

PASCAL 语言应用丛书之三

IBM-PC PASCAL

工程技术应用

李启炎 陈福生 杪 桢 编著

同济大学出版社

PASCAL 语言应用丛书之三

IBM-PC PASCAL

工程技术应用

陈福生 李启炎 杪桢 编著



同济大学出版社

内 容 提 要

本书结合各个工程领域中常用的三十个程序,介绍了 PASCAL 程序设计风格、程序设计技巧以及结构化程序设计方法。同时,还结合介绍了 IBM-PC PASCAL 语言数据文件的建立、源程序分离编译、动态数组处理以及与 dBASE III 的连接等实用技术。

书中所介绍的程序都在微机上调试通过,并给出使用说明及实例运行结果,读者可根据需要直接引用。

本书可作为工科大学高年级及研究生的教学参考书,也可供广大工程技术人员参考。

JS458/04

责任编辑 王建中

封面设计 王肖生

IBM-PC PASCAL 工程技术应用

陈福生 李启炎 杨程 编著

同济大学出版社出版

(上海四平路 1233 号)

新华书店上海发行所发行

吴县人民印刷二厂印刷

开本 850×1168 1/32 印张 13.5 字数 385 千字

1991年2月第一版 1991年2月第一次印刷

印数 1—5000 定价 6.00元

ISBN 7-5608-0729-1/TP·61

引 言

本书系 PASCAL 语言应用丛书之三,介绍了 PASCAL 语言的工程实际应用。PASCAL 语言以其丰富的数据结构,小巧严谨的语法规则,优良的程序风格受到广大程序设计人员和科技工作者的欢迎。各种 PASCAL 语言的实现,如 MS-PASCAL、UCSD PASCAL、Turbo PASCAL...等又大大扩充了标准 PASCAL 的功能,使其应用范围更为拓宽。从系统软件、到各种应用软件;从字符处理到各种事务处理以及图形、图像处理和各种工程应用,PASCAL 语言皆有其用武之地。本书以线性代数、回归分析、优化技术、数字滤波器设计等内容从几个不同的侧面奉献给广大读者,以介绍 PASCAL 语言的工程应用。

本书的特点是将算法介绍、程序设计方法、程序设计风格、源程序模块及其应用举例融为一体。特别将一些软件工程的基本指导思想,“自顶而下”的程序设计方法,以及追求良好程序风格的具体做法贯穿于全书的始终。这样,使读者不仅了解具体的解题算法和程序,更为重要的是能受到良好的程序设计风格的训练。为增加本书的实用性,书中所介绍的算法都是在实际应用中广为采用的、比较成熟的。给出了完整的 PASCAL 源程序和应用实例。读者如遇实际问题类同于书中实例,即可借用本书所介绍程序解决之。书中内容分为四章和二个附录。

第一章为线性代数,介绍了线性方程组的解法、矩阵求逆、特征值和特征向量的计算等内容。其中包括解线性方程组的全主元与列主元高斯消去法、平方根法、改进平方根法、高斯-赛德尔迭代法,以及大型线性方程组的分块求解法;平方根法求逆、行主元消

去法求逆;幂法和雅可比法求特征值及特征向量。

第二章为回归分析,介绍了一元与多元线性回归、逐步回归与多项式回归。

第三章介绍的是最优化和计划评审技术。其中包括线性规划、非线性规划、动态规划以及计划评审。并且加入了背包问题、运输问题、线路选择、投资优化等实际问题,使读者能领略到实际应用之精华。

第四章集中介绍了 IBM PC-PASCAL 的实用技术。其中包括分离编译技巧、高级数组的使用、文件系统的扩充、系统功能调用以及汇编语言子程序的调用。这些都反映了 PC-PASCAL 对标准 PASCAL 的扩充功能。有了这些扩充, PASCAL 语言功能更强,应用更广。

附录一和附录二分别介绍了求特征值和特征向量的 QR 方法,设计线性相位 FIR 数字滤波器的 PASCAL 程序。

全书提供了三十个有一定实用价值的 PASCAL 源程序,并附有应用实例。希望读者能从中得到一些启示和帮助。对于书中的错谬之处恳请广大读者批评指正。

目 录

第一章 线性代数	1
1.1 线性方程组解法	1
1.2 矩阵求逆	69
1.3 特征值与特征向量的计算	87
第二章 回归分析	112
2.1 一元线性回归分析	113
2.2 多元线性回归分析	122
2.3 逐步回归分析	136
2.4 多项式回归	152
第三章 最优化和计划评审技术	159
3.1 线性规划	159
3.2 非线性规划	225
3.3 动态规划	258
3.4 计划评审技术	273
第四章 IBM PC-PASCAL 实用技术	285
4.1 IBM PASCAL 分离编译单位	286
4.2 高级数组的使用	305
4.3 PASCAL 文件的使用	317
4.4 系统功能调用	351
4.5 PASCAL 程序调用汇编语言程序	352
附录一 QR 方法求实矩阵全部特征值,特征向量程序	356
附录二 设计线性相位 FIR 数字滤波器程序	378

程 序 目 录

1. 全主元高斯消去法	1
2. 列主元高斯消去法	14
3. 平方根法	14
4. 改进平方根法	22
5. 高斯-赛德尔迭代法	35
6. 分块求解大型线性方程组方法	42
7. 对称矩阵的平方根法求逆	70
8. 行主元消去法求逆矩阵(兼求行列式值)	76
9. 幂方法计算实矩阵绝对值最大特征值及特征向量	88
10. 雅可比法求实对称矩阵的特征值和特征向量	97
11. 一元线性回归分析	113
12. 多元线性回归分析	122
13. 逐步回归分析	136
14. 多项式回归	152
15. 单纯形法	161
16. 匈牙利法	177
17. 隐枚举法	190
18. 分配问题的 MACK 方法	209
19. 外推内插法	226
20. 共轭梯度法	232
21. 单纯形加速法	240
22. 拉格朗日乘子法	248
23. 投资最优化问题	266

24. 计划评审技术.....	273
25. 中值滤波.....	
26. 屏幕操作程序.....	
27. 快速富里埃变换(FFT).....	307
28. 库存管理程序.....	331
29. QR 方法求实矩阵全部特征值,特征向量程序	356
30. 设计线性相位 FIR 数字滤波器程序	378

第一章 线性代数

线性代数的数值计算是工程实践中经常遇到的问题。线性代数计算主要包括:解线性代数方程组;求逆矩阵;计算行列式值和求矩阵的特征值,特征向量等。这一章主要介绍线性代数计算中几个常用的算法及其 PASCAL 程序设计和应用。

1.1 线性方程组解法

在工程实践中提出的计算问题,差不多一半以上包括求解线性代数方程组。例如,结构应力分析问题,电力传输网分析问题,大地测量问题,数据拟合问题,电路分析问题以及其他有关非线性方程组与微分方程数值解等问题。

线性代数方程组解法很多,大致分为两大类:一类是直接法,直接法是经过有限步运算,就可求得问题的精确解;另一类是迭代法,迭代法是一种试探的方法,它是通过构造的迭代序列来逐步地逼近方程组的精确解。两种方法各有优缺点。前者仅适宜用于系数矩阵阶数不太高的问题,而后者主要用于某些高阶问题。下面所介绍的全主元高斯消去法,列主元高斯消去法,平方根法及改进平方根法,LDLT 分块求解大型线性方程组方法都是直接法;而高斯-赛德尔迭代法是迭代法中一种简单有效的方法。

一、全主元高斯消去法

1. 方法简介

消去法是解线性代数方程组的最古典,最常用的方法。消去法的基本思想是:对方程组的一些方程进行运算,把其中一部分方程

变成含有较少未知量的方程,反复运用消去过程,使原来方程组变成每个方程只含有一个未知量的方程组,于是就得到方程组的解。

全主元消去法的基本思想是在每一次消元过程中,选择系数矩阵中绝对值最大的元素(称为该次的主元),用它除所在的行,然后把它所在列上的其他非零元素均消为零。在整个消去过程中规定同一行上的元素只能一次被选作主元素。

现假定线性方程系数矩阵是 $n \times n$ 阶的矩阵 $[A]$,即线性方程组为 $[A] \cdot \{x\} = \{B\}$,则求解过程如下:

1) 选择主元素

在第一步消元前,要在 $n \times n$ 个元素中选择主元素,即在 n^2 个数中找出绝对值最大者作为主元素,记为 $a_{i_0 j_0}$ 。选择主元素采用逐行比较的方法而得到。即先从 a_{11} 开始与同一行的其他元素进行比较其绝对值,每次保留较大绝对值的元素,并记下相应的下标,然后再与第二行各元素相比,……这样直到最后一行比较结束,即得到本次的最大主元素 $a_{i_0 j_0}$,接着可消去 j_0 列上其他非零元素。在找第二次的主元素前,由于规定同一行上的元素只能一次被选作主元,故一开始就把上次选到的主元素所在行的元素排除在外。同时,经过第一步消元后,在 j_0 列上其他元素已全部为零,故第二步是在 $(n-1)^2$ 个元素中选择主元素。找到主元素后,同样进行消去 j_0 列上的非零元素,如此直到选择第 n 个即最后一个主元素并进行消去运算为止。

2) 消去运算

一旦找到主元素后,便可存放在一简单变量 c 中,而相应的行号和列号分别存放在变量 i_0 和 j_0 中,这样主元素记为 $a_{i_0 j_0}$ 。在消去运算时,主元素所在的行中各元素都要除以主元素,形成新的元素仍然放回原来位置,即该行的元素都应变为:

$$\frac{a_{i_0 j}}{a_{i_0 j_0}} \longrightarrow a_{i_0 j} \quad (j = 1, 2, \dots, n)$$

$$\frac{b_{i_0}}{a_{i_0 j_0}} \Longrightarrow b_{i_0}$$

完成上面的运算后,要把 j_0 列上的元素(除 $a_{i_0 j_0}$ 外)消为零,这只需要把第 i_0 行乘以 $(-a_{i j_0})$ 加到第 i 行即可。这是因为在第 i_0 行上 $a_{i_0 j_0}$ 已成为 1 了。所以以上运算就可以把 j_0 列上其他元素都消为零。同时,除 i_0 行外,其余各行的所有元素就变成新的元素。同样,新元素也应放回原来的位置,即:

$$a_{ij} - a_{i j_0} \cdot a_{i_0 j} \Longrightarrow a_{ij}$$

$$b_i - a_{i j_0} \cdot b_{i_0} \Longrightarrow b_i$$

$$\text{其中} \quad i = 1, 2 \cdots i_0 - 1, i_0 + 1, \cdots n$$

$$j = 1, 2, \cdots n$$

2. 结构化程序设计方法(一)

结构化程序设计是为了对付软件危机而提出的一套程序设计的原则和方法。按照这种方法设计程序,结构清晰,易于阅读、修改、调试。用这种原则和方法编出的程序称为结构化程序,按这种原则和方法定义的程序设计语言称为结构化程序语言。

结构化程序的设计包括以下内容:

- 1) 自顶向下逐步求精
- 2) 模块设计
- 3) 结构化编码

自顶向下逐步求精

结构化程序的构成反映了从抽象到具体的思维方式,自顶而下地开发程序。当我们要求解一问题时,只要对问题有足够清晰的了解和描述就可以进行设计。首先,将问题划分成若干子问题,对应于若干功能步骤(功能结点)。然后,把每个功能步骤(功能结点)再加以细化,逐步求精。凡能用伪代码形式表示出来的都写成伪代码,不能写的保留自然语言,直到什么时候自然语言全部变为伪代码,这个程序就设计出来了。在用 PASCAL 语言设计程

序时,决定一个计算步骤是否进一步精细化的关键在于:计算机执行这一步骤需要哪些 PASCAL 语句是否明确了,如果还不能立即明显地看出需要哪些 PASCAL 语言的语句,则该步算法还要进一步精细化。在每一步精细化时,一方面加入算法,一方面明确数据。因此,自顶向下逐步求精实际上是自顶向下展开一个程序,即把要求达到的功能展开成基本的程序步骤和简单的功能模块,层层细化,并且每一步都可以检验其正确性。

算法的精细化,在某种程度上与设计人员有关,当你在设计算法并把算法转换成 PASCAL 程序方面取得经验时,就会发现你要做的精细化工作就越来越少。

下面根据自顶向下逐步求精的程序设计方法来设计全主元高斯消去法 PASCAL 子程序的语句部分。

第一步:

BEGIN

置有解标志初值

逐行选择主元素并进行消去运算

END;

第二步求精:

BEGIN

置有解标志初值 flag := true;

逐行选择主元素并进行消去运算:

FOR K := 1 TO n DO

BEGIN

选择第K个主元素

判断主元素是否小于不能小于的值

进行消去运算

END;

排列解的顺序

END;

第三步求精:

BEGIN

flag: = true;

FOR k: = 1 TO n DO

BEGIN

选择第 K 个主元素:

c: = 0;

FOR i: = 1 TO n DO

BEGIN

跳过已选主元所在的行

在未选过主元素的行中找第 K 个主元

END;

保存第 K 个主元所在的行号和列号

判断主元素是否小于不能小于的值:

IF abs(c) <= eps THEN

BEGIN

显示无唯一解信息

置无唯一解标志值

转到程序结尾

END;

进行消去运算:

FOR j: = 1 TO i DO

BEGIN

主元素所在行中 j 列元素除以主元素

除主元素所在行外,其余各行进行消元

```

    END,
  { 右端项进行消去运算
END,

```

排列解的顺序:

```

    FOR k:=1 TO n DO
      a[1,jk[k]]:=b[ik[k]],
    FOR k:=1 TO n DO
      b[k]:=a[1,k],
END,

```

第四步求精:

```

BEGIN
flag:=true;
FOR k:=1 TO n DO
  BEGIN
    c:=0;
    FOR i:=1 TO n DO
      BEGIN
        跳过已选主元所在的行:
        FOR l:=1 TO k-1 DO
          { IF i=ik[l] THEN
            GOTO 100;
        在未选过主元素的行中找第K个主元:
        FOR j:=1 TO n DO

```

```

      BEGIN
         $a_{i_0 j_0} := a[i, j];$ 
        IF  $\text{abs}(a_{i_0 j_0}) > \text{abs}(c)$  THEN
          BEGIN
             $c := a_{i_0 j_0};$ 
             $i_0 = i; j_0 = j$ 
          END
        END;

```

100:

END;

保存第 K 个主元所在的行号和列号:

```

{  $ik[k] := i_0; jk[k] := j_0;$ 

```

IF $\text{abs}(c) \leq \text{eps}$ THEN

BEGIN

显示无唯一解信息:

writeln('too small');

置无唯一解标志值:

flag := false;

转到程序结尾: GOTO 300;

END;

FOR $j := 1$ TO n DO

BEGIN

主元素所在行中 j 列元素除以主元素,

FOR $l := 1$ TO k DO

IF $j = jk[l]$ THEN

GOTO 200;

$a_{10j_0} := a[i_0, j];$

IF $a_{10j_0} = 0$ THEN

GOTO 200;

$a_{10j_0} := a_{10j_0}/c;$

$a[i_0, j] := a_{10j_0};$

FOR $i := 1$ TO n DO

IF $i < j_0$ THEN

$a[i, j] := a[i, j] - a[i, j_0] * a_{10j_0}$

200;

END;

右端项进行消去运算:

$a_{10j_0} := b[i_0];$

IF $a_{10j_0} < > 0$ THEN

BEGIN

$a_{10j_0} := a_{10j_0}/c;$

$b[i_0] := a_{10j_0};$

FOR $i := 1$ TO n DO

IF $i < j_0$ THEN

$b[i] := b[i] - a[i, j_0] * a_{10j_0}$

END;

FOR $i := 1$ TO n DO

$a[i, j_0] := 0;$

END;

FOR $k := 1$ TO n DO

$a[1, jk[k]] := b[i_k[k]];$


```
FOR k:=1 TO n DO
```

```
    b[k]:=a[1,k];
```

```
300:
```

```
END;
```

3. PASCAL 子程序及说明

1) 形式参数说明

n——线性方程组阶数

a——存放系数矩阵数组,二维并按行存放

b——存放右端项一维数组,最后存放解结果。

eps——消去过程中最大主元所不能小于的值。

i_k, j_k ——工作数组,分别存放最大主元素所在的行号和列号。

flag——存放是否有唯一解标志值,flag=true,有唯一解;flag=false,无唯一解,即主元素小于不能小于的值eps。

2) PASCAL 子程序:

```
PROCEDURE gj(n: integer;eps:real;VAR a: art1;
```

```
    VAR b: art2;
```

```
    VAR flag:boolean);
```

```
LABEL 100, 200, 300;
```

```
VAR i, j, k,l,i0, j0:integer;
```

```
    ik, jk:art3;
```

```
    c, ai0j0: real;
```

```
BEGIN
```

```
    flag:=true;
```

```
    FOR k:=1 TO n DO
```