

COMMON/A/ D(500),V(500),X(501),XM(502)
COMMON/B/
DK 高等学校“十一五”规划教材

COMMON/C/ MN(101),H(101)

COMMON/D/ V0,VN1

COMMON/E/ NN,VB

COMMON/F/ EF,TK

COMMON/G/ I(101),A(101)

COMMON/H/ TK

OPEN(2,FILE='DT.DAT')

OPEN(7,FILE='DT1.DAT')

WRITE(*,555)

555 FORMAT(2X,'IE,EF,T(K)=?')

READ(*,666) IE,EF,TK

666 FORMAT(I5,2F15.7)

CCC=9.162260398D4*TK

EPS=1000D0*10.D0**(-IE)

IJK=2

NN=3

EF=EF

SX=0.1D0

V0=0.D0

MN(1)=20

MN(2)=50

MN(3)=20

D(1)=4.D0

D(2)=10.D0

D(3)=4.D0

WRITE(2,321) EPS,SX,EF,TK

FORMAT(2X,'EPS=' ,E15.7,2X,'SX,EF,TK=' ,3F15.7)

WRITE(2,322) NN,MN(1),MN(2),MN(3)

FORMAT(2X,'NN=' ,I4,2X,'NM(3)=',3I5)

WRITE(2,323) CCC,D(1),D(2),D(3)

323 FORMAT(2X,'CCC=' ,E15.7,2X,'D(3)=',3E15.7)

CALL SDM

CALL SXM

DO 1000 IJJ=1,100

VB=0.00400D0*DBLE(FLOAT(IJJ))

COMMON/H/ TK

OPEN(2,FILE='DT.DAT')

OPEN(7,FILE='DT1.DAT')

WRITE(*,555)

555 FORMAT(2X,'IE,EF,T(K)=?')

READ(*,666) IE,EF,TK

666 FORMAT(I5,2F15.7)

CCC=9.162260398D4*TK

EPS=0.5D0*10.D0**(-IE)

IJK=2

NN=3

EF=EF

SX=0.1D0

V0=0.D0

MN(1)=20

MN(2)=50

MN(3)=20

D(1)=4.D0

D(2)=10.D0

D(3)=4.D0

WRITE(2,321) EPS,SX,EF,TK

321 FORMAT(2X,'EPS=' ,E15.7,2X,'SX,' ,

WRITE(2,322) NN,MN(1),MN(2),MN(3)

322 FORMAT(2X,'NN=' ,I4,2X,'NM(3)=',

WRITE(2,323) CCC,D(1),D(2),D(3)

323 FORMAT(2X,'CCC=' ,E15.7,2X,'D(3)=',

CALL SDM

CALL SXM

井孝功 赵永芳 蒿凤有 编著

VB=0.00400D0*DBLE(FLOAT(IJJ))

VN1=-VB

CALL SV(IJK)

CALL ROMBG(1.0D-30,SX,EPS,20

IF(NKK.LT.0) STOP

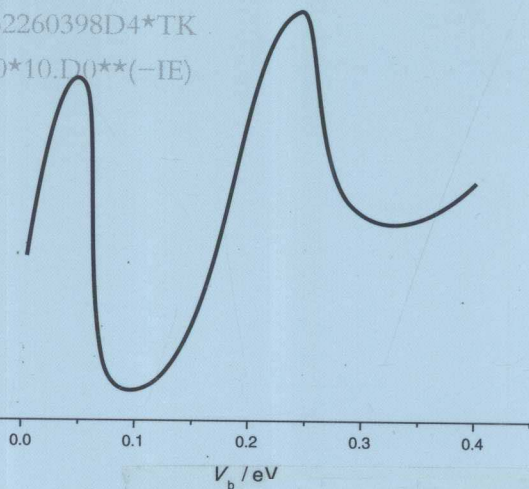
SJ=CCC*S

WRITE(*,551) NKK,VB,SJ

WRITE(2,551) NKK,VB,SJ

551 FORMAT(2X,'NKK=' ,I10,2X,'VB,S

量子物理学中的 常用算法与程序



哈尔滨工业大学出版社

高等学校“十一五”规划教材

量子物理学中的 常用算法与程序

井孝功 赵永芳 蒿凤有 编著

哈尔滨工业大学出版社

内 容 提 要

本书简明扼要地讲述了计算物理的基础知识,并给出量子物理学中常用算法及相应的 FORTRAN 程序。本书既可作为计算物理的入门书,又可作为量子物理计算的工具书。

全书由 13 章组成,主要包括 FORTRAN 算法语言简介,代数公式,常用特殊函数,3j、6j 和 9j 符号,一元方程,线性代数,函数插值与微商,常微分方程,数值积分,本征问题,递推与迭代,蒙特卡罗方法,快速傅里叶变换等,同时还有与内容相对应的大小程序 82 个。

本书内容偏重于量子物理学,其中一些内容是作者与同事们在计算物理领域的研究成果。例如,矩阵元的计算,主值积分,薛定谔方程的辛算法,定态薛定谔方程的有限差分法,微扰论的递推形式,最陡下降法,透射系数的递推公式, $I-V$ 曲线等,这些新的算法都具有较高的实际应用价值。

图书在版编目(CIP)数据

量子物理学中的常用算法与程序/井孝功,赵永芳,
蒿凤有编著. —哈尔滨:哈尔滨工业大学出版社,2009.12

ISBN 978-7-5603-2981-9

I. ①量… II. ①井…②赵…③蒿… III. ①量子力
学-算法理论②FORTRAN 语言-程序设计 IV. ①O413.1
②TP312

中国版本图书馆 CIP 数据核字(2009)第 232191 号

责任编辑 张秀华

封面设计 张孝东

出版发行 哈尔滨工业大学出版社

社 址 哈尔滨市南岗区复华四道街 10 号 邮编 150006

传 真 0451-86414749

网 址 <http://hitpress.hit.edu.cn>

印 刷 哈尔滨工业大学印刷厂

开 本 787mm×960mm 1/16 印张 23 字数 410 千字

版 次 2010 年 2 月第 1 版 2010 年 2 月第 1 次印刷

书 号 ISBN 978-7-5603-2981-9

印 数 1~2 000 册

定 价 45.00 元

前 言

传统的物理学分为理论物理与实验物理两大分支。理论物理是立足于全部物理实验的总和,以演绎归纳和逻辑推理方法揭示物质世界的基本规律,预见新现象,推动整个物理学乃至自然科学的发展。实验物理则以实验和观测为手段来发现新的物理现象,为理论物理提供找出新规律的素材,并检验理论物理推论的正确程度及适用范围等。

计算物理学起源于第二次世界大战中美国研制核武器工作的需要,随着电子计算机的出现和不断更新而迅速发展起来。由于计算物理学的性质、研究方法和所需要的设备与上述两大分支有着明显的差异,因而,逐渐形成了物理学的一个独立分支。计算物理学将物理学与计算数学有机地结合为一体,开辟了物理学的一个新的领域。它的特点是,以物理问题为出发点和归宿,一方面吸取专门从事计算方法研究者所建立的好方法,另一方面结合实际物理问题的需要,创造出富有特色的实用计算方法。计算物理学的出现使原来的理论物理和实验物理的研究现状大大改观,不仅使理论工作者从繁杂的解析推导的束缚中解放出来,而且使实验工作者的研究手段得到根本改革。在某种意义上说,计算物理学的研究方法及其研究风格更接近于实验科学,有人称其为数值实验或纸上的实验物理学,是有一定道理的。

量子理论是 20 世纪两个最伟大的科学发现之一,理论计算与实验观测结果的一致性,充分反映出它的正确性。当前的热门学科(生命、材料和信息)的进一步发展都离不开量子理论。量子理论导出的公式可以加深理性认识,对公式的数值计算可以做到“心中有数”,从而增加对物理问题的感性认识。虽然国内也有一些计算物理方面的书籍问世,但偏重于量子物理学的尚属少见。本书是为具备大学物理与高等数学基础知识的读者编写的,对量子理论中的常用算法既介绍该算法的由来,又给出相应的计算程序。书中与量子力学相关的例题可以在作者编著的《量子力学》或《高等量子力学导论》中找到,在这个意义上讲,本书又是上述两本书的配套书。作者期望本书既可作为计算物理的入门教材,又能成为量子物理计算的工具书。

在诸多的算法语言中,FORTRAN 语言是一种特别适合理论计算的算法语言,书中的程序全部使用 FORTRAN 语言编制。为了方便读者使用,首先对

FORTRAN 语言做了简单概括的介绍,然后由浅入深地讲解了量子物理学中常用到的一些算法及相应的程序,其中包括,代数公式,常用特殊函数,3j、6j 和 9j 符号,一元方程,线性代数,函数插值与微商,常微分方程,数值积分,本征问题,递推与迭代,蒙特卡罗方法,快速傅里叶变换等内容,同时还有与内容相对应的大小程序 82 个。

在介绍这些算法时力求简洁适用,而不刻意追求数学形式上的完美,只是期望读者能掌握相关的基本方法和计算程序,更深入的讨论与研究留给读者经过一段实际应用后自行解决。对每种算法的介绍都包括数学上的演绎、计算公式、计算程序、程序的使用说明、例题和计算结果等几个部分。读者若能学以致用,在计算机上进行实际操作,将会收到立竿见影的效果。

书中的一些常见算法,取自参考文献中列出的一些相关的书籍,而一些全新的量子物理学的内容,例如,矩阵元的计算、主值积分、薛定谔方程的辛算法、定态薛定谔方程的有限差分法、微扰论的递推形式、最陡下降法、透射系数的递推公式、 $I-V$ 曲线等,都具有较高的实用价值,都是作者与吴式枢教授、姚玉洁教授、刘曼芬教授、赵国权教授、吴兆颜教授、吴连勋教授等人的研究成果。作者在此向他们表示诚挚的感谢。

由于作者水平有限,书中难免有不当之处,敬请读者批评指正。

如需要本书中的程序请登录哈尔滨工业大学出版社网站。

井孝功 赵永芳 蒿凤有

2009 年 3 月 14 日

目 录

第 0 章	FORTRAN 语言简介	1
0.1	FORTRAN 语言概述	1
0.2	变量的类型	5
0.3	程序的结构	9
第 1 章	代数公式	15
1.1	排序与求和	15
1.2	阶乘、排列与组合	23
1.3	复数运算	28
第 2 章	常用特殊函数	42
2.1	伽马函数与贝塔函数	42
2.2	正交多项式	46
2.3	贝塞尔函数	57
第 3 章	3j、6j 和 9j 符号	65
3.1	CG 系数与 3j 符号	65
3.2	U 系数与 6j 符号	69
3.3	广义拉卡系数与 9j 符号	73
3.4	数值计算的验证功能	76
第 4 章	一元方程	82
4.1	直接公式解法	82
4.2	迭代法	93
4.3	二分法	99
4.4	牛顿法与弦截法	105
第 5 章	线性代数	110
5.1	高斯消元法	110
5.2	迭代法	117

5.3	追赶法	124
5.4	矩阵求逆	131
第 6 章	函数插值与微商	136
6.1	拉格朗日插值公式	136
6.2	差分、差商与数值微商	140
6.3	牛顿插值公式	144
6.4	厄米插值公式	148
6.5	曲线拟合	151
第 7 章	常微分方程	158
7.1	常微分方程的初值问题	158
7.2	薛定谔方程的辛算法	169
7.3	常微分方程的边值问题	175
7.4	有限元法	182
第 8 章	数值积分	189
8.1	辛普生求积公式	189
8.2	龙贝格积分法	196
8.3	二重积分	200
8.4	主值积分	204
8.5	积分转化为有限项求和	208
第 9 章	本征问题	221
9.1	乘幂法	221
9.2	雅可比方法	227
9.3	实对称矩阵的 QL 解法	234
9.4	有限差分法	245
第 10 章	递推与迭代	250
10.1	无简并微扰论公式的递推形式	250
10.2	简并微扰论公式的递推形式	253
10.3	微扰论递推公式应用举例	257
10.4	最陡下降法	302
10.5	透射系数的理论计算	311
10.6	$I-V$ 曲线	325
第 11 章	蒙特卡罗方法	330
11.1	蒙特卡罗方法的基本原理	330

11.2 随机变量抽样值的产生	333
11.3 蒙特卡罗方法计算积分	335
第 12 章 快速傅里叶变换	339
12.1 傅里叶变换	339
12.2 快速傅里叶变换	344
程序一览表	353
参考文献	356

第 0 章 FORTRAN 语言简介

复杂的数值计算通常是利用程序来完成的,而程序是用算法语言编制的,在诸多的算法语言中,虽然 FORTRAN 语言的功能不是最强大的,但是它特别适合于科学计算。本章将介绍 FORTRAN 语言的基本语句和程序的基本结构,使初学者能够顺畅地阅读程序,并可以自己编写一些简单的程序。

0.1 FORTRAN 语言概述

0.1.1 计算机

在科学研究工作中,推导出理论公式之后,常常需要借助计算工具对公式进行数值计算。随着科学技术的不断进步与发展,计算工具从算盘、计算尺、手摇计算机逐步演变到电动计算机。20 世纪 40 年代中期,电子管计算机的出现使理论计算实现程序化成为可能,它是计算工具变革的一个里程碑。后来,晶体管计算机、集成电路计算机、巨型计算机陆续出现,如今,计算机已经成为更新换代最快的产品。

现代的计算机主要由存储器、中央处理器和外围设备三个部分构成,存储器是存放指令和数据的地方,中央处理器(运算器)是执行程序指令的机器,外围设备是用于输入和输出数据的仪器。

随着计算机的不断更新换代,使其不断向大容量、高速度、小型化和智能化发展。计算机的内存迅速扩大且运算速度越来越快,它的功能也逐渐超出了人们最初的设想。除了本书将介绍的数值计算之外,还可以利用相应的软件进行公式的推导和数值模拟。特别是互联网的出现,计算机不仅仅只是服务于科学研究,还可以利用它来获取信息及传播信息,它已经成为人们生活中不可缺少的一部分。

0.1.2 算法

为了进行数值计算,必须事先选定正确的算法,所谓算法就是一步一步解决问题的过程。

正确的算法必须满足如下的条件:

- (1) 算法是分步实现的;
- (2) 每一步必须是一条可以执行的指令;
- (3) 各步的顺序必须是确定的;
- (4) 算法不能无休止地进行下去,即必须能结束。

为了加深对算法的理解,举一个通俗的例子。某人急于知道 $\sin 5^\circ + \ln 4$ 的计算结果,却苦于手头没有计算工具,只好给家里打电话求助。不巧的是妻子不在家,而刚上小学的儿子又不会使用计算器,于是,他先让儿子找来计算器,然后,再让儿子按照如下的步骤进行操作

[ON] → [5] → [sin] → [+] → [4] → [ln] → [=]

最后,儿子告诉他的结果是 1.473 450 104。这里,父亲对儿子下达的一连串指令就是用自然语言表达的算法。

顾名思义,算法就是计算的方法,原则上它与所使用的语言无关,但是,在不同的语言环境中,它的表现形式可能会不同。

0.1.3 算法语言

如前所述,算法是由一系列指令构成的,为了确切地表述这些指令,则需要选定一种操作者和计算机都能理解的语言,这种能实现人-机对话的语言就是算法语言。在确定的算法语言环境下算法就是所谓的程序。程序是由一系列语句构成的,执行程序的过程通常是,按照语句的顺序由前到后逐条进行。

算法语言是为编制程序服务的,现有的算法语言种类很多,不同的算法语言重点服务的对象不同。最基本的算法语言是汇编语言,对于非计算机专业的人来说,它不便于掌握和使用。为了解决这个问题,人们陆续编制出一些高级算法语言的软件,从最早的 BASIC 语言到 FORTRAN 语言,直至目前流行的 C 语言等,从而使许多计算机领域的门外汉也能充分地享受计算机带来的方便。

尽管各种算法语言之间有千差万别,但是有一点是相同的,那就是,它们都是由语句构成的,不同算法语言允许使用的语句个数会有差别。

0.1.4 FORTRAN 语言

FORTRAN 是英文 Formulae Translation 的缩写,显然,FORTRAN 语言的原意是公式翻译语言。

FORTRAN 语言有如下特点:

- (1) FORTRAN 语言特别适合于数值计算;
- (2) 语句个数不多,且全部用相应的英文单词表示;

(3) 语句分为说明语句和执行语句两类,且说明语句在前;

(4) 其程序由一个主程序和若干个子程序与函数子程序构成,这种独立的块状结构使得程序的结构十分清晰。

为了使读者有一个初步的印象,下面给出一个最简单的 FORTRAN 程序。

```
PROGRAM ABC
  A = 2.0 + 5.0 * (3.0 - 0.5)
  B = 3.0 - 7.0/3.0
  C = A + B * * 2
  WRITE( *, 1) A, B, C
1  FORMAT(3E15.7)
  STOP
  END
```

在上述程序中,所有的英文单词都是 FORTRAN 语言允许使用的语句用词。符号 + 与 - 即通常的加号与减号, * 为乘号, / 为除号, * * 表示幂次,赋值语句中的小括号 () 与通常情况相同,它可以有多重,但必须配对使用,程序中不允许使用中括号和大括号。操作的次序与正常的数学运算相同。

上面程序的意思是,一个名字为 ABC 的主程序,按照如下的次序运行:

- (1) 计算 $2.0 + 5.0 * (3.0 - 0.5)$ 的数值,并将其赋予变量 A;
- (2) 计算 $3.0 - 7.0/3.0$ 的数值,并将其赋予变量 B;
- (3) 计算 $A + B * * 2$ 的数值,并将其赋予变量 C;
- (4) 将 A, B, C 中的数值按照语句标号 1 所设定的格式在屏幕上显示出来;
- (5) STOP 表示主程序运行结束, END 表示退出计算程序。

0.1.5 程序的编制与运行

为了能够使用 FORTRAN 语言进行数值计算,必须事先在所用的计算机上安装 Fortran Power Station 软件,然后按下列基本步骤进行操作。

1. 选择算法、编制程序

- (1) 根据具体的问题选择适用的算法;
- (2) 按照算法的主要步骤绘出程序的流程图(框图);
- (3) 按照框图编制出相应的程序。

2. 程序的键入、检查与运行

- (1) 打开 Fortran Power Station 软件;
- (2) 把已经写好的程序按规定键入计算机,按“Save all”键将文件名为 ABC

扩展名为 FOR 的文本文件 ABC.FOR 保存;

(3) 按“Build”键,再按其中的“Build all”键,进行语法检查,如果有语法错误,则需要改正,直至无语法错误为止,形成扩展名为 EXE 的可执行文件 ABC.EXE;

(4) 按“Build”键,再按“Execute abc.exe”键,执行计算。

对于前面给出的程序而言,屏幕上将显示如下结果

.1450000E + 02 .6666667E + 00 .1494444E + 02

应该特别说明的是,在键入编制好的程序时,前 5 个格是为语句标号准备的,语句标号可以选取 1 ~ 99999 中的任意一个正整数值,但是,在同一个程序块中,不能出现相同的语句标号。第 6 个格是为延续行的标号准备的,所谓延续行是指对上一行的延续,延续行的标号可以在 0 ~ 9 之间任意选择,也可以选为 *。正常的语句应从第 7 个格开始键入。

如果在第一个格键入字母 C,则表示这一行为注释行,程序不执行 C 后面的语句。注释行中可以写任何内容,它只是程序编制者的一个说明或者备忘录,可以出现在程序的任意行。

在已经录入的程序中,正确的语句标号会自动变成红色,正确的语句用词会自动变成兰色,正确的注释行会自动变成绿色。

0.1.6 程序质量的判断

1. 程序正确性的验证

对一个程序的基本要求是,计算结果惟一且准确无误。一个刚刚编制好的程序即使没有语法错误,也不一定能得到正确的计算结果,为了保证计算结果的可信性,必须对程序进行验算。

如果一个程序由若干个程序块构成,则首先需要对每一个子程序或函数子程序进行验算。若最后的计算结果仍然不对,则通常采用所谓“分区停电”的方法来处理。这个方法的基本思想是,在程序的适当位置将中间结果显示出来,若能判断它是对的,则问题出在程序的后半部分,接着在程序的后半部分重复上面的操作,直至找到错误为止;若中间结果是错的,则问题出在程序的前半部分,接着在程序的前半部分重复上面的操作,直至找到错误为止。

2. 程序质量的判别标准

除了计算结果惟一且准确无误外,一个高质量的程序还要满足如下条件:

- (1) 占用的内存小和运算的时间短;
- (2) 结构清晰,易读;
- (3) 通用性好。

0.2 变量的类型

0.2.1 变量名

在 FORTRAN 语言中,数据是保存在变量中的,变量的名字是由程序编制者选定的,它可以由英文字母与数字的组合构成,字母与数字的总个数不要超过 6 个,并且,一定要以字母开始,例如,ABC, A24, B4A, HF56 等皆可作为变量名使用。

应该特别说明的是,英文字母用正体,大写字母与小写字母是没有区别的。如果直接用公式中的符号作为变量名,则会给程序的编制和阅读带来方便。但是,应该将公式中的大写与小写字母区别开来,例如,公式中的大写字母 B 可以在程序中改记为 DB,公式中的小写字母 b 可以在程序中改记为 XB,这样一来,既便于记忆又不容易混淆。变量名中不能出现下标,若要用一个变量标记公式中的 A_{24} ,则可以将其记为 A24。另外,不要用后面将介绍的内部函数名作为变量名,例如, SIN, ABS, SQRT 等。

0.2.2 简单变量与数组

按照存放数据个数的多少,变量分为简单变量与数组两类。

1. 简单变量

简单变量只能存放一个数值,例如, I, AB, EPS 等。

2. 数组

数组可以存放多个数值,数组可以是一维、二维与三维的。若用到数组,则必须事先利用说明语句进行维数说明,例如

```
DIMENSION A(8), J(5,7), B(4,4,4)
```

上述语句的意思是说, A(8) 表示 A 是一个可以容纳 8 个数的一维数组, J(5,7) 表示 J 是一个 5 行 7 列可以容纳 35 个数的二维数组, B 是一个可以容纳 64 个数的三维数组。

应该特别强调的是,出现在说明语句中的数组符号与可执行语句中的数组符号含意完全不同。例如,说明语句中的 A(8) 表示 A 是一个可以容纳 8 个数的一维数组,而可执行语句中的 A(8) 表示数组 A 的第 8 个元素的数值。

数组也可以利用公用块来说明,例如

```
COMMON/AB/ C(3), A(2,2)
```

其中,两条斜线中的 AB 为公用块的名称。关于公用块的其他用途将在下一节中

介绍。

0.2.3 变量的类型

在 FORTRAN 语言中,不同类型的数据是用不同类型的变量来表示的。常用的变量可分为整型、实型、双精度型、复型和逻辑型几种。原则上,所用变量的类型都需要事先利用说明语句给予说明。

1. 整型

若不加特殊说明,则以 I, J, K, M, N, L 六个字母打头的变量表示整型数,称之为整型变量的隐式说明,例如, $I = 2$, $MN = 100$, $LA = 5$, $K(1, 2) = 9$ 等。若要用上述字母以外的字母来表示整型量,则必须对其事先说明,例如,如果用 DJ 表示整数,则必须在说明语句中加入

INTEGER DJ

两个整型量的四则运算结果仍然是整型量,例如, $5/2 \neq 2.5$, 而是 $5/2 = 2$ 。

不同计算机系统允许的 FORTRAN 整数最大值不同,以 VAX 机为例,可用的最大整数为 2147483647。详见表 0.1。

表 0.1 不同计算机系统允许的整数最大值和实数的有效位数以及实数的范围

计算机类型	整数最大值	实数有效位数	实数的范围
IBM360/370	2147483647	6	$10^{-77} \sim 10^{75}$
VAX	2147483647	7	$10^{-38} \sim 10^{38}$
CRAY - 1	2.8×10^{14}	13	$10^{-2465} \sim 10^{2465}$
CDC6000/7000	9.2×10^{18}	14	$10^{-293} \sim 10^{322}$
IBM PC(长城 0520)	32767	7	$10^{-38} \sim 10^{38}$

2. 实型

若不加特殊说明,则除了以 I, J, K, M, N, L 六个字母打头的变量以外,所有的变量都表示实型数,称之为实型量的隐式说明。实型量的表示方法有两种,一是直接表示法,例如, $A = 1.0$, $B = 3.5$ 等;二是科学计数法,例如, $P = 1.2$ 表示为 $P = 1.2E0$, $Q = 1.05 \times 10^{-8}$ 表示为 $Q = 1.05E - 8$ 等。

不同计算机系统的 FORTRAN 实数有效位数及数域也不同,以 VAX 机为例,其有效位数为 7 位,实数域为 $10^{-38} \sim 10^{38}$ 。详见表 0.1。

3. 双精度型

为了提高计算精度,有时需要用到双精度变量。在使用双精度型变量之前

必须加以说明,说明的方式有两种,若只有少数几个双精度型变量,则可以分别具体说明,例如,A 与 B 为双精度型变量,只要在说明语句中加入

DOUBLE PRECISION A,B

即可;若除了以 I,J,K,M,N,L 字母打头的变量以外,所有的变量都表示双精度型变量,则可以在说明语句中加入双精度变量的隐式说明,即

IMPLICIT DOUBLE PRECISION (A - H,O - Z)

由于,双精度变量与单精度变量运算后的结果的精度是单精度,所以通常选用第二种方式进行说明。

双精度变量必须使用科学记数法,例如, $A = 0.5$ 应该记为 $A = 0.5D0$, $B = -2.0 \times 10^{-5}$ 记为 $B = -2.0D - 5$ 。

一般情况下,双精度变量的有效位数会比实型变量提高近一倍。

4. 复型

复型变量的说明类似于双精度变量,只不过用 COMPLEX 代替双精度型变量说明中的 DOUBLE PRECISION。

5. 逻辑型

若使用逻辑变量,则必须事先进行说明。例如,若 A 为逻辑变量,则必须在说明语句中加入

LOGICAL A

它表示 A 为逻辑变量,它只可能取两个值,真或者假。若 A 为假,则 $A = .FALSE.$,若 A 为真,则 $A = .TRUE.$ 。

0.2.4 变量的赋值

要给一个变量赋值,可以采用如下三种方法。

1. 利用赋值语句赋值

例如, $A = 8.0$; $I = 500$; $G = G + 0.2D - 8$; $B(1,2) = 5.0 * G$; $D = A * 3/G + C - G$ 等,上述的每一个语句都是赋值语句。应该特别说明的是,虽然在赋值语句中用到了等号,但并不意味着左右两端相等,只是表示将等式右边的数值赋于等式左边的变量。应该注意的是,等式两端的数值类型应该是相同的。

2. 利用 DATA 语句赋值

DATA 语句也是一个可执行语句,应该放在说明语句之后。

对于实型简单变量 A 与 B,具体的写法为

DATA A,B /2.5,3.6/

其作用相当于如下两个赋值语句

```
A = 2.5
```

```
B = 3.6
```

DATA 语句与赋值语句的差别是,在程序开始执行时,DATA 语句就将 2.5 和 3.6 分别赋予了变量 A 和 B,而赋值语句必须到执行它们时才能为相应的变量赋值。

利用 DATA 语句对 2×2 的整型数组 J 赋值,应该写成

```
DATA J /1,2,3,4/
```

其作用相当于如下 4 个赋值语句

```
J(1,1) = 1
```

```
J(2,1) = 2
```

```
J(1,2) = 3
```

```
J(2,2) = 4
```

应该注意的是,赋值的次序是先列后行,与通常的理解不同。

3. 利用 READ 语句赋值

按照输入方式的不同,READ 语句分为屏幕输入和文件输入两种。

第一种输入方式是由屏幕输入。例如

```
READ( *,5) A,B
```

```
5    FORMAT(2F15.7)
```

其中,READ 称为读语句,它后面括号中的 * 特指屏幕的通道号,5 为格式语句标号。FORMAT 称为格式语句,它后面括号中的 2F15.7 中的 2 表示输入两个实数,每一个实数占用 15 个字符,其中小数点后有 7 位数字。如果用科学记数法,则只要将 F 换成 E 即可。

当程序执行到上述 READ 语句时会停止运行,只有按语句标号为 5 的格式由屏幕输入变量 A 与 B 的数值后,程序才会继续运行。

第二种输入方式为文件输入。首先,在说明语句之后打开那个已经存有数据的文件,即

```
OPEN(3,FILE = 'ABC.DAT')
```

其中,OPEN 称为打开文件语句,它后面括号中的 3 为通道号,'ABC.DAT' 表示该文件名为 ABC、扩展名为 DAT。然后利用下列读语句对变量赋值,例如

```
READ(3,6) A,B
```

```
6    FORMAT(2F15.7)
```

上述语句表示,按语句标号为 6 的格式由存在通道号为 3 的数据文件 ABC.DAT 中读入变量 A 与 B 的数值。应该注意的是,读入时的变量的个数、顺序、类型和格式应该与数据文件中的完全一致。

总之,上述三种方法都能达到给变量赋值的目的。

0.2.5 数据的输出

1. 屏幕显示

屏幕显示的语句为

```
WRITE(*,15) A,B,C  
15    FORMAT(2X,'A,B,C = ',3F15.7)
```

在上述格式语句中,2X 表示空两个字符,'A,B,C = ' 表示将引号中的内容显示在屏幕上。另外,如果在格式语句中出现 /,则表示换行。

2. 记录文件

若将数据记录在名为 ABC、扩展名为 DAT 的数据文件上,也必须在程序的可执行语句的开始打开一个相应的文件,即

```
OPEN(2,FILE = 'ABC.DAT')
```

其中,2 为设定的通道号。然后利用写语句将数据记录其上,即

```
WRITE(2,17) A,B,C  
17    FORMAT(2X,3F15.7)
```

如前所述,在读出上述数据时,文件名、通道号、数据的个数、顺序、类型与格式都应写入时相同。

需要特别强调的是,计算机中数据的存取与通常意义下的存取不同,它遵循“取之不尽”和“后人为主”两条原则。例如,有如下一段程序

```
A = 6.0  
B = 5.0 * A  
B = B - A
```

在执行 $B = 5.0 * A$ 时,需要从变量 A 中取出数值 6.0,而在执行 $B = B - A$ 时,还可以从变量 A 中取出数值 6.0,此即所谓“取之不尽”。严格地说,这里的“取”如果换成“拷贝”会更贴切。在执行 $B = 5.0 * A$ 后,变量 B 中的数值为 30.0,在执行 $B = B - A$ 后,变量 B 中的数值变成 24.0,如果这时输出变量 B 中的数值,则一定是 24.0,而不是原来的 30.0,此即所谓的“后人为主”。

0.3 程序的结构

一个用 FORTRAN 语言编制的程序是由一个主程序和若干个子程序、函数子程序构成的,其中的每一个程序称之为程序块,因此其具有块状结构。