Ю.С.Кушлинис, Т.Т.Битвинскас, И.И.Гикис

СИНХРОНИЗАЦИЯ СЕРИЙ ГОДИЧНЫХ КОЛЕЦ ПРИ ПОМОЩИ ЭВМ БЭСМ-6

Проблема синхронизации двух серий годичных колец возникает в том случае, если эти серии частично перекрываются во времени, и датировка колец одной (или обеих) серий неизвестна. Таким образом, в общем случае задача состоит из двух частей:

1. определение, перекрываются ли серии во времени,
2. определение перекрывающихся участков обоих серий.

Для синхронизации серии последовательно сдвигаются одна относительно другой, и при каждом сдвиге определяется некоторый - коэффициент сходства. Считается, что серии синхронны при сдвиге, дающем наибольший коэффициент сходства, при условии, что этот коэффициент превышает некоторый порог. В противном случае принимается, что серии не имеют общего участка, т.е. календарный год последнего кольца одной серии меньше календарного года первого кольца другой серии.

Как известно [1], на ширину годичных колец оказывают влияние как климатические факторы, так и возраст дерева. Если воздействие климатических факторов обычно синхронно для обеих серий, то влияние возраста дерева при синхронизации приходиться исключать. Это достигается тем, что сравниваются не исходные серии годичных колец, а преобразованные серии (обычно серии индексов годичных колец).

Ниже описывается комплекс программ, предназначенных для синхронизации серий. Каждое дерево, характеризуемое инвентарным номером $i = 1, 2, 3, \ldots$, представляется в виде двух серий годичных колец

$$R_j^{(i)} = \left(r_{j,1}^{(i)}, r_{j,2}^{(i)}, \ldots, r_{j,n^{(i)}}^{(i)} \right), \quad j = 1, 2,$$

где $R_1^{(i)}$ - серия колец ранней древесины, $R_2^{(i)}$ - серия колец поздней древесины, $r_{j,u}^{(i)}$ - ширина кольца, $n^{(i)}$ - количество колец в серии. Серии должны быть записаны в базу данных, находящуюся на магнитной ленте.

Для исключения влияния возраста дерева исходная серия нормализуется т.е. преобразовывается в

$$R_j^{*(i)} = \left(r_{j,1}^{*(i)}, r_{j,2}^{*(i)}, \ldots, r_{j,n^{*(i)}}^{*(i)} \right)$$

при помощи одного из следующих двух преобразований - (3) или (4)

$$r_{j,u}^{*(i)} = \begin{cases} 1, & \text{если } r_{j,u+1}^{(i)} - r_{j,u}^{(i)} \geq 0 \\ -1, & \text{в противном случае} \end{cases},$$

$$u = 1, 2, \ldots, n^{*(i)} = n^{(i)} - 1$$

После такой трансформации в преобразованной серии $R_j^{*(i)}$ остается только информация о знаке изменения ширины колец.

Второй способ нормализации ширины колец по отношению к возрасту дерева основан на вычислении индексов годичных колец, которым определяются по формуле

$$r_{j,u}^{*(i)} = \frac{r_{j,u}^{(i)}}{S_{j,u}^{(i)}}, \quad u = 1, 2, \ldots n^{*(i)} = n^{(i)},$$

где $S_{j,u}^{(i)}$, и элемент кривой роста

$$S_j^{(i)} = (S_{j,2}^{(i)}, \ldots S_{j,n^{(i)}}^{(i)}).$$

Кривые роста для отдельных серий $R_j^{(i)}$ отождествляются с низкочастотной составляющей этой серии. Обычно кривые роста аппроксимируются аналитическим выражением. Для аппроксимации кривой роста реализованы следующие алгоритмы:

A_I - алгоритм, производящий аппроксимирующую кривой роста отрицательной экспонентной по формуле $S_{j,u}^{(i)} = \alpha_j^{(i)} \cdot e^{-u\beta_j^{(i)}} + \gamma_j^{(i)}$

или, если такая аппроксимация не удается, производится аппроксимация прямой с отрицательным уклоном $S_{j,u}^{(i)} = u\delta_j^{(i)} + \eta_j^{(i)}, \quad \delta_j^{(i)} < 0$

A_2 - горизонтальной прямой через среднее

$$S_{j,u}^{(i)} = \frac{1}{n^{(i)}} \sum_{u=1}^{n^{(i)}} r_{j,u}^{(i)},$$

A_3 - произвольной прямой

$$S_{j,u}^{(i)} = u\delta_j^{(i)} + \eta_j^{(i)},$$

A_4 - отрицательной экспонентной; если такая аппроксимация не удается, производится аппроксимация прямой с произвольным уклоном,

A_5 - многочленом по формуле

$$S_{j,u}^{(i)} = \sum_{K=1}^{m^{(i)}} \varepsilon_K^{(i)} \cdot (u+10)^{K-1},$$

где $m^{(i)}$ - степень многочлена,

A_6 - скользящим средним по формулам

$$S_{j,u}^{(i)} = \frac{1}{q_u - P_u + 1} \sum_{K=P_u}^{q_u} r_{j,K}^{(i)},$$

где $P_u = \begin{cases} u - \varkappa_1, & \text{если } u - \varkappa_1 > 1 \\ 1, & \text{если } u - \varkappa_1 \leq 1 \end{cases}$

$$q_u = \begin{cases} P_u + \varkappa - 1, & \text{если } P_u + \varkappa - 1 \leq n^{(i)} \\ n^{(i)}, & \text{если } P_u + \varkappa - 1 \; n^{(i)}, \end{cases}$$

$$\varkappa_1 = \text{entier}(\varkappa/2).$$

здесь $\alpha, \beta, \gamma, \delta, \eta, \varepsilon, m^{(i)}$ – коэффициенты, получаемые во время аппроксимации, $\mathcal{H}$ – интервал интегрирования.

При использовании преобразования (3) коэффициент сходства $q(t)$ при сдвиге t (лет) между двумя сериями $R^{(1)}$ и $R^{(2)}$ вычисляется согласно выражению:

$$q(t) = \frac{50}{h(t)} \sum_{u=p(t)}^{q(t)} |r_u^{*(1)} + r^{*(2)}|,$$

$$t = -n^{*(2)} + h_{min}, -n^{*(2)} + h_{min} + 1, \ldots, n^{*(1)} - h_{min} - 1,$$

где

$$p(t) = max(1, t+1),$$
$$q(t) = min(n^{*(1)} - 1, n^{*(2)} + t - 1),$$
$$h(t) = q(t) - p(t) + 1,$$
$$n^{*(i)} = n^{(i)} - 1,$$

$r_u^{*(i)}$ – вычисляется из $r_u^{(i)}$, $r_{u+1}^{(i)}$ по форм. (3),

h_{min} – минимальная допустимая длина перекрывающегося участка серий $R^{(1)}$, $R^{(2)}$.

При использовании преобразования (4) в качестве коэффициента сходства используется коэффициент корреляции, вычисляемый согласно выражению:

$$q(t) = \frac{100 \sum_{u=p(t)}^{q(t)} (r_u^{*(1)} - v^{(1)})(r_{u-t}^{*(2)} - v^{(2)})}{\sqrt{\sum_{u=p(t)}^{q(t)} (r_u^{*(1)} - v^{(1)})^2} \sqrt{\sum_{u=p(t)}^{q(t)} (r_{u-t}^{*(2)} - v^{(2)})^2}},$$

где

$$v^{(1)} = \frac{1}{h(t)} \sum_{u=p(t)}^{q(t)} r_u^{*(1)},$$

$$v^{(2)} = \frac{1}{h(t)} \sum_{u=p(t)}^{q(t)} r_{u-t}^{*(2)}.$$

Таким образом, получается множество коэффициентов сходства (функция сходства)

$$q^{(1,2)} = [q(t), t = -n^{*(2)} h_{min}, -n^{*(2)} + h_{min} + 1, \ldots, n^{*(1)} - h_{min} - 1].$$

Считается, что серии $R^{(1)}$ и $R^{(2)}$ синхронны при сдвиге таком, что

$$q(t^*) = \max_t q(t)$$

Подпрограммы, выполняющие синхронизацию, составлены на алгоритмических языках ФОРТРАН [2] и АЛГОЛ [3] (подпрограммы ФОРТРАНа могут обращаться к независимо транслированным процедурам АЛГОЛ'а и наоборот). Магнитная лента с базой данных (исходными сериями годичных колец) состоится на МЦ 4-0. С перфокарт вводятся следующие данные:

А. форматом I5:

1. номер первой зоны базы данных,

2. номер последней зоны базы данных,

3. $e = \delta$ – количество печатаемых максимальных коэффициентов сходства,

4. σ – минимальное допустимое расстояние (в годах) между двумя максимумами функции сходства,

5. h_{min} – минимальная допустимая длина перекрывающегося участка двух серий (в годах),

6. коэффициент K_1, указывающий, какая древесина нужна:

$$K_1 = \begin{cases} 0 - \text{ранняя древесина}, \\ 1 - \text{поздняя древесина}, \\ 2 - \text{общая древесина (сумма толщины кольца ранней и поздней древесины)}, \end{cases}$$

7. коэффициент , определяющий тип преобразования исходной серии и аппроксимации A:

$$K_2 = \begin{cases} -1 - (3), \\ 0 - (4),\ A_1, \\ 1 - (4),\ A_2, \\ 2 - (4),\ A_3, \\ 3 - (4),\ A_4, \\ 5 - (4),\ A_5, \\ \geqslant 6 - (4),\ A_6. \text{ в этом случае принимается } \mathcal{H} = K_2 \end{cases}$$

8. номер первой зоны результатов (функций сходства) на МД 4–0,

9. номер последней зоны результатов (функции сходства) на МД 4–0,

Б. для каждого множества $M^{(j)}$ $j = 1, 2, \ldots,$ синхронизируемых серий вводятся следующие данные: 1. $m^{(j)}$ – количество строк

2. $I_i^{(j)}$ $i = 1, 2, \ldots, n^{(j)} + 1$ – инвентарные номера серий годичных колец, находящихся в базе данных. Тут $I_{n^{(j)}+1}^{(j)} = 0$ – признак окончания множества $M^{(j)}$. Должно выполняться условие $n^{(j)} \geqslant m^{(j)}$.

В. признак окончания данных – число (-1).

Программа работает следующим образом:

1. формируется множество

$$M = \bigcup_{j=1}^{\omega} M^{(j)} = \left(I_i, i = 1, 2, \ldots, n\right),$$

где

$$n \leq \sum_{j=1}^{\omega} n^{(j)},$$

2. считываются с базы данных серии $R_{K_1}^{(i)}$, $i = 1, 2, \ldots, n$, над ними производятся преобразования (3) (при $K_2 = -1$) или (4) (при $K_2 \geqslant 0$), и полученные преобразования серии $R_{K_1}^{*(i)}$ записываются на магнитный барабан.

3. для каждого множества $M^{(j)}$ выполняется:

а) для всех пар серий

$$R^{*(i_1)}, R^{*(i_2)}, \quad i_1 = 1, 2, \ldots, n^{(j)} - 1,$$
$$i_2 = n^{(j)} - m^{(j)} + 1, n^{(j)} - m^{(j)} + 2, \ldots, n^{(j)}$$

вычисляются и записываются на магнитную ленту функции сходства $Q^{(i_1, i_2}$

б) из каждой функции сходства

$$Q^{(i_1, i_2)} = (q(t),\ t = d_1,\ d_1+1,\ \dots,\ d_2)$$

выбираются с максимальных значений коэффициента сходства

$$q(t_1),\ q(t_2),\ \dots,\ q(t_e)$$

таким образом, что

$$q(t_1) \geq q(t_2) \geq \dots \geq q(t_e)$$

и

$$t_{i+1} - t_i \geq v,\quad i = 1, 2, \dots, e-1.$$

Если имеется последовательность

$$q(t_i),\ q(t'_2),\ \dots\ q(t'_p)$$

такая, что

$$t'_{i+1} - t'_i < v,\quad i = 1, 2, \dots, p-1$$

то все эти максимумы принимаются за один максимум

$$q(t') = \max_{i=1,2,\dots,p} q(t'_i)$$

Результаты печатаются в виде следующей таблицы.

PANASUMAS PAGAL KORELIACIJA
SUMARINES RIEVES
TRANSFORMACIJA= 1
PRADINE REZULT. ZONA= 431
GALINE REZULT. ZONA= 435

табл.1

SERIJA 1		SERIJA 2		MAXI NR.	PANAS. KOEF.	PERST. (RIEV)	PERS. DENG.
INV.NR.	RIEV.KIEK.	INV.NR.	RIEV.KIEK.				
29	129	23	129	1	63.7	103	27
				2	54.0	-4	126
				3	49.0	-107	23
				4	48.6	-110	20
				5	47.3	77	53
				6	43.9	-86	44
				7	27.0	-60	70
				8	21.0	-30	100
33	129	23	129	1	45.5	107	23
				2	45.2	-85	45
				3	31.4	-4	126
				4	30.5	-99	31
				5	28.6	90	40
				6	27.8	-30	100
				7	26.4	-56	74
				8	26.3	-60	70
40	129	23	129	1	65.2	104	26
				2	60.6	-5	125
				3	44.8	-110	20
				4	36.0	-87	43
				5	35.1	-71	59
				6	32.8	-61	69
				7	26.1	-107	23
				8	24.3	107	23

43	129	23	129	1	70.8	-109	21
				2	60.4	-4	126
				3	53.2	-106	24
				4	49.8	103	27
				5	34.7	-60	70
				6	25.2	-57	75
				7	21.8	-69	61
				8	21.3	-86	44
53	129	23	129	1	68.5	-5	125
				2	58.4	-110	20
				3	51.2	-87	43
				4	47.7	107	23
				5	44.6	-107	23
				6	38.2	-61	69
				7	31.8	90	40
				8	29.8	50	80
33	129	29	129	1	69.7	0	130
				2	49.8	3	127
				3	45.7	-13	117
				4	44.8	109	21
				5	43.6	16	114
				6	41.4	-26	104
				7	36.9	-107	23
				8	28.7	96	34

Здесь коэффициенты принимают значение:

В приложении I приводятся тексты программ на алгоритмических языках ФОРТРАН и АЛГОЛ для БЭСМ-6.

ЛИТЕРАТУРА

I. Т.Т.Битвинскас. Дендроклиматические исследования. Гидрометеоиздат, 1974.

2. ФОРТРАН для БЭСМ-6, Вильнюс, 1974.

3. Р.Хирр, Р.Штробель. АЛГОЛ в мониторной системе ДУБНА, ИПМ АН СССР, 1973.

4. James B. Campbell. Manuel for using basic chronology programs, Laboratory of Tree-Ring Research, University of Arizona, 1973.

PRIEDAS I

```
'ALGOL
'CODE':EXSTEP;
'ALGOL':READR,FINV,RSMJMR,RERSMB,PARSMX,RSSUPR,TITPAN;'INTEGER' RMAIL;
  'BEGIN' 'REAL' INV1,INV2;'BITS' MJ;
  'INTEGER' ZR0,ZRMX,HMIN,Q,NMAX,N,NLM,NL,IL,NINVM,NINV,NINM,
  JIN,NRSM,NRS1,NRS2,I1,I2,J,K,Q1,NREG,TREE,RZP,RZG,DIPR,
  DIPR1,DIGA,NPES,NRS3,NRS4,DVF;
  'ARRAY' INV[1:500],IN[1:600],RS1,RS2[1:200],P[1:8];
  'INTEGER' 'ARRAY' N1,N2[1:21],PRA,ILG[1:500],T,H[1:8];
  MJ:='40';NMAX:=8;NLM:=20;NINVM:=300;NINM:=500;NRSM:=300;
  'READ'''(I5)',ZR0,ZRMX,N,DVF,HMIN,Q,Q1,RZP,RZG;
  DIPP:=1024*RZP+1;DIPR1:=DIPR;DIGA:=1024*(RZG+1);
  'IF'(Q1'EQ'-1)'THEN''PRINT'''(///10X,24HPANASUMAS PAGAL POKICIUS//)'
  'ELSE' 'PRINT'''(///10X,27HPANASUMAS PAGAL KORELIACIJA)'';
  'IF'(Q'EQ'0)'THEN' 'PRINT'''(10X,19HANKSTYVOSIOS PIEVES)'';
  'IF'(Q'EQ'1)'THEN' 'PRINT'''(10X,17HVELYVOSIOS PIEVES)'';
  'IF'(Q'EQ'2)'THEN' 'PRINT'''(10X,16HSUMARINES PIEVES)'';
  'PRINT'''(10X,15HTRANSFORMACIJA=,I2
  /10X,21HPRADINE REZULT. ZONA=,I4
  /10X,20HGALINE REZULT. ZONA=,I4)''',Q1,RZP,RZG;
  NL:=NLM;NRS1:=JIN:=I1:=0;
  'FOR' IL:=IL+1 'WHILE'((NRS1'GE'0)'AND'(IL'LE'NLM)'AND'
  (JIN'LT'NINM)),'DO'
    'BEGIN' 'READ'''(I5)''',NRS1;
    'IF'(NRS1'LE'0)'THEN' NL:=IL-1 'ELSE'
      'BEGIN' N1[IL]:=NRS1;READR(INV,''GT'',0.0);
      NRS2:=RMAIL(INV,NINVM,0,0)-1;N2[IL]:=NRS2;
      'IF'(JIN+NRS2'GT'NINM)'THEN' 'BEGIN' NL:=IL-1;NRS1:=-1'END' 'ELSE'
        'BEGIN' 'FOR' J:=1 'STEP' 1 'UNTIL' NRS2 'DO' IN[JIN+J]:=INV[J];
        JIN:=JIN+NRS2;
        'END';
      'END';
    'END';
  NINV:=0;FINV(IN,JIN,INV,NINVM,NINV);
  RSMJMR(Q,Q1,MJ,ZR0,ZRMX,INV,INV,PRA,ILG);JIN:=0;
  'FOR' IL:=1 'STEP' 1 'UNTIL' NL 'DO'
    'BEGIN' J:=N2[IL]-N1[IL]+1;TITPAN;
    'FOR' I2:=1 'STEP' 1 'UNTIL' N2[IL]-1 'DO'
      'BEGIN' INV2:=IN[JIN+I2];K:=I2+1;'IF'(K'LT'J) 'THEN'K:=J;
      RERSMB(INV2,INV,PRA,ILG,NINV,RS2,NRSM,NRS2);
      'IF'(NRS2'GT'HMIN)'THEN' 'FOR' I1:=K 'STEP' 1 'UNTIL' N2[IL] 'DO'
        'BEGIN' 'INTEGER' QQ;INV1:=IN[JIN+I1];TREE:=INV1*1000+INV2;
        NREG:=(I2-1)*(N2[IL]-1)+I1-1;
        RERSMB(INV1,INV,PRA,ILG,NINV,RS1,NRSM,NRS1);
        'IF'(NRS1'LE'HMIN)'THEN' 'GOTO' Z1;
        PARSMX(RS1,NRS1,RS2,NRS2,Q1,HMIN,N,P,T,H,DVF,DIPR1,DIGA,
        TREE,NPES,MJ);
        NRS3:=NRS1;NRS4:=NRS2;
        'IF'(Q1'GE'0)'THEN' 'BEGIN' NRS3:=NRS3-1;NRS4:=NRS4+1 'END';
        'IF'(I1'LT'N2[IL]) 'THEN' QQ:=0 'ELSE' QQ:=1;
        RSSUPR(INV1,NRS3,INV2,NRS4,QQ,N,P,T,H);
      Z1:'END';
    'END';
  JIN:=JIN+N2[IL];
  'END';
EXSTEP;
'END'
'EOP'
```

```
'PROCEDURE' FINV(INV1,NINV1,INV,NINVM,NINV);
'INTEGER' NINV1,NINVM,NINV;'ARRAY' INV1,INV;
  'BEGIN''INTEGER' Q,M,I,J;N:=NINV;
  'FOR' I:=1 'STEP' 1 'UNTIL' NINV1 'DO'
    'BEGIN' Q:=0;J:=0;
    'FOR' J:=J+1 'WHILE'((Q'EQ'0)'AND'(J'LE'M))'DO'
    'IF'(INV[J]'EQ'INV1[I])'THEN' Q:=1;
    'IF'((Q'EQ'0)'AND'(M'LT'NINVM))'THEN'
    'BEGIN' M:=M+1;INV[M]:=INV1[I] 'END';
    'END';
  NINV:=M;
  'END'
  'EOP'


'CODE':SKZONA,INDEX;'ALGOL':TRANS1,RW;
'PROCEDURE' RSMJNB(Q,Q1,MJ,ZR0,ZRMX,NINV,INV,PRA,ILG);'BITS'MJ;
'INTEGER'Q,Q1,ZR0,ZRMX,NINV;'ARRAY'INV;'INTEGER''ARRAY'PRA,ILG;
  'BEGIN''REAL' INV1;'INTEGER' NB,NR,ISN,ZR,QQ,ILG1,ILG2,PRA1,PRA2,I,
  J,U,K1,K2,Z1,IZ1,NDEG,NNV;'BITS' B1;
  'ARRAY' ZZ[1:1024],MB[1:1000],R,R1[1:300],COEF[1:31],TEMP[1:300,1:5];
  NB:=1000;NR:=900;ISN:=-999;ZZ[1]:=0;B1:='16';NNV:=NR;
  'FOR' I:=1 'STEP' 1 'UNTIL' NINV 'DO' ILG[I]:=0;
  PRA1:=PRA2:=1;Z1:=0;IZ1:=1;ZR:=ZR0-1;
  'FOR' ZR:=ZR+1 'WHILE'((ZZ[1]'NE'ISN)'AND'(ZR'LE'ZRMX))'DO'
    'BEGIN' SKZONA(ZZ,ZR,MJ);'IF'(ZZ[1]'NE'ISN)'THEN'
      'BEGIN' ILG1:=0;I:=1;
      'FOR' I:=I+ILG1 'WHILE'((ZZ[I]'NE'ISN)'AND'(I'LE'1024))'DO'
        'BEGIN' INV1:=ZZ[I];ILG1:=ZZ[I+1];QQ:=J:=0;
        'FOR' J:=J+1 'WHILE'((J'LE'NINV)'AND'(QQ'EQ'0))'DO'
        'IF'(INV[J]'EQ'INV1)'THEN'
          'BEGIN' QQ:=1;ILG2:=(ILG1-2)/2;
          'IF'(ILG2'LE'NR)'THEN'
            'BEGIN' K1:=I+1;
            'FOR' U:=1 'STEP' 1 'UNTIL' ILG2 'DO'
              'BEGIN'
              'IF'(Q'EQ'0)'THEN' R[U]:=ZZ[K1+U];
              'IF'(Q'EQ'1)'THEN' R[U]:=ZZ[K1+U+ILG2];
              'IF'(Q'EQ'2)'THEN' R[U]:=ZZ[K1+U]+ZZ[K1+U+ILG2];
              'END';
            'IF'(Q1'EQ'-1)'THEN' TRANS1(R,ILG2);
            'IF'(Q1'GE'0)'THEN'
              'BEGIN'
              'FOR' U:=1 'STEP' 1 'UNTIL' ILG2 'DO' R1[U]:=R[U];
              INDEX(R1,ILG2,R,Q1,NDEG,COEF,TEMP,NNV);
              'END';
            'IF'(PRA1+ILG2'GT'NB)'THEN'
              'BEGIN' RW(MB,PRA1-1,Z1,IZ1,B1,1,K1,K2);
              Z1:=K1;IZ1:=K2;PRA1:=1;
              'END';
            K1:=PRA1-1;
            'FOR' U:=1 'STEP' 1 'UNTIL' ILG2 'DO' MB[K1+U]:=R[U];
            PRA[J]:=PRA2;ILG[J]:=ILG2;PRA1:=PRA1+ILG2;PRA2:=PRA2+ILG2;
            'END';
          'END';
        'END';
      'END';
  'IF'(PRA1'GT'1)'THEN' RW(MB,PRA1-1,Z1,IZ1,B1,1,K1,K2);
  'END'
  'EOP'
```

```
'PROCEDURE' TRAPS1(R,NR);'INTEGER' NR;'ARRAY' R;
  'BEGIN''INTEGER' I;NR:=NR-1;
  'FOR' I:=1 'STEP' 1 'UNTIL' NR 'DO'
  'IF'(R[I+1]-R[I])'GE'0)'THEN' R[I]:=1 'ELSE' R[I]:=-1;
  'END'
  'EOP'

'PROCEDURE' TRAPS2(DRN,DR1,R,NR);
'INTEGER' DRN,DR1,NR;'ARRAY' R;
  'BEGIN''INTEGER' DRN1,DR11,DR,DR1,I,J;'REAL' DRNI;
  DRN1:=(DRN-1)/2;DR11:=(DR1-1)/2;DRNI:=DRN/DR1;
  'IF'(DRN'GT'DR1)'THEN' 'BEGIN' DR:=DRN;DR1:=DRN1 'END'
  'ELSE' 'BEGIN' DR:=DR1;DR1:=DR11 'END';
  'FOR' I:=DR1+1 'STEP' 1 'UNTIL' NR-DR1 'DO'
    'BEGIN''REAL' R1,R2;R1:=R2:=0;
    'FOR' J:=-DR11 'STEP' 1 'UNTIL' DR11 'DO' R1:=R1+R[I+J];
    'FOR' J:=-DRN1 'STEP' 1 'UNTIL' DRN1 'DO' R2:=R2+R[I+J];
    R[I-DR1]:=DRNI*R1/R2;
    'END';
  NR:=NR-DR+1;
  'END'
  'EOP'

'ALGOL';RWP,RW;
'PROCEDURE' RERSMA(INV1,INV,PRA,ILG,NINV,RS,NRSM,NRS);
'REAL' INV1;'INTEGER' NINV,NRSM,NRS;'ARRAY' INV,RS;
'INTEGER''ARRAY' PRA,ILG;
  'BEGIN''INTEGER' I,PRA1,Z1,IZ1;'BITS' B1;NRS:=0;I:=0;B1:='16';
  'FOR' I:=I+1 'WHILE'((NRS'EQ'0)'AND'(I'LE'NINV))'DO'
  'IF'(INV[I]'EQ'INV1)'THEN' 'BEGIN' NRS:=ILG[I];PRA1:=PRA[I] 'END';
  'IF'((NRS'GT'0)'AND'(NRS'LE'NRSM))'THEN'
  'BEGIN' RWP(PRA1,Z1,IZ1);RW(PS,NRS,Z1,IZ1,B1,0,I,PRA1) 'END';
  'END'
  'EOP'

'PROCEDURE' RWP(A,Z1,IZ1);'INTEGER' A,Z1,IZ1;
  'BEGIN'
  Z1:=ENTIER((A-1)/1024);IZ1:=A-Z1*1024;
  'END'
  'EOP';
```

```
'CODE':ARW;'ALGOL':PARS,MAXMAS;
'PROCEDURE' PARSMX(R1,NR1,R2,NR2,Q1,HMIN,N,P,T,M,DVE,DIPR,DIGA,
TREE,NRE,MJ);
'INTEGER' NR1,NR2,Q1,HMIN,N,DIPR,DIGA,TREE,NRE,DVE;'ARRAY' R1,R2,P;
'INTEGER''ARRAY' T,H;'PITS' MJ;
  'BEGIN''REAL' P1,PAB;'INTEGER' T1,H1,NCOM,NCO;'ARRAY' CO[1:405];
  NCOM:=404;PAB:=-999.0;CO[1]:=TREE;CO[3]:=NRE;CO[4]:=-NR2+HMIN;
  'FOR' T1:=1 'STEP' 1 'UNTIL' N 'DO' P[T1]:=-9999.0;NCO:=4;
  'FOR' T1:=-NR2+HMIN 'STEP' 1 'UNTIL' NR1-HMIN+1 'DO'
    'BEGIN' PARS(R1,NR1,R2,NR2,Q1,T1,P1,H1);
    NCO:=NCO+1;'IF'(NCO'GT'NCOM)'THEN' 'GOTO' Z1;
    MAXMAS(P,T,M,N,DVE,P1,T1,H1);CO[NCO]:=P1;

  Z1:'END';
  'IF'(NCO'GT'NCOM)'THEN' NCO:=NCOM;CO[2]:=NCO;CO[NCO+1]:=PAB;
  'IF'((DIPR'GT'0)'AND'(DIPR+NCO'LT'DIGA))'THEN'
  'BEGIN' ARW(DIPR,NCO+1,CO,1,MJ);DIPR:=DIPR+NCO 'END';
  'END'
  'EOP'


'ALGOL':PARSPO,PARSKO;
'PROCEDURE' PARS(R1,NR1,R2,NR2,Q1,T,P,H);
'REAL' P;'INTEGER' NR1,NR2,Q1,T,H;'ARRAY' R1,R2;
  'BEGIN''INTEGER' UP,UG,U;H:=0;P:=0;
  'IF'((T'GT'-NR2)'AND'(T'LT'NR1))'THEN'
    'BEGIN' 'IF'(T'GT'0)'THEN' UP:=T+1 'ELSE' UP:=1;
    'IF'(NR1'LT'NR2+T)'THEN' UG:=NR1 'ELSE' UG:=NR2+T;H:=UG-UP+1;
    'IF'(Q1'EQ'-1)'THEN' PARSPO(R1,NR1,R2,NR2,UP,UG,T,P) 'ELSE'
    PARSKO(R1,NR1,R2,NR2,UP,UG,T,P);
    'END';
  'END'
  'EOP'


'PROCEDURE' PARSPO(R1,NR1,R2,NR2,UP,UG,T,P);
'REAL' P;'INTEGER' NR1,NR2,UP,UG,T;'ARRAY' R1,R2;
  'BEGIN''INTEGER' H,U;P:=0;H:=UG-UP+1;
  'FOR' U:=UP 'STEP' 1 'UNTIL' UG 'DO' P:=P+ABS(R1[U]+R2[U-T]);
  P:=P*50/H;
  'END'
  'EOP'


'PROCEDURE' PARSKO(R1,NR1,R2,NR2,UP,UG,T,P);
'REAL' P;'INTEGER' NR1,NR2,UP,UG,T;'ARRAY' R1,R2;
  'BEGIN''INTEGER' H,U;'REAL' V1,V2,D1,D2;
  P:=0;H:=UG-UP+1;V1:=V2:=0;
  'FOR' U:=UP 'STEP' 1 'UNTIL' UG 'DO'
  'BEGIN' V1:=V1+R1[U];V2:=V2+R2[U-T] 'END';
  V1:=V1/H;V2:=V2/H;D1:=D2:=0;
  'FOR' U:=UP 'STEP' 1 'UNTIL' UG 'DO'
    'BEGIN''REAL' B1,B2;B1:=R1[U]-V1;B2:=R2[U-T]-V2;
    D1:=D1+B1**2;D2:=D2+B2**2;P:=P+B1*B2;
    'END';
  D1:=SQRT(D1);D2:=SQRT(D2);P:=100*P/(D1*D2);
  'END'
  'EOP'
```

```
'PROCEDURE' MAXMAS(P,T,H,N,DVE,P1,T1,H1);
'REAL' P1;'INTEGER' N,DVE,T1,H1;'ARRAY' P,T,H;
  'BEGIN' 'INTEGER' I,J,Q,I1,DVEMIN;DVEMIN:=10**10;Q:=I:=0;
  'FOR' I:=I+1 'WHILE'((P[I]'GT'-9999.0)'AND'(I'LE'N))'DO'
  'IF'((ABS(T[I]-T1))'LE'DVEMIN)'THEN'
  'BEGIN' DVEMIN:=ABS(T[I]-T1);I1:=I 'END';
  'IF'(DVEMIN'LE'DVE)'THEN'
    'BEGIN' 'IF'(P[I1]'GE'P1)'THEN' Q:=1 'ELSE'
      'BEGIN' 'FOR' I:=I1 'STEP' 1 'UNTIL' N-1 'DO'
      'BEGIN' P[I]:=P[I+1];T[I]:=T[I+1];H[I]:=H[I+1] 'END';
      P[N]:=-9999.0;
      'END';
    'END';
  I:=0;'FOR' I:=I+1 'WHILE'((Q'EQ'0)'AND'(I'LE'N))'DO'
  'IF'(P1'GE'P[I])'THEN'
    'BEGIN' 'FOR' J:=N-1 'STEP' -1 'UNTIL' I 'DO'
    'BEGIN' P[J+1]:=P[J];T[J+1]:=T[J];H[J+1]:=H[J] 'END';
    P[I]:=P1;T[I]:=T1;H[I]:=H1;Q:=1;
    'END';
  'END'
  'FOP'
```

```
'PROCEDURE' TITPAS;
  'BEGIN'
  'PRINT''('////12X,72(1H-))',;
  'PRINT''('12X,62HI        SERIJA 1        I        SERIJA 2        I
   50HI MAXI PANAS.I PERST.I PERSI-I)',;
  'PRINT''('12X,1HI,20(1H-),1HI,20(1H-),1HI,
   50HI NR.I KOEF. I (RIEV)I DENG. I)',;
  'PRINT''('12X,2(2HI INV.NR.I RIEV.KIEK.),1HI,
   50HI    I       I       I       I)',;
  'PRINT''('12X,1HI,6(1H-),1HI,11(1H-),1HI,8(1H-),1HI,11(1H-),
   1HI,4(1H-),1HI,7(1H-),1HI,7(1H-),1HI,7(1H-),1HI,7(1H-),1HI)',;
  'END'
  'FOP'
```

```
      SUBROUTINE ARW1(IMBP,NI,KP,KG,JP,JG)
      NZ=1024 ◊ KP=INT((IMBP-1)/NZ) ◊ JP=IMBP-NZ*KP
      KG=INT((IMBP+NI-2)/NZ) ◊ JG=IMBP+NI-1-NZ*KG
      RETURN
      END

      SUBROUTINE ARW(IMBP,NS,S,JQ,MJ)
      DIMENSION S(NS),Z(1024)
      NZ=1024
      CALL ARW1(IMBP,NS,KP,KG,JP,JG) ◊ I=0
      DO 4 K=KP,KG
      J1=1 ◊ IF(K.EQ.KP) J1=JP ◊ J2=NZ ◊ IF(K.EQ.KG) J2=JG
      IF((JQ.EQ.0).OR.(K.EQ.KP).OR.(K.EQ.KG)) CALL SKZONA(Z,K,MJ)
      DO 3 J=J1,J2
      I=I+1 ◊ IF(JQ) 1,1,2
    1 S(I)=Z(J) ◊ GOTO 3
    2 Z(J)=S(I)
    3 CONTINUE
      IF(JQ.EQ.1) CALL WRZONA(Z,K,MJ)
    4 CONTINUE
      RETURN
      END
```

```
'ALGOL';'STRING' RST,STAC;
'PROCEDURE' RSSUBR(INV1,N1,INV2,N2,Q,N,P,T,H);
'REAL' INV1,INV2;'INTEGER' N1,N2,N,Q;'ARRAY' P;'INTEGER''ARRAY' T,H;
  'BEGIN' 'INTEGER' PRA,I,J,K,NN;'ARRAY' MR[1:9];
  'STRING' SE1,SE2,SE3,SE4,SE5,S1,S2,S3;
  'INTEGER''ARRAY' MI,MI1[1:9];'STRING''ARRAY' MS,MS1[1:9];
  PRA:=12;NN:=8;SE1:=' ';SE2:='-';SE3:='I';SE4:=',';
  SE5:='*';S1:=SE1;
  'FOR' I:=1 'STEP' 1 'UNTIL' PRA-2 'DO' S1:=S1&SE1;
  MI[1]:=8;MI[2]:=11;MI[3]:=8;MI[4]:=11;MI[5]:=4;MI[6]:=7;MI[7]:=7;
  MI[8]:=7;MI[9]:=0;MS[1]:='('15')';MS[2]:='('16')';MS[3]:='('15')';
  MS[4]:='('16)';MS[5]:='('12)';MS[6]:='(F5.1)';MS[7]:='('15')';
  MS[8]:='('14)';MI1[9]:=0;
  'FOR' I:=1 'STEP' 1 'UNTIL' 4 'DO'
  'BEGIN' MI1[I]:=MI[I];MS1[I]:=SE1 'END';
  'FOR' I:=5 'STEP' 1 'UNTIL' 8 'DO'
  'BEGIN' MI1[I]:=MI[I];MS1[I]:=MS[I] 'END';
  K:=1;'FOR' I:=1 'STEP' 1 'UNTIL' NN 'DO' K:=K&MI[I]+1;
  S3:=S1;'FOR' I:=1 'STEP' 1 'UNTIL' K 'DO' S3:=S3&SE5;
  S2:=S1&SE3;'FOR' I:=1 'STEP' 1 'UNTIL' NN 'DO'
    'BEGIN'
    'FOR' J:=1 'STEP' 1 'UNTIL' MI[I] 'DO' S2:=S2&SE2;S2:=S2&SE3
    'END';
  MR[1]:=INV1;MR[2]:=N1;MR[3]:=INV2;MR[4]:=N2;
  'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
   'BEGIN' MR[5]:=I;MR[6]:=P[I];MR[7]:=T[I];MR[8]:=H[I];
    'IF'(I'EQ'1)'THEN' S1:=STAC(PRA,MI,MS,MR,SE1,SE3)
    'ELSE' S1:=STAC(PRA,MI1,MS1,MR,SE1,SE3);
    PRINT(S1,NEWLINE);
    'END';
  'IF'(Q'EQ'0)'THEN' PRINT(S2,NEWLINE) 'ELSE' PRINT(S3,NEWLINE);
  'END'
  'EOP'


'CODE' :SKZONA,WRZONA;
'PROCEDURE' RW(M,NM,Z1,IZ1,B,Q,Z2,IZ2);
'INTEGER' NM,Z1,IZ1,Q,Z2,IZ2;'ARRAY' M;'BITS' B;
  'BEGIN' 'INTEGER' Z,IZ,J;'ARRAY' MZ[1:1024];
  SKZONA(MZ,Z1,B);Z:=Z1;IZ:=IZ1-1;
  'FOR' J:=1 'STEP' 1 'UNTIL' NM 'DO'
    'BEGIN' IZ:=IZ+1;'IF'(IZ'LE'1024)'THEN' 'GOTO' A1;
    'IF'(Q'EQ'1)'THEN' WRZONA(MZ,Z,B);
    Z:=Z+1;IZ:=1;SKZONA(MZ,Z,B);
  A1:'IF'(Q'EQ'0)'THEN' M[J]:=MZ[IZ];
    'IF'(Q'EQ'1)'THEN' MZ[IZ]:=M[J];
    'END';
  'IF'(Q'EQ'1)'THEN' WRZONA(MZ,Z,B);IZ2:=Z;IZ2:=IZ+1;
  'IF'(IZ2'GT'1024)'THEN' 'BEGIN' Z2:=Z2+1;IZ2:=1 'END';
'END'
  'EOP'
```

```
'STRING''PROCEDURE' RST(R,F);'REAL' P;'STRING' F;
  'BEGIN''INTEGER' N;'ARRAY' K[1:3];'STRING' S;
  N:=LENGTH(F);S:='' '';F[3];
  'IF'(N'GE'5)'THEN' 'BEGIN' 'IF'(F[4]'NE'','.'')'THEN' S:=F[3:4] 'END';
  STOA(K[1],K[1],S);'DECODE'(S,'('12')',K[1]) N;
  'IF'(N'GT'18),'THEN' 'GOTO' Z1;'ENCODE'(N,F,K) R;ATOS(K[1],K[3],S);
  RST:=S[1:N];'GOTO' Z2;
Z1:'PRINT'''(10A,22HKLAIDA:RST FORM.,I5,I5 ,I2)''',N;STOP;
Z2:
 'END'
  'EOP'

'ALGOL';'STRING' RST;
'STRING''PROCEDURE' STAC(P,MI,MS,MR,ST,SP);
'INTEGER' P;'ARRAY' MR;'INTEGER''ARRAY' MI;'STRING' ST,SP;
'STRING''ARRAY' MS;
  'BEGIN''INTEGER' I,J,N;'STRING' S,S1,S2;  J:=0;
  S1:=SP+ST;S:='' '';'FOR' I:=2 'STEP' 1 'UNTIL' P-1 'DO' S:=S+'' '';
Z1:J:=J+1;'IF'(MI[J]'EQ'0)'THEN' 'GOTO' Z2;S:=S+S1;
  'IF'(MI[J]'LT'0)'THEN' S2:=MS[J] 'ELSE' S2:=RST(MR[J],MS[J]);
  S:=S+S2;N:=ABS(MI[J])-LENGTH(S2)-1;
  'FOR' I:=I 'STEP' 1 'UNTIL' N 'DO' S:=S+ST;'GOTO' Z1;
Z2:S:=S+SP;STAC:=S;
 'END'
  'EOP'


'BOOLEAN''PROCEDURE' SAR(A,S,B);'REAL' A,B;'STRING' S;
 'BEGIN''BOOLEAN' K;K:='FALSE';'IF'(S'EQ'''EQ'')'THEN' K:=A'EQ'B;
 'IF'(S'EQ'''NE'')'THEN' K:=A'NE'B;'IF'(S'EQ'''GT'')'THEN' K:=A'GT'B;
 'IF'(S'EQ'''GE'')'THEN' K:=A'GE'B;'IF'(S'EQ'''LT'')'THEN' K:=A'LT'B;
 'IF'(S'EQ'''LE'')'THEN' K:=A'LE'B;SAR:=K;
 'END'
  'EOP'


'ALGOL';'BOOLEAN' SAR;
'PROCEDURE' READR(M,S,Q);'REAL' Q;'STRING' S;'ARRAY' M;
 'BEGIN''INTEGER' I;READ(M[1]);I:=1;
 'FOR' I:=I+1 'WHILE' SAR(M[I-1],S,Q)'DO' READ(M[I]);READ(NEWLINE);
 'END'
  'EOP'
```

```
      SUBROUTINE FITPOL(V,NV,V1,COEF,NDEG,PERCEN,TEMP,NNV)
      DIMENSION V(NNV),V1(NNV),COEF(31),TEMP(NNV,5)
      CALL COPOL(V,NV,COEF,NDEG,V1,PERCEN,TEMP,NNV)
      IF(NDEG.LE.15) GOTO 3
      PRINT 1,NDEG
    1 FORMAT(/10X,'***FITPOL*** DEGREE',I3,' IS TOO HIGH')
      DO 2 IX=1,NV
    2 V1(IX)=0.
      GOTO 6
    3 XP=10.
      DO 5 IX=1,NV
      X=XP+IX  $  X1=1.  $  V2=COEF(1)
      DO 4 J=2,NDEG
      X1=X1*X  $  V2=V2+X1*COEF(J)
    4 CONTINUE
      V1(IX)=V2
    5 CONTINUE
    6 CONTINUE
      RETURN
      END


      SUBROUTINE INDEX(V,NV,V1,LOPT,NDEG,COEF,TEMP,NNV)
      DIMENSION V(NNV),V1(NNV),COEF(31),TEMP(NNV,5)
      NDEG=4  $  KK=1  $  IF((LOPT.LT.0).OR.(LOPT.EQ.4))GOTO 2  $  KK=0  $  PER=0.05
      IF((LOPT.GE.0).AND.(LOPT.LE.3)) CALL FITEXP(V,NV,V1,COEF,LOPT)
      IF(LOPT.EQ.5) CALL FITPOL(V,NV,V1,COEF,NDEG,PER,TEMP,NNV)
      IF(LOPT.GE.6) CALL FITMEA(V,NV,V1,LOPT)
      DO 1 J=1,NV
      V1(J)=V(J)/V1(J)  $  IF(V1(J).LT.0) KK=1
    1 CONTINUE
    2 IF(KK.EQ.1) NDEG=-NDEG
      RETURN
      END
```

```
      SUBROUTINE COPOL(V,N,A,M,X,PERCEN,TEMP,NN)
      DIMENSION X(NN),V(NN),TEMP(NN,5),A(31),W(1),
     *SIGSR(31),ALPHA(32),BETA(32),B(32),D(31),COEF(3,31),
     *ACCUR(5),DELS0(31),LABEL(6),LABEL1(6),IFMT(14)
      DATA(DELLIM=0.0),(KBEVON=0),(NCALL=0)
C     SUBROUTINE   C O P O L   IS SLIGHTLY CHANGED SUBROUTINE
C     L S T S Q P   , WHICH IS DESCRIBED IN THE UNIVERSITY OF
C     WISCONSIN COMPUTER CENTER USERS MANUAL , VOLUME IV ,
C     SUPPLEMENT H , FEBRUARY 1969 .
      MIDDL=0 $ NCALL=NCALL+1 $ NPAGES=1 $ MAXSUB=31
     *MIN=1 $ MAX=SHOPTIM $ IPRINT=4 $ ISHIFT=0
      IOUT=IPRINT $ ISH=ISHIFT $ R=0.0 $ P=0.0
      DO 2210 I=1,10
      K=N+1-I $ R=R+V(I)
 2210 P=P+V(K)
      R=R/10. $ P=P/10.
      DO 2440 I=1,N
      K=N+1-I $ S=K+10 $ X(K+10)=S
 2440 V(K+10)=V(K)
      N=N+10
      DO 2441 I=1,10
      S=I-0 $ X(I)=S $ S=N+I $ X(N+I)=S $ V(I)=R
 2441 V(N+I)=P
      N=N+10 $ NUM=N $ MESSAG=0 $ KKK=-1 $ IRETUR=0
  110 KWT=1
  120 IF(MIN.GT.-1.AND.MIN.LE.MAXSUB-1.AND.MIN.LT.NUM) GOTO 140
      M=-1
      PRINT 124
  124 FORMAT(1H0,'$$$CALL TO LSTSQP WITH INVALID MINDEG$$$')
      RETURN
  140 IF(MAX.NE.SHOPTIM) GOTO 150
  145 KOPTIM=0
      MAXPOL=MAXSUB $ GOTO 160
  150 KOPTIM=1 $ MAXPOL=MAX+1
  160 IF(MAX.GT.MAXSUB-1.OR.MAX.LT.MIN) GOTO 170 $ GOTO 175
  170 M=-2
      PRINT 174
  174 FORMAT(1H0,'$$$CALL TO LSTSQP WITH INVALID MAXDEG$$$')
      RETURN
  175 IF(MAX.LE.NUM-1) GOTO 180
  176 M=-5
      PRINT 178
  178 FORMAT(1H0,'$$$CALL TO LSTSQP WITH MAXDEG .GE.N$$$')
      RETURN
  180 IF(IOUT.GE.0.AND.IOUT.LE.4) GOTO 190
      M=-3
      PRINT 184
  184 FORMAT(1H0,'$$$CALL TO LSTSQP WITH INVALID IPRINT$$$')
      RETURN
  190 IF(ISH.EQ.0) GOTO 195
      IF(ISH.EQ.1) GOTO 196
      M=-4
      PRINT 192
  192 FORMAT(1H0,'$$$CALL TO LSTSQP WITH INVALID LTRANS$$$')
      RETURN
  195 INTERV=6HORIGIN
      GOTO 200
  196 INTERV=5HSHIFT
  200 CALL SEARCH(0,X,NUM,MASK,BIG)
      CALL SEARCH(1,X,NUM,MASK,SMALL)
      IF(BIG.NE.SMALL) GOTO 220
      PRINT 214
```

```
214 FORMAT(1H0,'000CALL TO LSTSQP WITH EVERY ELEMENT
   *OF THE X ARRAY EQUAL000')
    M=-6
    RETURN
220 XBAR=.5*(BIG+SMALL)
    SHRINK=4./(BIG-SMALL)
222 DO 224 K=1,NUM
224 TEMP(K,4)=SHRINK*(X(K)-XBAR)
230 DO 240 K=1,NUM
    TEMP(K,1)=0.0 $ TEMP(K,2)=1.0
240 TEMP(K,5)=0.0
    BETA(1)=0.0
    S3=WV5M5P(KWT,V,V,W,NUM)
    DELREL=S3
    S1=WV5M5P(KWT,TEMP(1,2),TEMP(1,2),W,NUM)
    KK,ITFL=0
    DO 242 K=1,MAXPOL
    D(K)=0.0 $ A(K)=0.0
    DO 242 J=1,3
242 COEF(J,K)=0.0
    COEF(2,1)=1.0
    DO 600 I=1,MAXPOL
    IF(KKK.EQ.0) GOTO 700
260 S4=WV5M5P(KWT,V,TEMP(1,2),W,NUM)
    B(I)=S4/S1 $ S5=S3-B(I)**2*S1 $ DELSQ(I)=S5
    IF(S5.LE.0.0) GOTO 264
    IF(DELREL/S5.LT.1.0E+6) GOTO 278
264 IF(KWITFL.EQ.1) GOTO 270
    ST=B(I)
    DO 266 M=1,NUM
266 TEMP(M,3)=V(M)-(TEMP(M,5)+ST*TEMP(M,2))
  * GOTO 275
270 SS=B(I-1)
    ST=B(I)
    DO 272 M=1,NUM
272 TEMP(M,3)=V(M)-(TEMP(M,5)+SS*TEMP(M,1)+ST*TEMP(M,2))
275 S5=WV5M5P(KWT,TEMP(1,3),TEMP(1,3),W,NUM)
    DELREL=S5
    DELSQ(I)=S5
278 IF(NUM-I.NE.0) GOTO 280
    SIGSQ(I)=S5
    IRETUR=1
    GOTO 370
280 SIGSQ(I)=S5/FLOAT(NUM-I)
    IF(KOPTIM.EQ.1) GOTO 370
    IF(I.LE.MIN+1) GOTO 370
    IF(KKK.NE.-1) GOTO 370
    IF(MESSAG.NE.0) GOTO 370
285 IF(S5.GT.DELLIM) GOTO 300
290 MESSAG=1
    KKK=KBEVON
    IF(IOUT.NE.0) GOTO 370
300 IF(KWITFL.EQ.0) GOTO 330
310 IF(SIGSQ(I).LE.(1.0-2.0*PERCEN)*SIGSQ(I-2)) GOTO 320
312 IF(IOUT.EQ.0) GOTO 316
    NNN=I-3
    NOT=I-2
316 KKK=KBEVON
    KWITFL=0 $ KOPTIM=1
    IF(KKK.NE.0) GOTO 325
317 NOT=I-3 $ GOTO 710
319 KKK=KKK-1 $ GOTO 325
```

```
320 KWITFL=0
325 II=I-1
    ASSIGN 370 TO KATSUP
    SS=B(II)
    DO 328 M=1,NUM
328 TEMP(M,5)=TEMP(M,5)+SS*TEMP(M,1)
    GOTO 410
330 IF(I.EQ.1) GOTO 370
    IF(SIGSQ(I).GT.(1.0-PERCEN)*SIGSQ(I-1)) KWITFL=1
370 DO 380 M=1,NUM
380 TEMP(M,3)=TEMP(M,4)+TEMP(M,2)
    S0=WVSMSP(KWT,TEMP(1,3),TEMP(1,2),W,NUM)
    ALPHA(I+1)=S0/S1
    IF(KWITFL.EQ.1) GOTO 395
388 DO 390 M=1,NUM
    TEMP(M,5)=TEMP(M,5)+B(I)*TEMP(M,2)
390 CONTINUE
395 DO 397 M=1,NUM
    TEMP(M,3)=TEMP(M,3)-ALPHA(I+1)*TEMP(M,2)-BETA(I)*TEMP(M,1)
    TEMP(M,1)=TEMP(M,2) ◊ TEMP(M,2)=TEMP(M,3)
397 CONTINUE
    IF(KWITFL.EQ.1.AND.KKK.EQ.-1) GOTO 595
    II=I
    ASSIGN 595 TO KATSUP
410 DO 420 M=1,II
420 D(M)=D(M)+B(M)*COEF(2,M)
    IF(KKK.GT.0) KKK=KKK-1
    IF(ISH.NE.0) GOTO 440
430 CALL A4C190(A,SHRINK,XBAR,B,COEF,II)
    GOTO 450
440 DO 442 M=1,II
442 A(M)=D(M)
450 IF(KKK.EQ.0.OR.II.EQ.MAXPOL) GOTO 460 ◊ GOTO 465
460 IRETUR=1
465 IF(IRETUR.EQ.0) GOTO 500
    NOT=II-1 ◊ GOTO 710
500 COEF(3,1)=-ALPHA(II+1)*COEF(2,1)-BETA(II)*COEF(1,1)
    DO 510 M=2,II
510 COEF(3,M)=COEF(2,M-1)-ALPHA(II+1)*COEF(2,M)-BETA(II)*COEF(1,M)
    COEF(3,II+1)=1.0
    JRW=II+1
    DO 520 M=1,JRW
    COEF(1,M)=COEF(2,M)
520 COEF(2,M)=COEF(3,M)
    GOTO KATSUP,(370,595)
595 S0=WVSMSP(KWT,TEMP(1,2),TEMP(1,2),W,NUM)
    BETA(I+1)=S0/S1
    S1=S0 ◊ S3=S5
600 CONTINUE
    NOT=MAXPOL-1 ◊ GOTO 710
700 NOT=I-1
710 ACCUR(1)=0
    ACCUR(2)=RESSUM
    ACCUR(3)=SIGSQ(NOT+1)
    ACCUR(4)=SHRINK
    ACCUR(5)=XBAR
    N=N-20 ◊ M=NOT+1
    DO 900 I=1,N
900 V(I)=V(I+M)
    RETURN
    END
```

```
      FUNCTION WVSMSP(KK,X,V,W,N)
      DIMENSION X(1),V(1),W(1)
C     THIS SUBROUTINE IS DESCRIBED IN THE UNIVERSITY OF WISCONSIN
C     COMPUTER CENTER MANUAL , VOLUME IV , SUPPLEMENT H ,
C     FEBRUARY 1969.
      SUM=0.0 $ IF(KK.EQ.0) GOTO 200
      DO 100 I=1,N
  100 SUM=SUM+X(I)*V(I)
      GOTO 400
  200 DO 300 I=1,N
  300 SUM=SUM+X(I)*V(I)*W(I)
  400 WVSMSP=SUM
      RETURN
      END


      SUBROUTINE A4C190(A,XOMEGA,XLAMBD,B,C,MM)
      DIMENSION A(31),B(31),C(3,31)
C     THIS SUBROUTINE IS DESCRIBED IN THE UNIVERSITY OF WISCONSIN
C     COMPUTER CENTER MANUAL , VOLUME IV , SUPPLEMENT H ,
C     FEBRUARY 1969.
      M=MM $ COL=1.0 $ FACTOR=-XOMEGA*XLAMBD
      DO 200 J=1,M
      POWER=1.0 $ SUM=0.0 $ ENTRY=1.0 $ TJ=J
      DO 100 I=J,M
      TI=I $ SUM=SUM+C(2,I)*POWER*ENTRY $ POWER=POWER*FACTOR
  100 ENTRY=ENTRY*TI/(TI-TJ+1.0)
      A(J)=A(J)+SUM*COL*B(M)
  200 COL=COL*XOMEGA
      RETURN
      END


      SUBROUTINE SEARCH(K,A,N,L,C)
      DIMENSION A(1)
C     THIS SUBROUTINE IS DESCRIBED IN THE UNIVERSITY OF WISCONSIN
C     COMPUTER CENTER MANUAL , VOLUME IV , SUPPLEMENT H ,
C     FEBRUARY 1969.
      REAL LARGE
      IF(K.GT.0) GOTO 100
      LARGE=A(1) $ L=1
      DO 10 I=2,N
      IF(LARGE.GE.A(I)) GOTO 10
      LARGE=A(I) $ L=I
   10 CONTINUE
      C=LARGE
      RETURN
  100 SMALL=A(1) $ L=1
      DO 20 I=2,N
      IF(SMALL.LE.A(I)) GOTO 20
      SMALL=A(I) $ L=I
   20 CONTINUE
      C=SMALL
      RETURN
      END
```