Development of a Computer Program, MDGP, for Population Pharmacokinetic Analysis Written Using ANSI C Language on Wide Platforms

Koji TANIKAWA,^a Yoshiaki MATSUMOTO,^{*,b} Mitsuo MATSUMOTO,^c and Masamichi FUKUOKA^b

Department of Pharmacy, St. Marianna University School of Medicine, Yokohama-shi Seibu Hospital,^a 1197-1, Yasashi-cho, Asahi-ku, Yokohama 241-0811, Japan and Department of Clinical Pharmacology and Toxicology^b and Department of Pharmaceutics,^c Showa College of Pharmaceutical Sciences, 3-3165, Higashitamagawagakuen, Machida 194-8543, Japan. Received January 23, 1998; accepted May 15, 1998

MDGP, a computer program for population pharmacokinetics, has been developed. This program is based on maximum likelihood estimation with poly-dimensional normal distribution errors, and variable metric methods, direction set methods, conjugate gradient methods and the polytope method are provided as minimizations of objective function. The source codes of MDGP are described using ANSI C language. MDGP is able to run on wide platforms equipped with an ANSI C compiler. Users can use algebraic, ordinary differential and Laplace-transformed equations as descriptions of a pharmacokinetic model. Partial differential equations for each model parameter are not needed. Comparison of the estimated parameters, the iteration counts until convergence and objective function value, using MDGP and NONMEM, revealed that the calculation performance of the two programs are comparable. The MDGP is thought to be useful for the analysis of various pharmacokinetic models, including population pharmacokinetic analysis.

Key words population pharmacokinetic analysis; MDGP; NONMEM; C language; maximum likelihood estimation; ordinary differential equation

For optimizing drug regimens in individual patients, relevant knowledge of the pharmacokinetic behavior of medications is important. The information required from pharmacokinetic models and parameters is as follows: decision of initial drug regimens, prediction of individual pharmacokinetics using Bayes estimation and optimization of maintenance drug regimens.¹⁾ It is necessary to consider various factors that affect the pharmacokinetics of drugs, *i.e.* renal and hepatic dysfunction and factors such as age and sex. Population pharmacokinetic analysis (PPK) is a useful method for the evaluation of these variable factors. During the last few years, several computer programs for PPK have been developed and investigated, for example, MKMODEL,²⁾ NONMEM,³⁾ P-PHARM,⁴⁾ MULTI(ELS)⁵⁾ and MLG.⁶⁾ In PPK, variable factors are represented as mixed effects models: fixed effect and random effects. Fixed effects include population-typical values (mean), and random effects include population-variability values (standard deviation).¹⁾ Furthermore, using a few data points from each individual, a large population can be analyzed algebraically using PPK.

A new computer program MDGP has been developed for PPK. This program is based on maximum likelihood estimation with poly-dimensional normal distribution errors using variable metric methods, direction set methods, conjugate gradient methods and the polytope method. The features of MDGP are as follows: descriptions using C language based on ANSI C, source code derived from general purpose modules for numerical computations, provision of several algorithms for minimization of objective function, support of algebraic, ordinary differential and Laplace-transformed equations, and no necessity for the definition of partial differential equations. There is no program for PPK other than MDGP that can select algorithms, having these characteristics. The present paper focuses on the validation of MDGP. We compared MDGP and NONMEM using example data sets and

investigated the feasibility of using this program on several platforms.

THEORY AND METHODS

Objective Function Population parameters include inter-individual (η) and intra-individual (ε) variations. The objective function (*OBJ*), which is similar to MULTI(ELS),⁵⁾ is defined as follows.

$$OBJ = \sum_{i=1}^n ((y_i - f_i)^T V_i^{-1} (y_i - f_i) + \ln(\det V_i)) \quad (1)$$

where n is the number of individuals, y_i is the observed value vector in the i th individual, f_i is the predicted value vector using model equations in the i th individual, V_i is the variance-covariance matrix in the i th individual. V_i is represented as follows.

$$V_i = G_i \Omega_i G_i^T + \text{diag}(H_i \Sigma_i H_i^T) \quad (2)$$

$$\Omega_i = \begin{pmatrix} \eta_{11} & \eta_{12} & \cdots & \eta_{1m} \\ \eta_{21} & \eta_{22} & \cdots & \eta_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ \eta_{m1} & \eta_{m2} & \cdots & \eta_{mm} \end{pmatrix} \quad (3)$$

$$\Sigma_i = \begin{pmatrix} \varepsilon_{11} & & 0 \\ & \varepsilon_{22} & \\ 0 & & \cdots \\ & & & \varepsilon_{ll} \end{pmatrix} \quad (4)$$

where G is the Jacobian for η , Ω is the variance-covariance matrix for η , H is the Jacobian for ε , Σ is the diagonal matrix for ε .

Minimization of Objective Function Value The objective function is minimized using algorithms for a generalized

* To whom correspondence should be addressed.

function minimization. These are Variable metric methods, Direction set methods, Conjugate gradient methods and the Polytope method. Variable metric methods are gradient methods that use Hesse matrices as descent direction, and include DFP,⁷⁾ BFGS,⁸⁾ and rank one⁹⁾ formulae for the updating of Hessians. Direction set methods are conjugate direction procedures, which include the Powell and modified Powell methods.¹⁰⁾ These procedures do not use differentials of objective functions. Conjugate gradient methods are conjugate direction procedures using function gradients, which include the Fletcher-Reeves¹¹⁾ and Polak-Ribiere-Polyak¹²⁾ formulae. The polytope method, formerly called the Simplex method¹³⁾ depends on the comparison of function values at several points of a simplex.

Numerical Integration of Ordinary Differential Equations The solution of ordinary differential equations is related to numerical integration and includes the 4-order Runge-Kutta method (RK4) with an adaptive stepsize control.¹⁴⁾ RK4 is represented by the following steps:

$$f1 = hf'(t_n, y_n)$$

$$f2 = hf'\left(t_n + \frac{h}{2}, y_n + \frac{f1}{2}\right)$$

$$f3 = hf'\left(t_n + \frac{h}{2}, y_n + \frac{f2}{2}\right)$$

$$f4 = hf'(t_n + h, y_n + f3)$$

$$y_{n+1} = y_n + \frac{f1}{6} + \frac{f2}{3} + \frac{f3}{3} + \frac{f4}{6} \quad (5)$$

where h is the stepsize, t_n is the discrete time at n th, y_n is the independent variable vector at t_n , f' is the simultaneous ordinary differential equation and y_{n+1} is the independent variable vector at $t_n + h$.

Fast Inversion of Laplace Transform¹⁵⁾ The fast inversion of Laplace transform (FILT) is classified as a simple rule-procedure and numerical calculations. FILT assumes the following conditions for a Laplace transform $F(s)$ which is regular:

$$\begin{cases} \text{Re } s > 0 \\ \lim_{|s| \rightarrow \infty} F(s) = 0 \\ F^*(s) = F(s^*) \end{cases} \quad (6)$$

$$f_{ec}(t, a) = \frac{e^a}{t} \sum_{n=1}^{\infty} (-1)^n F_n \left(\frac{a + j(n-0.5)\pi}{t} \right) \quad (7)$$

where Re is the real part, s is the Laplace variable, the asterisk (*) is the complex conjugate, f_{ec} is a good approximation of FILT $F(s)$, t is the time, a is used in error estimation ($a \gg 1$), I_m shows the imaginary part of the complex number and j is the imaginary unit.

Program Description Running Platforms: When a C compiler based on the ANSI C standard is available, there is no necessity for specific hardware or operating systems. We confirmed the running of MDGP on platforms that are shown in Table 1.

Table 1. The Running Platforms Indicating Hardware, Operating Systems and C Compilers of MDGP

No.	Hardware (CPU)	Operating system	C compiler
1	Pentium (133 MHz)	MSDOS	DJGPP (gcc)
2	Pentium (133 MHz)	Windows95	Borland C++
3	Pentium (133 MHz)	WindowsNT	Borland C++
4	Pentium (133 MHz)	Linux	gcc
5	DEC Alpha 21164 (300 MHz)	WindowsNT	Visual C++
6	DEC Alpha 21164 (300 MHz)	Linux	gcc

Fig. 1. The Architecture of C Functions in Program Modules for the Computing Procedure in MDGP

Function Modules: The MDGP consists of some C source modules which are ObjFunc, Mend, Ode, Filt and user-defined functions. A schematic representation is shown in Fig. 1.

Function Mend: Function Mend minimizes the objective function value calculated using ObjFunc. Mend uses variable metric methods (BFGS, DFP or rank one formula), the Powell method, the modified Powell method, conjugate gradient methods (Fletcher-Reeves or Polak-Ribiere-Polyak formula) and the polytope method as minimization procedures. Users can choose any one of these procedures and use one out of the three procedures for parameter constraint based on no constraint, guarantee of positive values and sine limitation.

Function ObjFunc: ObjFunc is used to calculate the objective function value which is represented in Eq. 1. ObjFunc includes UserFunc, the function which is coded by users for pharmacokinetic models. Jacobian G and H are determined using difference analogues of UserFunc. Furthermore, $\det V$ is calculated using the LU decomposition.¹⁴⁾

Function Ode: Function Ode is the driver routine which is based on RK4 with an adaptive stepsize control, with arguments which are start t , end t and ordinary differential equations, and returns the value of y .

Function Filt: Function Filt calculates independent variables based on FILT for Laplace-transformed equations. Since it is difficult to perform accurate calculations using Eq. 7 which is an infinite series, the Euler transform was applied for calculations in the infinite series.¹⁵⁾ When Filt and modules related to only FILT are used, a C++ compiler is needed for complex numerical calculations.

User-Defined Function: Users can define the following

three functions. Function *AlgebUser* includes pharmacokinetic modeling equations which are derived from algebraic equations. Function *OdeUser* uses ordinary differential equations representing matter input-output equations of drugs based on their pharmacokinetic behavior. Function *FiltUser* uses equations involving Laplace variables, which are Laplace-transformed equations for a pharmacokinetic model.

RESULTS

Example 1 The following equations were used as the pharmacokinetic and pharmacostatistical models in the test runs.

$$\begin{aligned} P1 &= p[1] + p[3] \\ P2 &= p[2] + p[4] \\ y &= P1 \exp(-P2 \cdot x) + p[5] \end{aligned} \quad (8)$$

where y is the dependent variable (observed serum concentration), $P1$ is the initial value of y (initial serum concentration), $P2$ is the model parameter (constant of elimination) and x is the independent variable (sampling time). The absolute error model was assumed for the inter-individual variations of $P1$ and $P2$, and the intra-individual variation of y . $p[1]$ is the mean of $P1$, $p[2]$ is the mean of $P2$, $p[3]$ is the variation of $P1$, $p[4]$ is the variation of $P2$ and $p[5]$ is the residual variation. Moreover, it was assumed that Ω is a diagonal matrix. This model and test data were the same as those used as sample data in the MULTI(ELS) study.⁵⁾ The initial parameter values used were $p[1]=10.0$, $p[2]=1.0$, $p[3]=1.0$, $p[4]=1.0$ and $p[5]=1.0$. In MDGP, the guarantee of positive values was applied as a constraint on the parameters.

The coding example of the source program is shown in Fig. 2, where users have only to define the function *AlgebUser*. Users can freely code the source program in function *AlgebUser*, but *VEC v* and *VEC p*, which are array arguments of this function are fixed, v (size n) and p (size m) represent independent variables $v[1]$, $v[2]$, ..., $v[n]$ and model parameters $p[1]$, $p[2]$, ..., $p[n]$ respectively. This function must return a dependent value of a double precision. Though intermediate variables x , $p1$, $p2$ and y are used in this func-

```

1: /*-----
2: * User definition: Algebraic equation
3: * If you want to use algebraic, you must define AlgebUser.
4: * [Data format]
5: * v[1]: x
6: * p[1]: p1
7: * p[2]: p2
8: * p[3]: variation of p1
9: * p[4]: variation of p2
10: * p[5]: variation of y
11: *-----*/
12: double AlgebUser(VEC v, VEC p)
13: {
14:     double x, P1, P2, y ;
15:
16:     x = v[1] ;
17:     P1 = p[1] + p[3] ;
18:     P2 = p[2] + p[4] ;
19:     y = P1*exp(-P2*x) ;
20:     return y + p[5] ;
21: }
```

Fig. 2. Coding Example of Source Program for the Pharmacokinetic Model Defined by Users in *AlgebUser* Function

There are no left line numbers in actual source codes.

tion coding, users may define the function as follows:

```
return (p[1]+p[3])*exp(-(p[2]+p[4])*x)+p[5];
```

Figures 3 and 4 represent coding examples of ordinary differential equations, and Laplace-transformed formulae, respectively.

The NONMEM version IV, level 2.0 (double precision version) was used as a standard for comparing the calculation performance of MDGP. The results of comparison revealed comparable iteration counts until the calculation of convergence, model parameters and objective function value.

The example programs were evaluated on the following platforms.

```

1: /*-----
2: * User definition: Ordinary differential equation
3: * If you want to use ODE,
4: * you must define OdeInit, OdeUser and OdeOut.
5: * [Data format]
6: * p[1]: p1
7: * p[2]: p2
8: * p[3]: variation of p1
9: * p[4]: variation of p2
10: * p[5]: variation of y
11: *-----*/
12: double OdeInit(VEC y, VEC v, VEC p)
13: {
14:     y[1] = p[1] + p[3] ;
15: }
16:
17: void OdeUser(double t, VEC y, VEC p, VEC dydx, VEC v)
18: {
19:     double P2 ;
20:
21:     P2 = p[2] + p[4] ;
22:     dydx[1] = -P2*y[1] ;
23: }
24:
25:
26: double OdeOut(VEC y, VEC v, VEC p)
27: {
28:     return y[1] + p[5] ;
29: }
```

Fig. 3. Coding Example of Source Program for the Pharmacokinetic Model Defined by Users in *OdeUser* Function

There are no left line numbers in actual source codes.

```

1: /*-----
2: * User definition: Laplace-transformed formula
3: * If you want to use Laplace-transformed,
4: * you must define FiltUser and FiltOut.
5: * You must include complex library header(C++).
6: * [Data format]
7: * p[1]: p1
8: * p[2]: p2
9: * p[3]: variation of p1
10: * p[4]: variation of p2
11: * p[5]: variation of y
12: *-----*/
13: complex FiltUser(VEC v, VEC p, complex s)
14: {
15:     double P1, P2 ;
16:
17:     P1 = p[1] + p[3] ;
18:     P2 = p[2] + p[4] ;
19:
20:     return P1/(s + P2) ;
21: }
22:
23: double FiltOut(double y, VEC v, VEC p)
24: {
25:     return y + p[5] ;
26: }
```

Fig. 4. Coding Example of Source Program for the Pharmacokinetic Model Defined by Users in *FiltUser* Function

There are no left line numbers in actual source codes.

Table 2. Computation of Example Runs on Platform I

	BFGS	DFP	PWL	PWL2	FR	NONMEM	ELS
LOOP	18	34	13	15	94	20	—
$p[1]$	9.15E+00	9.15E+00	9.15E+00	9.15E+00	9.15E+00	9.15E+00	9.15
$p[2]$	4.90E-01	4.90E-01	4.90E-01	4.90E-01	4.90E-01	4.90E-01	0.490
$p[3]$	9.78E-01	9.78E-01	9.78E-01	9.78E-01	9.83E-01	9.78E-01	0.983
$p[4]$	8.90E-04	8.88E-04	8.90E-04	8.91E-04	9.08E-04	8.82E-04	0.001
$p[5]$	3.89E-01	3.89E-01	3.89E-01	3.89E-01	3.89E-01	3.89E-01	0.388
OBJ	12.651	12.882	12.651	12.651	12.651	12.651	12.7

The results of ELS are taken from MULTI (ELS).⁵⁾ BFGS, variable metric method (BFGS formula); DFP, variable metric method (DFP formula); PWL, direction set method; PWL2, direction set method (Powell's modification); FR, conjugate gradient method (Fletcher-Reeves formula); LOOP, iteration counts until the calculation of convergence; OBJ, objective function value.

Table 3. Computation of Example Runs on Platform II

	BFGS	DFP	PWL	PWL2	FR	NONMEM	ELS
LOOP	19	33	12	19	90	22	—
$p[1]$	9.15E+00	9.15E+00	9.15E+00	9.15E+00	9.16E+00	9.15E+00	9.15
$p[2]$	4.90E-01	4.90E-01	4.90E-01	4.90E-01	4.90E-01	4.90E-01	0.490
$p[3]$	9.78E-01	9.77E-01	9.78E-01	9.78E-01	9.86E-01	9.78E-01	0.983
$p[4]$	8.89E-04	8.91E-04	8.91E-04	8.91E-04	9.19E-04	8.91E-04	0.001
$p[5]$	3.89E-01	3.89E-01	3.89E-01	3.89E-01	3.88E-01	3.89E-01	0.388
OBJ	12.651	12.651	12.651	12.651	12.651	12.651	12.7

The results of ELS are taken from MULTI (ELS).⁵⁾

I. Pentium (133 MHz) CPU, Microsoft Windows NT 4.0, Borland C++ compiler, f2c

II. DEC Alpha 21164 (300 MHz) CPU, Linux (UNIX clone operating system), gcc, g77

In platform I, NONMEM was translated from Fortran to C source codes using the f2c Fortran-C translator, and its C source programs were compiled into executable files using a Borland C++ compiler based on the ANSI C standard. In platform II, NONMEM was compiled into executable files using the g77 ANSI Fortran 77 compiler.

The computed results on platforms I and II using algebraic equations are shown in Tables 2 and 3, respectively. Upon running MDGP and NONMEM on platform I, iteration counts until the calculation of convergence were different between each minimization algorithm, and ranged from 13 to 94. The objective function value was calculated as 12.651. With regard to MDGP and NONMEM running on platform II, iteration counts until the calculation of convergence were different between each minimization algorithm, and ranged from 12 to 90. The objective function value was calculated as 12.651. These results show that the estimation of parameters using MDGP, MULTI(ELS) and NONMEM, were comparable.

The computed results on platform I using ordinary differential and Laplace-transformed equations are shown in Table 4. The iteration counts until the calculation of convergence ranged from 8 to 29. The objective function value in each case ranged from 12.651 to 12.917. No difference was noted between the parameters estimated using ordinary differential equations and Laplace-transformed equations.

Example 2 This model and test data have been used as the NONMEM example data with the population pharmacokinetic model of theophylline.¹⁶⁾ The following equations were used as the pharmacokinetic and pharmacostatistical models in the test runs.

Table 4. Computation of Example Runs on Platform I Using Ordinary Differential and Laplace-Transformed Formulae

	ODE		FILT	
	BFGS	PWL2	BFGS	PWL2
LOOP	18	8	29	12
$p[1]$	9.15E+00	9.15E+00	9.16E+00	9.15E+00
$p[2]$	4.90E-01	4.90E-01	4.98E-01	4.90E-01
$p[3]$	9.63E-01	9.64E-01	9.99E-01	9.78E-01
$p[4]$	5.01E-04	5.13E-04	1.11E-03	8.89E-04
$p[5]$	3.99E-01	3.99E-01	3.87E-01	3.90E-01
OBJ	12.917	12.917	12.687	12.651

ODE, numerical solution of ordinary differential equations; FILT, fast inverse Laplace transform.

$$k_a = p[1] + p[4]$$

$$k_e = p[2] + p[5]$$

$$CL = p[3] + p[6]$$

$$V_d = CL/k_e$$

$$y = \frac{D}{V_d} \frac{k_a}{k_a - k_e} (\exp(-k_e \cdot x) - \exp(-k_a \cdot x)) + p[7]$$

where k_a is the absorption rate constant, k_e is the elimination rate constant, CL is the total clearance and V_d is the distribution volume. The absolute error model was assumed for the inter-individual variations of k_a , k_e and CL , and the intra-individual variation of y . $p[1]$ is the mean of k_a , $p[2]$ is the mean of k_e , $p[3]$ is the mean of CL , $p[4]$ is the variation of k_a , $p[5]$ is the variation of k_e , $p[6]$ is the variation of CL and $p[7]$ is the residual variation. Moreover, it was assumed that Ω is a diagonal matrix.

The initial parameters used were those of NONMEM example values. The results of comparison revealed compara-

Table 5. Computation of Example Runs on Platform I Using NONMEM Example Data with the Population Pharmacokinetic Model of Theophylline

	ALGEB PWL	ODE PWL	FILT PWL	NONMEM
LOOP	8	8	9	16
$p[1]$	2.72E+00	2.72E+00	2.72E+00	2.72E+00
$p[2]$	7.64E-02	7.59E-02	7.64E-02	7.64E-02
$p[3]$	3.67E-02	3.64E-02	3.67E-02	3.67E-02
$p[4]$	5.66E+00	5.66E+00	5.66E+00	5.66E+00
$p[5]$	1.14E-04	1.11E-04	1.14E-04	1.14E-04
$p[6]$	3.07E-01	3.01E-01	3.07E-01	3.07E-01
$p[7]$	4.17E-01	4.17E-01	4.17E-01	4.17E-01
OBJ	113.293	113.167	113.293	113.293

ALGEB, algebraic equations; ODE, numerical solution of ordinary differential equations.

Table 6. Computation of Example Runs on Platform I Using Trial Data with the Population Pharmacokinetic Model of Virtual Medicine

	ALGEB		NONMEM
	BFGS	PWL2	
LOOP	21	9	22
$p[1]$	1.33E+00	1.33E+00	1.33E+00
$p[2]$	5.17E-02	5.18E-02	5.18E-02
$p[3]$	4.82E-01	4.82E-01	4.82E-01
$p[4]$	7.60E-03	7.58E-03	7.55E-03
$p[5]$	1.36E-01	1.35E-01	1.36E-01
$p[6]$	2.22E-02	2.22E-02	2.22E-02
$p[7]$	1.45E-02	1.46E-02	1.45E-02
$p[8]$	1.52E-02	1.52E-02	1.52E-02
OBJ	1857.58	1857.58	1857.578

with standard deviations of 30, 20, 10 and 10% respectively.

The initial parameters used were $p[1]=1.5$, $p[2]=0.05$, $p[3]=0.5$, $p[4]=0.01$, $p[5]=0.01$, $p[6]=0.01$, $p[7]=0.01$ and $p[8]=0.01$. The results of comparison revealed comparable iteration counts until the calculation of convergence, model parameters and objective function value.

The example programs were evaluated on platform I in example 1. The pharmacokinetic model was described using algebraic equations. The minimization algorithm of object function values used a variable metric method (BFGS) and direction set method (Powell's modification).

The computed results are shown in Table 6. Upon running MDGP and NONMEM, iteration counts until the calculation of convergence using variable metric and direction set methods were 21 and 9 respectively. The objective function value was calculated as 1857.58. These results show that the estimated population parameters using MDGP and NONMEM were comparable.

DISCUSSION

Many PPK programs are in use in the field of clinical pharmacokinetics which have technically different characteristics. For example, the MKMODEL performs nonlinear extended least squares on MSDOS,²⁾ NONMEM performs nonlinear regression analysis for mixed effects modeling and is provided as Fortran source code,³⁾ P-PHARM is a population pharmacokinetic-dynamic data modeling program using a two-stage method and/or an EM-type algorithm,⁴⁾ MULTI (ELS) is based on extended least squares and is provided as BASIC source code⁵⁾ and MLG is based on maximization of likelihood using gradient methods and is provided as Fortran source code.⁶⁾ Furthermore, NONLIN¹⁷⁾ and ADAPT¹⁸⁾ are general programs for pharmacokinetic analysis that use the definition of ordinary differential equations as pharmacokinetic models. However, there are several drawbacks for using these programs: the limitation of data size and parameter numbers within the bounds of memory resources in a specific operating system (OS, e.g. on MSDOS), the dependence on a particular OS and language processor, not allowing modifications and additions of pharmacokinetic models by users because of the provision of executable binary files, difficulty in developing graphical user interface (GUI) applications when source modules are made using a programming language other than C and/or C++, which are currently the popular

ble iteration counts until the calculation of convergence, model parameters and objective function value. The example programs were evaluated on platform I in example 1. The pharmacokinetic model was described using algebraic, ordinary differential and Laplace-transformed equations. The minimization algorithm of object function values used a direction set method (Powell's modification).

The computed results using algebraic, ordinary differential and Laplace-transformed equations are shown in Table 5. Upon running MDGP and NONMEM, iteration counts until the calculation of convergence were different between each minimization algorithm, and ranged from 8 to 16. The objective function value was calculated as 113. No difference was noted between the parameters estimated using ordinary differential equations, Laplace-transformed equations and using algebraic equations. These results show that MDGP gave the same estimated values of parameters as NONMEM.

Example 3 This model and test data were used as trial data in the population pharmacokinetic model of virtual medicine. It supported the one-compartment model with first-order absorption, and the total clearance of this model is proportional to the creatinine clearance ($CLCr$) as population pharmacokinetic model. The following equations were used as the pharmacokinetic and pharmacostatistical models in the test runs.

$$k_a = p[1](1 + p[5])$$

$$CL = (p[2] + p[4] \cdot CLCr)(1 + p[6])$$

$$V_d = p[3](1 + p[7])$$

$$k_e = CL/V_d$$

$$y = \frac{D}{V_d} \frac{k_a}{k_a - k_e} (\exp(-k_e \cdot x) - \exp(-k_a \cdot x)) + p[8]$$

The relational error model was assumed for the inter-individual variations of k_a , CL and V_d , and the intra-individual variation of y . $p[1]$ is the mean of k_a , $p[2]$ is the mean of CL , $p[3]$ is the mean of V_d , $p[4]$ is the regression coefficient of $CLCr$, $p[5]$ is the variation of k_a , $p[6]$ is the variation of CL , $p[7]$ is the variation of V_d and $p[8]$ is the residual variation. Moreover, it was assumed that Ω is a diagonal matrix. Sample data yielded two hundred and forty points which included 8 points per individual and a total of thirty individuals, and the sampling time of the 8 points was 0.25, 0.5, 1, 2, 4, 8, 12 and 24 h. Then k_a , CL , V_d and y were normally distributed

languages for developing GUI applications, the computing costs are high for applications using interpreter or intermediate language, and a definition of partial differential equations is needed for the Jacobians of model parameters. We, in an attempt to find solutions for these problems, have developed a computer program and library modules having the following features: source codes written using ANSI C language, no specific platforms which include OS and compilers, flexible definition of pharmacokinetic models by users, use of differential approximations for Jacobian calculations, support of algebraic, ordinary differential, and Laplace-transformed equations as pharmacokinetic models and the provision of several modules for minimization of objective function.

The results of several example runs to obtain parameters using MDGP were in agreement with those obtained using NONMEM, and the values of objective function were equal in each minimization algorithm and NONMEM. The use of variable metric methods (BFGS formula) as the algorithm for NONMEM resulted in similar iteration counts until convergence. Therefore, we consider that the results obtained using MDGP are sufficiently reliable from the viewpoint of numerical calculations, and similar results are obtained on varied platforms. It is believed that any user can define the few pharmacokinetic model equations required, since definition of partial differential equations is not needed. Furthermore, we believe that MDGP is useful for the analysis of nonlinear models and complex models since ordinary differential equations and Laplace transform formulae can be applied as pharmacokinetic models.

From the comparison of MDGP and NONMEM on varied platforms, it is clear that the calculation performances of the two programs are similar. Its usability on varied platforms,

its portability and the possibility for modification by users of MDGP and these modules are useful features since source codes are described using ANSI C language. The MDGP is able to choose any one algorithm from several options. The MDGP is thought to be useful for the analysis of various pharmacokinetic models including population pharmacokinetic analysis.

REFERENCES

- 1) Evans W. E., Schentag J. J., Jusko W. J. (ed.), "Applied Pharmacokinetics," Applied Therapeutics, Vancouver, 1992.
- 2) Holford N., "MKmodel Manual ver 3. 13," Elsevier, New York, 1988.
- 3) Beal S. L., Sheiner L. B., "NONMEM Users Guide—Part I Users Basic Guide," NONMEM Project Group, UCSF, San Francisco, 1989.
- 4) "P-Pharm User's Guide," 2nd ed., SIMED, Creteil, France, 1994.
- 5) Yamaoka K., Tanaka H., Okumura K., Yasuhara M., Hori R., *J. Pharmacobio-Dyn.*, **9**, 161—173 (1986).
- 6) Amisaki T., Tatsuhara T., *J. Pharmacobio-Dyn.*, **11**, 335—348 (1988).
- 7) Fletcher R., Powell M. J. D., *Comput. J.*, **6**, 163—168 (1963).
- 8) Fletcher R., *Comput. J.*, **13**, 317—322 (1970).
- 9) Broyden C. G., *Math. Comp.*, **21**, 368—381 (1967).
- 10) Powell M. J. D., *Comput. J.*, **7**, 155—162 (1964).
- 11) Fletcher R., Reeves C. M., *Comput. J.*, **7**, 149—154 (1964).
- 12) Polak V. T., *USSR Comp. Maths. and Math. Phys.*, **9**, 94—112 (1969).
- 13) Nelder J. A., Mead R., *Comput. J.*, **7**, 308—313 (1965).
- 14) Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P., "Numerical Recipes in C: The Art of Scientific Computing," 2nd ed., Cambridge University Press, Cambridge, 1993.
- 15) Hosono T., *Radio Sci.*, **16**, 1015—1019 (1981).
- 16) Boeckmann A. J., Sheiner L. B., Beal S. L., "NONMEM Users Guide—Part V Introductory Guide," NONMEM Project Group, UCSF, San Francisco, 1992.
- 17) Metzler C., Tech. Rep. 7292/69/9292/005, The Upjohn Company (1969).
- 18) D'Argenio D. Z., Schumitzky A., *Comput. Prog. Biomed.*, **9**, 115—134 (1979).