A computing program derived from Chree’s method

M. Attolini (*) – L. Dall’Olio (**) – V. Guerrieri (***) – P. Randi (*)

Ricevuto il 23 Novembre 1968

SUMMARY. A computing program has been activated for having information from stationary time series.

The technique utilized is that of superposed epochs (Chree’s method), adapted by using large computers.

This technique gives evidence for a phenomenon and statistical consistency is provided.

The program described elaborates unlimited numbers of data and epochs the maximum length of which is 2000 numerical data.

INTRODUCTION.

In graphic registrations, or tabulations of some quantities, that constitute a stationary time series, the problem that arises is that of testing the consistency of certain particular phenomenon that are noticed in correspondence with determinate physical events (1) (2).


(**) Centro di Calcolo di Bologna. Comitato Nazionale per l’Energia Nucleare.
The graphed or tabulated quantities correspond in some way to these physical events that, by other methods, can be localized in time.

A typical example is constituted by the oscillations observed in recorded cosmic radiation in correspondence with the sudden commencement of magnetic storms, whose position in time can be found by observing the magnetograms.

The phenomenon we must test can be approximated to the order of the statistical fluctuation (or a little better) of the data series whose registration is examined, and therefore necessitates a particular analysis for determining the characteristics.

In this work we propose a solution to the problem by using the technique of Chree of superposed epochs (3) for using electronic computers and completing it by an posterior analysis for proving the statistical consistency of the results regarding the width and the duration of the phenomena. Besides, the maximum temporal position is localized which allows us to visualize the bond between the registered phenomena and the physical events which are bound to it.

**METHOD.**

A stationary time series can be thought of as a linear combination of type:

\[ I(t) = A(t) + \epsilon, \]

being \( A(t) \) the systematic part and \( \epsilon \) the random one of the series.

The aim of pointing out the phenomena according to the type of series which we want to separate, we must utilize a preventive filter operation (4) of which a computing program (5) already exists.

In the numerical filtering steps the reduction of the influences by the random part in the residual series allows to note the phenomenon without modifying the structure.

Let us call \( q(r = 1, 2, \ldots, n) \) the corresponding values in the recording of the physical events which we consider responsible for the phenomenon that we must examine.

Let us construct the matrix:

\[
\begin{pmatrix}
p_{1,1} & p_{1,2} & \cdots & p_{1,n} & q_{1} & d_{1,1} & d_{1,2} & \cdots & d_{1,m} \\
p_{2,1} & p_{2,2} & \cdots & p_{2,n} & q_{2} & d_{2,1} & d_{2,2} & \cdots & d_{2,m} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
p_{n,1} & p_{n,2} & \cdots & p_{n,n} & q_{n} & d_{n,1} & d_{n,2} & \cdots & d_{n,m}
\end{pmatrix}
\]
being \( p_{s,r}(s = 1,2, \ldots, l) \) the numerical values of the \( l \) positions preceding the position of the event \( q \), and \( d_{s,r} \) the values of the \( m \) positions following.

Each row of the matrix, \( n \times R \ (R = l + m + 1) \), contains the elements of each one of the \( n \) epochs \( (t) \).

As for random part, between the variance of \( y(s = 1,2, \ldots, r) \) (obtained by adding and by averaging per column the elements of the matrix:

\[
\begin{align*}
y_i &= \frac{1}{n} \sum_{r=1}^{n} p_{i,r} \quad (i = 1,2, \ldots, l) \\
y_{i+1} &= \frac{1}{n} \sum_{r=1}^{n} q_r \\
y_{i+m} &= \frac{1}{n} \sum_{r=1}^{n} d_{s,r} \quad (i = 1,2, \ldots, m)
\end{align*}
\]

of which \( S_2 \) is the estimate) and the mean variance of each of the \( n \) epochs (of which \( S_2 \) constitutes the estimate) is the relation

\[
\bar{S}^2 = n S_2^2.
\]

The standard deviation \( S_1 \) is

\[
x, d, (\bar{S}^2) = S_1 = \sqrt{\frac{2}{R - 1} - \frac{\bar{S}^2}{n}} \quad \sqrt{\frac{2}{R - 1}}.
\]

Finally:

\[
S = \sqrt{\frac{R - 1}{2} \left( \frac{n S_1^2}{\bar{S}^2} - 1 \right)}.
\]

Being \( n \) the number of the superposed epochs, represents, in standard deviation units the true shift of the data series from random behaviour, and \( S \) is the significance level of the phenomenon in the series.

The length \( R \) of the epoch depends from the physical problem examined. The preceding \( (p_{s,r}) \) and successive \( (d_{s,r}) \) data of the event may be equally populated \( (l = m) \).

Since the effects of certain physical event are generally allayed in the recordings, whether for the inertia of the system on which it acts or by the recorder that smoothes the various phenomena exploring them in a large width, a time contraction of the elements forming the epoch is elaborated. Such an operation permits a better
localisation of the phenomenon studied and ulteriorly decreases that statistical fluctuation of the series.

The problem studied will guide the choice of such a contraction. The relation (1) is modified:

\[ S(k) = \sqrt{\frac{k - 1}{2k}} \left( k \cdot \frac{S_n^2}{S^2} - 1 \right) \quad \text{for } k \text{ integer} \]

being \( k \) the order of the temporal contraction \( S_n \), the variance of the contracted mean epoch.

A right representation of \( S(k) \) will permits useful considerations on the localisation and duration of the phenomena studied.

**DESCRIPTION OF THE PROGRAM.**

The program has been carried out at “Centro di Calcolo del CNEN” in Bologna using the FORTRAN IV language versione 13 under the IBSYS 7094/7040 DCS monitor control.

The data number that can be elaborated is practically unlimited; the maximum allowed number of elements of the epoch is 2000. This number is extremely sufficient for many problems but it can also be increased by modifying the appropriate statements.

The number \( n \) of superposed epochs is 200. This number may be increased by modifying the appropriate statement, but attention must be made in order of the approximation due to the machine representation of the value of the elements, that arises when mean epoch is computed.

The program allows also to group, \( k \) by \( k \), the set of elements in the mean epoch for giving an epoch whose variance is \( k \) times less than that of the original mean epoch.

This procedure allows a better accuracy in the computing of the average width of the studied event without phase changes, because, if \( k \) is odd, \( y_{me} \) will always be in the middle of the interval of the obtained epoch.

The program can also be informed to ignore the not good elements of the series without modifying the input data series cards.

It is useful to note \( S^2 \) is computed not using the mean of the variances of each row, but the mean of the variances concerning the consecutive epochs, each \( R \) elements long, obtained considering all
the time data series (except the not good elements); the variance computed in this way tends to the true variance of the series.

OUTPUT.

Normally we can obtain a following list:

a) input tape (NTPI)
b) number of data per input card (N18)
c) input FORMAT (FR1)
d) print FORMAT (FR2)
e) punch FORMAT (FR3)
f) number of superposed epochs or q (NKPO)
g) number of p (NOCR)
h) number of d (NODO)
i) index of q positions in time-series (KPO)
j) number of utilized data
k) number of consecutive epochs
l) mean epoch information:
   1) mean variance (SIGR2)
   2) standard deviation (SB)
   3) variance (SIGMA)
   4) significance level (S)
   5) ΔIFVER = SIGMA — \frac{SIGR2}{NKPO}

m) length of mean epoch (R)

n) elements of mean epoch (ELM)

o) grouped epoch information:
   1) number of elements (R)
   2) GROUPS
   3) SIGMA
   4) S
   5) ΔIFVER

p) local maxima of S are computed and the following information are listed:
   1) S
   2) SIGMA
   3) type of grouping (GROUPS)
   4) R
If the restart procedure is required, the normal list is like the proceeding from 1) to p).
If the program recognizes errors during its flow appropriate diagnostics are printed.

Order of data cards deck set-up.

All cards must be included in the order shown below,

a) Start procedure (Fig. 1)

- A* Parameter card for starting and grouping
- B* Variable format card for time series input data
- C* Variable format card for printout mean epoch
- D* Variable format card for punched input epoch
- E* Time series data parameter card
- F* Epoch parameter card
- G* q. index card
- H* Time series data
- I* END punched in col. 1—3
- J* Number of groups to be discarded
- K* Index of group to be discarded

b) Restart procedure (Fig. 2)

- A* Parameter card for restarting and grouping
- B* Blank card
A computing program derived from Cheke’s method

*C* Like in a)
*D* Like in a)
*I* Mean epoch parameter card
*M* Mean epoch elements.

Fig. 2

CARD PREPARATION.

<table>
<thead>
<tr>
<th>Card columns</th>
<th>Corresponding variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>A</em> card</td>
<td></td>
<td>FORMAT (413)</td>
</tr>
<tr>
<td>1-2</td>
<td>KTBST</td>
<td>— 1 if restart procedure is required</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≠ 1 if start procedure is required</td>
</tr>
<tr>
<td>3-4</td>
<td>KRAI</td>
<td>Initial grouping. Must always be KRAI &gt; 1.</td>
</tr>
<tr>
<td>5-6</td>
<td>KRAP</td>
<td>Last grouping. Must always be KRAP &gt; KRAI.</td>
</tr>
<tr>
<td>7-8</td>
<td>KRAP</td>
<td>Step for executing intermediate groupings.</td>
</tr>
</tbody>
</table>

*H* card

<table>
<thead>
<tr>
<th>Card columns</th>
<th>Corresponding variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-80</td>
<td>FRI</td>
<td>FORMAT (20A4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>— Format of time series input data. It must be like (AS, specifications for input data). Columns 1-3 cannot be used. This restriction is used to allow an indefinite number of time series input data cards, the end of which is reached when the program read the <em>I</em> card.</td>
</tr>
<tr>
<td>Card columns</td>
<td>Corresponding variable</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| *C* card     | 1-80                    | FORMAT (20A4)  
|              | FE2                     | — Format of output data to be put on standard output tape. |
| *D* card     | 1-80                    | FORMAT (20A4)  
|              | FE3                     | — Format according which one can have the elements of mean epoch on logical tape 8, from the second record on. |
| *E* card     | 1-9                     | NDS         |
|              | 3-3                     | NTPI        |
|              | 5-6                     | KE          |
|              | 7-8                     | IT          |
|              |                         | — Number of time series data contained in each input card. |
|              |                         | — Logical tape number of input (5 if input is from punched cards). |
|              |                         | — Number of decimal figures if conversion is used (*).  
|              |                         | — > 0 if input conversion is required (*). |
|              |                         | — < 0 if input conversion not required (*). |
| *F* card     | 1-2                     | NKPO        |
|              | 3-4                     | NOPR        |
|              | 5-6                     | NODO        |
|              |                         | — Number of superposed epochs. |
|              |                         | — Number of elements preceding qi in the epoch. |
|              |                         | — Number of elements successive to qi in the epoch. |

(*) One must use conversion when time series input data are punched as integer constants having the sign (hole zone) over the less significant figure of the number. In this case each field of width w containing the number, is considered like it were formed of two sequential fields: the 1st of integer-type of width w-1, the 2nd of alphanumeric-type of width 1, i.e. if number 12.123 is punched according to the format specification /"7.3 no conversion is required, being a Fortran number (in this case IT = 0). The character string 1212N (where N means holes 5 and 11, minus sign overpunched) columns of the card, punched in a field of width w = 5 for having the same value as the above one, must be read according to the format specification 14, A1 and must be KE = 3 (decimal figures of the number) and IT ≠ 0.
<table>
<thead>
<tr>
<th>Card columns</th>
<th>Corresponding variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>H</em> card</td>
<td>KPO</td>
<td>FORMAT (1615) — Index of ( g_i ) in the time series input (<em>H</em> card).</td>
</tr>
<tr>
<td>1-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>I</em> card</td>
<td>BL</td>
<td>FORMAT (A3) — Time series input data.</td>
</tr>
<tr>
<td>4-6</td>
<td>(KA, KB)</td>
<td></td>
</tr>
<tr>
<td><em>J</em> card</td>
<td>KEND</td>
<td>FORMAT (I3) — The work END must be punched in the corresponding columns. This card notifies the program that all the cards containing time series input data have been read.</td>
</tr>
<tr>
<td>1-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>K</em> card</td>
<td>NGRU</td>
<td>FORMAT (8(215, 3X)) — Index of the initial and last elements for each group to be discarded among the time series input data.</td>
</tr>
<tr>
<td>NGRUI, NGRUP...</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>L</em> card</td>
<td>LKPO</td>
<td>FORMAT (14, 215, 3E14.6) — Number of elements of the mean epoch.</td>
</tr>
<tr>
<td>1-4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOPB</td>
<td>See <em>F</em> card.</td>
</tr>
<tr>
<td>5-7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NKPO</td>
<td>See <em>F</em> card.</td>
</tr>
<tr>
<td>8-10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Card columns | Corresponding variable | Description
---|---|---
*L* card (cont.)
11-24 | SIGMA (1) | Variance of the mean epoch.
25-38 | S (1) | Significance level of the mean epoch.
39-52 | SIGR2 | Mean variance of data series.
53-66 | SE | Standard deviation of data series.
67-80 | DIFVER (1) | SIGMA (1) - SIGR2/NKPO

*M* card
81-100 | RLM | FE3 format (see ‘*I*’ card)

**Description.**

In the following pages it has been reproduced the list of the described program.

The authors have mainly contributed each in their activity field.

**Timing.**

1) It takes 7" to elaborate 1000 elements of stationary time series, the length of the epoch being 100, corresponding to \( n - 10 \), to group from 1 to 10 step 1, and to find the local maxima.

2) It takes 1" to elaborate the restart procedure grouping from 4 to 9.

**Acknowledgement.**

Prof. E. Clementel made easy the IBM 7094/7040 DCS computers utilization at the Centro di Calcolo del CNEN in Bologna (SHARE Installation B1).
A COMPUTING PROGRAM DERIVED FROM CHIJKER'S METHOD

**DIMENSIONS**
- FR1(20), FR2(20), FR3(20)
- KP0(100), KB1(100), KB2(100)
- LEPP(100), SIGMA(100), SIGMA(100)
- LIMP(100), LIM(100), LIM(100)
- DATA KALT, KEND

**EPOCHS**

```
DIMENSION FR1(20), FR2(20), FR3(20), KP0(100), KB1(100), KB2(100),
LEPP(100), SIGMA(100), SIGMA(100),
LIMP(100), LIM(100), LIM(100),
DATA KALT, KEND
```

**INPUT OF PARAMETERS**

```
READ(5,5) KTEST, KRAP, KRAP
READ(5,5) FR1, FR2, FR3
IF(KFJN.EQ.3) GO TO 440
READ(5,5) NDS, NTP1, KE, IT
KE = 100 * KE
READ(5,5) NKPO, NODR, NODO
READ(5,5) (KPO(I), I = 1, NKPO)
WRITE(6,9C2) NTP1, NDS, FR1, FR2, FR3, NKPO, NODR, NODO
WRITE(6,9C3)(KPO(I), I = 1, NKPO)
```

**INPUT OF DATA (NDS PER CARD) AND RECORDING ON BINARY TAPE 3 (NDS PER**

```
REWIND 3
NREC = 0
IF(IT.LE.0) GO TO 500
501 READ(NTP1, FR1, KFIN(ELJ), KBJ(J), J = 1, NDS)
IF(KFIN.EQ.KALT) GO TO 502
DO 503 J = 1, NDS
KB(I) = MOD(KB(J), 10) + 10 + MOD(KB(J), 10)
NREC = NREC + 1
WRITE(3)(EL(J), J = 1, NDS)
GO TO 501
500 READ(NTP1, FR1, KFIN(ELJ), J = 1, NDS)
IF(KFIN.EQ.KALT) GO TO 502
NREC = NREC + 1
WRITE(3)(EL(J), J = 1, NDS)
GO TO 500
502 NTP1 = NREC + NDS
REWIND 3
```

**COMPUTATION OF VARIANCE CONCERNING EACH EPOCH**

```
LEPP = NODR + NODO
NTP0 + NTP1 / LEPO
NTP0 + NTP1 + LEPO
WRITE(6,9C4)
```
READI, NGRU
IF(NFNL, //NGRU==0) GO TO 72C
READI, //NGRUI (1, NGRUFI(1, 1=1, NGRU)
WRITEI, //NGRUI (1, 1=1, NGRU)
710 DC TOT = TOT + 1, NT0P
701 LEFF(I) = LEFF0
SIG2 = SIG2
SIGR = SIGR
K = 0
JJ = 1
I1 = 1
NC = 0
DC 40C NRE = NRE
READI, (E(I, JJ, J=1, NDS)
DC 40T J = J, NDS
IF(LEFF(I1, NGRUJG0 TO 704
NC = NC + 1
IF(NC <= NGRU(1)) II = II + 1
704 TCT = TCT + E(I)
SIG2 = SIG2 + E(I) + E(I)
705 K = K + 1
IF(K, NE, LEFPO) GO TO 401
SIG2 = SIG2 - TOT * TOT / (FLOAT(LEFF(JJ) - 1.)
SIGR = SIGR + SIG2
TCT = 0
SIG2 = 0
SIGR = 0
I1 = 1
401 CONTINUE
SIGR2 = SIGR / FLOAT(NTEPO)
C
C COMPUTATION OF MEAN EPOCH
C
411 ELMI(I) = GA
412 LEPM(I1, (I, NMPO
413 LEPO(I1, (I, LEPO
KPQK(I) = KPQK(I1) - NPO
IF(KPQK(I1), CT, J, GO TO 412
KF = IAS3 * KPQK(I1) + 1
DC 732 K = K + 1
732 KCIVK = KCIVK + 1
LEPO(I1, LEPO(I1) - KF
KPQK(I) = KPQK(I1) + KF
412 CONTINUE
DO 413 L = 1, NMPO
RCWNC 3
1=1
100
101
102
103
104
105
106
107
108
110
111
112
113
114
115
116
A COMPUTING PROGRAM DERIVED FROM CHEEKS’S METHOD

J=1+1
I=2+1
J=LEPO-L(EP0)
I=NGRI,GT0,JGO TO 711
I=2+1
J=2+1
711 DC W(I) L=1,L(REC
READ(IJ) (El12),L2=1,NOS
KI=NGRI+L11
L=MT4+2
IF(KI>LPO(I)) GO TO 414
J=2+1
IF(LP(I)+LEPO) GO TO 413
GO TO 705,705+1,1
705 DC W(I) IG=1,NGRU
J=1+1
I=2+1
706 GO TO (705,709),12
708 IF(KI>LENGRI+II1) AND KI,LENGRU(I)GO TO 707
IF(KI,LENGRU(I)) II=II+1
IF(KI,LENGRU(I) II=2
709 EML(II)+EML(I)+EL(I2)
GO TO 414
707 ICIV(I)+ICIV(I)+1
I=2+1
414 CONTINUE
IF(LP(I)+LEPO) GO TO 413
J=2+1
DC W(I) K=J+1,LEPO
ICIV(I)+ICIV(I)+1
416 CONTINUE
413 CONTINUE
C=======================================================================
C COMPUTATION OF STANDARD DEVIATION
C=======================================================================
TSIGR2=SIGR2/DFLOAT(NKPC)
SE-TSIGR2>SQR2/FLOAT(LEPO-1)
C=======================================================================
C LINEAR INTERPOLATION
C=======================================================================
IZFOO
EO W(I) L=LEPO
IF(DIV(I)+L(0)) GO TO 733
EI12=EI12/FLOAT(KIV(E))
GO TO 735
733 IF(ZEO1014,
I=1+1
415 CONTINUE
J=2+1
I=2+1
I=1+1
A COMPUTING PROGRAM DERIVED FROM CHEEKE'S METHOD

WRITE(*,643)KK 233
GO TO 678 234
644 DC 645 1=13,KK 235
11=1 236
IF!(TEST.EQ.1) KRA=KRA+1 237
CALL RAG1ELM,ELM1,LEPO,NOPR,KRA,LEPRA,IER) 241
WRITE(6,646)II-1),SIGMA(I I- 1),KRA,KRA,LEPRA 242
CONTINUE 243
678 WRITE(6,914) 244
END FILE 0 245
STOP 246
C 247
C======================================================================= 248
C IF TEST = 1 INPUT OF MEAN EPOCH TO RESTART A FORMER COMPUTATION 249
C======================================================================= 250
C 440 READ(5,908) NOPR,NKPO,SIGMA I 1 ), SIGR2,SE,DIFVER(I) 252
TKPO=NKPO 253
I3=KRA I  + 2 254
WRITE(6,916) 255
READ(5,FR3)(ELMI  I), 1=1 1 ,LEPO ) 256
GO TO 912 257
G 258
C==  = = = = = = = = = = = = = = = = = = = = = = ====  = = = = = = = = = = = = = ===  = === = === ======  = = = = === 259
C 260
4 FORMAT(13) 261
5 FORMAT(413) 262
15 FORMAT(6(215,3X1) 263
15 FORMAT(20A4) 264
4 FORMAT(25,5) 265
6 FORMAT(15,6) 266
601 FORMAT(1H1,10X,15HMEAN EPOCH R = I6/1 1 X,10 1H-) ,///) 267
641 FORMAT(1H1 10X,16HGR0LPPING NUMBE RI 3, 1 15H CANNOT BE MADE5X,6HNOPR = 268
11 16, 10X, 5HK P.A =15) 269
643 FORMAT(1H0,10X,15HTHE GROUPS ARE I2.41H " THE LOCAL MAXIMUM CANNOT 270
BE DETERMINED) 271
646 FORMAT(1H0,10X,3HS =E14.6, 8X,7HSIGMA =E14.6, 8X,SHGROUPING,14,3H 272
1BY,15,8X,3H ) 273
901 FORMAT(1H1,10X,15HE 0 0 C H S/ 11X,13H INPUT TAPE I3/11X,13HDATA PER CARD 274
1THE INPUT 275
902 FORMAT(1H1,10X,15HINDEX OF SUPERPOSED POINTS Q! I)//, 11X,5(216,3H 276
THE ELEMENTS PRECEDING QI 4/1 1X,34HNUMBER OF THE ELEMENTS FOLLOWING 277
4 QI 4 ) 278
4 QI 279
903 FORMAT(1H1,31HINDEXES OF THE FIRST AND THE LAST ELEMENT TO DISCARD 280
1 IN EACH GROUP//,11X,5(216,3H 281
904 FORMAT(1H1,10X,2HNUMBER OF UTILIZED ELEMENTS/11X,2HNUMBER OF C 282
TONESECTIVE EPOCHS ) 283
905 FORMAT(1H1,10X,5HNUMBER OF GROUPS OF CONSECUTIVE ELEMENTS TO DIS 284
C) 285
906 FORMAT(1H1,6HINDEXES OF THE FIRST AND THE LAST ELEMENT TO DISCARD 286
1 IN EACH GROUP//,11X,5(216,3H 287
907 FORMAT(1H1,10X,2HMEAN VARIANCE SIGMAE15.6/ 288
1 SIGR2E15.6/ 289
2 SIGMAE15.6/ 290
$ IBFC VRJ M49, K7, LIST
SUBROUTINE VARI(EPO, SIGMA, S, KK, LUP, TKPO, SIGR2, DIFVER)
C
DIMENSION EPO(1), SIGMA(1), S(1), DIFVER(1)
C
T = LUP
DIF = SIGR2 / TKPO
TCT = 0
SIGMA(KK) = 0
DC = 1 - 1 / LUP
TCT = TCT + EPO(I)
SIGMA(KK) = SIGMA(KK) + EPO(I) * EPO(I)
CONTINUE
SIGMA(KK) = SIGMA(KK) - TOT * TOT / TL / (TL - 1)
S = SQRT((TL - 1) / 2) * TKPO / SIGMA(KK) / SIGR2 - 1
DIFVER(KK) = SIGMA(KK) - DIF
IF (KK .EQ. 1) RETURN
WRITE (6, 2) LUP, KK, SIGMA(KK), S, DIFVER(KK)
RETURN
2 FORMAT (11G12.10,3E20.6)
END

$ IBFC VGR M49, K7, LIST
SUBROUTINE VGR(EPO, EPO, LEP0, NOPR, KRA, LEPRA, IER)
C
IER = 1
NO = NOPR - KRA / 2
IF (NO .LT. KRA) GO TO 1
IN = MOD(IN, KRA) + 1
GO TO 2
1 IF (NO .LT. KRA +1) GO TO 1
IN = IN + 1
2 LEPRA = (LEPO - IN + 1) / KRA
END

DIMENSION EPO(1), EPPO(1)
C
IER = 1
NO = NOPR - KRA / 2
IF (NO .LT. KRA) GO TO 1
IN = MOD(IN, KRA) + 1
GO TO 2
1 IF (NO .LT. KRA +1) GO TO 1
IN = IN + 1
2 LEPRA = (LEPO - IN +1) / KRA
END
A computing program derives from Chree's method

11=11+1
DO 4 J=11,12
  4 EPK(IJ)=EPK(IJ)+EP0(IJ)
  11=12+1
  12=12+KRA
3 CONTINUE
DO 5 I=1,LEPRA
  5 EPK(I)=EPK(I)/FLOAT(KRA)
RETURN
7 IER=2
RETURN
END

REFERENCES