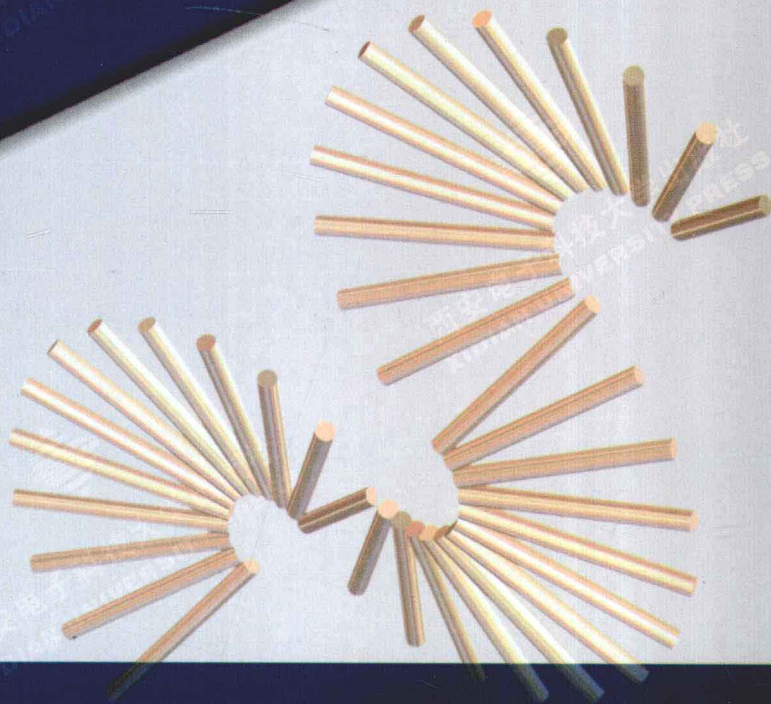




普通高等学校教材



计算物理学

■ 郭立新 李江挺 韩旭彪 编著



西安电子科技大学出版社
<http://www.xduph.com>



清华大学出版社



计算物理学

（第二版）



清华大学出版社

普通高等学校教材

计算物理学

郭立新 李江挺 韩旭彪 编著

西安电子科技大学出版社

2009

内 容 简 介

本书内容分为数值方法及其在物理学中的应用(上篇)和计算物理学(下篇)两篇。上篇主要讲述基本数值方法在大学物理中的应用,从 FORTRAN 语言和图形、图像的模拟出发,介绍了物理学中数值积分、常微分方程数值解、非线性方程求根及实验物理学中的插值和数据拟合。下篇则在上篇的基础上主要讲述有限差分方法、泛函和变分法、有限元方法、边界元方法和蒙特卡罗方法。本书内容丰富、推导详细,侧重讲述基本方法及其应用。书中的例题大部分来自物理学中的具体问题。作者在介绍具体算法的同时,附上了 FORTRAN 源程序,以供读者参考。

本书可作为本科应用物理学和电子信息科学与技术等专业的教材,也可作为物理类专业和其他非物理类理工专业本科生、研究生的教学参考书,同时,对于从事科学计算和工程设计的专业人员也具有一定的参考价值。

图书在版编目(CIP)数据

计算物理学/郭立新,李江挺,韩旭彪编著.

—西安:西安电子科技大学出版社,2009.9

普通高等学校教材

ISBN 978-7-5606-2333-7

I. 计… II. ①郭… ②李… ③韩… III. 计算物理学—高等学校—教材 IV. O411.1

中国版本图书馆 CIP 数据核字(2009)第 131407 号

策 划 马乐惠

责任编辑 任倍萱 马乐惠

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 陕西光大印务有限责任公司

版 次 2009年9月第1版 2009年9月第1次印刷

开 本 787毫米×1092毫米 1/16 印 张 18.125

字 数 424千字

印 数 1~4000册

定 价 26.00元

ISBN 978-7-5606-2333-7/C·0100

XDUP 2625001-1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

计算机的应用已遍及国民经济、科学技术和日常生活的各个领域。近代物理学与物理实验技术所获得的成果和进展，几乎都是与计算机科学相结合的产物。计算物理学是一门新兴学科，是随着计算机的出现和发展而逐步形成的物理学的一个分支。当今的物理学有三个分支，即理论物理、实验物理和计算物理，而计算物理是用计算机武装起来的理论物理，也是以计算机为仪器的实验物理。计算物理学可为理论物理提供模型和数据，也可为实验物理提供模型试验和数据。

本书分为上、下两篇，上篇为“数值方法及其在物理学中的应用”，下篇为“计算物理学”。最近几十年来计算物理发展很快，其内容 and 应用越来越广泛，而各个专业的要求又不尽相同，所以本书的主要内容是介绍处理物理问题时常用的数值算法以及它们在计算机上的实现过程。该过程可以具体为：物理问题—计算公式—数值方法—流程图、程序—上机操作—结果分析。

上篇主要面向用计算机进行工程和科学计算的大学理科及工科专业学生，内容包括物理图形图像的模拟、定积分的数值计算、常微分方程初值问题的数值解法、线性方程组的数值解法、非线性方程的求根问题以及数据插值拟合问题。其主要目的是要求学生掌握基本数值算法及其在大学物理学中的应用，力图把计算机作为学好大学物理、加强能力培养的一种手段，使学生能以数值算法为基础，以计算机为工具，加深对大学物理的基本概念、基本规律的理解，巩固已学过的算法语言课程内容，锻炼计算机编程和操作应用水平，学会应用计算机编程解决物理学或其它学科实际问题的方法。

下篇主要介绍物理学中一些偏微分方程的常用数值解法，包括有限差分、泛函与变分、有限元、边界元、蒙特卡罗方法等。对每一种方法的讲述都从实际物理问题出发，根据具体的偏微分方程，在计算机上编程求解，每一种算法都有各自的特点。例如，有限差分法把微分方程近似地用差分方程(代数方程)代替并进行求解；变分法通过求解一个相应的泛函的极小函数而得到偏微分方程边值问题的解；有限元方法则是基于变分原理的一种离散化方法，它将所要求解的边值问题转化为相应的变分问题，把问题的整体区域剖分为有限个基本单元进行离散，最终归结为一组多元的代数方程组进行求解；边界元法则以定义在边界上的边界积分方程为控制方程，通过对边界分元插值离散，化为代数方程组求解。由于计算机具有存储量大、运算速度快的优点，使用它能够方便地求解大型离散化的方程组，因此这些算法才得以实现。

计算物理学所包含的内容是相当广泛的。伴随着计算物理近些年来的飞速发展，计算物理涉及面也越来越广，它渗透到物理学的各个领域，对特定的物理问题相应的新思路、新算法也层出不穷，感兴趣的读者可以查阅相关资料。本书主要以讲述基本原理及其应用为目的，介绍一些基本的数值算法，使读者对计算物理的概念和方法有一个基本的了解。

本书各章均附有习题，可供选用。学习本课程前应先修“高等数学”、“算法语言”和“大学物理”等课程。课程教学由课堂教学和学生课外上机两部分组成。

讲授本课程上篇约需 46 学时，上机实习 20~24 小时；讲授本课程下篇约需 40 学时，上机实习 16~20 小时。

鉴于编者水平有限，书中疏漏在所难免，恳请广大读者批评指正。

编 者
2009 年 5 月

目 录

上篇 数值方法及其在物理学中的应用

第一章 FORTRAN 语言简介与误差分析初步	3
1.1 FORTRAN 语言简介	3
1.1.1 FORTRAN 语言的常量与变量	3
1.1.2 FORTRAN 基本语句	4
1.1.3 源程序语句的排列顺序	12
1.1.4 FORTRAN 常用内部函数和算术表达式	13
1.1.5 有关循环语句	14
1.1.6 FORTRAN 语言的特点	15
1.2 质点运动学问题的计算	16
1.2.1 瞬时性与极限	16
1.2.2 运动方程问题	16
1.3 误差及减小误差的原则	19
1.3.1 误差及其分类	19
1.3.2 绝对误差和相对误差	19
1.3.3 有效数字	20
1.3.4 数值计算中应注意的几个减小误差的原则	20
习题一	22
第二章 物理图形和图像的计算机模拟	24
2.1 简谐振动及其合成的模拟	24
2.1.1 简谐振动的位移—时间($x-t$)曲线和速度—时间($v-t$)曲线	24
2.1.2 简谐振动的合成	25
2.2 阻尼运动与阻尼振动的模拟	28
2.2.1 阻尼情况下物体运动的速度—时间($v-t$)曲线	28
2.2.2 阻尼振动	30
2.3 驻波的模拟	31
2.4 点电荷与点电荷系的电场模拟	34
2.4.1 等势线方程	34
2.4.2 等势线 $V(x, y)=V_0$ 的绘制	35
2.4.3 点电荷系电场线图像模拟	37
2.4.4 电偶极振子电场的模拟	38
2.4.5 带电粒子在电磁场中的运动	39
2.4.6 α 粒子散射实验	41
2.5 波的干涉和衍射图形模拟	42
2.5.1 波的干涉图形模拟	42
2.5.2 等厚干涉(牛顿环)	43

2.5.3 波的衍射图形模拟	45
2.5.4 圆孔的夫琅禾费衍射	46
2.5.5 矩形孔的夫琅禾费衍射	47
习题二	47
第三章 物理学中定积分的数值计算方法	49
3.1 定积分基本数值算法及其应用	49
3.1.1 矩形法、梯形法和抛物线法(辛普森法)	49
3.1.2 电磁学中数值积分的应用	53
3.1.3 分子物理中数值积分的应用	57
3.2 龙贝格法及其应用	58
3.2.1 变步长的梯形法	58
3.2.2 变步长的辛普森求积法	59
3.2.3 龙贝格求积法	60
3.3 高斯求积法	62
3.3.1 代数精度	62
3.3.2 高斯型代数求积公式	63
3.3.3 二维高斯求积法	66
习题三	66
第四章 物理学中常微分方程初值问题的数值解法	69
4.1 物理学中的常微分方程	69
4.1.1 力学中的常微分方程	69
4.1.2 电学中的常微分方程	70
4.1.3 常微分方程数值解法的原理	70
4.2 常微分方程初值问题的欧拉近似法	71
4.2.1 一级欧拉近似法	71
4.2.2 二级欧拉近似法	73
4.3 龙格-库塔法	79
4.3.1 龙格-库塔公式	79
4.3.2 常微分方程组的求解	80
4.3.3 高阶常微分方程的求解	81
习题四	82
第五章 物理学中线性方程组的数值解法	85
5.1 物理问题与线性方程组	85
5.2 高斯消去法与列主元消去法	87
5.2.1 高斯消去法	87
5.2.2 列主元消去法	88
5.3 解三对角方程组的追赶法	91
5.4 线性方程组的迭代解法	94
5.4.1 雅可比迭代法	94
5.4.2 高斯-塞德尔迭代法	95
5.4.3 超松弛迭代法(SOR法)	99
5.5 积分方程的数值解法	100

5.5.1 积分方程的定义及分类	100
5.5.2 有限求和方法求解积分方程	101
5.5.3 几点讨论	101
习题五	103
第六章 物理学中的非线性方程求根	105
6.1 物理问题中的非线性方程	105
6.2 根的搜索和二分法	107
6.2.1 根的搜索	107
6.2.2 二分法	107
6.3 函数迭代法	111
6.4 牛顿迭代法	114
6.5 非线性方程组的迭代法	120
习题六	124
第七章 实验物理学中的插值和数据拟合	125
7.1 实验数据的拉格朗日插值法	125
7.2 差商与牛顿插值公式	128
7.2.1 差商概念	128
7.2.2 牛顿插值多项式	129
7.3 Hermite 插值	130
7.3.1 Hermite 插值公式	131
7.3.2 分段两点三次 Hermite 插值	134
7.4 三次样条插值	136
7.4.1 三次样条函数	136
7.4.2 三次样条插值多项式	137
7.5 数值微分	142
7.5.1 插值型求导公式	142
7.5.2 样条求导公式	145
7.6 最小二乘曲线拟合法	146
7.6.1 最小二乘法的一般原理	146
7.6.2 用最小二乘法求解矛盾方程组	147
7.6.3 用多项式作最小二乘曲线拟合	149
习题七	152

下篇 计算物理学

第八章 有限差分方法	157
8.1 有关物理问题与数学物理方程	157
8.1.1 方程的导出	157
8.1.2 方程的分类	158
8.1.3 边界条件和初始条件	159
8.2 有限差分原理	160
8.2.1 差商公式	160
8.2.2 差分格式的收敛性和稳定性	161

8.3 矩形域中泊松方程的有限差分法	161
8.3.1 五点差分格式	161
8.3.2 矩形域的拉普拉斯方程	162
8.4 差分方程的迭代解法	163
8.5 非矩形边界区域泊松方程的有限差分法	167
8.5.1 圆形域中泊松方程的有限差分分解	167
8.5.2 轴对称场区域泊松方程的有限差分分解	170
8.6 一维扩散方程的有限差分法	171
8.6.1 隐式六点差分格式(C-N 格式)	171
8.6.2 边界条件的差分格式	171
8.6.3 差分方程组及其求解	172
8.6.4 计算程序	173
8.7 二维扩散方程的有限差分法	174
8.7.1 交替方向隐式差分格式(ADI 格式)	174
8.7.2 边界条件的差分格式	175
8.7.3 计算程序流程图	176
8.7.4 二维显式格式	177
8.8 一维波动方程的有限差分法	178
8.8.1 显式差分格式	178
8.8.2 初值、边界条件的差分格式	179
8.8.3 计算程序流程	179
习题八	181
第九章 泛函与变分法	183
9.1 泛函与变分的基本概念	183
9.1.1 泛函的定义	183
9.1.2 函数的变分和泛函的变分	184
9.2 最简泛函的极值问题	185
9.2.1 最简泛函的欧拉方程	185
9.2.2 欧拉方程的其他解法	188
9.2.3 瑞利-里兹法求解泛函的极值问题	189
9.3 其他类型泛函的极值问题	191
9.3.1 依赖于多个函数的泛函	191
9.3.2 依赖于函数的高阶导数的泛函	192
9.3.3 依赖于多元函数的泛函	193
9.4 泛函和变分法用于微分方程边值问题	194
习题九	198
第十章 有限元方法	200
10.1 有关物理问题的变分原理	200
10.2 泊松方程的有限元方法	201
10.2.1 静电场中二维泊松方程的有限元方法	201
10.2.2 有限元方法的具体实施	203
10.2.3 计算程序	211
10.3 扩散方程的有限元方法	217

10.4 波动方程的有限元方法	218
习题十	219
第十一章 泊松方程的边界元方法	221
11.1 边界积分方程	221
11.1.1 面积分化为边界积分	221
11.1.2 关于 δ 函数的基本知识	222
11.1.3 边界积分方程	223
11.2 边界元近似	224
11.2.1 常数边界元近似	224
11.2.2 边界元方程	224
11.2.3 矩阵 H 、 G 和 B 的计算	225
11.3 单一边界下的边界元法计算程序	228
11.4 两种介质情况下的边界元方法	230
习题十一	234
第十二章 蒙特卡罗方法	235
12.1 蒙特卡罗方法基础知识	235
12.2 基本随机数的产生与检验	236
12.2.1 随机数的产生	236
12.2.2 随机数的检验	239
12.3 任意分布随机数的产生	242
12.3.1 直接抽样方法	242
12.3.2 舍选抽样方法	244
12.4 蒙特卡罗模拟方法求解粒子输运问题	248
12.4.1 蒙特卡罗模拟	248
12.4.2 直接模拟方法	250
12.4.3 简单加权方法	252
12.4.4 统计估计方法	252
12.5 随机过程的蒙特卡罗模拟	254
12.6 梅氏抽样方法及其应用	255
12.7 蒙特卡罗方法在数值分析中的应用	259
12.7.1 定积分的计算	259
12.7.2 多维积分的计算	261
12.7.3 求方程的根	263
12.7.4 偏微分方程的第一类边值问题	264
习题十二	265
第十三章 定态薛定谔方程的数值解	267
13.1 定态薛定谔方程	267
13.2 一维方势阱中粒子能级和波函数的计算机求解	268
13.3 薛定谔方程的矩阵解法	271
习题十三	278
参考文献	279

上 篇

数值方法及其在物理学中的应用

第一章 FORTRAN 语言简介与误差分析初步

1.1 FORTRAN 语言简介

FORTRAN 是世界上最早出现的高级编程语言,从 1957 年第一套 FORTRAN 编译器诞生至今已有 50 多年的历史了。随着时间的推移,FORTRAN 自身也在不断发展,从 1966 年 ANSI(American National Standards Institute,美国国家标准局)制定的第一套 FORTRAN 66 标准到 FORTRAN 77 标准,FORTRAN 就一直是世界上广泛流行的一种最适用于数值计算的面向过程的高级语言。从 1992 年正式由国际标准组织 ISO 公布的 FORTRAN 90 标准到后来的 FORTRAN 95、FORTRAN 2000、FORTRAN 2003 等,FORTRAN 的功能在不断增强,提供了指针并加入了面向对象的概念,加强了数组的功能和并行运算方面的支持。

FORTRAN 是 Formula Translator(公式翻译器)的缩写,是为解决数学问题和科学计算而提出来的。多年来的应用表明,由于 FORTRAN 本身具有标准化程度高、便于程序互换、较易优化、计算速度快等特点,使得这种高级语言目前在科学计算领域仍被广泛使用。

1.1.1 FORTRAN 语言的常量与变量

1. 常量

常量是指在计算程序中其值始终不变的量。FORTRAN 语言中的常量分为以下七类:整数型(integer)、实型(real)、双精度实型(double precision 或 $\text{real} * 8$)、复型(complex)、双精度复型(D-P-C 或 $\text{complex} * 16$)、逻辑型(logical)及字符型(character)。

整数型常量又称整型常数或整数,FORTRAN 中的整数不应包括小数点,一般用 2 个字节(16 位)来存储一个整数。如 3、-17、+19 等。

实型常量可表示为小数和指数两种形式。小数形式如 10.18, -3.17, +2.2 等。小数点前或后可以不出现数字,但不能小数点前和后都不出现数字,如 36., .19 都合法。科学计算中用到的数和小数通常用指数形式表示,如 $9.5\text{e}+8$, $3.6\text{e}-30$, $1.8\text{e}6$ 等。其中数字部分可以是不带小数点的整数形式,也可以是带小数点的形式,但指数部分不能有小数出现。计算机内存中一般用 4 个字节(32 位)来存储一个实数。

2. 变量

计算程序中有一些量的值是可以变化的,系统程序为这样的量专门开辟了一个存储单元用来存放其值,这就是变量。变量是用来存储常量的,因此变量也分为整数型、实型、双精度实型、复型、双精度复型、逻辑型、字符型。每种类型的常数应放入同种类型的变量

中。在 FORTRAN IV 中变量只有前六种类型，而无字符型变量，字符型常数可放入各种类型的变量中；在 FORTRAN 77 中有字符型变量，字符型常数只能存储在字符型变量中。

计算程序要调用变量里存储的值就必须先对变量进行识别，也就是需要事先给变量起一个名字(变量名)。在 FORTRAN 中，变量的命名规则如下：

- (1) 变量名必须以字母开头；
- (2) 变量名中字母不区分大、小写；
- (3) 变量名有效长度为 6 个字符；
- (4) 变量名字符之间可以插入空格，但空格不起作用；
- (5) FORTRAN 77 没有规定保留字，但不建议使用有特定含义的字作变量名。

1.1.2 FORTRAN 基本语句

1. 书写格式

FORTRAN 程序有两种书写格式，即固定格式(Fixed Format)和自由格式(Free Format)。扩展名为 *.f 或 *.for 的文件，默认为固定格式；扩展名为 *.f90 的文件，默认为自由格式。

固定格式的 FORTRAN 程序必须严格按照一定格式书写。编译程序时，第 1 列如果有字符 C 或 * 时，此行为注释行，不参加编译和运行；第 1~5 列为标号区，可以写 1~5 位无符号整数，标号大小没有顺序要求；程序语句写在 7~72 列中，可以从第 7 列以后的任意位置书写，一行只能写一条语句，如果一行写不完可以续行，当第 6 列上有非零或非空格的字符时，表明此行为上行的继续行；第 73~80 列为注释区，注释区不参与程序的编译，只是在打印时照常打印，方便程序设计者调试程序与检查错误。

自由格式的 FORTRAN 程序编写非常自由，对每行每列的字符没有特殊规定。编译程序时，每行可写 132 个字符；在 FORTRAN 90 格式中，也可使用 ! 来标注注释，! 后的语句也不参加编译和运行；续行标志是 &，写在行末或要续行的行首；自由格式的行号放在每行程序的最前面即可。

2. 可执行语句

1) 赋值语句

赋值语句的作用是将一个确定的值赋给一个变量。

例如： $V(\text{变量}) = e(\text{表达式})$

需要注意的是，这里的“=”不是等号，而是赋值的符号，它将“=”右边的表达式赋予左边的变量。“=”左边只能是变量名而不能是表达式。如果“=”两边类型相同，则直接赋值；如果“=”两边类型不同，则先进行右边表达式的求值，将所得结果转化为被赋值变量的类型后再进行赋值。

2) 流程控制语句

在数值计算中经常遇到的问题是需要对给定条件作逻辑判断，根据判断结果决定是否要执行某段代码，这就需要用到流程控制语句。

(1) 无条件 goto 语句。

goto k(k 为语句标号)

例如：goto 100

该语句表示流程无条件地转去运行标号为 100 的语句。标号为 100 的语句在程序中的排列位置，可以在引导到它的 goto 语句之后，也可以在该 goto 语句之前。无条件跳转语句常和其他控制语句结合起来使用。

(2) 算术条件语句。

if(e) k1, k2, k3

其中, k1, k2, k3 为语句标号; e 必须是算术表达式。当表达式运算结果 $e < 0$ 时, 程序转向标号为 k1 的语句; 当 $e = 0$ 时, 程序转向标号为 k2 的语句; 当 $e > 0$ 时, 程序转向标号为 k3 的语句。

【例 1.1】 编程求 $Y = \begin{cases} -\pi/2 & (x < 0) \\ 0 & (x = 0) \\ \pi/2 & (x > 0) \end{cases}$ 的值。要求 x 为键盘输入。

程序如下:

```

      read(*,40) x
40    format(F8.2)
      if(x)10,20,30
10    y=-1.57079
      goto 100
20    y=0
      goto 100
30    y=1.57079
      goto 100
100   write(*,50)x, y
50    format(1x, 2Hx=, F10.6, 4H, y=F10.6)
      end

```

(3) 逻辑条件语句。

if(e) s

其中, e 为逻辑表达式; s 为内嵌语句, 内嵌语句是单独的一个可执行语句。逻辑 if 语句执行时, 首先计算逻辑表达式的值。如果逻辑表达式的值为“真”, 则执行内嵌语句。若内嵌语句是非转移语句, 则执行该语句后继续按顺序往下执行; 若内嵌语句是转移语句, 则转向指定的语句。如果逻辑表达式的值为“假”, 则不执行内嵌语句, 而直接执行该语句后面的语句。

【例 1.2】 编程求 $y = \begin{cases} 0.5x + 0.95 & (x \leq 2.1) \\ 0.7x + 0.53 & (x > 2.1) \end{cases}$ 的值。

错误表示:

```

if(x.le.2.1)y=0.5*x+0.95
y=0.7*x+0.53
write(*,*)x, y

```

正确表示:

```

if(x.le.2.1) y=0.5*x+0.95
if(x.gt.2.1) y=0.7*x+0.53

```

```
write( *, * ) x, y
```

或写为

```
if(x. le. 2. 1) then
  y=0. 5 * x+0. 95
else
  y=0. 7 * x+0. 53
end if
write( *, * ) x, y
```

在这里, if 语句与 else 搭配, 用来判断当逻辑判断式不成立时, 会去执行另一段代码。即

```
if(逻辑判断式)then
  逻辑成立时, 执行这一段程序。
else
  逻辑不成立时, 执行这一段程序。
end if
```

当程序通过 if 语句判断是否顺序执行下面的语句时, 要判断逻辑表达式是否为真。关系表达式是最简单的一种逻辑表达式。常用的关系运算符有 6 个, 即在 FORTRAN 77 标准中为 .gt. (大于)、.ge. (大于或等于)、.lt. (小于)、.le. (小于或等于)、.eq. (等于)、.ne. (不等于), 而在 FORTRAN 90 标准中为 > (大于)、>= (大于或等于)、< (小于)、<= (小于或等于)、==(等于)、/=(不等于)。

逻辑表达式除了可以单纯地对两个数字比较大小之外, 还可以对两个逻辑表达式的关系进行运算。例如:

```
A. ge. 0. 0. and. A. lt. 7. 0
```

其中, .and. 是逻辑与的意思, 即 A 大于等于 0.0 和 A 小于 7 两个简单条件的组合。

常用的逻辑运算符有 5 个, 即 .and. (逻辑与)、.or. (逻辑或)、.not. (逻辑非)、.eqv. (逻辑等)、.neqv. (逻辑不等)。

【例 1.3】 有三个数 x 、 y 、 z , 要求打印出其中最大的数。

程序如下:

```
read( *, 20) x, y, z
20  format(3F10. 4)
big=x
if(y. gt. big) big=y
if(z. gt. big) big=z
write( *, * ) 'big=', big
end
```

(4) 循环 do 语句。

计算程序有时候需要重复计算一段程序代码, 这个时候需要用到循环。当需要循环的次数为已知时, 可以用 do 语句实现循环。

```
do n i=m1, m2, m3 或 do n i=m1, m2
```

即 do 标号 循环变量=表达式 1, 表达式 2[, 表达式 3]。括号内为可选项。