

FORTRAN 算法汇编

第二分册

刘德贵 费景高 编
于泳江 李广元

国防工业出版社

内 容 简 介

FORTRAN 算法汇编共分三个分册。第一、二分册一般是供科学和工程计算的常用算法,也包括近年来出现的新算法;第三分册是求特征值和特征向量的程序包和结构分析有限元程序包。

本书为第二分册,分六章,即第七章,线性代数计算(2)——特征值与特征向量计算;第八章,拟合与平滑;第九章,特殊函数计算;第十章,最优化计算;第十一章,概率与统计计算;第十二章,补遗及其它。书中编写的程序,均以过程(Procedure)的形式出现。这里的过程是子程序段和函数段的统称。

本书可作为从事科学和工程计算的数值工作者、科研人员、工程技术人员和管理人员的工具书,也可作为高等院校有关专业师生的教学参考书。

FORTRAN 算法汇编

第 二 分 册

刘德贵 费景高 于泳江 李广元 编

*

国防工业出版社 出版

新华书店北京发行所发行 各地新华书店经售

国防工业出版社印刷厂印装

*

787×1092¹/₃₂ 印张 18 383千字

1983年1月第一版 1984年6月第二次印刷 印数:10,301—22,900册

统一书号:15034·2025 定价:1.85元

目 录

第七章 线性代数计算 (2)

——特征值与特征向量计算	1
--------------------	---

7.1 求实对称矩阵特征值及特征向量的雅可比方法	1
7.2 用二分法求实对称三对角矩阵的特征值	9
7.3 广义特征值 ($Ax = \lambda Bx$, $ABx = \lambda x$ 等) 问题的简化	17
7.4 化一般矩阵为赫申伯格型矩阵	31
7.5 求实赫申伯格型矩阵特征值的 QR 算法	35
7.6 求一般实矩阵的全部特征值和特征向量的 QR 算法	48
7.7 一般实对称矩阵化为三对角矩阵	63
7.8 求实对称三对角矩阵特征值的 QL 算法	67
7.9 求实对称矩阵全部特征值和特征向量的 QL 算法	75

第八章 拟合与平滑

8.1 指数曲线拟合	85
8.2 多项式曲线拟合	91
8.3 正交多项式曲线拟合 (I)	96
8.4 正交多项式曲线拟合 (II)	104
8.5 切比雪夫曲线拟合	110
8.6 曲线的分段拟合	116
8.7 关于一般的非线性函数的最小二乘法曲线拟合	129
8.8 五点三次平滑	141
8.9 样条函数平滑	145

第九章 特殊函数计算

9.1 零阶或一阶贝塞尔函数	156
9.2 零阶或一阶贝塞尔函数及其导数	159
9.3 任意整数阶贝塞尔函数	165
9.4 任意分数阶贝塞尔函数	169
9.5 成组整数阶贝塞尔函数	174
9.6 整数阶球贝塞尔函数	176

IV

9.7	整数阶柱汉克耳函数	178
9.8	整数阶球汉克耳函数	182
9.9	实 Γ 函数 $\langle I \rangle$	185
9.10	实 Γ 函数 $\langle II \rangle$	189
9.11	Γ 函数的自然对数	193
9.12	不完全 Γ 函数比	196
9.13	不完全贝塔函数比	199
9.14	实误差函数	202
9.15	正态分布函数 $\langle I \rangle$	205
9.16	正态分布函数 $\langle II \rangle$	207
9.17	复误差函数	212
9.18	两类完全椭圆积分	216
9.19	两类不完全椭圆积分	220
9.20	两类完全或不完全椭圆积分	222
9.21	正弦和余弦积分	226
9.22	弗莱斯那积分	229
9.23	指数积分	233
9.24	正交多项式	235
第十章	最优化计算	242
10.1	抛物线拟合—维寻找	243
10.2	黄金分割—维寻找	253
10.3	梯度—牛顿算法	264
10.4	DFP 变度量算法	274
10.5	BFS 变度量算法	283
10.6	联合应用 DFP 和 BFS 公式的变度量算法	292
10.7	布罗伊登变度量算法	301
10.8	利用差商的变度量算法	309
10.9	PRP 共轭梯度算法	320
10.10	FR 共轭梯度算法	327
10.11	包维尔算法	334
10.12	可变多面体方法	341
10.13	可变误差多面体算法	350
10.14	求线性规划问题的改进单纯形算法	372
第十一章	概率与统计计算	392
11.1	正态分布的上概率	393

11.2	正态分布的百分点	396
11.3	χ^2 分布的上概率	398
11.4	χ^2 分布的百分点	402
11.5	F 分布的上概率	409
11.6	F 分布的百分点	412
11.7	t 分布的上概率	415
11.8	t 分布的百分点	418
11.9	贝塔分布 $\left(\frac{1}{2}\right.$ 整数倍 $\left.)$ 的分布函数	422
11.10	贝塔分布 $\left(\frac{1}{2}\right.$ 整数倍 $\left.)$ 的百分点	426
11.11	T 函数	431
11.12	多因素方差分析	435
11.13	多元线性回归分析	444
11.14	均匀分布随机数的产生 I	455
11.15	均匀分布随机数的产生 II	458
11.16	均匀分布随机数的产生 III	461
11.17	均匀分布随机数的产生 IV	465
11.18	正态分布随机数的产生 I	468
11.19	正态分布随机数的产生 II	471
11.20	泊松 (Poisson) 分布随机数的产生	474
11.21	平稳正态随机过程的模拟	478
11.22	均匀分布随机数的检验	481
11.23	任意分布随机数的连检验	486
11.24	随机数的独立性检验	490
11.25	平稳正态随机过程的检验	493
11.26	蒙特卡罗方法求多重积分	497
11.27	蒙特卡罗方法求实根	501
11.28	蒙特卡罗方法求复根	505
11.29	蒙特卡罗方法求方程组的根	510
第十二章	补遗及其它	516
12.1	打印曲线程序 I	517
12.2	打印曲线程序 II	518
12.3	上半带存储高斯消去法	520
12.4	一维变带存储高斯消去法	526
12.5	上半带存储改进平方根法 II	531

VI

12.6	一维变带存储改进平方根法 I	537
12.7	上半带存储超松弛迭代法	541
12.8	一维变带存储超松弛迭代法	546
12.9	计算三维球体上的积分	549
12.10	快速傅里叶变换 (矩阵法)	554

第七章 线性代数计算(2)

——特征值与特征向量计算

矩阵特征值与特征向量的计算在工程设计、科学研究以及计算数学本身等方面均有着重要的意义。在本章中,收集了少量的这方面比较常用的算法。其中一部分算法是围绕求实对称矩阵的特征值和特征向量的,包括雅可比(Jacobi)算法和先将矩阵化成三对角矩阵,然后用二分法或QL方法来求解的算法。另一部分算法是围绕求一般实矩阵的特征值和特征向量的,其中包括将一般矩阵化为赫申伯格(Hessenberg)型矩阵,然后用QR算法求这类矩阵的特征值,最后用回代的方法求出原矩阵的特征向量。

本章还包含一个处理广义特征值问题的程序。

7.1 求实对称矩阵特征值及特征向量的雅可比方法

一、功能

本程序应用雅可比方法,求 $N \times N$ 实对称矩阵 A 的全部特征值和全部特征向量,或只求 A 的全部特征值。

二、使用说明

1. 子程序语句

SUBROUTINE JACOBI(N,A,D,IVEC,V)

2. 哑元说明

- N 输入参数,整变量,实对称矩阵 A 的阶。
A 输入参数, $N \times N$ 个元素的二维实数组。工作开始时要求在相应的单元存放矩阵 A 的上三角部分,其

它单元可以是任意的。工作结束时保持对角线及对角线以下元素不变，其它元素已破坏。

D 输出参数，N个元素的一维实数组，存放求得特征值，但不按大小顺序排列。

IVEC 输入参数，整变量，当 IVEC > 0 时，表示要求计算全部特征向量；否则不要求计算特征向量。

V 输出参数，N * N个元素的二维实数组。当要求计算特征向量时，数组 V 按列并按数组 D 中特征值的顺序，存放特征向量；当不要求计算特征向量时，保持 V 的状态不变。

三、方法简介

任一 n 阶实对称矩阵 A 均可经一系列初等旋转变换（正交变换）化为对角阵 D 。 D 的对角线元素即为 A 的特征值。令 $A_0 = A$ ，则变换关系式如下：

$$A_{k+1} = U_k^T A_k U_k \quad k = 0, 1, \dots \quad (1)$$

其中 $U_k = U_k(p, q, \varphi)$ 为正交旋转阵。它的元素除了

$$\begin{aligned} U_{pp} &= U_{qq} = \cos \varphi \\ U_{pq} &= -U_{qp} = \sin \varphi \end{aligned} \quad (2)$$

外，其余元素与 n 阶单位矩阵相同。

记 $A_k = [a_{pq}^{(k)}]$ ，则 A_{k+1} 的第 p ， q 行和第 p ， q 列上的元素与 A_k 的相应元素不同。一般可按下面的公式计算：

$$\begin{cases} a_{pp}^{(k+1)} = c^2 \times a_{pp}^{(k)} - 2cs \times a_{pq}^{(k)} + s^2 \times a_{qq}^{(k)} \\ a_{qq}^{(k+1)} = s^2 \times a_{pp}^{(k)} + 2cs \times a_{pq}^{(k)} + c^2 \times a_{qq}^{(k)} \\ a_{pq}^{(k+1)} = (c^2 - s^2) \times a_{pq}^{(k)} + cs \times (a_{pp}^{(k)} - a_{qq}^{(k)}) \end{cases} \quad q > p \quad (3)$$

$$\begin{cases} a_{pj}^{(k+1)} = c \times a_{pj}^{(k)} - s \times a_{qj}^{(k)} \\ a_{qj}^{(k+1)} = s \times a_{pj}^{(k)} + c \times a_{qj}^{(k)} \end{cases} \quad j \neq p, q \quad (4)$$

其中 $c = \cos \varphi$, $s = \sin \varphi$ 。而 A_{k+1} 的其它元素均与 A_k 相同。

现在设要通过第 $k+1$ 次的变换来消去元素 $a_{pq}^{(k+1)}$ (即令 $a_{pq}^{(k+1)} = 0$)，则式 (3)、(4) 中的 φ 可由下式确定：

$$\operatorname{tg} 2 \varphi = -2a_{pq}^{(k)} / (a_{pp}^{(k)} - a_{qq}^{(k)}) \quad (5)$$

以上就是通常求实对称矩阵特征值的雅可比方法。

本子程序的计算步骤如下：

1) 计算

$$i\theta = \cot 2 \varphi = (a_{qq}^{(k)} - a_{pp}^{(k)}) / (2a_{pq}^{(k)}) \quad (6)$$

则 $t = \operatorname{tg} \varphi$ 为方程式

$$t^2 + 2t \times i\theta = 1 \quad (7)$$

的绝对值较小的根，即

$$t = \begin{cases} 1 / (i\theta + (1 + (i\theta)^2)^{1/2}) & i\theta \geq 0 \\ -1 / (|i\theta| + (1 + (i\theta)^2)^{1/2}) & i\theta < 0 \end{cases} \quad (8)$$

由此可以计算

$$\begin{cases} c = 1 / (1 + t^2)^{1/2} \\ s = t \times c \\ \tau = \operatorname{tg}(\varphi/2) = s / (1 + c) \end{cases} \quad (9)$$

2) 利用公式 (3)、(4) 计算 $a_{pp}^{(k+1)}$, $a_{qq}^{(k+1)}$, $a_{pq}^{(k+1)}$, $a_{pj}^{(k+1)}$, $a_{qj}^{(k+1)}$, 即

$$\begin{cases} a_{pp}^{(k+1)} = a_{pp}^{(k)} - t \times a_{pq}^{(k)} \\ a_{qq}^{(k+1)} = a_{qq}^{(k)} + t \times a_{pq}^{(k)} \\ a_{pq}^{(k+1)} = 0 \end{cases} \quad q > p \quad (10)$$

$$a_{pj}^{(k+1)} = a_{pj}^{(k)} - s \times (a_{qj}^{(k)} + \tau \times a_{pq}^{(k)})$$

$$j \neq p, q \quad (11)$$

$$a_{qj}^{(k+1)} = a_{qj}^{(k)} + s \times (a_{pj}^{(k)} - \tau \times a_{pq}^{(k)})$$

当 A_{k+1} 的对角线以上 (不包括对角线) 的元素均消成零时, 则有

$$A_{k+1} = V^T A V = D = \text{diag}(\lambda_1, \dots, \lambda_n) \quad (12)$$

其中 $\lambda_i (i = 1, \dots, n)$ 即为 A 的特征值, 而

$$V = U_0 U_1 \cdots U_k \quad (13)$$

由式 (12) 可知

$$A V = V D \quad (14)$$

故 V 的第 i 列即为相应于特征值 λ_i 的特征向量。 V 的计算采用累积的方法。按以下步骤进行:

$$V_0 = I (n \times n \text{ 单位阵})$$

$$V_1 = V_0 U_0$$

$$V_{i+1} = V_i U_i \quad i = 1, 2, \dots, k$$

$$V = V_{k+1}$$

四、程序

```

SUBROUTINE JACOBI(N,A,D,IVEC,V)
  DIMENSION A(N,N),D(N),V(N,N),B(50),Z(50)
  IF (IVEC) 10,10,20
20 DO 21 I=1,N
   DO 22 J=1,N
22   V(I,J)=0.0
21   V(I,I)=1.0
10 DO 30 K=1,N
   D(K)=A(K,K)
   B(K)=D(K)
   Z(K)=0.0
30 CONTINUE
   DO 40 I=1,50
     SM=0.0
     N1=N-1
     DO 50 IP=1,N1

```

```

      IP1 = IP + 1
      DO 50 IQ = IP1, N
50    SM = SM + ABS(A(IP, IQ))
      IF(SM) 60, 70, 60
60    IF(I - 4) 80, 90, 90
80    TRESH = 0.2*SM/FLOAT(N)**2
      GO TO 100
90    TRESH = 0.0
100   DO 110 IP = 1, N1
      IP1 = IP + 1
      DO 110 IQ = IP1, N
      G = 100.0*ABS(A(IP, IQ))
      IF(I - 4) 120, 120, 130
130   IF((ABS(D(IP)) + G) - ABS(D(IP))) 120, 140, 120
140   IF((ABS(D(IQ)) + G) - ABS(D(IQ))) 120, 150, 120
150   A(IP, IQ) = 0.0
      GO TO 110
120   IF(ABS(A(IP, IQ)) - TRESH) 110, 110, 160
160   H = D(IQ) - D(IP)
      IF((ABS(H) + G) - ABS(H)) 170, 180, 170,
180   T = A(IP, IQ)/H
      GO TO 190
170   THETA = 0.5*H/A(IP, IQ)
      T = SIGN(1.0, THETA)/(ABS(THETA) + SQRT(1. + THETA*
      ↑ THETA))
190   C = 1./SQRT(1.0 + T*T)
      S = T*C
      TAU = S/(1. + C)
      H = T*A(IP, IQ)
      Z(IP) = Z(IP) - H
      Z(IQ) = Z(IQ) + H
      D(IP) = D(IP) - H
      D(IQ) = D(IQ) + H
      A(IP, IQ) = 0.0
200   IP2 = IP - 1
      IF(IP2) 210, 210, 201

```

```

201 DO 205 K = 1, IP2
    G = A(K, IP)
    H = A(K, IQ)
    A(K, IP) = G - S*(H + G*TAU)
    A(K, IQ) = H + S*(G - H*TAU)
205 CONTINUE
210 IQ1 = IQ - 1
    IF (IP1 - IQ1) 211, 211, 220
211 DO 215 K = IP1, IQ1
    G = A(IP, K)
    H = A(K, IQ)
    A(IP, K) = G - S*(H + G*TAU)
    A(K, IQ) = H + S*(G - H*TAU)
215 CONTINUE
220 IQ1 = IQ + 1
    IF (IQ1 - N) 221, 221, 230
221 DO 225 K = IQ1, N
    G = A(IP, K)
    H = A(IQ, K)
    A(IP, K) = G - S*(H + G*TAU)
    A(IQ, K) = H + S*(G - H*TAU)
225 CONTINUE
230 IF (IVEC) 110, 110, 231
231 DO 235 K = 1, N
    G = V(K, IP)
    H = V(K, IQ)
    V(K, IP) = G - S*(H + G*TAU)
    V(K, IQ) = H + S*(G - H*TAU)
235 CONTINUE
110 CONTINUE
    DO 240 K = 1, N
        B(K) = B(K) + Z(K)
        D(K) = B(K)
        Z(K) = 0.0
240 CONTINUE
40 CONTINUE

```

END

五、程序附注

1. 本程序实现时实际上是修正的雅可比方法。若按行顺序对每个非对角线元素都进行一次消去的旋转变换称为一次扫描的话（共有 C_n^2 个旋转变换），在前三次扫描中，设当前是为了消去 a_{pq} ，则只当

$$|a_{pq}| > tresh = \left(0.2 \times \sum_{i=1}^{n-1} \sum_{j=i+1}^n |a_{ij}| \right) / n^2$$

时，才进行变换；否则跳过该变换不执行。从第四次扫描起，置 $tresh = 0$ 。从第五次扫描起，若当前要消去的元素是 a_{pq} ，而 a_{pq} 的值小到使 $|a_{pp}|$ 和 $|a_{qq}|$ 的增长不起作用时，则置 $a_{pq} = 0$ ，而跳过当前的旋转变换不执行。

2. 本子程序进行到变换对所有对角线的元素绝对值的的增长无贡献时为止。此时，对角线元素即为原矩阵 A 的特征值。

3. 当矩阵 A 的阶 N 大于 50 时，需将维数语句中的 50 改成相应的数。

六、例题

求矩阵

$$A = \begin{pmatrix} 10 & 1 & 2 & 3 & 4 \\ 1 & 9 & -1 & 2 & -3 \\ 2 & -1 & 7 & 3 & -5 \\ 3 & 2 & 3 & 12 & -1 \\ 4 & -3 & -5 & -1 & 15 \end{pmatrix}$$

的特征值和特征向量。

计算结果

$$\lambda_1 = 0.6994837830_{10}^1$$

$$\lambda_2 = 0.9365554918_{10}^1$$

$$\lambda_3 = 0.1655266206_{10}^1$$

$$\lambda_4 = 0.1580892074_{10}^2$$

$$\lambda_5 = 0.1917542026_{10}^2$$

V_1	V_2	V_3
0.6540829840 ₁₀ ⁰	-0.5215111784 ₁₀ ⁻¹	-0.3872968746 ₁₀ ⁰
0.1996812688 ₁₀ ⁰	0.8599638666 ₁₀ ⁰	0.3662210210 ₁₀ ⁰
0.2565104562 ₁₀ ⁰	-0.5055750724 ₁₀ ⁰	0.7043772652 ₁₀ ⁰
-0.6604027222 ₁₀ ⁰	-0.2011666316 ₁₀ ⁻³	-0.1189262220 ₁₀ ⁰
-0.1742798634 ₁₀ ⁰	0.4621919960 ₁₀ ⁻¹	0.4534231080 ₁₀ ⁰
V_4	V_5	
0.6237024996 ₁₀ ⁰	0.1745051092 ₁₀ ⁰	
0.1591011206 ₁₀ ⁰	-0.2473025186 ₁₀ ⁰	
0.2272974940 ₁₀ ⁰	-0.3616417392 ₁₀ ⁰	
0.6926843856 ₁₀ ⁰	-0.2644108530 ₁₀ ⁰	
0.2328222836 ₁₀ ⁰	0.8412440690 ₁₀ ⁰	

试通程序

PAGE 1

SUBROUTINE JACOBI (N,A,D,IVEC,V)

: 本子程序段体部分

END

PAGE 10

MASTER EXAMPI

DIMENSION A(5,5),D(5),V(5,5)

READ(1)A

CALL JACOBI(5,A,D,1,V)

WRITE(4,1)D,V

STOP

1 FORMAT(30 E 17.10)

END

FINISH

参 考 资 料

- [1] H. Rutishauser, The Jacobi method for real symmetric matrices, Numer. Math, 9, 1966, pp. 1-10.

7.2 用二分法求实对称三对角矩阵的特征值

一、功能

本程序可用来计算 n 阶实对称三对角矩阵 A 的全部或部分特征值。当矩阵的阶数较高且只需计算一部分特征值时, 用这程序比较有利。对于非对称 n 阶三对角矩阵 $A=[a_{ij}]$, 若 $a_{i,i+1}=f_i, a_{i+1,i}=g_i$ 满足 $f_i g_i \geq 0 \quad i=1, 2, \dots, n-1$ 时, 则取 $b_{i+1}^2=f_i g_i$ 也可以用本程序处理。

二、使用说明

1. 子程序语句

SUBROUTINE BISECT(N,D,B,BETA,M1,M,EPS1,EPS2,X)

2. 哑元说明

- N 输入参数, 整变量, 实对称三对角矩阵 A 的阶。
- D 输入参数, N 个元素的一维实数组, 存放 A 的主对角线元素。
- B 输入参数, N 个元素的一维实数组, 在 $B(2), \dots, B(N)$ 中存放 A 的次对角线的元素, 而 $B(1)=0$ 。
- BETA 输入参数, N 个元素的一维实数组, 存放 A 的次对角线元素的平方, 即 $BETA(I)=(B(I))^2, I=1, 2, \dots, N$ 。
- M1 输入参数, 整变量。当 A 的全部特征值按值的大小顺序排列时 (最小的序号为 1), $M1$ 为所要计算的

相继 M 个特征值的第一个特征值的序号数。

- M 输入参数, 整变量, M 为所要计算的特征值的个数。
若以 λ_i 表示第 i 个特征值, 则要求本程序计算出 $\lambda_{M1}, \lambda_{M1+1}, \dots, \lambda_{M1+M-1}$ 。
- EPS1 输入参数, 实变量。当 EPS1 是大于零的充分小的数时, 它作为特征值精度 (绝对误差) 控制的一个参量; 当 EPS1 小于或等于零时, EPS1 不起作用, 由程序本身给出精度参数来代替 EPS1。
- EPS2 输出参数, 实变量, 所求的特征值的误差的一个上界。
- X 输出参数, M 个元素的一维实数组, 存放所求得特征值。

三、方法简介

设要求特征值的实对称三对角矩阵 A 的主对角线元素为 $d_1, d_2, \dots, d_n$, 次对角线元素为 $b_2, b_3, \dots, b_n$ (可令 $b_1 = 0$ 而成一 n 维向量), 则对任意 λ , 可得一特征多项式的斯脱姆 (Sturm) 序列 $\{p_i(\lambda)\}_{i=0}^n$, $p_i(\lambda)$ 可有递推式

$$\begin{aligned} p_0(\lambda) &= 1 & p_1(\lambda) &= d_1 - \lambda \\ p_i(\lambda) &= (d_i - \lambda)p_{i-1}(\lambda) - b_i^2 p_{i-2}(\lambda) \end{aligned} \quad (1)$$

$$i = 2, \dots, n$$

若序列 $\{p_i(\lambda)\}_{i=0}^n$ 的相邻两个数之间的变号次数为 $a(\lambda)$, 则表示矩阵 A 小于 λ 的特征值的个数有 $a(\lambda)$ 个。斯脱姆序列的这个性质, 可以用来分隔 A 的每个特征值。

假定 A 的所有特征值为 $\lambda_1 < \lambda_2 < \dots < \lambda_n$, 并设当前要计算第 k 个特征值 λ_k , 并已知其上下界为 b_0, a_0 ($b_0 > a_0$), 于是用二分法经过 p 步找出 λ_k 所在的区间 (a_p, b_p) , 则区

间的长度为 $(b_0 - a_0)/2^p$ 。这个逐次二分的过程可叙述如下：

设当前是进行第 r 步二分，令 c_r 为区间 $[a_{r-1}, b_{r-1}]$ 的中点，即

$$c_r = \frac{1}{2}(a_{r-1} + b_{r-1}) \quad (2)$$

于是可计算序列 $\{p_i(c_r)\}_{i=0}^n$ ，并确定变号数 $a(c_r)$ 。再按条件

若 $a(c_r) \geq k$ ，令 $a_r = a_{r-1}$ ， $b_r = c_r$ ；

若 $a(c_r) < k$ ，令 $a_r = c_r$ ， $b_r = b_{r-1}$ ；

确定区间 $[a_r, b_r]$ 。由于一定有

$$a(a_r) < k \text{ 和 } a(b_r) \geq k,$$

所以， λ_k 在区间 $[a_r, b_r]$ 上。

关于特征值 λ_k 的初始上下界 b_0 ， a_0 ，在本程序中分别由下式确定。

$$\begin{cases} b_0 = \max\{d_i \pm (|b_i| + |b_{i+1}|)\} \\ a_0 = \min\{d_i \pm (|b_i| + |b_{i+1}|)\} \end{cases} \quad i = 1, 2, \dots, n \quad (3)$$

其中令 $b_1 = b_{n+1} = 0$ 。

上述方法一般来说是很稳定的。但仍有可能产生“下溢”或“上溢”。特别当一些特征值非常接近时，可能性更大。为了克服这个缺点，本程序采用下面的序列 $\{q_i(\lambda)\}$ 代替序列 $\{p_i(\lambda)\}$ ，即

$$q_i(\lambda) = p_i(\lambda)/p_{i-1}(\lambda) \quad i = 1, 2, \dots, n \quad (4)$$

它满足下面的关系：

$$q_1(\lambda) = d_1 - \lambda$$

$$q_i(\lambda) = (d_i - \lambda) - b_i^2/q_{i-1}(\lambda) \quad i = 2, \dots, n \quad (5)$$

这时， $a(\lambda)$ 就由序列 $\{q_i(\lambda)\}$ 中负的 $q_i(\lambda)$ 的个数确定。