# Fortran Programs for Printing General Forms of the Expansion Formulae in the Damping Theory

**Itsuko Sato***

The Doctoral Research Course in Human Culture,
Ochanomizu University, Bunkyo-ku, Tokyo 112

(Received September 9, 1986)

The damping theoretical framework is useful in treating nonequilibrium systems. Expansion formulas in this theory are written in terms of the "partial cumulant" and "ordered cumulant". A FORTRAN program to print out these cumulants is given in this paper.

## § 1. Introduction.

In the field of nonequilibrium statistical mechanics, the damping theoretical method plays an important role. By this method, we can extract necessary information on the relevant system which interacts with its surrounding environment.

Shibata and Arimitsu[1] obtained the two kinds of formulas: One is the expansion formula for the known time convolution equation (TCE) and the other is for the newly derived time convolutionless equation (TCLE). The TCE is expressed in terms of the "partial cumulant" whereas the TCLE is written in terms of the "ordered cumulant". These equations are used to treat various physical systems. For instance, we could solve exactly a model of low field magnetic resonance with the use of the TCE[2]. We could also treat a problem of exciton migration quite satisfactory by TCE[3]. The TCLE was also used in treating such phenomena like exchange dephasing[4] and spin relaxation[5].

Thus in applying each formula to actual problems, explicit forms of these cumulants are necessary. Of course, from the general formulas of ref. 1 we can write down necessary expression. Nonetheless, it is more convenient to use the computer. This is because, especially in higher orders, we must treat a large number of terms to obtain these cumulants.

First we summarize the two kinds of expansion formulas and give the nature of "partial cumulant" and explain the method of "how to use the programs". In the last section we give the summary of this work and possible applications to other problems.

* Present Address: Toshiba R & D Center, Toshiba Corporation, Komukai, Saiwai-ku, Kawasaki 210.

## § 2. Summary of the expansion formulas.

*2-1. TCE*

We sometimes treat an equation like

$$\frac{\partial}{\partial t} W(t) = gL_1(t) W(t) . \tag{2.1}$$

We separate a relevant part $\mathcal{P}W(t)$ from $W(t)$ where $\mathcal{P}$ is a projection operator which extracts necessary information; the remaining irrelevant part is $QW(t)$ where $Q=1-\mathcal{P}$.

After eliminating $QW(t)$, we have

$$\frac{\partial}{\partial t} \mathcal{P}W(t) = g\mathcal{P}L_1(t)\mathcal{P}W(t) + \int_{t_0}^{t} \mathrm{d}\tau \Phi(t,\tau)\mathcal{P}W(\tau) + I(t) \tag{2.2}$$

where

$$\Phi(t,\tau) = g^2 L_1(t)\mathcal{G}(t,\tau)QL_1(\tau) \tag{2.3}$$

and

$$I(t) = g\mathcal{P}L_1(t)\mathcal{G}(t,t_0)QW(t_0) , \tag{2.4}$$

$$\mathcal{G}(t,\tau) = \exp_{-}\left[g\int_{\tau}^{t} \mathrm{d}sQL_1(s)\right] . \tag{2.5}$$

By expanding these quantities in powers of $g$, we find

$$\begin{aligned}&\int_{t_0}^{t} \mathrm{d}\tau \Phi(t,\tau)\mathcal{P}W(\tau)\\ &= \sum_{n=2}^{\infty} g^n \int_{t_0}^{t} \mathrm{d}t_1 \int_{t_0}^{t_1} \mathrm{d}t_2 \cdots \int_{t_0}^{t_{n-2}} \mathrm{d}t_{n-1} \mathcal{P}L_1(t)QL_1(t_1)\cdots QL_1(t_{n-1})\\ &\qquad \times \mathcal{P}W(t_{n-1}) .\end{aligned} \tag{2.6}$$

For a special kind of $\mathcal{P}$ defined by

$$\mathcal{P}X = \langle X \rangle_B$$

where $\langle \cdots \rangle_B$ denotes a certain average, (2.6) can be written in the form

$$(2.6) = \int_{t_0}^{t} \mathrm{d}t_1 \int_{t_0}^{t_1} \mathrm{d}t_2 \cdots \int_{t_0}^{t_{n-2}} \mathrm{d}t_{n-1} \langle L_1(t)L_1(t_1)\cdots L_1(t_{n-1})\rangle_{B,P.C.} \langle W(t_{n-1})\rangle_B \tag{2.7}$$

where we have defined the "partial cumulant" by

$$\begin{aligned}&\langle L_1(t)L_1(t_1)\cdots L_1(t_{n-1})\rangle_{B,P.C.}\\ &= \Sigma'(-)^q \langle L_1(t)\cdots\rangle_B \langle L_1(t_j)\cdots\rangle_B \langle L_1(t_l)\cdots\rangle_B .\end{aligned} \tag{2.8}$$

In (2.8), the sum is taken over all possible divisions keeping the chronological order; $q$ is the number of divisions in the term.

Here we write down the first few terms:

$$\langle L_1(t)\rangle_{B,P.C.} = \langle L_1(t)\rangle_B , \tag{2.9a}$$

$$\langle L_1(t)L_1(t_1)\rangle_{B,P.C.}=\langle L_1(t)L_1(t_1)\rangle_B-\langle L_1(t)\rangle_B\langle L_1(t_1)\rangle_B\,, \tag{2.9b}$$

$$\begin{aligned}&\langle L_1(t)L_1(t_1)L_1(t_2)\rangle_{B,P.C.}\\&=\langle L_1(t)L_1(t_1)L_1(t_2)\rangle_B-\langle L_1(t)L_1(t_1)\rangle_B\langle L_1(t_2)\rangle_B\\&\quad-\langle L_1(t)\rangle_B\langle L_1(t_1)L_1(t_2)\rangle_B+\langle L_1(t)\rangle_B\langle L_1(t_1)\rangle_B\langle L_1(t_2)\rangle_B\,.\end{aligned} \tag{2.9c}$$

*2-2. TCLE*

In the TC formalism we obtain (2.2) starting from (2.1). We proceed, however, in a different way: After integrating the equation for $\mathcal{P}W(t)$, we can renormalize the memory effect in (2.2) to obtain the TCL equation given by

$$\frac{\partial}{\partial t}\mathcal{P}W(t)=K(t)\mathcal{P}W(t)+I(t)\,. \tag{2.10}$$

Where

$$K(t)=g\mathcal{P}L_1(t)[1-\Sigma(t)]^{-1}\,, \tag{2.11}$$

$$I(t)=g\mathcal{P}L_1(t)[1-\Sigma(t)]^{-1}\mathcal{G}(t,t_0)\,, \tag{2.12}$$

$$\Sigma(t)=g\int_{t_0}^{t}\mathrm{d}\tau\,\mathcal{G}(t,\tau)\mathcal{Q}L_1(\tau)\mathcal{P}G(t,\tau) \tag{2.13}$$

with

$$\mathcal{G}(t,\tau)=\exp_{\leftarrow}\left[g\int_{\tau}^{t}\mathrm{d}s\,\mathcal{Q}L_1(s)\right]\,, \tag{2.14}$$

$$G(t,\tau)=\exp_{\rightarrow}\left[-g\int_{\tau}^{t}\mathrm{d}s\,L_1(s)\right]\,. \tag{2.15}$$

In (2.14) the function is time-ordered from the right while in (2.15) it is from the left.

It has been shown that a general structure of $K(t)$ is expressed in terms of "ordered cumulant" when $K(t)$ is expanded in powers of $g$:

$$K(t)=\sum_{n=1}^{\infty}g^nK_n(t) \tag{2.16}$$

where

$$K_n(t)=\int_{t_0}^{t}\mathrm{d}t_1\int_{t_0}^{t_1}\mathrm{d}t_2\cdots\int_{t_0}^{t_{n-2}}\mathrm{d}t_{n-1}\langle L_1(t)L_1(t_1)\cdots L_1(t_{n-1})\rangle_{B,O.C.} \tag{2.17}$$

We write down the first few terms:

$$\langle L_1(t)\rangle_{B,O.C.}=\langle L_1(t)\rangle_B\,, \tag{2.18a}$$

$$\langle L_1(t)L_1(t_1)\rangle_{B,O.C.}=\langle L_1(t)L_1(t_1)\rangle_B-\langle L_1(t)\rangle_B\langle L_1(t_1)\rangle_B\,, \tag{2.18b}$$

$$\begin{aligned}&\langle L_1(t)L_1(t_1)L_1(t_2)\rangle_{B,O.C.}\\&=\langle L_1(t)L_1(t_1)L_1(t_2)\rangle_B-\langle L_1(t)L_1(t_1)\rangle_B\langle L_1(t_2)\rangle_B\\&\quad-\langle L_1(t)L_1(t_2)\rangle_B\langle L_1(t_1)\rangle_B-\langle L_1(t)\rangle_B\langle L_1(t_1)L_1(t_2)\rangle_B\\&\quad+\langle L_1(t)\rangle_B\langle L_1(t_1)\rangle_B\langle L_1(t_2)\rangle_B+\langle L_1(t)\rangle_B\langle L_1(t_2)\rangle_B\langle L_1(t_1)\rangle_B\,.\end{aligned} \tag{2.18c}$$

## § 3. Partial Cumulant.

In § 2 we gave the partial cumulant in the form of (2.8). A first few lower order terms are explicitly given by (2.9). Higher order terms are obtained by the following prescription:

( i ) $L_1(t), L_1(t_1), \cdots, L_1(t_{n-1})$ are ordered in this way (namely, in the order of $t \geqq t_1 \geqq \cdots \geqq t_{n-1}$).

(ii) In succeeding $L_1(t_j)$'s, there are two possibilities; whether they are divided or not. All possible cases must be realized for $(n-1)$ possible divisions. When there is no division between two other succeeding divisions, an enclosing part is an averaged quantity separated from others.

(iii) The $+(-)$ sign is attached when the number of division is even (odd).

These rules can be transcribed into the ones for computer.

*3-1. Treatment in the computer*

Let us first give a transcription rule for (ii). Once $L_1(t), L_1(t_1), \cdots,$ $L_1(t_{n-1})$ are given, we have $(n-1)$ spaces among succeeding $L_1(t_j)$'s. For each space ($\Rightarrow$ "box") we have two possibilities; namely, whether it is occupied (divided) or not. We assign "1" for an occupied box and "0" for an unoccupied box. Therefore there are 2 possible configurations each of which is associated with a corresponding binary digit. A method of converting the decimal system to the binary system can easily be accomplished with the use of a single DO-loop which does not violate the usual FORTRAN rule.

A program which makes a series of numbers composed of "1" and "0" is then called "DVCM".

In actual applications, we sometimes treat a case of $\langle L_1(t_j)\rangle_B=0$. In order to eliminate these contributions, we are asked from DVCM that "whether $\langle L_1(t_j)\rangle_B$ vanishes or not". When $n$ is even: input 0 corresponds to $\langle L_1(t_j)\rangle_B=0$ whereas 1 to non-zero. When $n$ is odd, the corresponding partial cumulant vanishes when $\langle L_1(t_j)\rangle_B=0$.

Next we present a program which gives a series of $L_1(t), L_1(t_1), \cdots,$ $L_1(t_{n-1})$ in a suitable form. The quantity $L_1(t)$ and $L_1(t_j)$ are represented by 0 and $j$, respectively.

When succeeding $L_1(t_j)$'s are divided (that is, the corresponding "box" is occupied; the case of "1"), we associate "$\rangle\langle$" like

$$0123 \longrightarrow \langle 01\rangle\langle 23\rangle\,.$$

When there is no division (unocupied box; the case of "0"), a symbol " " is assigned:

$$0123 \longrightarrow \langle 0123 \rangle .$$

These characters are printed with the use of the A-transformation taking into account the rule (iii). This program is called "PC13".

### *3–2. Programs List**

DVCM

```
C**** COMBINATION OF DIVISION ****
C
         DIMENSION IB(11)
C
         WRITE(2,100)
100      FORMAT('ORDER(2-12)=@')
         READ(1,101) ID
101      FORMAT(I2)
         LD1=ID-1
C
         IN=1
         JD=ID/2*2
         IF(JD.NE.ID) GO TO 2
C
         WRITE(2,200)
200      FORMAT('FIRST ORDER AVERAGE IS ZERO OR NOT ZERO(0/1)@')
         READ(1,201) IN
201      FORMAT(I1)
C
2        JF=2**(ID-1)
         REWIND 9
         WRITE(9) JF,ID
C
         JC=0
         DO 30 J=1,JF
C
         J0=J-1
         DO 15 I=1,LD1
         L=LD1+1-I
         IB(L)=J0-J0/2*2
         J0=J0/2
         IF(J0.LT.1) GO TO 3
15       CONTINUE
C
3        IF(IN.NE.0) GO TO 4
         DO 25 I=1,LD1,2
         IF(IB(I).NE.0) GO TO 30
25       CONTINUE
C
4        WRITE(2,300) (IB(I),I=1,LD1)
300      FORMAT(11I3)
         WRITE(9) (IB(I),I=1,LD1)
         JC=JC+1
```

* The programs presented here can treat problems up to the 12-th order, because of a large number of terms.

```
30        CONTINUE
C
          REWIND 9
          WRITE(9) JC,ID
C
          STOP
          END


  PC13

          DIMENSION IA(12),IB(11)
     *      ,AA(12),AC(2),BC(2)
     *      ,INT(12)
          DATA INT/' 0',' 1',' 2',' 3',' 4',' 5',' 6',' 7'
     *      ,' 8',' 9','10','11'/
C
          AC(1)='  '
          AC(2)='><'
          BC(1)=' +'
          BC(2)=' -'
          CC='c='
C
          REWIND 9
          READ(9) MF,ID
          WRITE(2,200) ID,MF
200       FORMAT('ORDER=',I2,2X,'MF=',I5)
C
          LD1=ID-1
          DO 10 I=1,ID
          IA(I)=I-1
          AA(I)='  '
10        CONTINUE
          AA(ID)='>p'
C
          WRITE(6,400) ID
400       FORMAT(1H ,'**** PARTIAL CUMULANT **** ORDER=',I2/)
          WRITE(6,410) (INT(IA(I)+1),AA(I),I=1,ID),CC
410       FORMAT(1H ,'<',25A2/)
C
          REWIND 9
          READ(9) MF,LD
C
          DO 30 M=1,MF
          READ(9) (IB(L),L=1,LD1)
C
          IBT=0
          DO 15 L=1,LD1
          IBT=IBT+IB(L)
15        CONTINUE
          IBT0=IBT-IBT/2*2+1
C
          LC=IA(1)
          DO 20 I=2,ID
          IF(IA(I).LT.LC) GO TO 30
          LC=IA(I)
          IF(IB(I).EQ.1) LC=0
```

```
20        CONTINUE
          DO 25 L=1,LD1
          IL=IB(L)+1
          AA(L)=AC(IL)
25        CONTINUE
          AA(ID)='> '
C
          WRITE(6,300) BC(IBT0),(INT(IA(I)+1),AA(I),I=1,ID)
300       FORMAT(1H ,A2,'<',24A2)
C
30        CONTINUE
C
          STOP
          END
```

*3-3. Examples*

Now we present the results using the programs which are given in the previous subsection. In the case of $\langle L_1(t_j)\rangle_B \neq 0$, full expressions from the 2nd to 6th order are given. While the case of $\langle L_1(t_j)\rangle_B = 0$, expression up to 12th order are also given.

```
**** PARTIAL CUMULANT **** ORDER= 2
< 0   1>pc=
 +< 0   1>
 -< 0>< 1>


**** PARTIAL CUMULANT **** ORDER= 3
< 0   1   2>pc=
 +< 0   1   2>
 -< 0   1>< 2>
 -< 0>< 1   2>
 +< 0>< 1>< 2>


**** PARTIAL CUMULANT **** ORDER= 4
< 0   1   2   3>pc=
 +< 0   1   2   3>
 -< 0   1   2>< 3>
 -< 0   1>< 2   3>
 +< 0   1>< 2>< 3>
 -< 0>< 1   2   3>
 +< 0>< 1   2>< 3>
 +< 0>< 1>< 2   3>
 -< 0>< 1>< 2>< 3>
```

```
**** PARTIAL CUMULANT **** ORDER= 5
< Ø   1   2   3   4>pc=
 +< Ø   1   2   3   4>
 -< Ø   1   2   3>< 4>
 -< Ø   1   2>< 3   4>
 +< Ø   1   2>< 3>< 4>
 -< Ø   1>< 2   3   4>
 +< Ø   1>< 2   3>< 4>
 +< Ø   1>< 2>< 3   4>
 -< Ø   1>< 2>< 3>< 4>
 -< Ø>< 1   2   3   4>
 +< Ø>< 1   2   3>< 4>
 +< Ø>< 1   2>< 3   4>
 -< Ø>< 1   2>< 3>< 4>
 +< Ø>< 1>< 2   3   4>
 -< Ø>< 1>< 2   3>< 4>
 -< Ø>< 1>< 2>< 3   4>
 +< Ø>< 1>< 2>< 3>< 4>
```

```
**** PARTIAL CUMULANT **** ORDER= 6
< Ø   1   2   3   4   5>pc=
 +< Ø   1   2   3   4   5>
 -< Ø   1   2   3   4>< 5>
 -< Ø   1   2   3>< 4   5>
 +< Ø   1   2   3>< 4>< 5>
 -< Ø   1   2>< 3   4   5>
 +< Ø   1   2>< 3   4>< 5>
 +< Ø   1   2>< 3>< 4   5>
 -< Ø   1   2>< 3>< 4>< 5>
 -< Ø   1>< 2   3   4   5>
 +< Ø   1>< 2   3   4>< 5>
 +< Ø   1>< 2   3>< 4   5>
 -< Ø   1>< 2   3>< 4>< 5>
 +< Ø   1>< 2>< 3   4   5>
 -< Ø   1>< 2>< 3   4>< 5>
 -< Ø   1>< 2>< 3>< 4   5>
 +< Ø   1>< 2>< 3>< 4>< 5>
 -< Ø>< 1   2   3   4   5>
 +< Ø>< 1   2   3   4>< 5>
 +< Ø>< 1   2   3>< 4   5>
 -< Ø>< 1   2   3>< 4>< 5>
 +< Ø>< 1   2>< 3   4   5>
 -< Ø>< 1   2>< 3   4>< 5>
 -< Ø>< 1   2>< 3>< 4   5>
 +< Ø>< 1   2>< 3>< 4>< 5>
 +< Ø>< 1>< 2   3   4   5>
 -< Ø>< 1>< 2   3   4>< 5>
 -< Ø>< 1>< 2   3>< 4   5>
 +< Ø>< 1>< 2   3>< 4>< 5>
 -< Ø>< 1>< 2>< 3   4   5>
 +< Ø>< 1>< 2>< 3   4>< 5>
 +< Ø>< 1>< 2>< 3>< 4   5>
 -< Ø>< 1>< 2>< 3>< 4>< 5>
```

```
**** PARTIAL CUMULANT **** ORDER= 2
< 0   1>pc=
 +< 0   1>


**** PARTIAL CUMULANT **** ORDER= 4
< 0   1   2   3>pc=
 +< 0   1   2   3>
 -< 0   1>< 2   3>


**** PARTIAL CUMULANT **** ORDER= 6
< 0   1   2   3   4   5>pc=
 +< 0   1   2   3   4   5>
 -< 0   1   2   3>< 4   5>
 -< 0   1>< 2   3   4   5>
 +< 0   1>< 2   3>< 4   5>


**** PARTIAL CUMULANT **** ORDER= 8
< 0   1   2   3   4   5   6   7>pc=
 +< 0   1   2   3   4   5   6   7>
 -< 0   1   2   3   4   5>< 6   7>
 -< 0   1   2   3>< 4   5   6   7>
 +< 0   1   2   3>< 4   5>< 6   7>
 -< 0   1>< 2   3   4   5   6   7>
 +< 0   1>< 2   3   4   5>< 6   7>
 +< 0   1>< 2   3>< 4   5   6   7>
 -< 0   1>< 2   3>< 4   5>< 6   7>


**** PARTIAL CUMULANT **** ORDER=10
< 0   1   2   3   4   5   6   7   8   9>pc=
 +< 0   1   2   3   4   5   6   7   8   9>
 -< 0   1   2   3   4   5   6   7>< 8   9>
 -< 0   1   2   3   4   5>< 6   7   8   9>
 +< 0   1   2   3   4   5>< 6   7>< 8   9>
 -< 0   1   2   3>< 4   5   6   7   8   9>
 +< 0   1   2   3>< 4   5   6   7>< 8   9>
 +< 0   1   2   3>< 4   5>< 6   7   8   9>
 -< 0   1   2   3>< 4   5>< 6   7>< 8   9>
 -< 0   1>< 2   3   4   5   6   7   8   9>
 +< 0   1>< 2   3   4   5   6   7>< 8   9>
 +< 0   1>< 2   3   4   5>< 6   7   8   9>
 -< 0   1>< 2   3   4   5>< 6   7>< 8   9>
 +< 0   1>< 2   3>< 4   5   6   7   8   9>
 -< 0   1>< 2   3>< 4   5   6   7>< 8   9>
 -< 0   1>< 2   3>< 4   5>< 6   7   8   9>
 +< 0   1>< 2   3>< 4   5>< 6   7>< 8   9>
```

```
**** PARTIAL CUMULANT **** ORDER=12
< Ø   1   2   3   4   5   6   7   8   9  1Ø  11>pc=
 +< Ø   1   2   3   4   5   6   7   8   9  1Ø  11>
 -< Ø   1   2   3   4   5   6   7   8   9><1Ø  11>
 -< Ø   1   2   3   4   5   6   7><8   9  1Ø  11>
 +< Ø   1   2   3   4   5   6   7><8   9><1Ø  11>
 -< Ø   1   2   3   4   5><6   7   8   9  1Ø  11>
 +< Ø   1   2   3   4   5><6   7   8   9><1Ø  11>
 +< Ø   1   2   3   4   5><6   7><8   9  1Ø  11>
 -< Ø   1   2   3   4   5><6   7><8   9><1Ø  11>
 -< Ø   1   2   3><4   5   6   7   8   9  1Ø  11>
 +< Ø   1   2   3><4   5   6   7   8   9><1Ø  11>
 +< Ø   1   2   3><4   5   6   7><8   9  1Ø  11>
 -< Ø   1   2   3><4   5   6   7><8   9><1Ø  11>
 +< Ø   1   2   3><4   5><6   7   8   9  1Ø  11>
 -< Ø   1   2   3><4   5><6   7   8   9><1Ø  11>
 -< Ø   1   2   3><4   5><6   7><8   9  1Ø  11>
 +< Ø   1   2   3><4   5><6   7><8   9><1Ø  11>
 -< Ø   1><2   3   4   5   6   7   8   9  1Ø  11>
 +< Ø   1><2   3   4   5   6   7   8   9><1Ø  11>
 +< Ø   1><2   3   4   5   6   7><8   9  1Ø  11>
 -< Ø   1><2   3   4   5   6   7><8   9><1Ø  11>
 +< Ø   1><2   3   4   5><6   7   8   9  1Ø  11>
 -< Ø   1><2   3   4   5><6   7   8   9><1Ø  11>
 -< Ø   1><2   3   4   5><6   7><8   9  1Ø  11>
 +< Ø   1><2   3   4   5><6   7><8   9><1Ø  11>
 +< Ø   1><2   3><4   5   6   7   8   9  1Ø  11>
 -< Ø   1><2   3><4   5   6   7   8   9><1Ø  11>
 -< Ø   1><2   3><4   5   6   7><8   9  1Ø  11>
 +< Ø   1><2   3><4   5   6   7><8   9><1Ø  11>
 -< Ø   1><2   3><4   5><6   7   8   9  1Ø  11>
 +< Ø   1><2   3><4   5><6   7   8   9><1Ø  11>
 +< Ø   1><2   3><4   5><6   7><8   9  1Ø  11>
 -< Ø   1><2   3><4   5><6   7><8   9><1Ø  11>
```

## § 4. Ordered Cumulant.

The ordered cumulants are written according to the following rules:

( i ) Make all possible permutations composed of $L_1(t), L_1(t_1), \cdots, L_1(t_{n-1})$; $L_1(t)$ is always in the first position of the terms.

(ii) As in the case of the partial cumulant, one makes all possible combinations of divisions among $L_1(t_j)$'s. A product of $L_1(t_j)$'s enclosed by two divisions is considered to be an average. In this average, $L_1(t_j)$'s are time ordered from the right.

(iii) Attach $+(-)$ sign to the term when the number of the divisions are even (odd).

These rules are transcribed into the following computer handling.

*4-1. Treatment in the computer*

In contrast with the partial cumulant, there are several ways of orderings of $\{L_1(t_j)\}$ for the ordered cumulant. But in a certain average, $\langle\cdots\cdot\rangle$, the time sequence is ordered from the right.

To manage the rule (i), we first make all the numbers of $n$ figures with $n$ scale beginning with 0 ($\equiv t$). In these numbers, we eliminate the ones which have plural same digits. This program is called "OC00".

Corresponding to rule (ii), we make a series of division using DVCM. In the averages thus derived, we next exclude the ones which violate the time ordering rule.

Taking into account the rule (iii), results are printed out similarly as in the previous section. We call this program "OC13".

### *4-2. Programs list*

```
OC00

      DIMENSION IA(12)
C
      WRITE(2,100)
100   FORMAT('ORDER(2-12)=@')
      READ(1,101) ID
101   FORMAT(I2)
C
      N=ID
      JF=N**(N-1)
      JC=0
      IC=(ID-1)*ID/2
C
      REWIND 5
      WRITE(5) JF,ID
C
      DO 30 J=1,JF
      DO 10 I=2,ID
      IA(I)=0
10    CONTINUE
C
      J0=J-1
      DO 15 I=1,ID
      L=ID+1-I
      IA(L)=J0-J0/N*N
      J0=J0/N
      IF(J0.LT.1) GO TO 2
15    CONTINUE
2     IT=0
      DO 20 I=1,ID
      IT=IT+IA(I)
20    CONTINUE
      IF(IT.NE.IC) GO TO 30
C
      DO 25 I1=2,ID
      DO 26 I2=2,ID
      IF(I1.EQ.I2) GO TO 26
      IF(IA(I1).EQ.IA(I2)) GO TO 30
```

```
26        CONTINUE
25        CONTINUE
C
          WRITE(2,300) (IA(I),I=1,ID)
300       FORMAT(12I3)
          JC=JC+1
          WRITE(5) (IA(I),I=1,ID)
30        CONTINUE
C
          REWIND 5
          WRITE(5) JC,ID
C
          STOP
          END
```

OC13

```
          DIMENSION IA(12),IB(11)
     *      ,AA(12),AC(2),BC(2)
     *      ,INT(12)
          DATA INT/' 0',' 1',' 2',' 3',' 4',' 5',' 6',' 7'
     *      ,' 8',' 9','10','11'/
C
          AC(1)='  '
          AC(2)='><'
          BC(1)=' +'
          BC(2)=' -'
          CC='c='
C
          REWIND 5
          READ(5) JF,ID
          WRITE(2,100) ID,JF
100       FORMAT('ORDER=',I2,2X,'JF=',I5)
C
          REWIND 9
          READ(9) MF,LD
          WRITE(2,200) LD,MF
200       FORMAT('ORDER=',I2,2X,'MF=',I5)
          IF(LD.NE.ID) STOP
C
          LD1=ID-1
          DO 10 I=1,ID
          IA(I)=I-1
          AA(I)='  '
10        CONTINUE
          AA(ID)='>o'
C
          WRITE(6,400) ID
400       FORMAT(1H ,'**** ORDERED CUMULANT **** ORDER=',I2/)
          WRITE(6,410) (INT(IA(I)+1),AA(I),I=1,ID),CC
410       FORMAT(1H ,'<',25A2/)
C
          DO 40 J=1,JF
          READ(5) (IA(I),I=1,ID)
C
```

```
            REWIND 9
            READ(9) MF,LD
C
            DO 30 M=1,MF
            READ(9) (IB(L),L=1,LD1)
C
            IBT=0
            DO 15 L=1,LD1
            IBT=IBT+IB(L)
15          CONTINUE
            IBT0=IBT-IBT/2*2+1
C
            LC=IA(1)
            DO 20 I=2,ID
            IF(IA(I).LT.LC) GO TO 30
            LC=IA(I)
            IF(IB(I).EQ.1) LC=0
20          CONTINUE
            DO 25 L=1,LD1
            IL=IB(L)+1
            AA(L)=AC(IL)
25          CONTINUE
            AA(ID)='> '
C
            WRITE(6,300) BC(IBT0),(INT(IA(I)+1),AA(I),I=1,ID)
300         FORMAT(1H ,A2,'<',24A2)
C
30          CONTINUE
40          CONTINUE
C
            STOP
            END
```

*4-3. Examples*

We present the results with the use of the programs which are given in the previous subsection. In the case of $\langle L_1(t_j)\rangle \neq 0$, the expressions is from 2nd to 4th are given. While, in the case of $\langle L_1(t_j)\rangle = 0$, they are given up to 6th order.

```
**** ORDERED CUMULANT **** ORDER= 2
< 0   1>oc=
 +< 0   1>
 -< 0>< 1>


**** ORDERED CUMULANT **** ORDER= 3
< 0   1   2>oc=
 +< 0   1   2>
 -< 0   1>< 2>
 -< 0>< 1   2>
 +< 0>< 1>< 2>
 -< 0   2>< 1>
 +< 0>< 2>< 1>
```

```
**** ORDERED CUMULANT **** ORDER= 4
< 0    1    2    3>oc=
 +< 0    1    2    3>
 -< 0    1    2>< 3>
 -< 0    1>< 2    3>
 +< 0    1>< 2>< 3>
 -< 0>< 1    2    3>
 +< 0>< 1    2>< 3>
 +< 0>< 1>< 2    3>
 -< 0>< 1>< 2>< 3>
 -< 0    1    3>< 2>
 +< 0    1>< 3>< 2>
 +< 0>< 1    3>< 2>
 -< 0>< 1>< 3>< 2>
 -< 0    2>< 1    3>
 +< 0    2>< 1>< 3>
 +< 0>< 2>< 1    3>
 -< 0>< 2>< 1>< 3>
 -< 0    2    3>< 1>
 +< 0    2>< 3>< 1>
 +< 0>< 2    3>< 1>
 -< 0>< 2>< 3>< 1>
 -< 0    3>< 1    2>
 +< 0    3>< 1>< 2>
 +< 0>< 3>< 1    2>
 -< 0>< 3>< 1>< 2>
 +< 0    3>< 2>< 1>
 -< 0>< 3>< 2>< 1>


**** ORDERED CUMULANT **** ORDER= 2
< 0    1>oc=
 +< 0    1>


**** ORDERED CUMULANT **** ORDER= 4
< 0    1    2    3>oc=
 +< 0    1    2    3>
 -< 0    1>< 2    3>
 -< 0    2>< 1    3>
 -< 0    3>< 1    2>
```

```
**** ORDERED CUMULANT **** ORDER= 6
< Ø    1    2    3    4    5>oc=
 +< Ø    1    2    3    4    5>
 -< Ø    1    2    3>< 4    5>
 -< Ø    1>< 2    3    4    5>
 +< Ø    1>< 2    3>< 4    5>
 -< Ø    1    2    4>< 3    5>
 +< Ø    1>< 2    4>< 3    5>
 -< Ø    1    2    5>< 3    4>
 +< Ø    1>< 2    5>< 3    4>
 -< Ø    1    3    4>< 2    5>
 +< Ø    1>< 3    4>< 2    5>
 -< Ø    1    3    5>< 2    4>
 +< Ø    1>< 3    5>< 2    4>
 -< Ø    1    4    5>< 2    3>
 +< Ø    1>< 4    5>< 2    3>
 -< Ø    2>< 1    3    4    5>
 +< Ø    2>< 1    3>< 4    5>
 +< Ø    2>< 1    4>< 3    5>
 +< Ø    2>< 1    5>< 3    4>
 -< Ø    2    3    4>< 1    5>
 +< Ø    2>< 3    4>< 1    5>
 -< Ø    2    3    5>< 1    4>
 +< Ø    2>< 3    5>< 1    4>
 -< Ø    2    4    5>< 1    3>
 +< Ø    2>< 4    5>< 1    3>
 -< Ø    3>< 1    2    4    5>
 +< Ø    3>< 1    2>< 4    5>
 +< Ø    3>< 1    4>< 2    5>
 +< Ø    3>< 1    5>< 2    4>
 +< Ø    3>< 2    4>< 1    5>
 +< Ø    3>< 2    5>< 1    4>
 -< Ø    3    4    5>< 1    2>
 +< Ø    3>< 4    5>< 1    2>
 -< Ø    4>< 1    2    3    5>
 +< Ø    4>< 1    2>< 3    5>
 +< Ø    4>< 1    3>< 2    5>
 +< Ø    4>< 1    5>< 2    3>
 +< Ø    4>< 2    3>< 1    5>
 +< Ø    4>< 2    5>< 1    3>
 +< Ø    4>< 3    5>< 1    2>
 -< Ø    5>< 1    2    3    4>
 +< Ø    5>< 1    2>< 3    4>
 +< Ø    5>< 1    3>< 2    4>
 +< Ø    5>< 1    4>< 2    3>
 +< Ø    5>< 2    3>< 1    4>
 +< Ø    5>< 2    4>< 1    3>
 +< Ø    5>< 3    4>< 1    2>
```

## § 5. Conclusion.

In the present paper, we have given a method of printing out the partial and the ordered cumulants which play important roles in the theory of nonequilibrium statistical mechanics.

These cumulants can be calculated in principle, if we were patient enough. However, in higher orders it becomes exceedingly difficult to write down explicit expressions. Thus use of computer is inevitable.

We introduce a systematic division method using "1" and "0" for a given time sequence of $L_1(t), L_1(t_1), \cdots, L_1(t_{n-1})$. Use of binary-digit and/or $n$-digit was the key step of the method. Thanks to this method, we have been able to handle the higher order cumulants without using the multi-Do-loops in the program.

We can apply these procedures to actual problems. Indeed, we used the method to solve a difficult problem of the low field resonance[2]. This problem can never be solved exactly if the present method was not used.

Thus, for actual problems, our method can cope with computational difficulties inherent to higher order terms which sometimes give essential effects in certain physical systems. Our method in this paper will be useful in solving these problems.

## Acknowledgement

The author is indebted to Professor F. Shibata for valuable discussions and to Professor N. Hashitsume for encouragement throughout the work.

## References

[1] F. Shibata and T. Arimitsu: J. Phys. Soc. Jpn., **49** (1980), 891.
[2] F. Shibata and I. Sato: Physica A, submitted.
[3] I. Sato and F. Shibata: Physica **135A** (1986) 139, 388.
[4] F. Shibata and T. Arimitsu: J. Phys. Soc. Jpn., **51** (1982), 1070, 2075.
[5] Y. Hamano and F. Shibata: J. Phys. Soc. Jpn., **51** (1982), 1727, 2085, 2721, 2728.