计算机科学资料

# DJS——21机

# 标准程序汇编

## 第三分册

中国科学院数学研究所计算站

1975.1

# 第三分册

插　　值

数值积分

特殊函数的计算

函数逼近

# 序

随着我国社会主义建设的发展，电子计算机在我国的使用已日益普遍。近年来，由于语言程序的采用，节约了大量编制程序的人力，从而为更加广泛地使用电子计算机开辟了新的前景。

为了普及语言程序，推进程序自动化，减少编制程序时的重复劳动，我们收集了国外近年来发表的常用ALGOL—60标准程序以及国内一些兄弟单位编制的程序，并且结合我们自己的计算实践汇编成本书。

本书共分五个分册。第一分册为线性代数部分（包括矩阵求逆、求行列式的值、线性方程组求解、求特征值及特征向量、求解线性规划等内容）。第二分册包括常微分方程、超越方程及多项式方程的求解。第三分册有插值、数值积分、函数逼近、特殊函数计算等内容。第四、五分册分别收集了一些有关运筹学、概率统计方面的程序。在各分册中，均举例说明如何使用这些程序。

由于我们水平有限，工作做得也不多，加之是集体编写，分头执笔所以各篇中所用的符号与叙述方式都不尽一致，错误也在所难免。希望读者给予批评与指正。并将使用中发生的问题和改进方案告诉我们。尤其希望读者能不断地向我们提供新的程序，以便编制标准程序的工作逐步完善。

最后附带说明几点：

1．本书提供的程序是根据国产DJS—21机的算法语言编写的，各篇均可作标准过程运用于程序之中。并在我所机器上考核通过。对于其他型号的机器，需根据机器的特殊要求略加修改。

2．由于现在DJS—21机的编译程序尚不十分完备，非正常出口 FAIL 在编译过程中会打出错误讯息F(⊠Z96……当仅有这类错误时，请把选择开关6按下（$K_6=1$），程序即可通过。

3．我站对DJS—21机的编译程序作了一些修改与补充，例如原编译程序最多只允许有九个形参而我们的编译程序增至十个形参等等。希读者在使用时注意。

编　者

1974年元月

# 目　录

## 插　值



## 数 值 积 分



## 特 殊 函 数 的 计 算

## 函 数 逼 近

# 拉格朗日插值算法

## §1 数学方法简述

已知一元函数 $F(x)$ 的 $n+1$ 个结点 $a_i(i=0,1,\cdots,n)$ 及其对应的结点函数值 $y_j(j=0,1,\cdots,n)$，对于插值区间内任一点 $x$ 都可按下列公式求出与之对应的函数值 $F$。

$$F=\sum_{j=0}^{n}\prod_{\substack{i=0\\ i\neq j}}^{n}\left(\frac{x-a_i}{a_j-a_i}\right)y_j$$

## §2 LAG1过程

### 2.1 参数表

**输入参数**

N　已知插值点的个数减1。

A　一维数组A[0:N]，即已知插值结点值。

Y　一维数组Y[0:N]，即已知插值结点上的函数值。

X　被插值点值。

**输出参数**

F　所求之被插值点X的函数值。

### 2.2 程　序

```
'PROC'  LAG1(N,A,Y,X,F);
  'VALUE' N,X;          'INTEG' N;
  'REAL'  X,F;          'ARRAY' A,Y;
'BEGIN' 'REAL' I,J,L;
  F:=0;
  'FOR' J:=0 'STEP' 1 'UNTIL' N 'DO'
  'BEGIN'  L:=1;
    'FOR' I:=0 'STEP' 1 'UNTIL' N 'DO'
```

```
    'IF' I 'NQ' J 'THEN'
    L:=L×(X-A[I])/(A[J]-A[I]);
    F:=F+L×Y[J]
  'END'
'END';
```

## §3 数值结果

例：已知插值结点及其对应的函数值如下：

$a_i$：　−2；　−0.4；　　−0.2；　　1；　　4；

$y_i$：　24；　−0.2688；　−0.0768；　0；　480；

求当X=−1.5；　−1；　0.42时的函数值F.

套用 LAG 1 过程 DJS—21机计算结果：

F(−1.5)=5.6249999，　F(−1)=$-0.34924596\times10^{-9}$，

F(0.42)=−0.29056607.

# 埃特金逐步插值法

## §1 数学方法简述

用拉格朗日公式计算插值多项式在某点上的数值的缺点是：当我们增加插值点时计算工作必须从头做起。例如：我们已经用插值点：$a_0$，$a_1$，$a_2$ 计算出了 $l_2(x_0)$以后，还想再计算用 $a_0$，$a_1$，$a_2$，$a_3$ 作插值点的插值多项式 $l_3(x)$的值 $l_3(x_0)$时，很显然，原来的 $l_2(x_0)$ 就不能使用，这时需重新代入拉格朗日公式进行计算，这样做给计算工作带来很大不便，埃特金逐步插值法弥补了这个缺点，计算过程如下：

设：$I_0(x)=Y_0$，$I_1(x)=Y_1$，…，

$I_{0,1}(x)$　　表示用 $a_0$，$a_1$ 作插值点的一次插值多项式。

$I_{0,2}(x)$　　表示用 $a_0$，$a_2$ 作插值点的一次插值多项式。

$I_{1,2}(x)$　　表示用 $a_1$，$a_2$ 作插值点的一次插值多项式。

……

$I_{0,1,2}(x)$　　表示用 $a_0$，$a_1$，$a_2$ 作插值点的二次插值多项式。

$I_{0,1,3}(x)$　　表示用 $a_0$，$a_1$，$a_3$ 作插值点的二次插值多项式。

……

$I_{0,1,\cdots,k}(x)$　　表示用 $a_0$，$a_1$，…，$a_k$ 作插值点的 $k$ 次插值多项式。

则有下面关系式成立：

$$I_{0,1}(x)=\frac{1}{a_0-a_1}\begin{vmatrix} I_0(x) & x-a_0 \\ I_1(x) & x-a_1 \end{vmatrix}=\frac{(x-a_0)y_1-(x-a_1)y_0}{a_1-a_0}$$

$$I_{0,1,2}(x)=\frac{1}{a_1-a_2}\begin{vmatrix} I_{0,1}(x) & x-a_1 \\ I_{0,2}(x) & x-a_2 \end{vmatrix}=\frac{1}{a_0-a_2}\begin{vmatrix} I_{0,1}(x) & x-a_0 \\ I_{1,2}(x) & x-a_2 \end{vmatrix}=\frac{1}{a_0-a_1}\begin{vmatrix} I_{0,2}(x) & x-a_0 \\ I_{1,2}(x) & x-a_1 \end{vmatrix}$$

……

$$I_{0,1,\cdots,k}(x)=\frac{1}{a_q-a_p}\begin{vmatrix} I_{0,1,\cdots,p-1,p+1,\cdots,k}(x) & x-a_q \\ I_{0,1,\cdots,q-1,q+1,\cdots,k}(x) & x-a_p \end{vmatrix}$$

故由上面各关系式可知：求 $I_{0,1,\cdots,k}(x)$的值的问题实际上已经化成一系列线性插值问题了。

## §2 AITKEN 过程

### 2.1 参 数 表

**输入参数**

N　已知结点的个数减1.

A　一维数组 A[0:N]，即已知的插值结点值。

Y　一维数组 Y[0:N]，即已知插值结点上的函数值。

X　被插值点值。

**输出参数**

F　所求之被插值点X的函数值。

## 2.2 程　序

```
'PROC'  AITKEN(N,A,Y,X,F);
  'VALUE' N,X;        'INTEG' N;
  'REAL' X,F;         'ARRAY' A,Y;
'BEGIN' 'INTEG' I,J;
  'FOR' J:=0 'STEP' 1 'UNTIL' N-1 'DO'
  'FOR' I:=J+1 'STEP' 1 'UNTIL' N 'DO'
  Y[I]:=((X-A[J])×Y[I]-(X-A[I])×Y[J])/(A[I]-A[J]);
  F:=Y[N]
'END';
```

# §3 数值结果

例：求正弦积分：$\mathrm{sin}(x)=-\int_x^{\infty}\frac{\sin t}{t}dt$

当 $x=x_0=0.462$ 时的值。

已知：$A_i$：0.3；　0.4；　0.5；　0.6；　0.7；

$Y_i$：0.29850；0.39646；0.49311；0.58813；0.68122；

X：0.462；

N：4；

套用 AITKEN 过程 DJS—21 机计算结果、F(0.462)=0.45655811.

# 一 元 三 点 插 值 算 法

## §1 数学方法简述

已知一元函数 $F(x)$ 的 $n$ 个插值结点 $a_j(j=1,\cdots,n)$ 及其对应的结点函数值 $y_i(i=1,\cdots,n)$ 对于插值区间内任一点 $x$，由下式可求得与之对应的函数值 $F$。

$$F=\sum_{i=k}^{k+2}\prod_{\substack{j=k\\ j\neq i}}^{k+2}\left(\frac{x-a_j}{a_i-a_j}\right)y_i$$

其中 $a_k, a_{k+1}, a_{k+2}$ 为最靠近 $x$ 的三个已知结点。

## §2 LAG13 过程

### 2.1 参 数 表

#### 输入参数

N　已知插值结点的个数。

A　一维数组 A[1:N]，即已知插值结点值。

Y　一维数组 Y[1:N]，即已知插值结点上的函数值。

X　被插值点值。

#### 输出参数

F　所求之被插值点的函数值。

### 2.2 程　　序

```
'PROC' LAG13(N,A,Y,X,F);
  'VALUE' N,X;        'REAL' X,F;
  'ARRAY' A,Y;        'INTEG' N;
'BEGIN' 'REAL' I,K,H,J;
  'FOR' I:=1 'STEP' 1 'UNTIL' N-1 'DO'
  'IF' (X-A[I])×(X-A[I+1]) 'LQ' 0 'THEN'
  'BEGIN' K:=I; 'GOTO' L1 'END';
```

```
'IF' ABS(X-A[1]) 'LS' ABS(X-A[N]) 'THEN' K:=1
'ELSE' K:=N-1;
L1: 'IF' K=N-1 'OR' K 'NQ' 1 'AND' ABS(X-A[K]) 'LS'
          ABS(X-A[K+1])
  'THEN' K:=K-1;
  F:=0;
  'FOR' I:=K 'STEP' 1 'UNTIL' K+2 'DO'
  'BEGIN' H:=1;
    'FOR' J:=K 'STEP' 1 'UNTIL' K+2 'DO'
    'IF' J 'NQ' I 'THEN'
    H:=H×(X-A[J])/(A[I]-A[J]);
    F:=F+H×Y[I]
  'END'
'END';
```

### 2.3 说　明

1. 本过程的功能是：根据一元函数的已知结点及其对应的结点函数值从中选取与 $x$ 最靠近的三点及其对应的结点函数值按§1所述之公式计算出函数值 $F$.

2. 本过程只适用于结点较密以及结点函数值相差较小的情况。

3. 如果仅仅对函数 $F(x)$ 进行内插时，关于K的选取可去掉：

```
'IF' ABS(X-A[1]) 'LS' ABS(X-A[N]) 'THEN' K:=1
'ELSE' K:=N-1;
```

其结果不变。

但当 $x=0.421$ 时，(见例题) 若去掉上列语句将会产生动态错(即：下标超界)。

## §3 数值结果

例：已知一元函数 $F(x)$ 的六个结点及其对应的结点函数值（如下表）。求当 $x=0.29$；0.38；0.42 时的函数值。

```
A[I]: 0.2;      0.24;     0.28;     0.32;     0.36;     0.40;
Y[I]: 0.19867;  0.23770;  0.27636;  0.31457;  0.35227;  0.38942;
N:    6;
X:    0.29;     0.38;     0.42;
```

套用LAG13过程DJS—21机计算结果：

F(0.29)=0.285954680,

F(0.38)=0.370913750,

F(0.42)=0.407788750.

# 二 元 n 点 插 值 算 法

## §1 数学方法简述

设已知二元函数 $w(x,y)$ 的第一变元结点值为 $a_i(i=1,\cdots,n)$，第二变元的结点值为 $b_j(j=1,\cdots,m)$，其对应结点上的函数值为 $z_{ij}(i=1,\cdots,n;\ j=1,\cdots,m)$，对于给定的不是结点的变元值 $p(x,y)$可用下列公式计算出与其对应的函数值$w$。

$$w=\sum_{j=1}^{m}\sum_{i=1}^{n}\prod_{\substack{k=1\\k\neq i}}^{n}\left(\frac{x-a_k}{a_i-a_k}\right)\prod_{\substack{l=1\\l\neq j}}^{m}\left(\frac{y-b_l}{b_j-b_l}\right)z_{ij}$$

## §2 LAG2 过程

### 2.1 参 数 表

**输入参数**

N　已知 $x$ 向插值结点的个数。

M　已知 $y$ 向插值结点的个数。

P　被插值点值。

A　一维数组 A[1:N]，即已知 $x$ 向插值结点的值。

B　一维数组 B[1:M]，即已知 $y$ 向插值结点的值。

Z　二维数组 Z[1:N,1:M]，即已知插值结点上的函数值。

**输出参数**

W　所求之被插值点的函数值。

### 2.2 程　　序

```
'PROC' LAG2(N,M,X,Y,A,B,Z,W);
  'VALUE' N,M,X,Y;          'INTEG' M,N;
  'REAL' X,Y,W;             'ARRAY' A,B,Z;
'BEGIN' 'REAL' I, LP, K, J, LR, L;
  W:=0;
  'FOR' J:=1 'STEP' 1 'UNTIL' M 'DO'
```

```
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
'BEGIN' LP:=1;
  'FOR' K:=1 'STEP' 1 'UNTIL' N 'DO'
  'IF' K 'NQ' I 'THEN'
  LP:=LP×(X-A[K])/(A[I]-A[K]);
  LR:=LP;
  'FOR' L:=1 'STEP' 1 'UNTIL' M 'DO'
  'IF' L 'NQ' J 'THEN'
  LR:=LR×(Y-B[L])/(B[J]-B[L]);
  W:=W+LR×Z[J,I]
'END'
'END';
```

## §3 数值结果

例：已知结点 $(x,y)$ 及其对应的结点函数值 $Z(x,y)$（如下表），求下列各点：$(-0.9,-0.7)$；$(0.5,-0.3)$；$(0.1,0.1)$；$(0.7,0.9)$；$(0.2,-0.6)$的函数值$W$.

| $Z(x,y)$ $y$ \ $x$ | -1 | -0.8 | -0.6 | -0.4 | -0.2 | 0 | 0.2 | 0.4 | 0.6 | 0.8 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| -1 | 9 | 8.04 | 7.16 | 6.36 | 5.64 | 5 | 4.44 | 3.96 | 3.56 | 3.24 | 3 |
| -0.8 | 9.04 | 8.12 | 7.28 | 6.52 | 5.84 | 5.24 | 4.72 | 4.28 | 3.92 | 3.64 | 3.44 |
| -0.6 | 9.16 | 8.28 | 7.48 | 6.76 | 6.12 | 5.56 | 5.08 | 4.68 | 4.36 | 4.12 | 3.96 |
| -0.4 | 9.36 | 8.52 | 7.76 | 7.08 | 6.48 | 5.96 | 5.52 | 5.16 | 4.88 | 4.68 | 4.56 |
| -0.2 | 9.64 | 8.84 | 8.12 | 7.48 | 6.92 | 6.44 | 6.04 | 5.72 | 5.48 | 5.32 | 5.24 |
| 0 | 10 | 9.24 | 8.56 | 7.96 | 7.44 | 7 | 6.64 | 6.36 | 6.16 | 6.04 | 6 |
| 0.2 | 10.44 | 9.72 | 9.08 | 8.52 | 8.04 | 7.64 | 7.32 | 7.08 | 6.92 | 6.84 | 6.84 |
| 0.4 | 10.96 | 10.28 | 9.68 | 9.16 | 8.72 | 8.36 | 8.08 | 7.88 | 7.76 | 7.72 | 7.76 |
| 0.6 | 11.56 | 10.92 | 10.36 | 9.88 | 9.48 | 9.16 | 8.92 | 8.76 | 8.68 | 8.68 | 8.76 |
| 0.8 | 12.24 | 11.64 | 11.12 | 10.68 | 10.32 | 10.04 | 9.84 | 9.72 | 9.68 | 9.72 | 9.84 |
| 1 | 13 | 12.44 | 11.96 | 11.56 | 11.24 | 11 | 10.84 | 10.76 | 10.76 | 10.84 | 11 |

套用LAG2 过程 DJS—21 机计算结果：

W(-0.9,-0.7)=8.63,

W(0.5,-0.3)=5.29,

W(0.1,0.1)=7.1299999,

W(0.7,0.9)=10.23,

W(0.2,-0.6)=5.08.

# 二 元 三 点 插 值 算 法

## §1 数学方法简述

设已知二元函数$W(x,y)$的第一变元结点值为 $a_i(i=1,\cdots,n)$，第二变元的结点值为 $b_j(j=1,\cdots,m)$，其对应结点上的函数值为$Z_{ij}(i=1,\cdots,n;\ j=1,\cdots,m)$，对于给定的不是结点的变元值 $P(x,y)$可用下列公式计算出与其对应的函数值$W$.

$$W=\sum_{j=q}^{q+2}\sum_{i=p}^{p+2}\prod_{\substack{k=p\\k\neq i}}^{p+2}\left(\frac{x-a_k}{a_i-a_k}\right)\prod_{\substack{l=q\\l\neq j}}^{q+2}\left(\frac{y-b_l}{b_j-b_l}\right)Z_{ij}$$

其中 $a_p,a_{p+1},a_{p+2}$ 和 $b_q,b_{q+1},b_{q+2}$ 分别为最靠近 $x$ 和 $y$ 的三个结点值

## §2 LAG23 过程

### 2.1 参 数 表

**输入参数**

N　已知 $x$ 向插值点的个数。

M　已知 $y$ 向插值点的个数。

P　被插值点值。

A　一维数组A[1:N]，即已知 $x$ 向的插值结点值。

B　一维数组B[1:M]，即已知 $y$ 向的插值结点值。

Z　二维数组Z[1:N,1:M]，即已知插值结点上的函数值。

**输出参数**

W　所求之被插值点的函数值。

### 2.2 程　　序

```
'PROC' LAG23(N,M,X,Y,A,B,Z,W);
  'VALUE' N,M,X,Y;          'REAL' X,Y,W;
  'INTEG' M,N;              'ARRAY' A,B,Z;
'BEGIN' 'REAL' I,P,Q,J,L,K,H,H1;
  'FOR' I:=1 'STEP' 1 'UNTIL' N-1 'DO'
```

```
    'IF' (X-A[I])×(X-A[I+1]) 'LQ' 0 'THEN'
    'BEGIN' P:=I; 'GOTO' L1 'END';
    'IF' ABS(X-A[1]) 'LS' ABS(X-A[N])
    'THEN' P:=1 'ELSE' P:=N-1;
L1: 'FOR' J:=1 'STEP' 1 'UNTIL' M-1 'DO'
    'IF' (Y-B[J])×(Y-B[J+1]) 'LQ' 0 'THEN'
    'BEGIN' Q:=J; 'GOTO' L2 'END';
    'IF' ABS(Y-B[1]) 'LS' ABS(Y-B[M])
    'THEN' Q:=1 'ELSE' Q:=M-1;
L2: 'IF' P=N-1 'OR' P 'NQ' 1 'AND' ABS(X-A[P]) 'LS' ABS(X-A[P+1])
      'THEN' P:=P-1;
    'IF' Q=M-1 'OR' Q 'NQ' 1 'AND' ABS(Y-B[Q]) 'LS' ABS(Y-B[Q+1])
    'THEN' Q:=Q-1;      W:=0;
    'FOR' J:=Q 'STEP' 1 'UNTIL' Q+2 'DO'
    'BEGIN' H:=1;
      'FOR' L:=Q 'STEP' 1 'UNTIL' Q+2 'DO'
      'IF' L 'NQ' J 'THEN'
      H:=H×(Y-B[L])/(B[J]-B[L]);
      'FOR' I:=P 'STEP' 1 'UNTIL' P+2 'DO'
      'BEGIN' H1:=H;
        'FOR' K:=P 'STEP' 1 'UNTIL' P+2 'DO'
        'IF' K 'NQ' I 'THEN' H1:=H1×(X-A[K])/(A[I]-A[K]);
        W:=W+H1×Z[J,I]
      'END'
    'END'
'END';
```

### 2.3 说　明

1. 本过程的功能是：根据二元函数$W(x,y)$的已知结点以及对应的结点函数值，分别从中选取与$P(x,y)$最为靠近的三个结点以及对应的结点函数值按§1所述之公式对二元函数$W(x,y)$进行插值。

2. 本过程只适用于结点较密以及结点函数值相差较小的情况。

3. 如果仅仅是对函数$W(x,y)$进行内插时，关于$K$值的选取那段程序中可以去掉：

```
'IF' ABS(X-A[1]) 'LS' ABS(X-A[N])
'THEN' P:=1 'ELSE' P:=N-1;
'IF' ABS(Y-B[1]) 'LS' ABS(Y-B[M])
'THEN' Q:=1 'ELSE' Q:=M-1;
```

其结果不变。

## §3 数值结果

例：将二元 $n$ 点插值算法一节中所举之例题，用本节算法重新进行计算（数据见P 8 ）并将所得结果与上节进行比较。

套用LAG23 过程 DJS—21 机计算结果：

$W(-0.9,-0.7)=8.63$,

$W(0.5,-0.3)=5,2899999$,

$W(0.1,0.1)=7.1299999$,

$W(0.7,0.9)=10.23$,

$W(0.2,-0.6)=5.08$.

故与上节算法所得之结果基本相同。

# 带导数值的HERMITE插值

## §1 数学方法简述

给定 $n$ 个已知的互不相同的点 $x_1, x_2, \cdots, x_n$，要求一个次数不高于 $2n-1$ 的多项式 $H(x)$，使 $H(x)$ 和导数 $H'(x)$ 在这些点都取事先给定的值：

$$H(x_i)=y_i,\ H'(x_i)=y'_i,\ i=1,2,\cdots,n.$$

计算公式：

$$H(x)=\sum_{i=1}^{n} y_i h_i(x)+\sum_{i=1}^{n} y'_i H_i(x)$$

其中

$$H_i(x)=(x-x_i)\frac{(x-x_1)^2\cdots(x-x_{i-1})^2(x-x_{i+1})^2\cdots(x-x_n)^2}{(x_i-x_1)^2\cdots(x_i-x_{i-1})^2(x_i-x_{i+1})^2\cdots(x_i-x_n)^2}$$

$$h_i(x)=\left\{1-2(x-x_i)\sum_{\substack{j=1\\ j\neq i}}^{n}\frac{1}{x_i-x_j}\right\}\times$$

$$\times\frac{(x-x_1)^2\cdots(x-x_{i-1})^2(x-x_{i+1})^2\cdots(x-x_n)^2}{(x_i-x_1)^2\cdots(x_i-x_{i-1})^2(x_i-x_{i+1})^2\cdots(x_i-x_n)^2}$$

## §2 HM过程

### 2.1 参 数 表

#### 输入参数

N　已知插值结点的个数。

X　横座标数组，共 $n$ 个元素，为已知插值结点的值。

Y　纵座标数组，共 $n$ 个元素，为已知插值结点上的函数值。

Y1　导数的数组，共 $n$ 个元素，为已知插值结点上函数的导数值。

U　给定的插值结点值。

#### 输出参数

ANS 所求的插值结点U上的函数值。