是把一个结构在外因作用下的反应分析出来,不研究设计的理论。现在计算结构力学已经迈出了有力的一步,把结构优化设计作为自己的任务<sup>[5]</sup>。它不仅分析和说明结构,随着计算机辅助设计 CAD 的进展<sup>[6][7]</sup>还进一步设计和改造结构。

结构力学的基础理论是早已建立起来的。平衡条件、连续条件、物理性质三者的统一产生了力法和位移法(变位法),还有各种能量原理和变分原理。过去由于来自计算工作方面的障碍、各种特殊的计算方法层出不穷,而基础理论的作用却没有能尽情地发挥。计算结构力学使计算方法趋于统一。如以位移法为基础的直接刚度法目前几乎成了统一的方法,而各种变分原理增添了新的活力。

结构力学的计算模型关系到力学工作能否真实反映实际。以往总是把空间问题简化为平面问题,多维变少维,把非线性的问题简化为线性问题,把不均匀简化为均匀,把不连续的改为连续的,把动态的改为静态的。改来改去,无非是如何使问题便于计算。计算机化以后,结构力学可以处理很复杂的计算模型,因此便于反映真实的情况。

结构力学使用的数学工具也有了变化。以往杆件结构用线性代数,板、壳和连续体用偏微分

方程。现在计算机的工作是离散化的数值计算。矩阵数学成为有效的数学工具。计算结构力学离不开矩阵数学,这是因为矩阵数学本身的表达能力强,运算推演简洁方便,更重要的是它非常适应电子计算机的工作。凡是电子计算机的专长,诸如矩阵数学、循环迭代、逻辑判别等都是计算结构力学的重要工具。

### § 1-3 结构分析程序设计的原则

近几年来我国微型机数量有了很快的发展,微机速度和容量不断增长,体积和价格不断下降急需各种应用软件以充分发挥其作用。本书是在 IBM—PC 微机上编制杆件结构一系列的程序,重点是如何设计这些程序。

编写结构分析程序,就是人们将结构力学中的有关计算问题(静力、动力、稳定等等),预先设计好计算步骤,每一个步骤计算的内容,用计算机能接受的算法语言通知计算机去完成这些计算。

编制结构分析程序是一项耗资很大的高度智力劳动,它不仅需要掌握结构力学基本理论,结构矩阵分析原理,而且还要熟悉数值计算方法,程序设计和算法语言以及有关计算机方面的知识。

编写一份简单程序是容易的,但编写好一份大的程序,特别是复杂的程序就不那么容易。其困难程度不一定随着程序规模的增大而成比例地增长。一般而言,研制一个程序所需要的时间与该程序的复杂程度密切相关,与这个程序的大小只是间接地有关。如果不能设法控制这种相互关系,使之尽量逻辑化,显式化,而且减少其相关性,那么将会使其复杂程度爆炸性地增加,使得本来能在短期中解决的课题而在几年中也难以完成。

这种对复杂度的控制成为软件工程中的一个中心问题。应当说,人们对于复杂度太高的问题至少在目前还是能力有限的。软件工程师采取的各种方法、技术与工具都是用来控制软件复杂性的。

为了便于理解和掌握一份程序的整体,如同研究其它学科一样,须要建立模型和进行抽象。对于一个整体程序可用一个总体的模型加以描述,并且用一些子程序来阐明下一层较细致的模型,逐步精细化。于是在一个大的程序中,就会遇到许多层的抽象,只有用这个方法方能控制程序复杂度的指数式扩展。模型抽象的选择,应在描写实际问题的条件下愈简单愈好,使模型抽象既可充分描述所企求的程序性能,而又简单易理解。

建立研制一个好的程序没有一个绝对的规则,但它必须使程序妥当地完成其预期的功能,同时也应当使成本不太昂贵。遵循一般的原则<sup>[8]</sup>,就可得出一个好的实用程序。这些原则是:程序设计方法论;结构化程序设计;软件工程规则;软件工具等。

## 一、程序设计方法论的主要内容

随着电子工业的迅速发展,电子计算机硬件发展很快。资料表明,自 1945 年以来,硬件的成本平均每隔 2~3 年下降一半,而软件研制成本则不断上升。图 1-1 所示,为美国软硬件投资比重曲线。由于软件是一种逻辑产品而不是物质产品,软件主要靠人工研制,生产周期长,再加上软件的平均复杂性程度逐年增加,因此软件的发展速度远远赶不上硬件发展的速度。这样,导致了软、硬件之间比例失调,出现了所谓“软件危机”。为了改变这种状态,可以通过:

- (1) 增加对软件的投资;
- (2) 加强对程序设计方法论以及加速其它软件研制的研究。

程序设计方法论主要研究软件研制过程的各个环节,包括软件的需要性分析,程序功能规定,程序设计,说明书写,编写程序,调试与纠错,维护等。总的目的是力求软件正确、可靠、降低成本,并把设计思想由强调灵活的技巧和局部效率转向强调程序结构化和使用集成化方向。

初学者常会感到程序设计的中心问题就是学会一种语言,以编写程序。然后用语言编写程序(编程)只是上述各个环节中的一个步骤。由图 1-2 可以看到编程在程序研制中的比例。

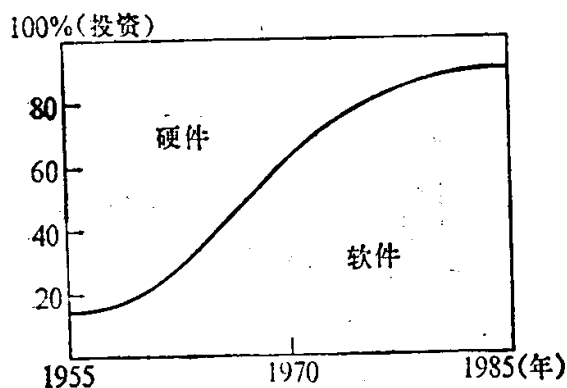


图 1-1

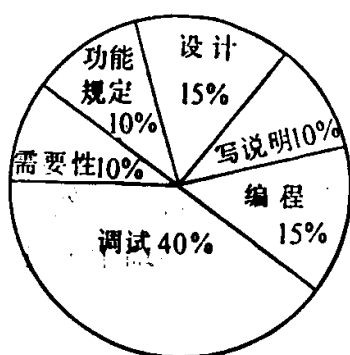


图 1-2

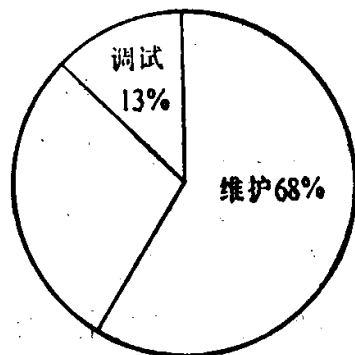


图 1-3

一个大程序投入应用后,其维护的工作量比起原先研制的工作量几乎要大一倍(图 1-3)。如此大的工作量使我们注意到,在程序研制阶段应当为以后的维护工作提供方便。

自从荷兰软件学者 Dijkstra, E. W. 提出结构化程序设计(structured programming)的概念和方法以来,经过十几年的发展,现在已包括广泛的内容,其要点可归纳为:

- (1) 编程结构化(structured coding)。即程序的编写应层次分明,逻辑清楚,使人易懂。
- (2) 分层输入输出处理技术,即 HIPO(Hierarchical Input Process Output)技术。

其主要内容为将程序划分成各个层次的许多模块,各个模块的功能要预先确定。把每个模块孤立起来,只保留外部接口(interface)与其他模块连接。对于每个模块,如有必要,又可以再分成几个小模块,而每个小模块的处理与上面一样。图 1-4 给出了 HIPO 技术示意图。对于每一模块的功能设计,则按图 1-5 所示进行,即按输入输出条件,进行某种处理。对于一个较大的软件系统要由许多人合作,必须将系统分成许多模块,每个人只能作一个或几个模块。HIPO 是模块程序设计(modular programming)中的一项技术,相当于模块的说明及功能设计。

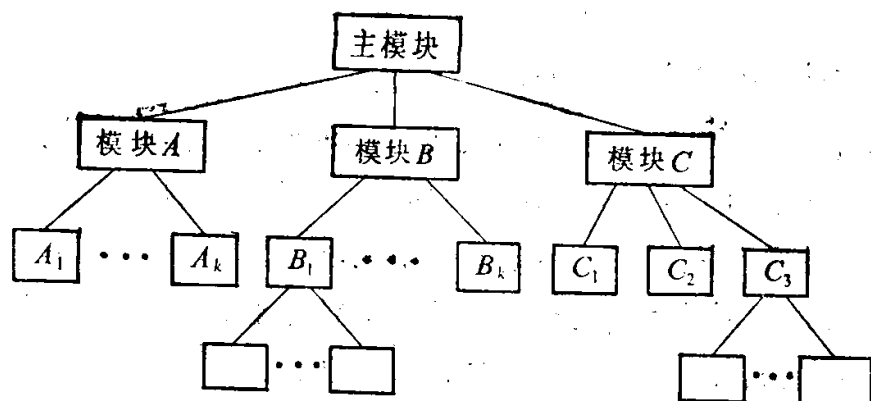


图 1-4

输入 (INPUT)	处理 (PROCESS)	输出 (OUTPUT)
入口条件	1. ... 2. ... 3. ...	出口条件

图 1-5

(3) 自顶向下(top-down)的设计方法与关键部分优先(critical component first)考虑相结合。

程序设计自顶向下，逐步精化已是公认的高效率方法。图1-4所示给了一个形象。但是有些实际问题是比较复杂的，为了设计更有把握起见，常常伴随使用关键部分优先考虑的设计方法。采用自顶向下的方法比较容易控制程序的复杂度，可以避免一些“积重难返”的错误设计。

(4) 避免过多使用 GOTO 语句，特别是逆转的 GOTO 语句，有人建议采用称为无 GOTO 语句的程序设计。

(5) 在设计，编程时，程序的主要功能用元语言(meta code)代替框图，再逐步精化成为计算机能接受的源程序。

所谓元语言表示，就是用程序设计语言与自然语言(如中文，英文)相结合来表示某些程序段落的功能(做什么)。它是软件设计中的一项重要技术。

#### (6) 程序结构定型化

程序由3种基本控制结构组成，即顺序结构(sequence)、选择结构(IF-THEN-ELSE)、循环结构(iteration)，如图1-6。其中S、S<sub>1</sub>、S<sub>2</sub>表示一些程序段落，BE(Boolean Expression)是一个表示判定条件的逻辑表达式。

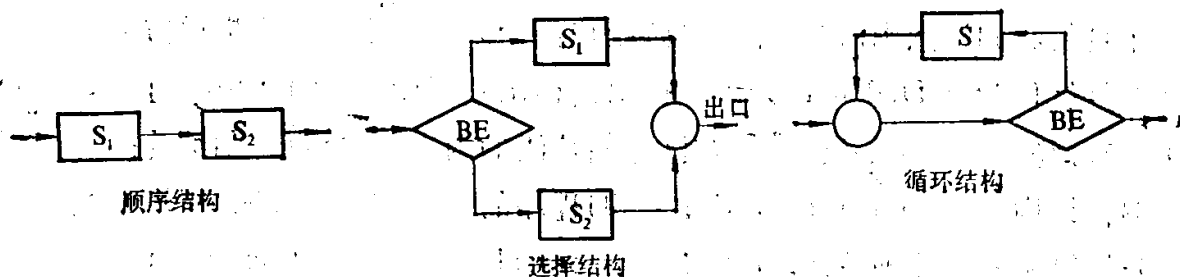


图 1-6

#### (7) 主程序员团(chief programmer team)

可以由几个人组成，最多不超过7人，但常常是一个人。主程序员相当于中央处理机；其他程序员都是助手，相当于分层输入输出装置(I/O)。这是一项行之有效的软件管理技术。当然这也不是唯一可行的。

应当指出，程序设计方法论仍在发展探索之中。上列几点也并不是结构化程序设计的全部

内容，更不能把它当作绝对正确的教条，而应通过实践来发展，丰富其内容。

经验证明，采用结构化程序设计的方法研制软件，一般可显著提高软件的正确程度，增加其可读性与可修改性，缩减调试与维护费用，从而达到降低软件成本的目的。

## 二、程序的结构化及逐次精化的概念

程序的展开或逐步求精是程序从抽象(做什么)到具体(如何做)的发展过程。在图 1-6 中表示程序结构由 3 种基本控制结构所组成。它们是程序展开的最基本形式。下面结合 PASCAL 语言予以进一步说明。

### (1) 顺序结构

采用顺序结构把功能框 S (用虚线表示) 展开成两个相继的功能框 S<sub>1</sub> 和 S<sub>2</sub> (图 1-7)

程序表示:

$\xrightarrow{\text{入口}} S_1; S_2; \xrightarrow{\text{出口}}$  (1-1)

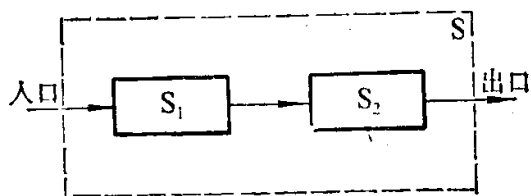


图 1-7

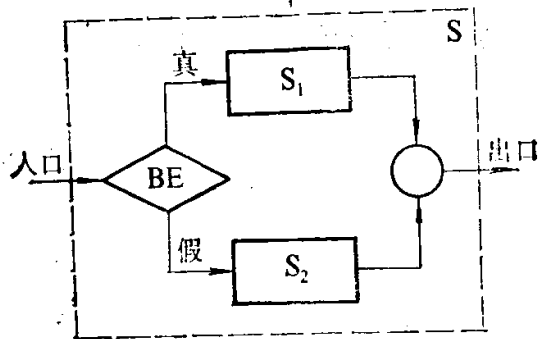


图 1-8

### (2) 选择结构

采用选择结构把功能框展开成供选择的(二择一)两个功能框 S<sub>1</sub> 和 S<sub>2</sub> (图 1-8)。

程序表示:

```

 $\xrightarrow{\text{入口}}$  IF (BE) THEN BEGIN
    S1
END ELSE BEGIN
    S2
END;
    
```

(1-2)

### (3) 循环结构

采用循环结构把功能框 S 展开成需要重复执行若干次的循环体 S<sub>1</sub> (图 1-9)。

程序表示:

```

 $\xrightarrow{\text{入口}}$  WHILE (BE) DO BEGIN
    S1
END;
    
```

(1-3)

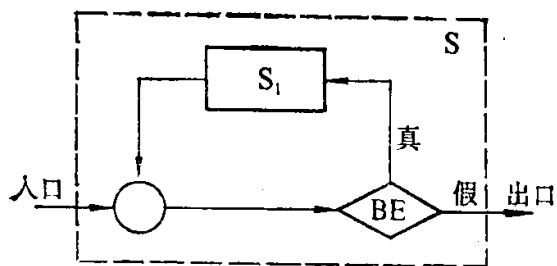


图 1-9

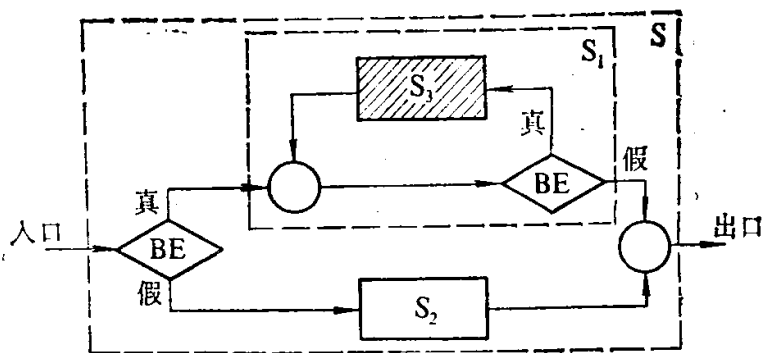


图 1-10

在 *WHILE* 语句中, 若 *BE* 为真, 则执行  $S_1$  一次, 显然循环体  $S_1$  中应包括隐式或显式地修改 *BE* 中的变量, 一旦 *BE* 为假, 则立刻退出循环, 执行紧接于 *END* 后的语句。

所谓结构化程序, 就是其中的任一功能方框仅用图 1-7、图 1-8 和图 1-9 所示的顺序、选择和循环结构而展开的程序。例如图 1-10 给出的即是一个结构化程序。其逐级展开过程为:

```

S ⇒ IF (BE) THEN BEGIN
        S1
      END ELSE BEGIN
        S2
      END;
⇒ IF (BE) THEN
  WHILE (BE) DO BEGIN
    S3
  END ELSE BEGIN
    S2
  END;

```

式中, 首先使用选择结构把  $S$  展成  $S_1$  或  $S_2$ ; 然后再用循环结构把  $S_1$  展成  $S_3$ 。

由图 1-10 可见, 结构化程序是一种嵌套式的分层结构。  $S$  作为第 0 层;  $S_1$  和  $S_2$  位于第 1 层;  $S_3$  位于第 2 层。某一程序段的层次等于框住它的方框(实线或虚线)的数目减 1。既然是分层结构, 因此结构化程序也可以表示成树形结构。

图 1-11 即为与图 1-10 对应的树形结构。

图 1-12 给出了另一个结构化程序的例子; 图 1-13 为其相应的树形结构。

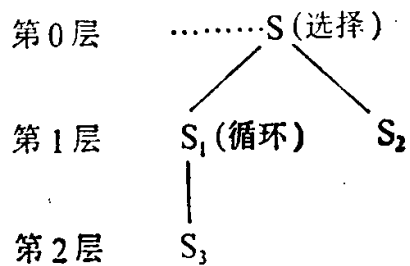


图 1-11

为了清楚起见, 我们把图 1-12 与 1-13 所示结构, 按层次用 PASCAL 语言的记号逐级展开如下:

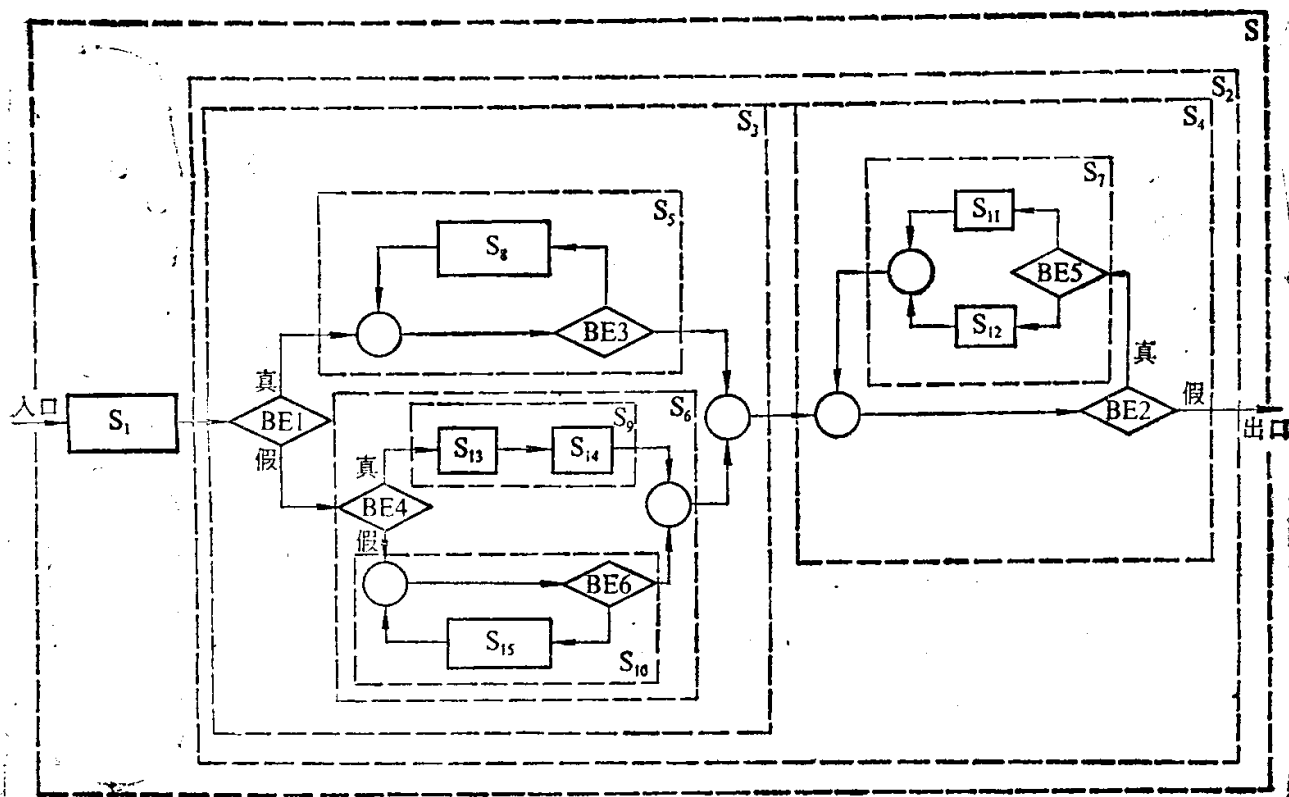


图 1-12

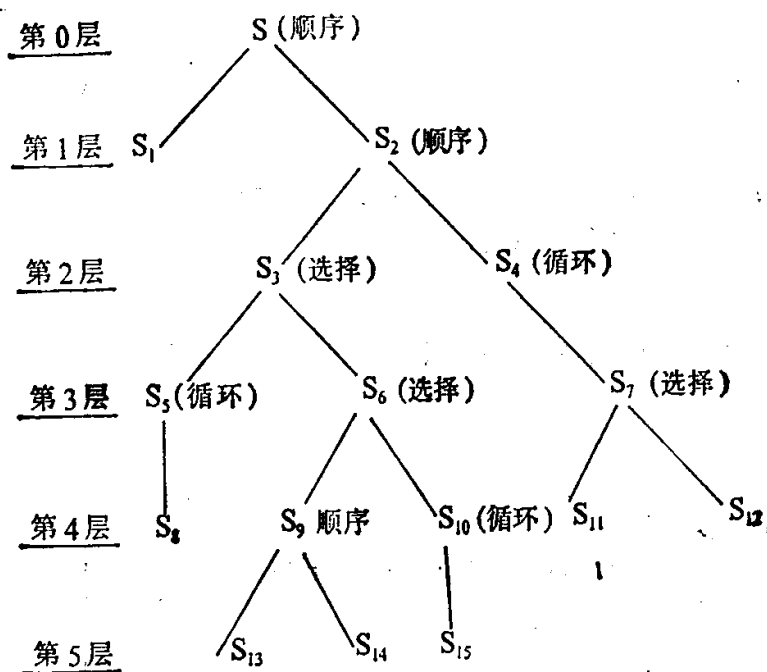


图 1-13

S  
 S<sub>1</sub>; S<sub>2</sub>  
 S<sub>1</sub>; S<sub>3</sub>; S<sub>4</sub>

(I)  
 (II)  
 (III)

```

      S1
    IF (BE1) THEN BEGIN
      S5
    END ELSE BEGIN
      S6
    END;
    WHILE (BE2) DO BEGIN
      S7
    END;

```

(IV)

```

      S1
    IF (BE1) THEN BEGIN
      WHILE (BE3) DO BEGIN
        S8
      END ELSE IF (BE4) THEN BEGIN
        S9
      END ELSE BEGIN
        S10
      END;
    END;
    WHILE (BE2) DO BEGIN
      S7
    END;

```

(V)

```

      S1
    IF (BE1) THEN BEGIN
      WHILE (BE3) DO BEGIN
        S8
      END ELSE IF (BE4) THEN BEGIN
        S13
        S14
      END ELSE BEGIN
        WHILE (BE6) DO BEGIN
          S15
        END;
      END;
    END;
    WHILE (BE2) DO BEGIN
      S7
    END;

```

(VI)



上述公式(I)至(VI)显示了这个例子的逐步求精过程。如前所述,这里的 $S, S_1, \dots, S_{15}$  可以用元语言表示。开始时,往往只用自然语言(中文或英文)说明这些段落的功能(做什么),然后逐步精化为计算机上能实现的程序语言。当公式(VI)中 $S_1, S_8, S_{13}, S_{14}, S_{15}, S_7$  均用程序设计语言表示时,精化过程就全部完成了。应该指出,(II)或至(VI)中的 $S_1$ ,虽然用同一记号表示,但精化程度可以不一。(I)到(VI)式的精化过程是严格按结构层次进行的。实际问题的精化过程可按难易程度灵活掌握。显然,若按相反的次序:(VI) $\rightarrow$ (V) $\rightarrow$ (IV) $\rightarrow$ (III) $\rightarrow$ (II) $\rightarrow$ (I),则就是程序的逐级抽象过程。

现在,再举一个具体的结构化程序例子。设 $I$ 为一整数(正、负或零均可)。试编制一段程序,使得这段程序执行后,有

$$I = \begin{cases} 0 & \text{当原来的 } I \text{ 为零或正、负偶数时} \\ 1 & \text{当原来的 } I \text{ 为正或负的奇数时} \end{cases}$$

这段程序可编制如下:

```

IF (I>0) THEN
    I:=I-I DIV 2*2
ELSE
    I:=I DIV 2*2-I
    
```

其中利用了整除的商自动取整的性质。

如注意到一个数加2或减2不改变奇、偶数的特性,上述程序也可以改写如下:

```

IF (I>0) THEN
    WHILE (I>1) DO BEGIN
        I:=I-2;
    END;
ELSE
    WHILE (I<0) DO BEGIN
        I:=I+2;
    END;
    
```

对应这个程序的框图及相应的树形展开可由图 1-14 给出。

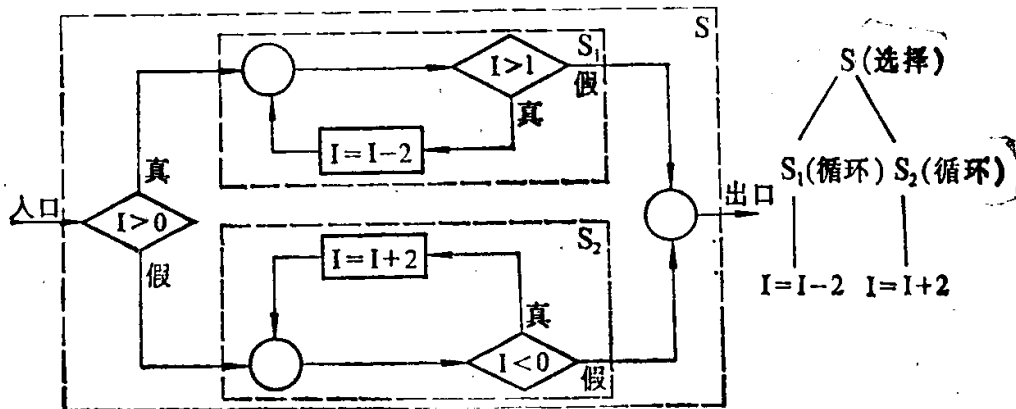


图 1-14