J.Pharm.Dyn., **4**, 879–885 (1981)

A PHARMACOKINETIC ANALYSIS PROGRAM (MULTI) FOR MICROCOMPUTER

KIYOSHI YAMAOKA , YUSUKE TANIGAWARA, TERUMICHI NAKAGAWA AND TOYOZO UNO

Faculty of Pharmaceutical Sciences, Kyoto University, Sakyo-ku, Kyoto 606, Japan

(Received May 29, 1981)

A nonlinear least squares program (MULTI) for microcomputers was developed. The program is written in BASIC programming language. Four algorithms, (1) Gauss-Newton method, (2) damping Gauss-Newton method, (3) modified Marquardt method and (4) simplex method, can be used for nonlinear curve fitting in MULTI. Up to five pharmacokinetic equations, which are voluntarily defined by the user, are simultaneously fitted to observed time courses. The executions of MULTI are demonstrated for time courses of ampicillin and oxacillin in man.

Keywords —pharmacokinetics; MULTI; SALS; AIC; curve fitting; microcomputer; least squares

INTRODUCTION

Recent development of microcomputers has attracted many engineers' and researchers' attention. Personal computers on which BASIC language can be used, are rapidly permeating not only into homes but also into research laboratories. In the field of pharmacokinetics, a nonlinear curve fitting is becoming indispensable to extract information from time course data of drugs. Considering this requirement, a nonlinear least squares program (MULTI) was developed for personal computers.

A. Hardware

MULTI is written in BASIC programming language. A PET-2001 personal computer (Commodore) with an LP-80 printer (Super Brain) was used to develop MULTI. MULTI can be executed on CBM series computers without any modification. Slight changes in the program are required for other personal computers, especially with respect to the commands related to printer. MULTI occupies about 5K bytes for its own memory area which, however, can be reduced to about 4K bytes if the remark lines are omitted.

B. Nonlinear Least Squares Algorithms

Four nonlinear least squares algorithms can be applied to pharmacokinetic calculations in MULTI.

(1) Gauss-Newton method
(2) damping Gauss-Newton method[1)]
(3) modified Marquardt method[2,3)]
(4) simplex method[4)]

Method(1) is the classical algorithm, from which numerous modifications have been derived. Methods(2) and (3) which are also the modified Gauss-Newton methods have been adopted in SALS program by Nakagawa *et al.* [1)] for large computers. Personal computers, however, are too slow in calculation and too limited in memory to directly translate SALS into MULTI. Hence, the flexibility in SALS is considerably restricted in MULTI. The simplex method, which was introduced by Nelder and Mead,[4)] is widely accepted in chemical engineering[5)] because of its robustness against the divergence in computation. The detailed explanation of the simplex method is found in the book of Kowalik and Osborne.[6)]

C. Residual Sum of Squares and Akaike's Information Criterion

Residual sum of squares, SS, is defined by Eq. 1 for simultaneous multi-lines fittings.

$$SS = \sum_i \sum_j W_{ij}(C_{i,j} - f_i(t_j, P))^2 \qquad (1)$$

```
10 PRINT"* MULTI-LINES FITTINGS MARCH 29,1981 *"
20 PRINT:PRINT"DEFINE EQUATIONS AT 1000, 1100, 1200,   1300, 1400.":PRINT
30 PRINT"CP AND T ARE DEPENDENT AND INDEPENDENT  VARIABLES, RESPECTIVELY."
35 PRINT:PRINT"P(1), P(2),... ARE PARAMETERS TO FIT."
36 PRINT:PRINTTAB(8);"GOTO840 IF DIVERGED.":FP=0
37 DIMME$(3):ME$(0)="GAUSS-NEWTON":ME$(1)="DAMPING GAUSS-NEWTON"
38 ME$(2)="MARQUARDT":ME$(3)="SIMPLEX"
40 PRINT:FORI=0TO3:PRINT"(";I;") ";ME$(I);" METHOD":NEXT
45 PRINT:INPUT"WHICH ALGORITHM DO YOU SELECT";AL
49 PRINT:PRINT"* I BELIEVE YOU HAVE DEFINED EQUATIONS *"
50 INPUT"SUBJECT NAME";N$:INPUT"PRINT(Y/N)";P$
54 INPUT"NUMBER OF LINES";LN:DIMNL(LN)
55 INPUT"WEIGHT OF DATA(0,1,2)";IW:INPUT"NUMBER OF PARAMETERS";M
56 FORI=1TOLN:PRINT"NUMBER OF POINTS(";I;")";:INPUTNL(I):NEXT
60 N=0:FORI=1TOLN:N=N+NL(I):NEXT:DIMTX(N),CY(N),A(M,M+1),P(M),X(M,M)
65 NL(0)=0:BS=0:FORJ=1TOLN:BS=BS+NL(J-1):PRINT
70 FORI=1TONL(J):PRINT"T";J;"(";I;"), CP";J;"(";I;")";:INPUTTX(BS+I),CY(BS+I)
75 NEXTI,J:IFAL=3THEN3000
77 PC=.0001:CF=100:IFFP=0THENDIMCS(N,M):FP=1
78 PRINT:INPUT"DT FOR JACOBIAN (0.1-0.0001)";DT:PRINT:FORI=1TOM
80 PRINT"INITIAL P(";I;")=";:INPUTA(I,0):P(I)=A(I,0):NEXT:GOSUB4000:S1=SS
140 PRINT"INITIAL SS=";SS:FORK=1TO100:GOSUB7000:GOSUB7400:GOSUB6000:JJ=0
490 JJ=JJ+1:IFJJ>25THEN730
500 FORI=1TOM:P(I)=A(I,0)+A(I,M+1):NEXT:GOSUB4000
510 DS=ABS(S1-SS):IFAL<>2ORSS=0THEN590
515 REM FLETCHER MODIFICATION
520 PW=0:FORI=1TOM:PW=PW+X(I,0)*A(I,M+1)+CF*A(I,M+1)*A(I,M+1):NEXT
530 IFDS/PW>.75THENCF=CF/2
540 IFDS/PW<.25THENCF=5*CF
590 IFDS<=PC*S1THEN730:REM CHECK OF CONVERGENCE
595 REM DAMPING
600 IFAL=1ANDSS>S1THENFORI=1TOM:A(I,M+1)=.5*A(I,M+1):NEXT:GOTO490
630 FORI=1TOM:A(I,0)=P(I):NEXT:S1=SS:PRINT:PRINT"LOOP=";K
640 IFAL=1THENPRINT"DAMP=";JJ
660 FORI=1TOM:PRINT"P(";I;")=";P(I):NEXT:PRINT"SS=";SS:NEXT
730 IFP$="Y"THENOPEN4,4:CMD4:REM CHANGE TO PRINTER
731 PRINT:PRINT"*";N$;"*  BY ";ME$(AL);" METHOD":PRINT"WEIGHT=1/CP^(";IW;")"
733 IFAL<>3ANDN>MTHENGOSUB8000
734 IFSS=0THENPRINT"AIC=-INFINITE":GOTO740
735 PRINT"AIC=";N*LOG(SS)+2*M
740 IFAL=3THENPRINT"ALPHA=";AA;"  BETA=";BB;"  GAMMA=";CC:PRINT
742 IFAL<>3THENPRINT"DT=";DT
745 IFAL=2THENPRINT"FACTOR=";CF
750 FORI=1TOM:PRINT"FINAL P";I;"=";P(I);
760 IFAL<>3ANDX(I,0)>0ANDN>MTHENPRINT"   S.D.=";SQR(X(I,0)*SS/(N-M));
810 PRINT:NEXT:PRINT"FINAL SS=";SS:BS=0:FORJ=1TOLN:BS=BS+NL(J-1)
820 PRINT:FORI=1TONL(J):T=TX(BS+I):ONJGOSUB1000,1100,1200,1300,1400
830 PRINT"T";J;"=";T;"    CP";J;"=";CP;"  (";CY(BS+I);")":NEXTI,J
835 IFP$="Y"THENPRINT#4:CLOSE4:REM CHANGE TO CRT
840 PRINT:PRINT"WHICH ALGORITHM DO YOU SELECT"
850 INPUT"(0,1,2,3 OR -1)";AL:IFAL<0THENEND
860 IFAL=3THEN3000
870 GOTO77:REM************************
1000 DEFINE CP=F1(T,P(I)):RETURN
1100 DEFINE CP=F2(T,P(I)):RETURN
1200 DEFINE CP=F3(T,P(I)):RETURN
1300 DEFINE CP=F4(T,P(I)):RETURN
1400 DEFINE CP=F5(T,P(I)):RETURN
1999 REM JACOBIAN ********************
2000 FORJS=1TOM:PT=P(JS):P(JS)=PT+DT:ONJGOSUB1000,1100,1200,1300,1400
2020 DD=CP:P(JS)=PT-DT:ONJGOSUB1000,1100,1200,1300,1400
2030 CS(BS+I,JS)=(DD-CP)/(2*DT):P(JS)=PT:NEXT:RETURN
2999 REM SIMPLEX METHOD
```

```
 3000 AA=1:BB=.5:CC=2:SG=1E10:PC=.00001
 3025 PRINT:FORI=1TOM:PRINT"INITIAL P(";I;")";:INPUTA(I,1):NEXT
 3030 FORJ=2TOM+1:FORI=1TOM:A(I,J)=2*RND(1)*A(I,1)+.01*(RND(1)-.5):NEXTI,J
 3040 FORK=1TOM+1:FORI=1TOM:P(I)=A(I,K):NEXT:GOSUB4000:A(0,K)=SS:NEXT
 3070 PRINT:FORI=1TOM+1:PRINT"SS";I;"=";A(0,I):NEXT:GOTO5000
 3080 SR=0:SL=1E10:FORJ=1TOM+1:IFSR<A(0,J)THENJH=J:SR=A(0,J)
 3090 IFSL>A(0,J)THENJL=J:SL=A(0,J)
 3100 NEXT:SR=0:FORJ=1TOM+1:IFJ<>JHANDSR<A(0,J)THENJS=J:SR=A(0,J)
 3110 NEXT:FORI=1TOM:X(0,I)=0:FORJ=1TOM+1:IFJ<>JHTHENX(0,I)=X(0,I)+A(I,J)
 3120 NEXT:X(0,I)=X(0,I)/M:NEXT:FORI=1TOM:A(I,0)=(1+AA)*X(0,I)-AA*A(I,JH)
 3130 P(I)=A(I,0):NEXT:GOSUB4000:SR=SS:IFSR<=A(0,JS)THEN3300
 3160 IFSR<A(0,JH)THENFORI=1TOM:A(I,JH)=A(I,0):NEXT:A(0,JH)=SR
 3170 FORI=1TOM:A(I,0)=BB*A(I,JH)+(1-BB)*X(0,I)
 3180 P(I)=A(I,0):NEXT:GOSUB4000:SR=SS
 3190 IFSR<A(0,JH)THENFORI=1TOM:A(I,JH)=A(I,0):NEXT:A(0,JH)=SR:GOTO3070
 3200 FORK=1TOM+1:FORI=1TOM:A(I,K)=(A(I,K)+A(I,JL))/2:P(I)=A(I,K):NEXT
 3210 GOSUB4000:A(0,K)=SS:NEXT:GOTO3070
 3300 IFSR<A(0,JL)THEN3500
 3320 FORI=1TOM:A(I,JH)=A(I,0):NEXT:A(0,JH)=SR:GOTO3070
 3500 FORI=1TOM:X(1,I)=CC*A(I,0)+(1-CC)*X(0,I):P(I)=X(1,I):NEXT:GOSUB4000:SL=SS
 3510 IFSL<SRTHENFORI=1TOM:A(I,JH)=X(1,I):NEXT:A(0,JH)=SL:GOTO3070
 3520 GOTO3320
 3999 REM CALCULATION OF SS
 4000 SS=0:BS=0:FORJ=1TOLN:BS=BS+NL(J-1):FORI=1TONL(J):T=TX(BS+I)
 4020 ONJGOSUB1000,1100,1200,1300,1400:SS=SS+(CY(BS+I)-CP)^2/CY(BS+I)^IW
 4030 NEXTI,J:RETURN
 4999 REM CHECK OF CONVERGENCE
 5000 SR=0:FORI=1TOM+1:SR=SR+A(0,I):NEXT
 5030 IFABS(SR-SG)>PC*SGTHENSG=SR:GOTO3080
 5040 FORI=1TOM:P(I)=A(I,JL):NEXT:SS=A(0,JL):GOTO730
 5999 REM GAUSS ELIMINATION WITH PIVOT
 6000 IFNP=1THENA(1,2)=A(1,2)/A(1,1):RETURN
 6020 RM=ABS(A(1,1)):FORIS=1TONP:FORJS=1TONP
 6050 IFRM<ABS(A(JS,IS))THENRM=ABS(A(JS,IS))
 6060 NEXTJS,IS:FORKS=1TONP-1:W=0:FORIS=KSTONP
 6100 IFABS(A(IS,KS))<WTHEN6130
 6110 W=ABS(A(IS,KS)):JS=IS
 6130 NEXT:IFJS=KSTHEN6200
 6150 FORIS=KSTONP+1:W=A(KS,IS):A(KS,IS)=A(JS,IS):A(JS,IS)=W:NEXT
 6200 P=1/A(KS,KS):FORJS=KS+1TONP+1:A(KS,JS)=A(KS,JS)*P:W=-A(KS,JS)
 6250 IFW=0THEN6290
 6260 FORIS=KS+1TONP:A(IS,JS)=A(IS,JS)+A(IS,KS)*W:NEXT
 6290 NEXT:NEXT:A(NP,NP+1)=A(NP,NP+1)/A(NP,NP):FORIS=2TONP
 6350 LS=NP-IS+1:W=-A(LS,NP+1):FORJS=LS+1TONP
 6400 W=W+A(LS,JS)*A(JS,NP+1):NEXT:A(LS,NP+1)=-W:NEXT:RETURN
 6999 REM NORMAL EQUATION
 7000 NL(0)=0:BS=0:FORJ=1TOLN:BS=BS+NL(J-1):FORI=1TONL(J):T=TX(BS+I)
 7300 ONJGOSUB1000,1100,1200,1300,1400:CS(BS+I,0)=CY(BS+I)-CP:GOSUB2000
 7310 NEXTI,J:FORI=1TOM:FORJ=ITOM:A(I,J)=0:FORL=1TON
 7390 A(I,J)=A(I,J)+CS(L,I)*CS(L,J)/CY(L)^IW:NEXT:A(J,I)=A(I,J):NEXTJ,I:RETURN
 7400 FORI=1TOM:A(I,M+1)=0:FORJ=1TON
 7440 A(I,M+1)=A(I,M+1)+CS(J,I)*CS(J,0)/CY(J)^IW:NEXTJ,I:NP=M
 7450 REM MARQUARDT MODIFICATION
 7460 IFAL=2THENFORI=1TOM:A(I,I)=A(I,I)+CF:X(I,0)=A(I,M+1):NEXT
 7470 RETURN
 7999 REM VARIENCE
 8000 GOSUB7000:FORI=1TOM:FORJ=1TOM:X(I,J)=A(I,J):NEXTJ,I
 8010 FORK=1TOM:FORI=1TOM:FORJ=1TOM:A(I,J)=X(I,J):NEXTJ,I:FORI=1TOM
 8020 A(I,M+1)=0:NEXT:A(K,M+1)=1:GOSUB6000:X(K,0)=A(K,M+1):NEXT:RETURN
READY.
```

FIG. 1. *Program List of MULTI*

where $C_{i,j}$ is the value of the jth point on the ith experimental time course, $f_i(t_j,P)$ is the theoretical equation to fit, which is derived from a pharmacokinetic model, t_j is time at the jth point, $P$are pharmacokinetic parameters to be estimated and W_{ij} is the weight of data points. 1, $(C_{i,j})^{-1}$ or $(C_{i,j})^{-2}$ is often adopted as the weight in pharmacokinetics. The number of lines to simultaneously fit is limited to five in MULTI. The least squares operation means to estimate the pharmacokinetic parameters, P, that minimize SS.

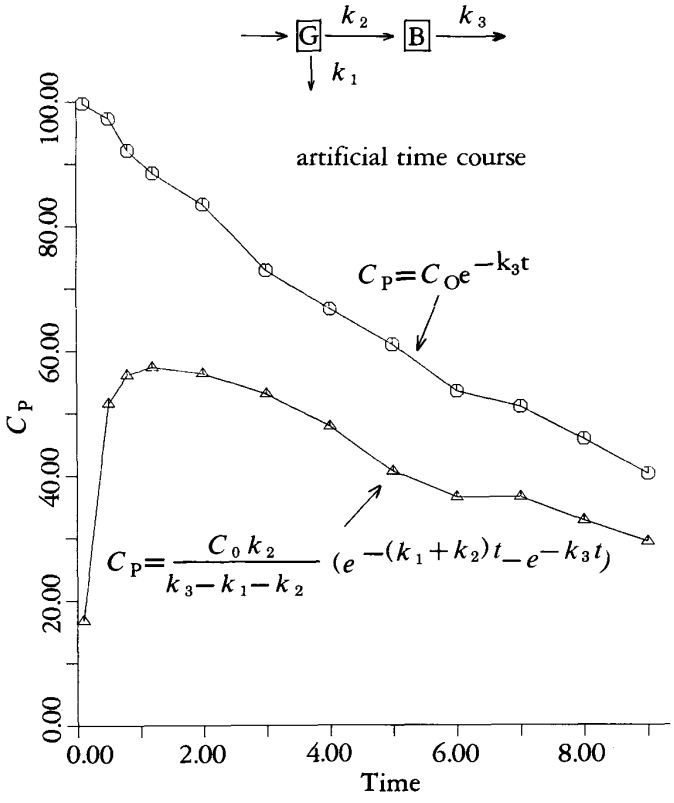

FIG.2. *Artificial Time Courses following Intravenous Dose(o) and Oral Dose (△)*

Akaike's information criterion, AIC[7,8] is defined by Eq. 2.

$$AIC = N \ln (SS) + 2M \quad (2)$$

where N is the number of observed data points and M is the number of parameters to estimate. The pharmacokinetic model that gives the minimum AIC value is supposed to be the best.

D. Input to MULTI

MULTI gives the detailed prompts for input data (*i.e.* selection of an algorithm and numbers of parameters, data points and curves to simultaneously fit, *etc.*) by making the most of the conversational features of BASIC language.

E. Output from MULTI

The following results are printed out after convergence.

(1) Estimated parameters converged and their standard deviations.

TABLE I. *Calculating Time, Memories for Calculation and Converged Parameters by Four Algorithms using Artificial Time Course Data*

Algorithm	Time(min)	Memory(K bytes)	Convergence
G-N	3.33	1.46	OK
Damping G-N	3.33	1.46	OK
Marquardt	5.00	1.47	OK
Simplex	6.53	0.88	OK

Parameter	Initial	Final	True
k_1	0.8	0.976	1.0
k_2	2.4	1.95	2.0
k_3	0.05	0.0997	0.1
C_0	80	100	100

(2) Minimum SS and AIC values
(3) Input time course data and predicted values from the model.

When the simplex method is adopted as the least squares algorithm, the standard deviations are not calculated because the principle of this algorithm is completely different from the Gauss-Newton methods.

F. Definition of Equations to fit

Fig.1 shows the program list of MULTI. Pharmacokinetic equations to fit must be defined before the execution of MULTI. The equations to simultaneously fit are defined from 1000 through 1400. An equation may be defined over several lines so far as it does not invade the line for the next equation. At that time, a RETURN command is needed at the end of each equation. For example, the two-compartment model equation,

$C_P = A e^{-at} + B e^{-bt}$, becomes

```
1000 CP=P(1)*EXP(-P(2)*T)
1010 CP=CP+P(3)*EXP(-P(4)*T):
RETURN
```

where P(1), P(2), ... are pharmacokinetic parameters to estimate, CP and T are dependent and independent variables, respectively.

G. Modification of MULTI

The variable, PC, which appears at 77 and 3000 is a criterion for convergence. When the relative change in SS becomes less than PC, MULTI terminates the iterative computation. The accuracy of estimated parameters may be improved with the decrease of PC to zero.

The subroutine 2000 numerically calculates the Jacobian matrix which is required in the Gauss-Newton methods. The variable, DT, is

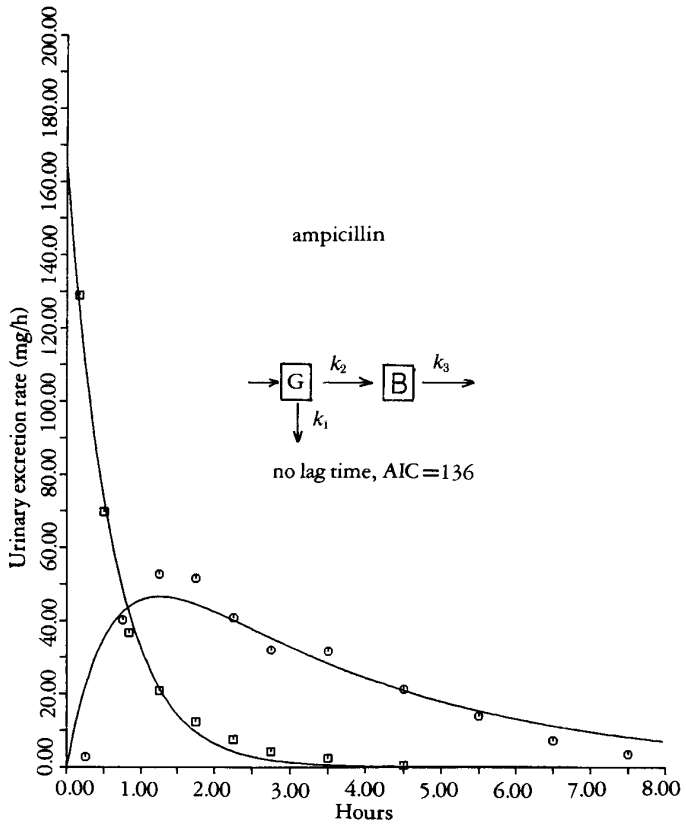

FIG.3. *Urinary Excretion Rate of Ampicillin following Oral and Intravenous Doses in Man Model includes no lag time.*

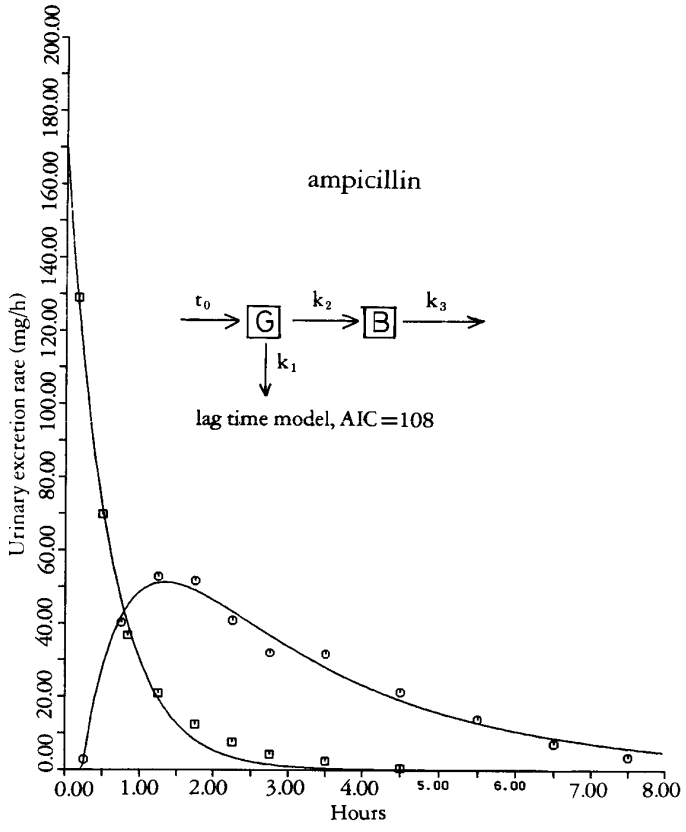

FIG.4. *The Same as in Fig. 3 except Model includes Lag Time Process*

used to give the perturbation to parameters as the absolute value. If the perturbation as the ratio to parameters is needed, DT from 2000 through 2030 must be replaced by PT*DT. The Jacobian matrix can be defined here in the analytical manner.

0.5 is used as the damping factor in the damping Gauss-Newton method (line 600). SALS also adopts this damping factor.

The lines from 730 through 835 which relate to the printer must be changed to corresponding commands peculiar to other personal computers.

H. Examples of Execution of MULTI

Fig. 2 shows two time courses which are artificially generated by giving random errors to theoretical equations from the model in the figure. The time course shown by open circles corresponds to intravenous dosing, and that by triangles denotes oral dosing. The defined equations in these cases are

```
1000 CP=P(4)*EXP(-P(3)*T) :RETURN
1100 CP=P(4)*P(2)/P(3)-(P(1)-P(2) )*
(EXP(-(P(1)+P(2))*T)
      -EXP(-P(3)*T)): RETURN
```

where $P(1)=k_1$, $P(2)=k_2$, $P(3)=k_3$ and $P(4)=C_0$.

Table I shows the results by MULTI. The data are weighted by unity in all calculations here. Pharmacokinetic parameters which were converged to the same values by all algorithms in about 5 minutes coincided with the results computed by SALS.

Fig. 3 shows the time courses of the urinary excretion rate of ampicillin in a human subject.[9] Open circles and rectangles indicate the time courses following an intravenous dose of 125 mg sodium ampicillin and following an oral dose of 500 mg ampicillin capsule, respectively. The lines mean the fitted curves by MULTI using the model in the figure. The initial points of the time course following oral dose, however, do not appear to fit well (AIC=136). Therefore the lag time was introduced into the model as shown in Fig. 4. Using this model, the AIC value is reduced to 108 that supports the existence of lag time in the absorption of ampicillin from the GI tract.

Fig. 5 shows the time courses of cumulative urinary excretion of oxacillin and its metabolites in a human subject following an oral dose of 500 mg oxacillin capsule. The details about analytical procedures and pharmacokinetic explanations are found elsewhere.[10] The lines are theoretical curves fitted by the simplex method. The other algorithms gave memory-over when 8K PET was used. The numbers on arrows in the chart are the first-order rate constants (h^{-1}) estimated by MULTI.

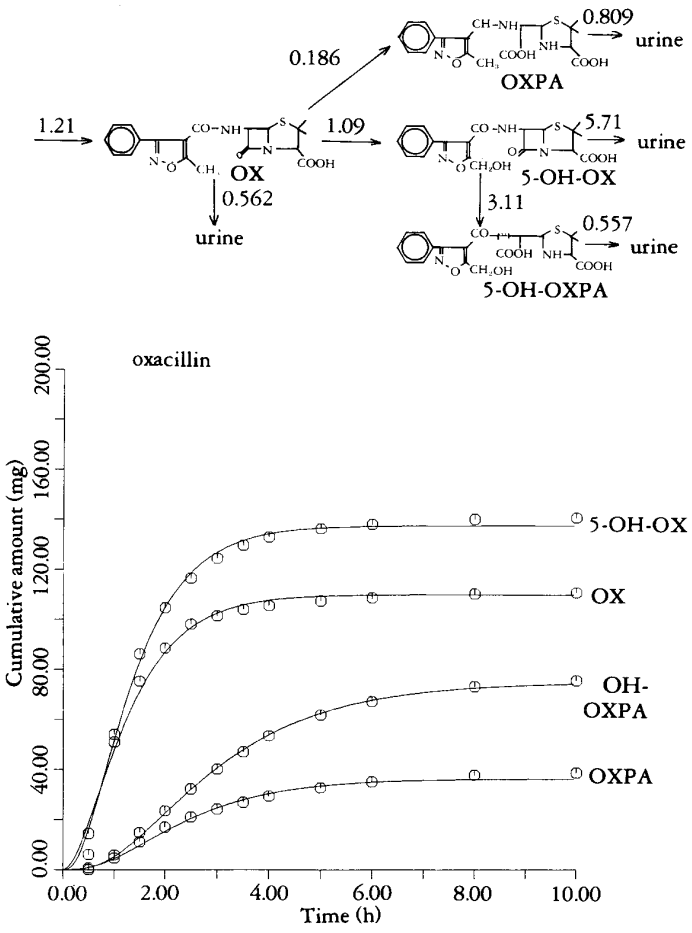

FIG. 5. *Cumulative Excretions of Oxacillin and Its Metabolites in Man following Oral Dose of Oxacillin*

Where OX is oxacillin, OXPA is penicilloic acid of oxacillin, 5-OH-OX is 5-hydroxymethyl derivative of oxacillin and 5-OH-OXPA is penicilloic acid of 5-hydroxymethyl derivative. The unit of rate constant on the arrow is h^{-1}.

I. Discussion

The simplex method shows a good convergency even when the initial values for parameters severely deviate from the final converged values. This algorithm, however, has a disadvantage that when SS meets an extreme value in the computing process, the simplex method tends to stop the calculation at that point. On the other hand, the Gauss-Newton methods show the most efficient convergence if the initial values are close to the final values. Hence, the cooperative use of these algorithms is expected to give correct results.

Although the minimum AIC estimation (MAICE) may suggest the selection of a pharmacokinetic model, this test has no validity to detect the existence of a flip-flop model. For example, the model for oxacillin metabolism in Fig. 5 has a flip-flop, which is reported elsewhere.[10]

REFERENCE

1) T.Nakagawa and Y.Oyanagi: SALS user manual, Feb. 1, 1981.
2) R.Flecher: A modified Marquardt subroutine for nonlinear least squares, Tech. Report AERE R6799, U.K.Atomic Energy Authority Research Establishment, Harwell, U.K., 1971.
3) D.W.Marquardt: An algorithm for least-squares estimation of nonlinear parameters, *J.Soc. Indus. Appl. Math.*, **11**, 431–441 (1963).
4) J.A.Nelder and R.Mead: A simplex method for function minimization, *Computer J.*, **7**, 308–313 (1965).
5) D.M.Himmelblau: Process analysis by statistical methods, 1970 by John Wiley & Sons Inc.
6) J.Kowalik and M.R.Osborne: Method for unconstrained optimization problem 1968, Americal Elsevier Publishing Company.
7) K.Yamaoka, T.Nakagawa and T.Uno: Application of Akaike's information criterion (AIC) in the evaluation of linear pharmacokinetic equations, *J.Pharmacok. Biopharm.*, **6**, 165–175 (1978).
8) H.Akaike: A new look at the statistical model identification, *IEEE Trans. Aut. Contr.*, **19**, 716–723 (1973).
9) Y.Tanigawara, K.Yamaoka, T.Nakagawa and T.Uno: *J.Pharm. Sci.*, under submission.
10) Y.Murai, K.Yamaoka, T.Nakagawa and T.Uno: *Chem.Pharm.Bull.*, **29**, in press.