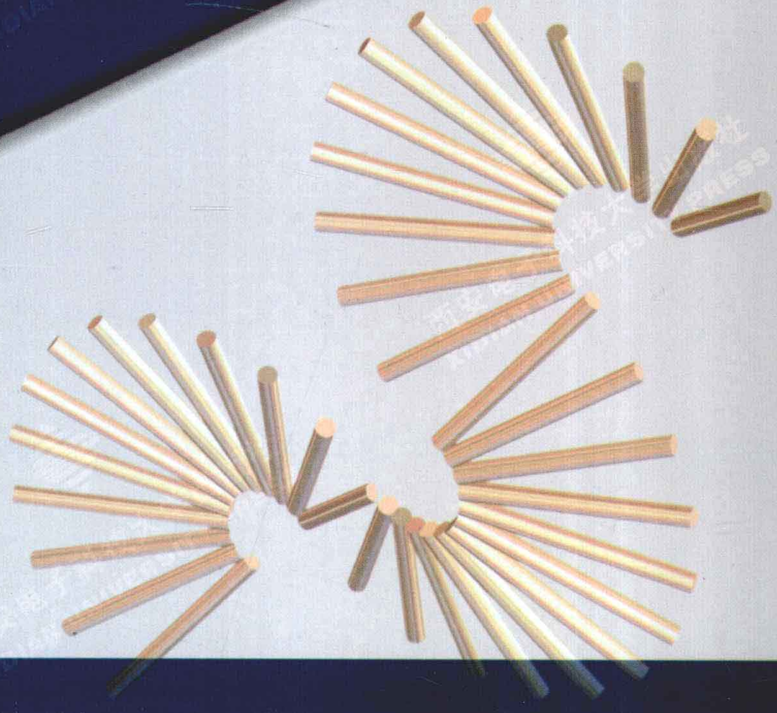
普通高等学校教材



# 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

# 计算物理学

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/O·0100

**XDUP 2625001-1**

# 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，其内容和应用越来越广泛，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

# 下篇 计算物理学

# 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，是为解决数字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 real * 8)、复型(complex)、双精度复型(D-P-C 或 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。科学计算中用到的大数和小数通常用指数形式表示，如 9.5e＋8，3.6e－30，1.8e6 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(变量)＝e(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 $y=\begin{cases}0.5x+0.95 & (x\leqslant 2.1)\\ 0.7x+0.53 & (x>2.1)\end{cases}$ 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。常用的关系运算符号有 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=$m_1$, $m_2$, $m_3$或 do n i=$m_1$, $m_2$

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。