NONMEM Users Guide - Part V

Introductory Guide

December 2017

by

Alison J. Boeckmann

Lewis B. Sheiner

Stuart L. Beal

NONMEM Project Group
University of California at San Francisco

ICON plc, Gaithersburg, Maryland

Preface

This edition of "NONMEM Users Guide - Part V Introductory Guide" is distributed with NONMEM 7.4. It revises the version of November 2013, which appeared with NONMEM 7.3. Details that have changed since the previous edition have been corrected, and some new features have been added.

Significant changes since the previous version are marked with bars.

Examples of NONMEM outputs have not been updated. They remain as they were in 1994 and are from NONMEM IV. With later versions of NONMEM there are changes in the outputs. In some cases the wording has been changed; there is new content; and the numerical results may have changed slightly. But none of this affects the features and methodology that Lewis Sheiner described in chapters 2, 10, and 11.

True to its purpose as an instructional guide for new users of NONMEM, this Guide remains oriented to the classic NONMEM methods and basic features (through NONMEM VI). References to even earlier versions of NONMEM and PREDPP have been deleted.

Chapter 12 (Brief Descriptions of Other Features) has been revised. Sections 1-5 have been expanded to be more complete. A new section, 2.8. Output-Type Compartments, describes a feature that has always been part of PREDPP, but was not documented. Note sub-sections titled "More About ...". Section 6 ("Supplemental List of Features through NONMEM 7.4") is is a summary of all features of NONMEM not mentioned elsewhere in this guide.

Appendix 4 is new to this version. It contains implementation details for the records listed in Appendix 3.

Table of Contents

Chapter 1 - Introduction to NONMEM, PREDPP, and NM-TRAN . . . 1
What This Chapter is About . . . 1
Introducing NONMEM . . . 1
What is NONMEM? . . . 1
What is PREDPP? . . . 1
What is NM-TRAN? . . . 2
Scope of this Introductory Guide . . . 2
Contents of this Introductory Guide . . . 3
How to Read this Guide . . . 3
A Brief Technical Overview . . . 4
NONMEM . . . 4
PREDPP and the PREDPP Library . . . 4
Subroutines PK and ERROR . . . 4
Building an Executable Module for NONMEM . . . 5
NM-TRAN . . . 5
Control Language Translation . . . 5
Model Specification via Abbreviated Code . . . 5
Partial Differentiation . . . 6
Data Preprocessor . . . 6
Additional Documentation . . . 6
Chapter 2 - NONMEM Examples . . . 8
What This Chapter is About . . . 8
An Individual's Theophylline Kinetics . . . 8
The NM-TRAN Control Records . . . 8
The Model . . . 9
The Output . . . 10
A Population Model for Phenobarbital . . . 12
The NM-TRAN Control Records . . . 13
The Model . . . 13
The Output . . . 14
Overview . . . 21
Chapter 3 - Models for Individual Data . . . 23
What This Chapter is About . . . 23
Pharmacokinetic Structural Models for Individual Data . . . 23
Alternative Parameterizations . . . 24
The Scale Parameter, **S** . . . 24
S Depends on a Known Constant . . . 24
S Depends on a Parameter . . . 25
S Depends on an Element of **x** . . . 25
Statistical Model for an Individual's Observations . . . 25
The Additive Error Model . . . 27
The Constant Coefficient of Variation and Exponential Models . . . 27
Combined Additive and CCV Error Model . . . 28
The Power Function Model . . . 29
Two Different Types of Measurements . . . 29
Use of an Indicator Variable . . . 29
Two Different Types of Observations . . . 30
More Than One Indicator Variable . . . 30

The General Mixed Effects Model for an Individual . . . 31
Chapter 4 - Models for Population Data . . . 32
What This Chapter is About . . . 32
General . . . 32
Structural Parameter Models . . . 33
Linear Models . . . 33
Multiplicative Models . . . 34
Saturation Models . . . 34
Models with Indicator Variables . . . 35
Combinations . . . 35
Time Varying *z* . . . 35
Structural Kinetic Models . . . 36
Population Random Effects Models . . . 36
Models for Interindividual Errors . . . 36
Additive/Multiplicative Models and the Exponential Model . . . 37
Other Models . . . 37
General Form for the Parameter Model . . . 38
Statistical Models for an Individual's Observations . . . 39
The Population Mixed Effects Model . . . 39
Chapter 5 - Estimates, Confidence Intervals, and Hypothesis Tests . . . 41
What This Chapter is About . . . 41
Model Fitting Criterion . . . 41
Least Squares for Individual Data with an Additive Error Model . . . 41
Least Squares for Individual Data with Other Types of Error Models . . . 41
Least Squares for Population Data . . . 42
Parameter Estimates . . . 43
Precision of Parameter Estimates . . . 43
Distribution of Parameters vs Distribution of Parameter Estimates . . . 44
Confidence Interval for a Single Parameter . . . 44
Estimating a Parameter's Standard Error . . . 45
Relating the Confidence Interval to the **SE** . . . 45
A Confidence Interval for a Function of a Single Parameter . . . 45
Multiple Parameters . . . 45
Correlation of Parameter Estimates . . . 45
Confidence Intervals for a Function of Several Parameters . . . 46
Hypothesis Testing . . . 46
Hypothesis Testing Using the **SE** . . . 47
Hypothesis Testing Using the Likelihood Ratio . . . 47
Definition — Full/Reduced Models . . . 47
Reduced/Full Models Express the Null/Alternative Hypotheses . . . 47
The Likelihood Ratio Test . . . 48
Choosing Among Models . . . 48
Chapter 6 - Data Sets, $DATA and $INPUT Records, and the Data Preprocessor . . . 50
What This Chapter is About . . . 50
Data Sets for NONMEM . . . 50
Data Records . . . 50
Data Items . . . 50
Clinical Data and Data Conversion . . . 51
Data Sets for PREDPP . . . 52

The $DATA Record . . . 52
The $INPUT Record . . . 54
Data Item Labels . . . 54
Reserved Labels and Synonyms . . . 54
Dropping Data Items via DROP . . . 55
NONMEM Data Items . . . 55
DV Data Item . . . 55
ID Data Item for Population Data . . . 55
MDV data item . . . 55
PREDPP Data Items . . . 56
TIME Data Item . . . 56
AMT, RATE, SS, II: Dose-related Data Items . . . 56
EVID Data Item . . . 56
CMT and PCMT Data Items . . . 57
CALL Data Item . . . 58
Describing Doses to PREDPP . . . 58
Dose-related Data Items . . . 58
Different Kinds of Doses . . . 59
Instantaneous Bolus Doses . . . 59
Infusions . . . 59
Steady-State Doses . . . 60
Steady-State with Multiple Bolus Doses . . . 60
Steady-State with Multiple Infusions . . . 61
Steady-State with Constant Infusion . . . 62
Multiple Steady-State Doses . . . 63
Combining Non-Steady-State Doses with Steady-State Doses . . . 64
The Output Compartment: Urine Collections and Observations . . . 64
The Data Preprocessor . . . 65
Day-time Translation . . . 65
TIME Data Item . . . 65
DATE Data Item . . . 66
Calendar Dates . . . 66
Converting Hours to Days and More General Conversions . . . 67
The Year 2000 - LAST20 . . . 68
Leap Year Warning - LYWARN . . . 68
Interdose Interval (II) Conversion . . . 69
Data Items Generated by the Data Preprocessor . . . 69
When Must a Format Specification be Included or Omitted? . . . 69
Skipping Data Items . . . 70
Chapter 7 - $SUBROUTINE Record and $PK Record . . . 71
What This Chapter is About . . . 71
The $SUBROUTINE Record . . . 71
Choosing an ADVAN Subroutine: Standard Pharmacokinetic Models . . . 71
Choosing a TRANS Subroutine: Alternative Parameterizations . . . 72
$PK Abbreviated Code . . . 73
Syntax . . . 73
When are $PK Statements Evaluated? . . . 74
Time Varying PK parameters . . . 74
$PK Statements for Individual Data . . . 75
Basic and Additional Parameters . . . 75

Alternative Parameterizations using $PK Statements . . . 76
Scale Parameters . . . 77
Scaling by a Known Constant . . . 77
Scaling by a Parameter: Conditional Statements and Indicator Variables . . . 77
Scaling by a Data Item . . . 78
Bioavailability Fraction Parameters . . . 79
Output Fraction . . . 79
$PK Statements for Population Data . . . 80
Structural Part of Parameter Models . . . 80
Linear Models . . . 80
Multiplicative Models . . . 80
Saturation Models . . . 80
Models with Indicator Variables . . . 81
Population Random Effect Models . . . 81
Models for Interindividual Errors . . . 82
Additive/Multiplicative Models . . . 82
Other Models . . . 82
Restrictions on Random Variables . . . 82
Chapter 8 - $ERROR Record . . . 84
What This Chapter is About . . . 84
$ERROR Abbreviated Code . . . 84
Syntax . . . 84
When are $ERROR Statements Evaluated? . . . 84
Error Models . . . 85
The Additive Error Model . . . 85
The Constant Coefficient of Variation and Exponential Models . . . 85
Combined Additive and CCV Error Model . . . 85
The Power Model . . . 85
Two Different Types of Measurements . . . 85
Two Different Types of Observations . . . 86
More than One Indicator Variable . . . 86
Chapter 9 - Additional NM-TRAN Records . . . 87
What This Chapter is About . . . 87
Providing Initial Estimates For θ: The $THETA Record . . . 87
Providing Initial Estimates For Elements Of θ . . . 87
Providing Constraints for Elements of θ . . . 87
Fixing Elements of θ . . . 88
How to Obtain Initial Estimates for θ . . . 88
Providing Initial Estimates for Ω and Σ: the $OMEGA and $SIGMA Records . . . 88
$OMEGA Record With Individual Data . . . 89
$OMEGA Record With Population Data . . . 89
The $SIGMA Record . . . 89
Fixing Elements of Ω or Σ . . . 89
How to Obtain Initial Estimates for Ω and Σ . . . 90
Specifying Optional Tasks . . . 91
Requesting the Estimation Step: the $ESTIMATION Record . . . 91
Requesting the Covariance Step: the $COVARIANCE Record . . . 92
Specifying Optional Output . . . 92
Requesting the Table Step: the $TABLE Record . . . 93

Requesting Scatterplots: the $SCATTERPLOT Record . . . 93
Placement and Order of Records . . . 93
INCLUDE records . . . 94
Chapter 10 - Reading the Output . . . 95
What This Chapter is About . . . 95
NONMEM Describes its Inputs . . . 95
PREDPP Describes Its Inputs . . . 97
Diagnostic Output from the Estimation Step . . . 99
Intermediate Output from the Estimation Step . . . 99
Summary Output from the Estimation Step . . . 101
Minimum Value of the Objective Function and Final Parameter Estimates . . . 101
Output from the Covariance Step . . . 102
Additional Output: Tables and Scatterplots . . . 103
Output from the Table Step . . . 104
Output from the Scatterplot Step . . . 104
Chapter 11 - Model Building . . . 105
What This Chapter is About . . . 105
The Stages of Model Building . . . 105
Check-out — Index Plots . . . 107
Building the Structural Part of the Model . . . 112
A General Approach . . . 112
The Minimal Model . . . 113
Use of Constraints . . . 114
Diagnostic Tools . . . 114
Plot of DV vs PRED . . . 115
Residual Plots . . . 115
Index Plots of Residuals . . . 116
Plot of WRES vs Independent Variable . . . 117
Judging Goodness of Fit . . . 118
A Global Measure — Change in the Objective Function . . . 119
Decrease in Unexplained Variability . . . 119
Improvement in Plots . . . 119
Using the Tools: Further Improvement . . . 121
An Additional Effect of WT . . . 121
The Effect of APGR . . . 123
Building the Statistical Model . . . 125
Judging Among Alternatives . . . 125
Unexplained Variability . . . 126
Residual Plots . . . 126
Refine Model . . . 128
Use of Standard Errors and Confidence Intervals . . . 129
A Model Refinement . . . 130
Testing the Model . . . 131
Chapter 12 - Brief Descriptions of Other Features . . . 132
What This Chapter is About . . . 132
Advanced Features of PREDPP . . . 132
Pharmacodynamic Modeling Using the $ERROR Record . . . 132
Other Pharmacokinetic Models: ADVAN5-9, ADVAN13-ADVAN15 . . . 133
Zero-Order Bolus Doses . . . 134
The Additional Dose Data Item: ADDL . . . 135

Lagged doses: the ALAG Parameter . . . 135
Model Event Times: MTIME . . . 135
Controlling Calls to PK and ERROR . . . 136
Output-Type Compartments . . . 137
Transgeneration of Input Data: the INFN Subroutine . . . 138
User-written PRED Subroutines . . . 138
Required Data Items . . . 138
An Example of $PRED Statements: Pharmacodynamic Modeling . . . 139
Advanced Features of NONMEM . . . 139
Full Covariance Matrices: $OMEGA BLOCK and $SIGMA BLOCK . . . 139
More About $OMEGA and $SIGMA . . . 140
Grouping Related Observations: The L1 and L2 Data Items . . . 141
Continuing a NONMEM Run: MSFO and MSFI . . . 142
NONMEM Can Obtain Initial Estimates for θ, Ω, Σ . . . 142
Improving Parameter Estimates: REPEAT and RESCALE . . . 143
The Covariance Step: $\mathbf{R^{-1}}$, $\mathbf{S^{-1}}$, Special Computation . . . 143
More About $COVARIANCE . . . 143
Multiple Problems in a Single NONMEM Run . . . 144
Simulation Using NONMEM: The $SIMULATION Record . . . 144
More About $SIMULATION . . . 145
Files for Subsequent Processing: the $TABLE Record . . . 146
More about $TABLE and $SCATTER . . . 146
Data Checkout Mode . . . 147
Obtaining Individual Parameter Estimates - Conditional Estimates of ηs . . . 147
Population Conditional Estimation Methods . . . 147
Displaying PRED-Defined Variables and Conditional Estimates of ηs . . . 148
Mixture Models . . . 148
PRED Error Return Codes and Error Messages in File PRDERR . . . 148
User-Written Subroutines . . . 149
PRIOR . . . 149
Observations of Two Different Types . . . 150
Supplemental List of Features through NONMEM 7.4 . . . 152
NONMEM Features . . . 152
Miscellaneous Features . . . 153
Changes to NONMEM Outputs . . . 154
PREDPP . . . 155
NM-TRAN . . . 155
General Features . . . 155
Data Preprocessor . . . 156
Abbreviated Code . . . 156
Reserved Variables in Abbreviated Code . . . 157
Utility Routines . . . 158
All Options for $ESTIMATION . . . 159
All Options for $COVARIANCE . . . 166
Chapter 13 - Errors in NONMEM Runs . . . 168
What This Chapter is About . . . 168
Abnormal Termination of the Estimation Step . . . 168
"DUE TO MAX. NO. OF FUNCTION EVALUATIONS EXCEEDED" . . . 168
"DUE TO ROUNDING ERRORS (ERROR=134)" . . . 169
Abnormal Termination of the Covariance Step . . . 169

Miscellaneous Problems 170
Proportional Error Model 170
Errors in the Pharmacokinetic Model 170
Errors with PREDPP 171
Error Messages from a TRANS Routine 171
Error Messages from ADVAN Routines 171
Numeric difficulties in PREDPP 171
Index 197

Chapter 1 - Introduction to NONMEM, PREDPP, and NM-TRAN

1. What This Chapter is About

This chapter introduces a computer program called NONMEM. It also introduces two programs that are distributed with NONMEM and make it easier to use: PREDPP and NM-TRAN. The scope of this text itself is described, and suggestions are made for reading it. A somewhat detailed technical description of the components of NONMEM is then given. The final section is a list of additional references.

2. Introducing NONMEM

2.1. What is NONMEM?

NONMEM stands for "Nonlinear Mixed Effects Model." NONMEM is a computer program, written in FORTRAN 90/95, designed to fit general statistical (nonlinear) regression-type models to data.

NONMEM was developed by the NONMEM Project Group at the University of California at San Francisco for analyzing population pharmacokinetic data in particular.† These are data typically collected from clinical studies of pharmaceutic agents, involving the administration of a drug to individuals and the subsequent observation of drug levels (most often in the blood plasma). Proper modeling of these data involves accounting for both unexplainable inter- and intra-subject effects (random effects), as well as measured concomitant effects (fixed effects). NONMEM allows this mixed effect modeling. Such modeling is especially useful when there are only a few pharmacokinetic measurements from each individual sampled in the population, or when the data collection design varies considerably between these individuals. However, NONMEM is a general program which can be used to fit models to a wide variety of data.

Like many nonlinear regression programs, NONMEM does not have any "built in" models (such as the linear model) with which it can compute a predicted value given the current values of the regression parameters. Instead, NONMEM calls a subroutine having entry name PRED ("prediction") to obtain a predicted value. PRED also must compute for NONMEM partial derivatives with respect to certain random variables. Depending on the model and the kinds of doses, PRED may be very simple or may be very complicated. A user can write his own PRED subroutine. This can be as simple or complicated as is necessary, and may involve calls to its own subprograms.

2.2. What is PREDPP?

PREDPP stands for "PRED for Population Pharmacokinetics". It is a PRED subroutine for use with NONMEM and is the second major component distributed with NONMEM. Whereas NONMEM is a general nonlinear regression tool, PREDPP is specialized to the kinds of predictions which arise in pharmacokinetic data analysis. It can compute predictions according to many different pharmacokinetic models, according to a great variety of dosing regimens. Almost all the examples in this guide use PREDPP.

---

† NONMEM versions up through VI are the property of the Regents of the University of California, but ICON Development Solutions has exclusive rights to license their use. NONMEM 7 is the current version of the software and is the property of ICON Development Solutions.

2.3. What is NM-TRAN?

NM-TRAN stands for "NONMEM Translator". It is the third major component distributed with NONMEM. It is a separate, "stand-alone" control language translator and data preprocessor. When NM-TRAN is used, a NONMEM run includes two separate steps: first the NM-TRAN step, in which a file of NM-TRAN records (which begin with "$") are translated into several NONMEM input files, and second the NONMEM step itself. All the examples in this guide use NM-TRAN. We strongly recommend its use.

Note that neither NM-TRAN nor NONMEM-PREDPP run interactively. Files of commands and data are created by means of (say) the operating system editor. Then NM-TRAN and NONMEM are executed, using these files as input. Figure 1.1 shows the relationship between NONMEM, PREDPP, and NM-TRAN.

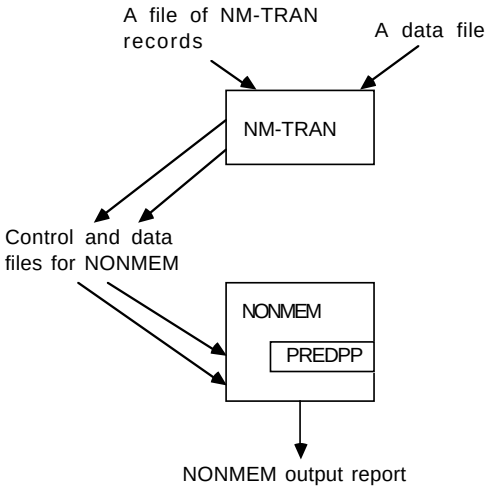

Fig 1.1. NONMEM, PREDPP, and NM-TRAN. A user-written PRED subroutine could be included instead of PREDPP.

2.4. Scope of this Introductory Guide

This Guide is intended to be read by new users of NONMEM-PREDPP. Typically, such users have pharmacokinetic data, either from a population or from a single individual†, to be fit to a standard pharmacokinetic model (e.g., a one or two compartment linear mammillary model). However, new users with nonstandard models, or with pharmacokinetic/pharmacodynamic data, may also find this guide helpful.

It is assumed that NONMEM and its components are already installed on the user's computer and that the user wants to learn to use them as quickly as possible. This guide does not tell how to perform the installation or how to run an installed NONMEM under a particular operating system; the new user will have to ask experienced users what the local commands are. However, someone who is installing NONMEM at a new site may find it useful to review this guide to get a quick overview of NONMEM, its component programs, its inputs, and its outputs.

This guide is not a text book in pharmacokinetics or statistics. Readers should be familiar with basic concepts in pharmacokinetics and statistical data analysis. We also assume a

† The terms "population" and "single individual" are used in this guide. NM-TRAN and NONMEM outputs refer to POPULATION and SINGLE-SUBJECT data and analysis.

very basic familiarity with FORTRAN.

2.5. Contents of this Introductory Guide

Chapter 2 contains two examples of the use of NONMEM. The first presents data from a single individual; estimates are obtained of his pharmacokinetic parameters. The second presents data from a group of individuals; estimates are obtained of the pharmacokinetic parameters of the population which this group represents. The examples serve to introduce NONMEM notation, input and output, and to provide an idea of what is possible using the system.

Chapter 3 presents the notation and definitions we will use to discuss models for individual data. The relationship of these models to data is discussed, and the distinction between so-called fixed effects and random effects is made.

Chapter 4 extends this discussion to models for population data.

Chapter 5 discusses NONMEM's fitting criterion, the parameter estimates obtained by using this criterion, and the standard errors of these estimates. It then discusses how to do hypothesis tests with NONMEM.

Chapter 6 tells how to create data files for NONMEM-PREDPP and how to describe them using the $DATA and $INPUT records of NM-TRAN. It also discusses the Data Preprocessor feature of NM-TRAN.

Chapter 7 tells how to use NM-TRAN to write simple $SUBROUTINE records for PREDPP, how to write $PK records for individual data, and how to write $PK records for population data.

Chapter 8 tells how to write simple $ERROR records for PREDPP. Chapters 7 and 8 are meant to be read in parallel with Chapters 3 and 4.

Chapter 9 tells how to use NM-TRAN to specify the remaining choices for an analysis. It tells how to assign initial values to parameters ($THETA, $OMEGA, $SIGMA records), how to specify what analysis tasks to perform ($ESTIMATION, $COVARIANCE records), and how to specify certain additional output ($TABLE, $SCATTERPLOT records).

Chapter 10 describes NONMEM's output in detail.

Chapter 11 outlines the process of model building, showing how a simple model can be made more complex to better fit the data.

Chapter 12 briefly describes a variety of features of PREDPP and NONMEM that are somewhat advanced for this text but are of interest to most users of NONMEM. References are given to other documents in which additional information can be found.

Chapter 13 discusses errors that can occur during a NONMEM run.

Appendix 1 describes PREDPP's most commonly used pharmacokinetic models (ADVAN subroutines).

Appendix 2 describes alternative parameterizations (TRANS subroutines) for these models.

Appendix 3 is a list of NM-TRAN records.

2.6. How to Read this Guide

Readers who are completely new to NONMEM should read this guide starting with Chapter 2; the examples presented are used again in the later chapters. Chapters 2-5 are theoretical in nature. Chapters 6-12 describe the details of building the input for

NONMEM-PREDPP and interpreting the output. Readers who have non-pharmacokinetic data to fit can skip (or skim) Chapters 3, 4, 7, and 8. Readers who already have some familiarity with certain topics (e.g., who have used other nonlinear analysis programs to analyze data) can concentrate on the chapters of interest to them. We strongly recommend that all users "graduate" to the more thorough NONMEM documentation listed in Section 4 of this chapter.

Throughout the guide, examples are given of NM-TRAN records. These examples appear in boldface:

```
$THETA .01
```

Examples are also given of (fragments) of input data files. They appear as follows:

```
ID      AMT         TIME        DV
 2      320.        0.         0.
 2        0.         .27       1.71
```

Alphabetic characters such as ID, AMT, etc., are shown for descriptive purposes. They are *not* part of the actual data file.

3. A Brief Technical Overview

In this section we discuss the components of NONMEM in some detail. First-time readers may prefer to skip this section and go directly to Chapter 2, which gives an example of a NONMEM run, and return to this section later (if at all).

3.1. NONMEM

NONMEM is written (almost) entirely in ANSI FORTRAN 90/95. It is distributed on CD-ROM as FORTRAN source code, some of which is encrypted. It can be compiled and run on any computer which has a FORTRAN 90/95 compiler and sufficient memory and speed to run a large, computationally intensive program.

NONMEM consists of a main program and many subroutines, all of which are required for each NONMEM run. As discussed above, one subroutine, PRED, is not included in NONMEM itself.

3.2. PREDPP and the PREDPP Library

PREDPP is not a single subroutine. It is a collection of FORTRAN subroutines. Some of these are always needed but must be supplied by the user himself (see PK and ERROR below). Others are always needed and are supplied; these are called the kernel routines. Others (subroutines ADVAN and TRANS, for example) are also always needed, and are supplied, but are chosen from different versions corresponding to different pharmacokinetic models. The collection of supplied routines constitutes the PREDPP Library.

3.3. Subroutines PK and ERROR

Two very important subroutines of PREDPP are called PK and ERROR. PK computes the values of the population or individual pharmacokinetic parameters (e.g., CL and V) of a given model and accounts for the "differences" between individual and population values. ERROR accounts for the "differences" between predicted and observed values. These two subroutines are where the basic task of modelling is carried out; this task is the user’s responsibility. Chapters 7 and 8 are devoted to a description of these subroutines.

Figure 1.2 shows the major components of PREDPP.

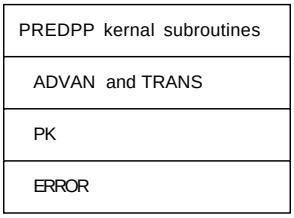

Figure 1.2. Components of PREDPP. ADVAN and TRANS are chosen from the PREDPP library. PK and ERROR are user-supplied.

3.4. Building an Executable Module for NONMEM

Whether PREDPP is used or a special purpose PRED subroutine is written, the PRED subroutine must be combined ("linked") with NONMEM; this process (which is sometimes is called "link editing" or "loading") must take place before the actual NONMEM run. The NONMEM-PRED combination is generally called a "load module" or "executable module". Compiling and linking are processes which are operating system dependent; each installation must supply its own commands and procedures for these tasks. They may be done before the NM-TRAN step or between it and the NONMEM step. This choice is discussed in Section 3.7 below. For certain platforms, a front-end interface provided by the NONMEM Project Group (nmfe74.bat for MS/DOS; nmfe74 C-shell script for Unix-type) can be used to perform these steps, and can create all both types of load modules described below: generated subroutines and user-written subroutines.

3.5. NM-TRAN

NM-TRAN provides the following services: control language translation, model specification via FORTRAN-like statements (called abbreviated code), partial differentiation, and preprocessing of the data. They are discussed separately.

3.6. Control Language Translation

NM-TRAN includes a language for communicating control information to NONMEM. NM-TRAN records are free-form (i.e., spacing between options within a record and the order of the records and their options is flexible) and use English words (or their abbreviations) for options. For example, the record name $ESTIMATION may be abbreviated to $EST; the option name SIGDIGITS may be abbreviated to SIG. Either spaces or commas may be used to separate options. Defaults are understood for most options, allowing the records to be relatively compact. Considerable error checking is performed by NM-TRAN. This reduces the number and severity of the errors that can occur during the NONMEM run. NM-TRAN also produces messages that warn the user of possible errors in the data and/or control stream.

NM-TRAN translates a file of NM-TRAN control records into NONMEM control records, which use a fixed-field, predominately numerical control language.

3.7. Model Specification via Abbreviated Code

With PREDPP, FORTRAN subroutines PK and ERROR are needed to specify parts of the pharmacostatistical model. In most cases, these specifications can be directly expressed within NM-TRAN records $PK and $ERROR, using FORTRAN-like assignment and

conditional statements called abbreviated code. These statements are implemented by NM-TRAN as complete FORTRAN subroutines in file FSUBS, incorporating the abbreviated code. An intermediate step between the NM-TRAN and NONMEM steps is needed to compile these subroutines and link them with NONMEM-PREDPP.

The message "Recompiling certain components" will be displayed at the console at this step.

Figure 1.3 shows how the compile and link step relates to the two steps of Figure 1.1.

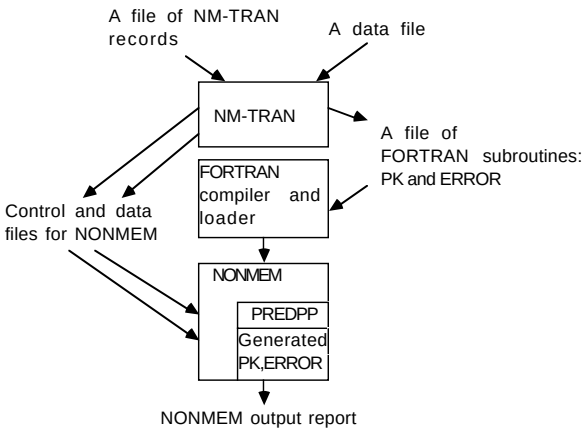

Figure 1.3. Building a NONMEM load module with generated FORTRAN subroutines. An intermediate step is placed between the two steps of Figure 1.1.

If the user supplies complete PK and ERROR subroutines (i.e., $PK and $ERROR records are not used), then the NONMEM load module can be built at any time.

Note that even when PREDPP is not used, the same options exist. For example, if the desired model can be expressed via a $PRED record, then NM-TRAN will generate a complete PRED subroutine. However, whereas NM-TRAN's FORTRAN-like syntax is sufficient for most purposes of writing PK and ERROR subroutines, it is not sufficient for writing any but the simplest of PRED subroutines.

3.8. Partial Differentiation

NONMEM requires that PRED (whether PREDPP or user-written) compute more than just predicted values. It must also compute certain partial derivatives with respect to the random variables η and ε described in Chapters 3 and 4. When $PK, $ERROR, or $PRED records are used, NM-TRAN performs symbolic differentiation to generate the code needed to compute these derivatives. This relieves the user of a major burden.

3.9. Data Preprocessor

NM-TRAN includes a Data Preprocessor program which allows the user greater flexibility in constructing his data file than is allowed in a data file input directly into NONMEM. This is discussed in Chapter 6.

4. Additional Documentation

More information can be found in the other parts of the NONMEM Users Guide, all of which may be found as pdf files on the NONMEM distribution medium.

Part I - Users Basic Guide
A thorough, step by step discussion of the various features and some of the statistical concepts involved in using NONMEM, including many examples.

Part II - Users Supplemental Guide
A continuation of Part I which includes advanced features of NONMEM.

Part III - NONMEM Installation Guide
A guide for installing NONMEM, PREDPP, and NM-TRAN.

Part IV - NM-TRAN Guide
A complete reference guide to NM-TRAN and the Data Preprocessor.

Part V - Introductory Guide
The present document.

Part VI - PREDPP Guide
A complete reference guide to PREDPP.

Part VII - Conditional Estimation Methods
A description of these methods and some guidelines for their use.

Part VIII - Help Guide
A fast way to locate information on a given word or topic. The content of the Help Guide is also supplied on the NONMEM distribution medium as both text files ("on-line help") and html files for on-line searching.

NONMEM V Supplemental Guide
Describes new features of NONMEM V.

Introduction to Version VI
Describes new features of NONMEM VI.

Introduction to NONMEM 7.4.0
Describes new features of NONMEM 7.1, 7.2, 7.3, and 7.4

NONMEM7_Technical_Guide
Technical Guide on the Expectation-Maximization Population Analysis Methods in the NONMEM 7 Program. New with NONMEM 7.2; revised for NONMEM 7.3 and for NONMEM 7.4

useful_variables
New with NONMEM 7.3; revised for NONMEM 7.4 A description of variables that are available via the NM-TRAN include file util\nonmem_general_reserved.

Chapter 2 - NONMEM Examples

1. What This Chapter is About

In this chapter, two examples of the use of NONMEM will be given. The first estimates pharmacokinetic parameters of an individual from his data; the second estimates so-called population parameters from data from a group of individuals. The examples serve to introduce NONMEM notation, input and output, and to provide an idea of what is possible using the system. The second example will be discussed again in Chapter 11.

2. An Individual's Theophylline Kinetics

Figure 2.1 shows the input used to fit a model to observations of theophylline plasma concentration *vs* time in a single individual after a single dose of 320 mg.

```
$PROB  SIMPLE NONLINEAR REGRESSION - THEOPHYLINE
$INPUT ID AMT TIME DV
$DATA  P2DATA
$SUBROUTINE ADVAN2
$PK
 KA=THETA(1)
 K=THETA(2)
 V=THETA(3)
 S2=V
$ERROR
 Y=F+ERR(1)
$THETA  (0, 1.7)  (0, .102)  (0, 29.)
$OMEGA  1.2
$ESTIMATION PRINT=5
$COVARIANCE
$TABLE ID AMT TIME
$SCATTER PRED VS DV  UNIT
```

Figure 2.1. The input (i.e., NM-TRAN control records) for analysis of some individual theophylline data.

The first line (record) gives a name to the problem. The rest of the lines (records) discuss the data, the model, and the desired output. Before going into these in some greater detail, you may want to look right now at figures 2.1 and 2.2, and then 2.4 and 2.5. Figure 2.2 shows the data for this problem, and figures 2.4 and 2.5 show some of NONMEM's output. All you need to know to get a good idea of what this analysis shows is that the one-compartment model with first-order absorption has been used; the observed concentrations and the times of observation after the bolus dose are in columns 4 and 3, respectively, of figure 2.2; and that the symbol DV stands for dependent variable (the observed concentrations, in this case). You should, for example, even at this point, be able to tell that the estimate of Volume of Distribution (V in figure 2.1, and THETA(3) in figure 2.4) is 32 liters (L), with a standard error of ± 1.26 L. Now consider the figures in greater detail.

2.1. The NM-TRAN Control Records

The second record of figure 2.1 names the data items that appear on each data record, and the third record gives the name of the file containing the data records, P2DATA in this example. Figure 2.2 shows the contents of P2DATA.

```
2     320.       0.        0.
2       0.        .27      1.71
2       0.        .52      7.91
2       0.       1.        8.31
2       0.       1.92      8.33
2       0.       3.5       6.85
2       0.       5.02      6.08
2       0.       7.03      5.4
2       0.       9.        4.55
2       0.      12.        3.01
2       0.      24.3        .90
```

Figure 2.2. The contents of the data file containing the data records.

According to the second record of figure 2.1, the third data item (column) of a data record is TIME, the time associated with the event described by that data record. The event at a given time (for this simple data set) can either be the administration of a dose or the acquisition of an observation. The second data item of a data record is AMT, amount (in this case in mg) of the dose given at TIME, the time of the record. Apparently, 320 mg is given at time zero (first record of figure 2.2), and no further doses are given (all zeros in column 2 thereafter). The fourth data item (column) in P2DATA is named DV, for Dependent Variable (the measured plasma theophylline concentration), as already mentioned. So, all of the data records, except the first, give the time after the 320 mg dose, and the concentration of theophylline (in mg/L) measured in a plasma sample drawn at that time. The first data item is labelled ID for the IDentification number of the patient. Here it happens to be 2.

2.2. The Model

The fourth record of figure 2.1 identifies the pharmacokinetic model PREDPP is to use: the one-compartment model with first-order absorption. It is implemented by an ADVAN subroutine (see Chapter 1, Section 3.2) which is called ADVAN2 (See Chapter 7). Figure 2.3 shows the part of the output of NONMEM for this problem that verifies the user's choice of model. It also describes the features of the model in terms of its compartments. Of relevance to this problem are the DEPOT compartment (where the dose goes, and from which drug enters the central compartment by a first order process), and the CENTRAL compartment itself. Note, for example, that the default compartment for doses (i.e., where PREDPP will add doses if not otherwise instructed) is the DEPOT compartment, as it should be.

```
ONE COMPARTMENT MODEL WITH FIRST-ORDER ABSORPTION (ADVAN2)

MAXIMUM NO. OF BASIC PK PARAMETERS:   3

BASIC PK PARAMETERS (AFTER TRANSLATION):
  ELIMINATION RATE (K) IS BASIC PK PARAMETER NO.:  1
  ABSORPTION RATE (KA) IS BASIC PK PARAMETER NO.:  3


COMPARTMENT ATTRIBUTES
COMPT. NO.   FUNCTION   INITIAL    ON/OFF      DOSE      DEFAULT    DEFAULT
                        STATUS     ALLOWED     ALLOWED    FOR DOSE   FOR OBS.
   1         DEPOT        OFF        YES         YES        YES        NO
   2         CENTRAL      ON         NO          YES        NO         YES
   3         OUTPUT       OFF        YES         NO         NO         NO
```

Figure 2.3. The PREDPP output that verifies the user's choice of model. Features of the model are discussed, such as the names and numbering of parameters, and the attributes of the various compartments in the model.

The fifth input record (figure 2.1) signals the start of the user's specification of the model for the pharmacokinetic parameters. This specification is given in the next 4 lines of so-called abbreviated code (the $PK record, along with this abbreviated code is called the $PK block). Some of the parameters that NONMEM estimates are denoted by θ herein, and are labeled THETA in NONMEM input and output. The model specified in figure 2.1 is very simple. It says that a different unknown constant (NONMEM parameter) is to be assigned to each pharmacokinetic parameter: first-order absorption rate, KA (line 1 of the PK block, after the $PK record - THETA(1)), rate constant of elimination, K (line 2 - THETA(2)), and volume of distribution, V (line 3 - THETA(3)). The S2 parameter (a so-called scale parameter) is discussed in Chapter 3, Section 2.2.

The sixth input record (figure 2.1 - 11^{th} line) signals the start of the user's specification of the (statistical) model for the lack of fit of the pharmacokinetic model to the data. This specification is given in the next line of abbreviated code (the $ERROR record, along with this line of abbreviated code is called the $ERROR block). The model here says that observations differ from predictions by an additive error (ERR(1)).

Record 7 ($THETA) gives NONMEM information about possible values of each element of θ in the format: lower bound, initial estimate, upper bound. When, as in this particular record, only two numbers are given for an element of θ, these are understood to mean the lower bound and initial estimate; the upper bound is unlimited. Record 8 ($OMEGA) gives NONMEM an initial estimate of the variance of ERR(1). This statistical parameter is often denoted by σ^2 in statistical discussions, but with data from a single individual, it is denoted by Ω in NONMEM documentation, and by OMEGA in NONMEM input and output. It is understood that a variance parameter is always nonnegative. The $OMEGA record is further discussed in Chapter 9, Section 3.

2.3. The Output

Record 9 ($ESTIMATION) instructs NONMEM to obtain estimates of the parameters, and the next record ($COVARIANCE) asks that it also obtain standard errors of the parameter estimates. The output is shown in figure 2.4. It requires little discussion. The first item, the minimum value of the objective function, is a goodness of fit statistic, much like a sum of squares (and as with a sum of squares, the lower the value, the better the fit). The parameter estimates (the parameter values at which the objective function is minimized) and their standard errors follow. Note that the estimate of OMEGA, too, has a standard error. Unlike most fitting programs, NONMEM treats this parameter like any

other.

```
********************************************************************************************************************************
********************                                                                                        ********************
********************                          MINIMUM VALUE OF OBJECTIVE FUNCTION                           ********************
********************                                                                                        ********************
********************************************************************************************************************************

************************************************          8.940     ************************************************

********************************************************************************************************************************
********************                                                                                        ********************
********************                                  FINAL PARAMETER ESTIMATE                              ********************
********************                                                                                        ********************
********************************************************************************************************************************

THETA - VECTOR OF FIXED EFFECTS  **********

           TH 1      TH 2      TH 3

         1.94E+00  1.02E-01  3.20E+01

OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********

           ETA1

ETA1     8.99E-01

********************************************************************************************************************************
********************                                                                                        ********************
********************                                STANDARD ERROR OF ESTIMATE                              ********************
********************                                                                                        ********************
********************************************************************************************************************************

THETA - VECTOR OF FIXED EFFECTS  **********

           TH 1      TH 2      TH 3

         6.28E-01  7.35E-03  1.25E+00

OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********

           ETA1

ETA1     5.44E-01
```

Figure 2.4. NONMEM output giving the goodness of fit statistic (the minimum value of the objective function) the parameter estimates, and their standard errors.

The next to last control record asks NONMEM to print a table displaying the input data and certain computed quantities. A portion of a NONMEM table is shown in figure 10.10 of Chapter 10. The last control record asks NONMEM to make a scatterplot of the prediction of each plasma concentration (PRED) VS the observed value (DV) and to draw the line of identity (UNIT, for "unit slope" line) on the plot. The plot is shown in figure 2.5.

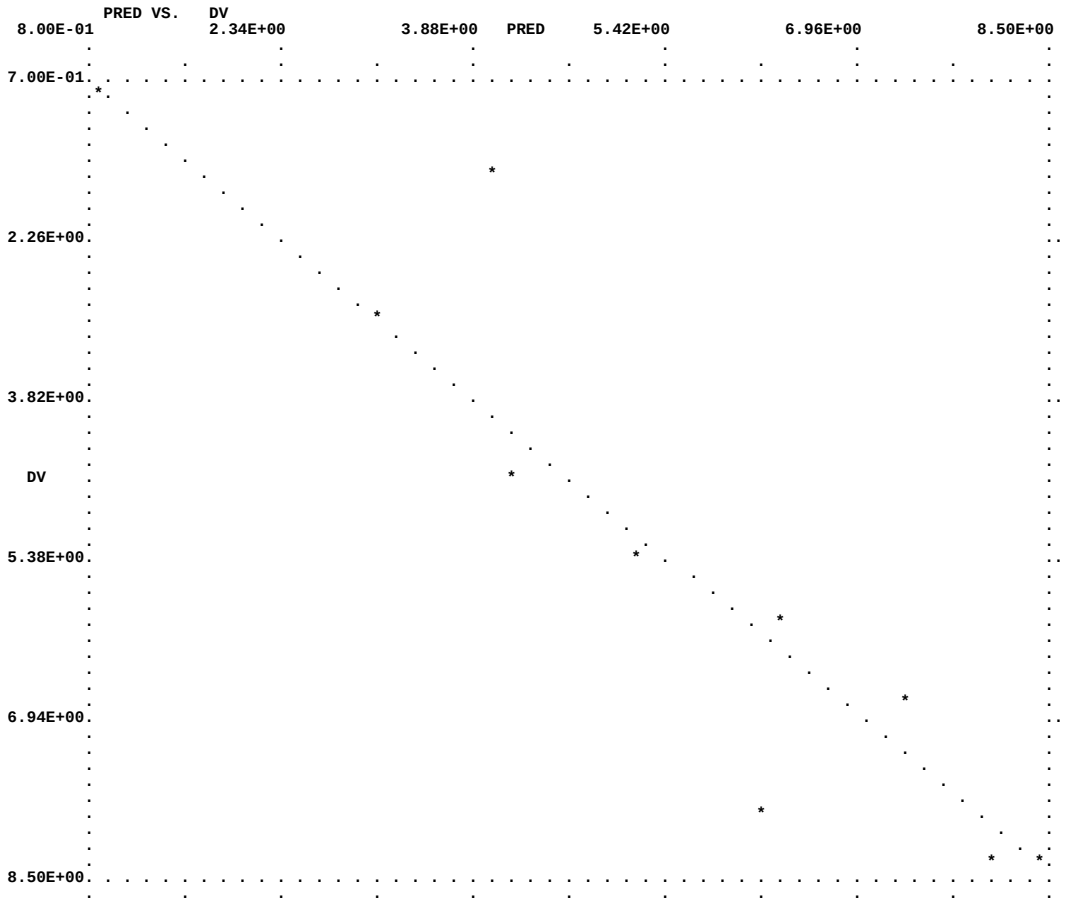

Figure 2.5. A scatterplot of the observed data (DV) vs the predictions of the best-fitting model parameters (PRED). The line of identity (intercept = 0; slope = 1) is drawn. If all points fell on that line, the fit would be perfect.

3. A Population Model for Phenobarbital

About 60 infants were given phenobarbital therapeutically. A plasma concentration was measured in each some hours after the first (loading) dose, followed by multiple maintenance doses. A second, and sometimes a third concentration were measured later. In all, 155 concentrations were observed. Figure 2.6 gives the NM-TRAN control records. The data are too lengthy to show in full, but figure 2.7 shows the data for the first individual†. Figures 2.8 - 2.10 have some relevant output. Again, most of the analysis results are apparent from the figures, and you should try to see if you can figure them out before going further. Note that the $INPUT record now defines a new data item, WT, the patient weight. It's value is given on every data record for an individual, in the column indicated. This is so despite the fact that WT may not change within an individual. This is a bit repetitious, but convenient.

† File PHENO of NONMEM distribution medium contains the complete data set.

```
$PROBLEM PHENOBARB
$INPUT   ID TIME AMT WT APGR DV
$DATA    PHENO
$SUBROUTINE ADVAN1
$PK
   TVCL=THETA(1)
   CL=TVCL+ETA(1)
   TVVD=THETA(2)
   V=TVVD+ETA(2)
                    ; THE FOLLOWING ARE REQUIRED BY PREDPP
   K=CL/V
   S1=V
$ERROR
   Y=F+ERR(1)
$THETA  (0,.0047) (0,.99)
$OMEGA  .0000055, .04
$SIGMA  25
$ESTIMATION  PRINT=5
$TABLE     ID TIME AMT WT APGR
$COVARIANCE
$SCATTER   PRED VS DV  UNIT
$SCATTER   RES  VS WT
```

Figure 2.6. NM-TRAN control records for analysis of some population phenobarbital data.

```
1    0.    25.0    1.4      7      .
1    2.0      .    1.4      7   17.3
1   12.5    3.5    1.4      7      .
1   24.5    3.5    1.4      7      .
1   37.0    3.5    1.4      7      .
1   48.0    3.5    1.4      7      .
1   60.5    3.5    1.4      7      .
1   72.5    3.5    1.4      7      .
1   85.3    3.5    1.4      7      .
1   96.5    3.5    1.4      7      .
1  108.5    3.5    1.4      7      .
1  112.5      .    1.4      7   31.0
```

Figure 2.7. The first individual's phenobarbital data.

3.1. The NM-TRAN Control Records

The records are very similar to those for the theophylline problem. The new features are that the model has changed (it is implemented by ADVAN1, not ADVAN2), the model for the pharmacokinetic parameters is more complicated, and an additional scatterplot is requested. The data for each infant is similar to those shown in figure 2.7; however, now all of the data records for each infant start with the *same* value for the ID data item (column 1), but this value differs *between* infants.

3.2. The Model

ADVAN1 implements the one-compartment (monoexponential) model, without first order absorption. No absorption model was needed for this problem because all concentrations were measured many hours after the last (oral) dose, so absorption could be considered to be complete, and, for the purposes of data analysis, immediate.

The parameters of the one-compartment model are defined by the abbreviated code following the $PK statement: Clearance (CL) and Volume of Distribution (V). However, here each parameter is not simply equal to one of NONMEM's parameters (an element of THETA). Rather, CL, for example, is equal to a parameter (THETA(1)) plus a new term,

ETA(1). The latter expresses interindividual variability, and stands for the deviation of the individual's true clearance (CL) from the population value (TVCL, Typical Value of CLearance, which, in turn, is simply THETA(1)). This model is essentially different from the theophylline model, because it incorporates interindividual variability (something that an individual model need not do). Note that since PREDPP ultimately needs the values of microconstants, rather than physiological-based pharmacokinetic parameters such as clearance, code must be given for K, the rate constant of elimination. There is, though, a simple alternative to writing this additional line of code. It is discussed in Chapter 3 Section 2.1.

The abbreviated code after the $ERROR record is exactly the same as that with the theophylline data and expresses the same model for lack-of fit between observations and predictions for an individual. The $OMEGA and $SIGMA records, which give NONMEM information about the estimated variances of the ETA and ERR variables, are discussed in Chapter 9, Section 3. Previously the initial estimate of the variance of ERR(1) was given on a $OMEGA record. Here it is given on a $SIGMA record. This difference in NONMEM conventions between individual type data and population type data will be discussed more fully in Chapters 3 and 4.

3.3. The Output

NONMEM is again instructed to estimate the parameters and their standard errors. The results are shown in figures 2.8 and 2.9.

```
********************************************************************************************************************************
********************                                                                                        ********************
********************                            MINIMUM VALUE OF OBJECTIVE FUNCTION                         ********************
********************                                                                                        ********************
********************************************************************************************************************************

************************************************       717.203     ************************************************

********************************************************************************************************************************
********************                                                                                        ********************
********************                                  FINAL PARAMETER ESTIMATE                              ********************
********************                                                                                        ********************
********************************************************************************************************************************

THETA - VECTOR OF FIXED EFFECTS  ********************

           TH 1      TH 2

         5.48E-03  1.40E+00

OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********

           ETA1      ETA2

ETA1     6.85E-06

ETA2     0.00E+00  2.86E-01

SIGMA - COV MATRIX FOR RANDOM EFFECTS - EPSILONS  ****

           EPS1

EPS1     8.01E+00
```

Figure 2.8. NONMEM output giving the goodness of fit statistic (the minimum value of the objective function) and the parameter estimates for the phenobarbital problem.

```
********************************************************************************************************************************
********************                                                                                        ********************
********************                            STANDARD ERROR OF ESTIMATE                                  ********************
********************                                                                                        ********************
********************************************************************************************************************************

THETA - VECTOR OF FIXED EFFECTS  **********************

           TH 1      TH 2

        4.86E-04  7.84E-02

OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********

           ETA1      ETA2

ETA1    2.27E-06

ETA2   .........  8.34E-02

SIGMA - COV MATRIX FOR RANDOM EFFECTS - EPSILONS  ****

           EPS1

EPS1    1.49E+00
```

Figure 2.9. NONMEM output giving the standard errors of the parameter estimates for the phenobarbital problem.

Note that now there are estimates of the variances of the interindividual differences in CL (OMEGA - ETA1) and V (OMEGA - ETA2), as well as of the residual error variance (denoted by SIGMA in NONMEM output when the data are from a population; again, see Chapters 3 and 4). There are also standard errors for these estimates.

The next-to-last control record asks NONMEM to make the same kind of scatterplot as in the theophylline problem: a plot of the predictions vs the observations. Here, a prediction for an individual's observation is based on typical (population) values of the pharmacokinetic parameters (see figure 2.8), rather than the values of the pharmacokinetic parameters for the specific individual. The plot is shown in figure 2.10.

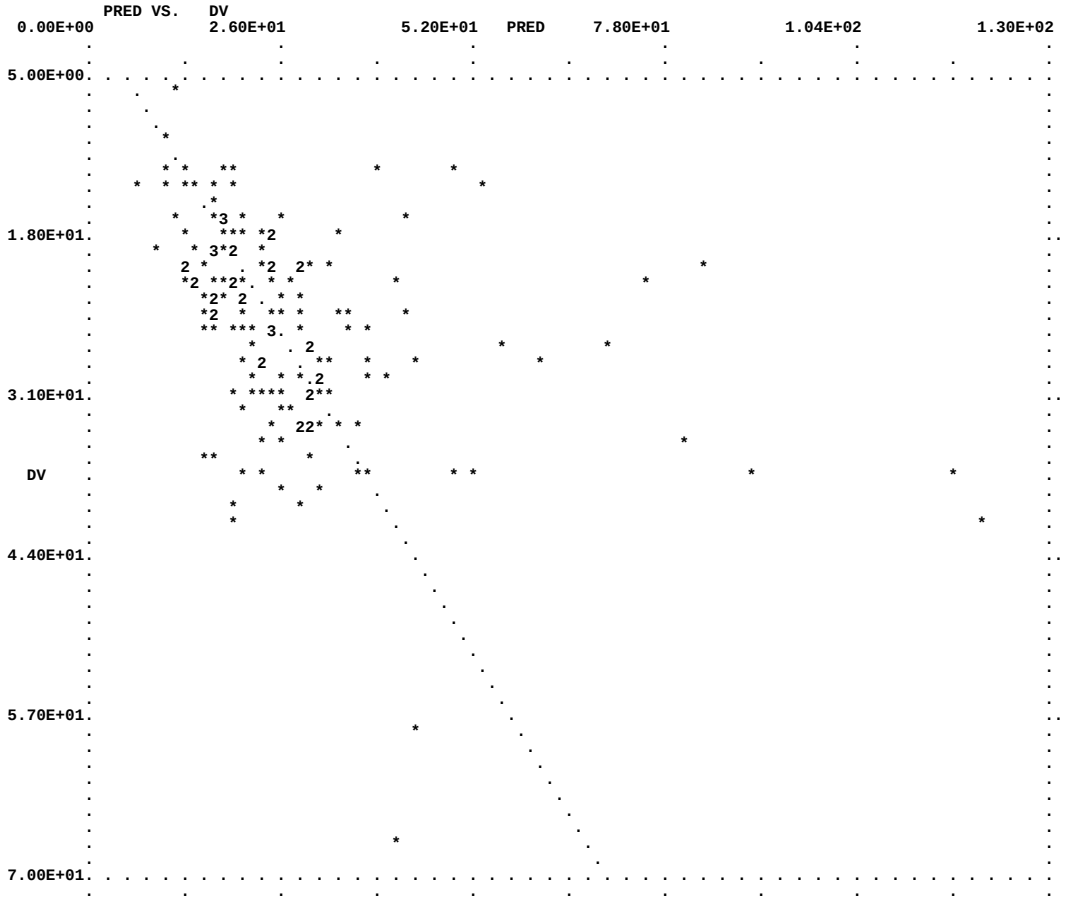

Figure 2.10. A scatterplot of the observed data (DV) vs the predictions with the best fitting model parameters (PRED). The line of identity (intercept = 0; slope = 1) is drawn. If all points fell on that line, the fit would be perfect. Here, in contrast to figure 2.5, the data arise from many different individuals. One cannot tell which data came from which infant.

Although the fit is fairly good, the points far to the right of the line of identity of figure 2.10 indicate that there are many predictions (PRED) that are much higher than their corresponding observations (DV). This is seen from another point of view in the second scatterplot. This scatterplot plots residuals (RES) vs patient weight (from the data item, WT — see figure 2.6). A residual is the difference between an observed concentration and its prediction (the same prediction used in the scatterplot of figure 2.10). The residuals reflect not only lack of fit between observations and predictions for a given individual (the variance SIGMA), but also interindividual variability (the variances comprising OMEGA). They can be thought of as reflecting the part of the data that the model does not explain. As can be seen from figure 2.11, there is a clear relationship between the sign and magnitude of the residuals and patient weight. Here, the patients with the largest weights have the most negative residuals; i.e., their predictions are much larger than their observations. These are the same points that fell on the far right of figure 2.10.

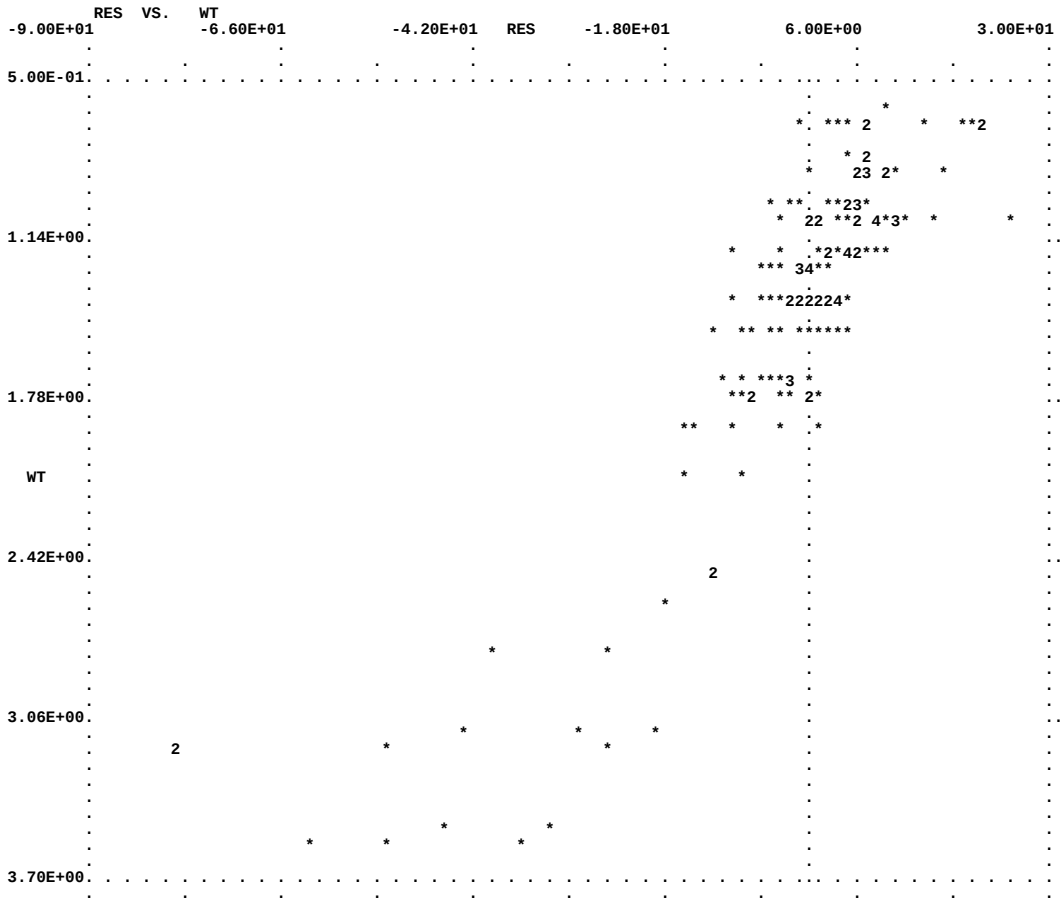

Figure 2.11. A scatterplot of the residuals (RES) vs patient weight (WT). The pattern suggests that observations are underpredicted in infants with low weight, and overpredicted in those with higher weights.

An obvious explanation is that Clearance or Volume, or both, increase with weight, so that without weight being taken into account, too high a prediction is being made for a larger infant and too low a prediction is being made for a smaller infant, all other things (i.e., dose) being equal. To see if accounting for weight improves the fit, the run specified in figure 2.12 can be done.

```
$PROBLEM PHENOBARB WITH WEIGHT IN MODELS FOR CL AND V
$INPUT   ID TIME AMT WT APGR DV
$DATA    PHENO
$SUBROUTINE ADVAN1
$PK
   TVCL=THETA(1)+THETA(3)*WT
   CL=TVCL+ETA(1)
   TVVD=THETA(2)+THETA(4)*WT
   V=TVVD+ETA(2)
                        ; THE FOLLOWING ARE REQUIRED BY PREDPP
   K=CL/V
   S1=V
$ERROR
   Y=F+ERR(1)
$THETA  (0,.0027) (0,.70) .0018  .5
$OMEGA  .000007, .3
$SIGMA   8
$ESTIMATION PRINT=5
$COVARIANCE
$TABLE ID TIME AMT WT APGR DV
$SCATTER   PRED VS DV  UNIT
$SCATTER   RES  VS WT
```

Figure 2.12. NM-TRAN control records for fitting a model taking into account the effect of patient weight to the population phenobarbital data.

Now both TVCL and TVVD are linear functions of weight with, in the case of TVCL, for example, intercept THETA(1), and slope THETA(3). Both slope and intercept are "population" parameters since they relate weight to typical population values of the pharmacokinetic parameter. Now we see why WT is given in every data record: the abbreviated PK code may need to be evaluated at each event time. If WT did not change over time within any patient, it could be given only on the first data record for each patient, but then slightly more complicated abbreviated code would be needed. The output from running the input of figure 2.12 is shown in figures 2.13 - 2.16.

```
********************************************************************************************************************************
********************                                                                                        ********************
********************                          MINIMUM VALUE OF OBJECTIVE FUNCTION                           ********************
********************                                                                                        ********************
********************************************************************************************************************************
**************************************************        609.134     **************************************************

********************************************************************************************************************************
********************                                                                                        ********************
********************                                  FINAL PARAMETER ESTIMATE                              ********************
********************                                                                                        ********************
********************************************************************************************************************************

THETA - VECTOR OF FIXED EFFECTS  ********************

           TH 1      TH 2      TH 3      TH 4

         1.43E-11  1.21E-01  4.77E-03  9.18E-01

OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********

           ETA1      ETA2

ETA1     1.36E-06

ETA2     0.00E+00  7.51E-02

SIGMA - COV MATRIX FOR RANDOM EFFECTS - EPSILONS  ****

           EPS1

EPS1     8.71E+00
```

Figure 2.13. The minimum objective function value and parameter estimates for the phenobarbital data, using the model of figure 2.12, which takes into account the effect of patient weight.

```
********************************************************************************************************************************
********************                                                                                    ********************
********************                            STANDARD ERROR OF ESTIMATE                              ********************
********************                                                                                    ********************
********************************************************************************************************************************

THETA - VECTOR OF FIXED EFFECTS  **********************

           TH 1      TH 2      TH 3      TH 4

        9.49E-11  1.46E-01  2.24E-04  1.13E-01

OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********

           ETA1      ETA2

ETA1    7.24E-07

ETA2   .........  3.63E-02

SIGMA - COV MATRIX FOR RANDOM EFFECTS - EPSILONS  ****

           EPS1

EPS1    1.71E+00
```

Figure 2.14. The standard errors of the parameter estimates for the phenobarbital data, using the model of figure 2.12, which takes into account the effect of patient weight.

Note the improvement in the minimum objective function value (it drops 108 points), and the profound decreases in the sizes of the estimates of the interindividual variances; now that weight is in the model, there is less unexplained interindividual variability. As will be discussed in Chapter 5, the decrease in the objective function can be used for a formal hypothesis test of the appropriateness of the new model (figure 2.12) for the effect of weight on the pharmacokinetic parameters.

Note also the very small values estimated for THETA(1) and for its standard error. The intercept term of TVCL does not appear to be an important part of the model. This model is refined in Chapter 10, Section 6.2, where it is seen that deleting THETA(1) and THETA(3) produces a model that fits as well as the model including them.

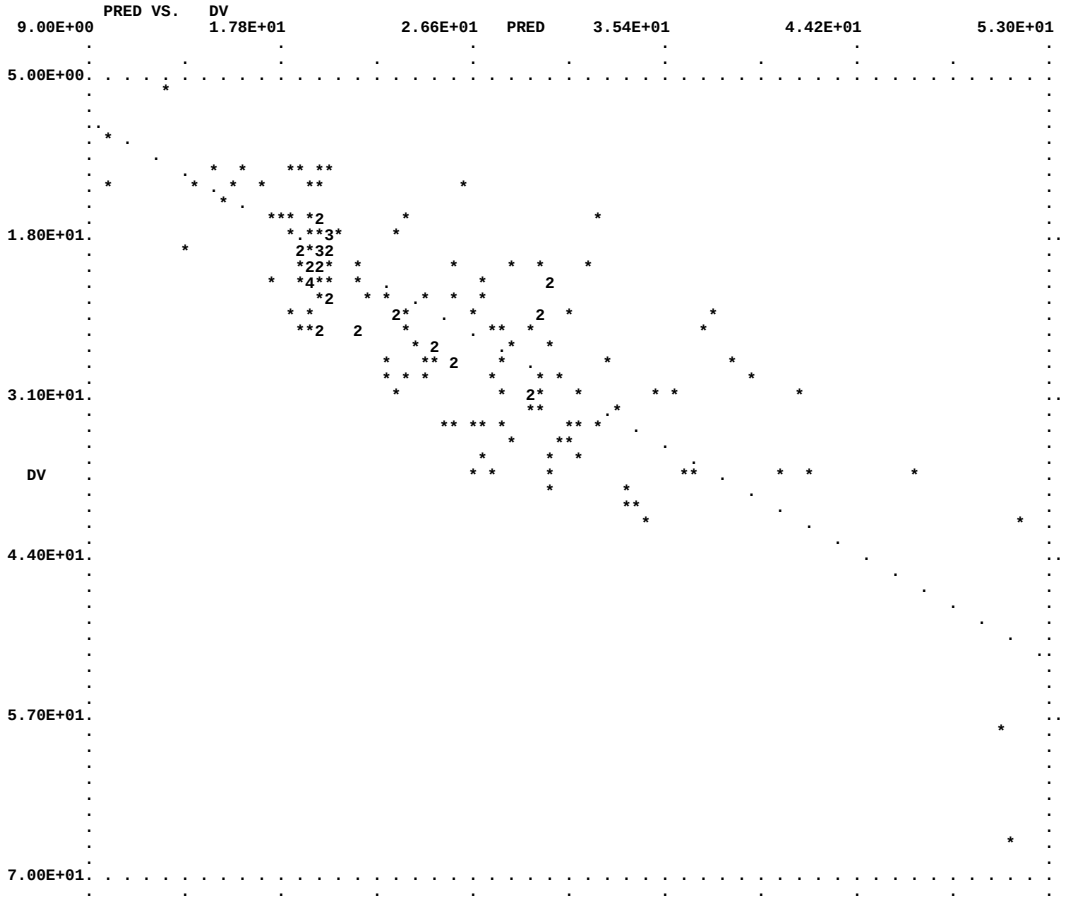

Figure 2.15. A scatterplot of predictions vs observations for the phenobarbital data, using the model of figure 2.12, which takes into account the effect of patient weight. Compare to figure 2.10.

The scatterplots (figures 2.15 and 2.16) confirm that the new model is an improvement: the group of points far to the right of the line of identity have disappeared from the PRED vs DV plot, and the plot of residuals vs weight no longer shows a pattern.

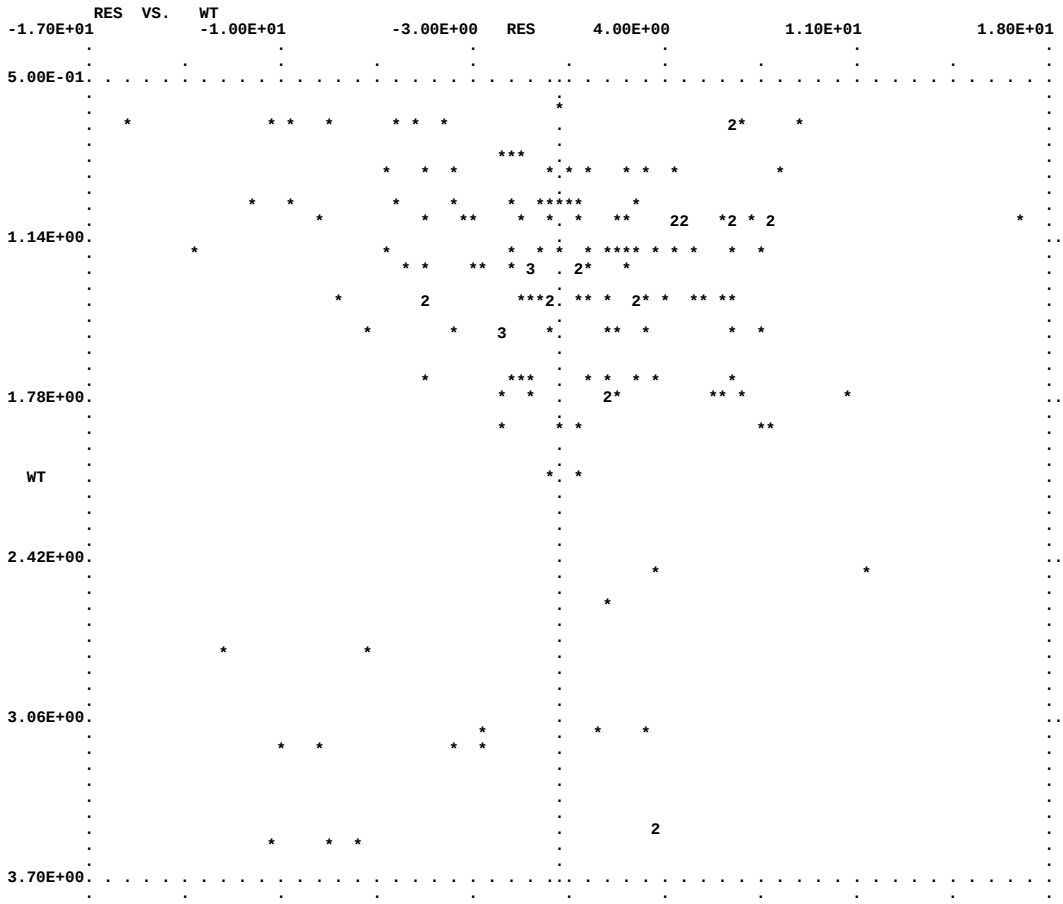

Figure 2.16. A scatterplot of residuals (RES) vs patient weight (WT) for the phenobarbital data, using the model of figure 2.12, which takes into account the effect of patient weight. Compare to figure 2.11.

4. Overview

The examples in this chapter illustrate some of the most important and useful features of NONMEM.

- NONMEM can fit both individual and population models.
- NONMEM has a menu of pharmacokinetic models from which the one appropriate to the problem at hand can be chosen.
- The user specifies the relationship of pharmacokinetic parameters to independent variables (such as WT in the phenobarbital example), using "population" parameters that will be estimated.
- The user also specifies which parameters vary between individuals, and the form (model) for this variability, as well as the form (model) for the differences between observations from an individual and their predictions for this individual.
- NONMEM estimates parameters describing both kinds of variability.
- NONMEM provides estimates (standard errors) of the precision of its parameter estimates, including those describing variability.
- NONMEM provides a means of deciding whether one model (e.g., that including weight's effect on CL and V) fits the data better than another using the minimum

objective function value, a goodness-of-fit statistic.

° NONMEM provides (limited) graphics, useful in judging the adequacy of the model currently fit to the data.

Chapter 3 - Models for Individual Data

1. What This Chapter is About

In this chapter, the notation and definitions we will use to discuss models for individual data will be presented. The relationship of these models to data will be discussed, and a distinction between pharmacokinetic structural models (that describe the underlying shape and form of the data) and statistical error models (that describe the "errors" or differences between observations and structural model predictions) will be made. Several error models will be discussed, as will a useful modelling device, the indicator variable.

2. Pharmacokinetic Structural Models for Individual Data

By individual data we usually mean data from a single individual (animal or human). One could also be concerned with data comprised of a pharmacokinetic response at just one time point from each of a number of individuals. Call this type of data single-response population data. This name comes from the fact that data such as these can, of course, be regarded as a particular instance of the more general data type, population data; i.e., data comprised of one *or more* pharmacokinetic responses at different time points from a number of individuals sampled from a population. Although one can discuss the treatment of single-response population data as population data, they are often treated just as are individual data.

A simple pharmacokinetic model for data from a single individual is the monoexponential ("one-compartment") model:

$$A_j = De^{-kt_j} \tag{3.1}$$

This model describes the typical time course of amount of drug in the body (A), as a function of initial dose (D), time (t), and a *parameter*, k. As we may be interested in A at several possible times, we explicitly note this by the subscript j which indexes a list of times, $t_0, t_1, \ldots, t_j, \ldots, t_n$.

A way to write a generic form for a structural model, omitting details of its structure, is

$$y_j = f(x_j, \phi) \tag{3.2}$$

where y stands for some "response" (dependent variable) of interest (A in (3.1)), the symbol f stands for the unspecified form of the model (a monoexponential such as in (3.1)), which is a function of known quantities, x (t_j and D in (3.1)), and *parameters*, ϕ (k in (3.1)). The quantities in x are *known*, because they are either measured or controlled, and therefore, are called fixed effects, in contrast to effects which are not known and are regarded as random (see below). The parameters in the parameter vector ϕ are called fixed effect parameters because they quantify the influence of the fixed effects on the dependent variable. Each one of an individual's pharmacokinetic parameters is a particular type of fixed effect parameter. With NONMEM, parameters comprising θ are (usually) fixed effect parameters, but these may or may not be an individual's pharmacokinetic parameters (contrast figures 2.1 and 2.6). Here we shall use the symbol ϕ for the parameter vector comprised specifically of an individual's pharmacokinetic parameters (although there will be some exception to this).

Aside from the fact that the values given by a structural model are usually not the values observed due to measurement error or model misspecification, an amount of drug (A of (3.1)) is usually not itself observable. Instead, we may observe a concentration (C) of drug. We need an "observation scaling" model to describe the relationship between A

and C. This might be

$$\tilde{C}_j = \frac{A_j}{V} \tag{3.3}$$

where V is another parameter, Volume of Distribution. (We denote the concentration in model (3.3) by the symbol $\tilde{C}$, to distinguish it, the model-predicted value, from the actually observed value, C. This will soon be discussed further.) PREDPP assumes that there is always an observation scaling model like (3.3) that relates an amount of drug (in some compartment of the body) to the observation, and therefore always expects a parameter, *Sn* that scales (i.e. divides) the predicted amount in the n^{th} compartment. In the example above, $S1$ is simply V. In other examples, to be discussed later, *Sn* can be more complicated. If a value for *Sn* is not specified, it is taken to be 1. For the rest of this discussion, it is convenient to assume that ϕ itself includes a scaling parameter (if such is needed, and even though such a parameter is not usually regarded as one of an individual's pharmacokinetic parameters) and that f actually includes observational scaling. Note, considering the example of (3.3), that $x \equiv (D, t)$, and $\phi \equiv (k, V)$. Thus x and ϕ of (3.1) are in general lists of things (vectors), not single things (scalars).

PREDPP implements a number of pharmacokinetic models, such as the one-compartment model (3.1), (3.3). These will be discussed more fully in Chapter 7. There is no need for further general discussion of kinetic models, as we assume the readers of this document are familiar with pharmacokinetics. However, two modelling features deserve further comment, alternative parameterizations and the special parameter, *Sn* .

2.1. Alternative Parameterizations

Recall the phenobarbital example of Chapter 2. For the first run, the input contained, among other things, some lines of code defining the variables *CL* and *V*, and then the line

```
K = CL/V
```

This line was needed because PREDPP expects the one-compartment model to be parameterized using the parameter *K*, the rate constant of elimination, not clearance and volume of distribution. However, we chose to estimate typical population values for *CL* and *V*, so we had to relate these parameters to THETA and then relate *K* to *CL* and *V*. This is an example of reparametrization of a model so that the pharmacokinetic parameters used are those of primary interest to the modeler. In fact, we may use any parameterization we wish, so long as we are willing to include the reparameterization line(s) that translate our parameters into those expected by PREDPP. (Chapter 7 discusses the parameters PREDPP expects for the various models it implements.) However, there is a program called TRANS which automatically does this translation. Different versions of TRANS exist in the PREDPP Library and correspond to translations of different parameterizations into that expected by PREDPP.

2.2. The Scale Parameter, S

Usually, observations are concentrations. So, as in model (3.3), *S* will usually be set identical to *V*. However, *S* is not always simply *V*. Some examples should clarify this point. (In the discussion below, we avoid the notation *Sn*, and use *S*, to refer to the scale term for the amount in the compartment in which concentrations are being measured.)

2.2.1. S Depends on a Known Constant

This almost trivial case occurs when one wishes to match the units of predicted responses to those of the data. For example, suppose *D* is in milligrams, but concentrations are in

ng/ml. If no scaling is done, the units of V will be kiloliters (i.e., V=1 corresponds to V=1000 liters). To avoid this, one might choose the model

$$S = V/1000$$

thereby converting the units of A into micrograms, and since mcg/L ≡ ng/ml, the units of V become liters. Of course, one could recode one's data, dividing all concentrations by 1000 (or multiplying the dose by 1000) and avoid this, but that may not be convenient.

2.2.2. S Depends on a Parameter

Later in this chapter we will discuss a model used when the data arise from two different assays (call them assay 1 and assay 2). In such a case, there may be a systematic (multiplicative) bias of one assay relative to the other. If we wish to allow for this possibility, we might need a model such as

$$S = \begin{cases} V, & \text{if } \textit{assay is } 1 \\ hV, & \text{if } \textit{assay is } 2 \end{cases}$$

where h is a new parameter that measures the proportional bias of the assays (i.e., bias causes the apparent volume of distribution to be different for data from the two assays). The parameter h is not really a pharmacokinetic parameter, but for the purpose of this discussion it can be included in ϕ.

2.2.3. S Depends on an Element of x

Later in this chapter we will describe a model useful when two kinds of responses are measured, plasma and urine concentrations. In the case of urine concentrations, the predicted total drug in the urine during a time period (available from an "output" compartment present in all models implemented by PREDPP; see Chapter 7) would have to be scaled by the actual urine volume during that time period. This volume would be an element of x, and S would be set equal to it.

3. Statistical Model for an Individual's Observations

One does not, in fact, ever observe the predicted plasma concentration (or any other predicted response). What one observes is a measured value which differs from the predicted value by some (usually small) amount called a residual error (also called intra-individual error). We regard this error as a random quantity (see below). We will want NONMEM to fit our model to our data, and in so doing, provide us with estimates of the model parameters. The way NONMEM's fit follows the data is determined largely by what we tell it about the nature of the errors (see Chapter 5). We must therefore provide NONMEM with another model, an error model.

There are many reasons that the actual observation may not correspond to the predicted value (e.g. $\tilde{C}$ as given by the right side of (3.3)) The structural model may only be approximate, or the quantities in x may have been measured with error, or, as is always true, pharmacokinetic responses may be measured with some error (assay error). It is too difficult to model all these sources of error separately, so we usually make the simplifying assumption that each difference between an observation and its *prediction* (i.e. each error) is a randomly occurring number. When the data are from a single individual, and the error model is the Additive error model (see Section 3.1, below), the error is denoted by η herein, by ETA in NONMEM output, and by ETA or ERR in NM-TRAN input. (When data are from a population, and the same error model is used, this error will be denoted ε; see Chapter 4.) Therefore a model for the jth observation, y_j, could be written

$$y_j = f(x_j, \phi) + \eta_j \tag{3.4}$$

Implicit in using the symbol η in this way is the assumption that all residual errors come from probability distributions with mean zero and the same (usually unknown) variance. (The error variance is the mean *squared* error.) More complicated error models involving η can be written (see below). A schematic of model (3.4) is shown for the structural model of (3.1), (3.3) in figure 3.1. Because this model describes the influence of both fixed effects (x_j) *and* random effects (η_j), it is called a Mixed Effects Model (hence the name, NONMEM: *NON*linear *M*ixed *E*ffects *M*odel). Mixed effects models, in general, may have more than one random effect, and more than one type of random effect (Chapter 4); (3.4) is only a particularly simple example.

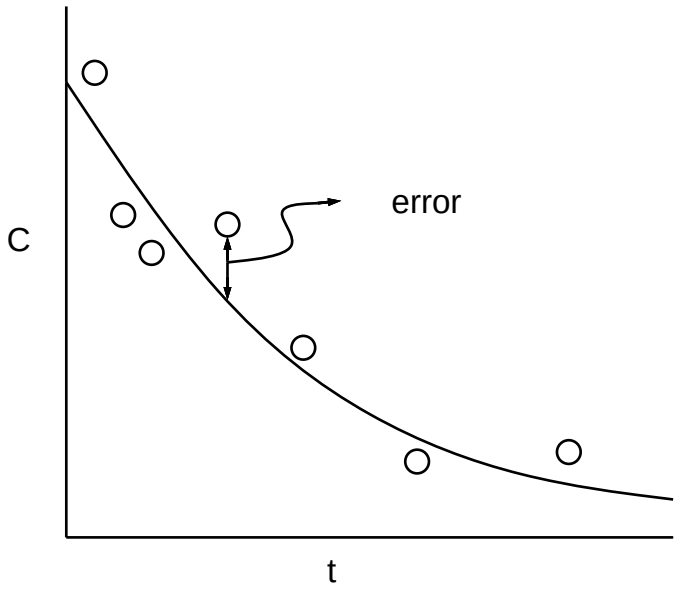

Figure 3.1. C vs t for a monoexponential model. The solid line is $f(x, \phi)$; the circles are the observed data points. An error is indicated.

Even though errors are unpredictable random quantities, some information about them is usually assumed, and some can be estimated. First, it is assumed that the mean error is zero. This simply means that were the true values for the parameters in ϕ known, the model would have no systematic overall bias (e.g., be systematically below or above the data points, on average).

A second aspect of the error, one that can be estimated by NONMEM, is its typical size. Since errors may be positive or negative, their typical size is not given by their mean (which is zero), but by their standard deviation, the square root of their variance. One can always simply convert the variance into the standard deviation, and conversely. NONMEM output gives estimates of the error variance. With individual data this variance is denoted in this text by ω^2, and by OMEGA in NONMEM input and output. The standard deviation (SD) of the error is denoted ω herein. The reason that OMEGA, rather than, for example, OMEGA SQ stands for ω^2 in NONMEM input and output will be discussed in Section 3.8. (We will see, in Chapter 4, that when the error is symbolized by ε, not η, its variance will be denoted σ^2 in this text, and SIGMA, not OMEGA, in NONMEM input and output.) Here, the parameter ω^2 quantifies the influence of the random effect, η on the observations, y. It is therefore called a random effects parameter.

3.1. The Additive Error Model

The symbol η is always used to denote a random quantity whose probability distribution has mean zero and variance ω^2. Model (3.4) says that the errors themselves can be so regarded, and since an observation equals its prediction (under the structural model) plus an error, model (3.4) is called the Additive error model. This model is illustrated in figure 3.2.

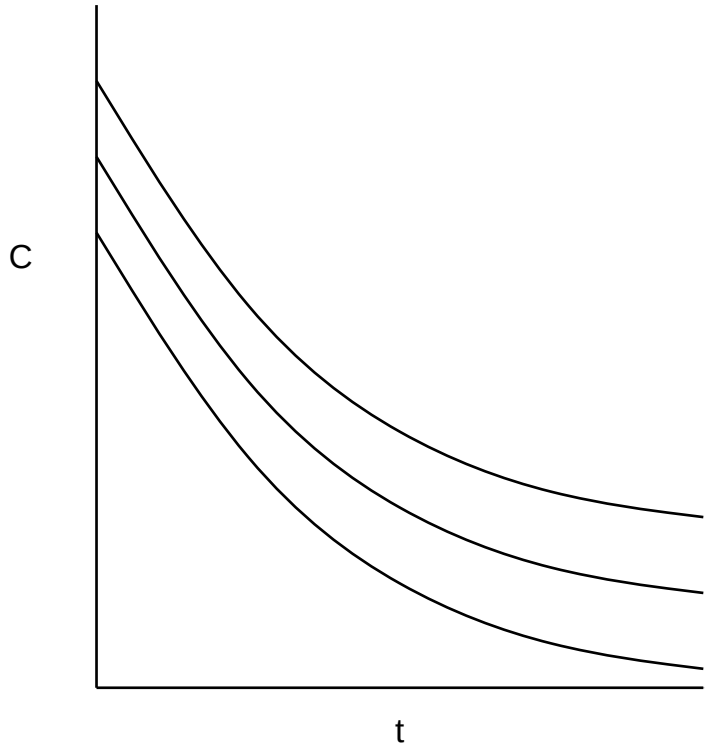

Figure 3.2. C vs t for a monoexponential model. The middle line is $f(x, \phi)$; the outer lines give the approximate "envelope" for additive errors. Don't be fooled by the apparent widening of the gap between the upper and lower curves as time increases: the vertical distance from the middle line to either outer line is everywhere the same.

3.2. The Constant Coefficient of Variation and Exponential Models

NONMEM allows an error model which can be more complicated than that of (3.4). One such more complicated, but useful model is the Constant Coefficient of Variation (CCV), or Proportional error model,

$$y_j = f(x_j, \phi) + f(x_j, \phi)\eta_j = f(x_j, \phi)(1 + \eta_j) \tag{3.5}$$

A fractional error is an error expressed as a fraction of the corresponding prediction. The CCV model says that a fractional error can be written as an η, i.e. as a random quantity with mean zero and variance ω^2. Under this model, the variance of an error itself is proportional to the squared prediction, with ω^2 being the proportionality factor, and so is not constant over observations. Since, under this model, the standard deviation of the error, and also of y, is $\omega f(x, \phi)$, and since the mean of y is $f(x, \phi)$ (when ϕ assumes its true value), the coefficient of variation of y is just the constant ω (the coefficient of variation of a random quantity is defined as its standard deviation divided by its mean). This is the

reason the CCV error model is so named. Also for this reason, ω^2 is dimensionless, in contrast to having units equal to those of the squared observation as with the Additive model. This error model is illustrated in figure 3.3.

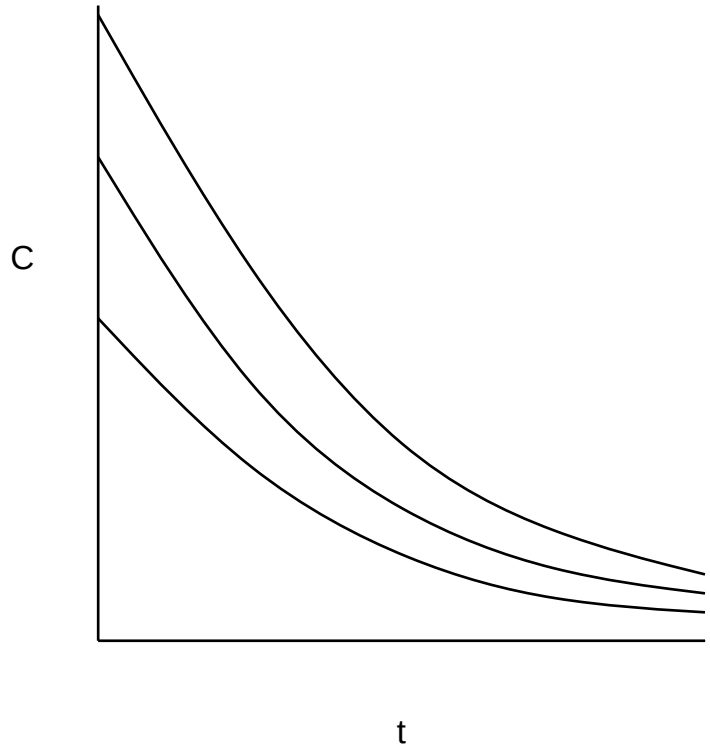

Figure 3.3. C vs t for a monoexponential model. The middle line is $f(x, \phi)$; the outer lines give the approximate "envelope" for constant coefficient of variation errors.

The exponential error model is

$$y_j = f(x_j, \phi) \exp(\eta_j) \tag{3.5a}$$

This model is sometimes referred to as the log-normal model, because it it is additive if logs are taken (and because eta sj is assumed to be normally distributed):

$$\log y_j = \log f(x_j, \phi) + \eta_j \tag{3.5b}$$

See Chapter 8, Section 3.2 for a discussion of this model.

3.3. Combined Additive and CCV Error Model

When most observations obey the CCV model but some observations may be near the lower limit of detection of an assay, a model which may be useful is one which is a combination of both the Additive and CCV error models:

$$y_j = f(x_j, \phi) + f(x_j, \phi)\eta 1_j + \eta 2_j. \tag{3.6}$$

Here there are two types of η's, $\eta 1$ and $\eta 2$. The first has variance ω_1^2; the second has a possibly different variance, ω_2^2. NONMEM permits several types of η's. Under this model, the variance of the error portion of the model is $\omega_1^2 f(x_j, \phi)^2 + \omega_2^2$. When the prediction is near zero, the variance is approximately constant, namely ω_2^2. This is the

smallest variance possible and corresponds, perhaps, to the limit of assay precision. When the prediction is considerably greater than zero, the variance is approximately proportional to the squared prediction.

3.4. The Power Function Model

A model that has both the additive and the CCV error models as special cases, and smoothly interpolates between them in other cases is the Power Function model:

$$y_j = f(x_j, \phi) + f(x_j, \phi)^p \eta_j. \tag{3.7}$$

Here $f(x, \phi)$ is raised to the p^{th} power in the error model, rather than the 0^{th} power (Additive error model; note $a^0 = 1$ for any number, a) or the first power (CCV model). The parameter p is a fixed effects parameter, even though its role in the overall model is to modify the variance model, not the structural model. With NONMEM all fixed effect parameters must be elements of the general parameter vector θ. If we want the Power Function Model to interpolate between the additive and CCV models, p must be constrained to lie between 0 and 1. NONMEM allows this (see Chapter 9). While one might be tempted to combine the Power Function model with the Additive model, much as the CCV and Additive model were combined above, such a combination model can lead to identifiability difficulties, and for this reason such a combination should be avoided.

3.5. Two Different Types of Measurements

Another more complicated error model can arise when more than one type of measurement is made. Suppose, for sake of illustration, that the observations are drug concentrations, but that they are measured with two different assays. If one assay may be more precise than the other, then this is equivalent to saying that one assay has a smaller ω^2 than the other. We would like to be able to take this into account in the analysis (i.e., not pay as much attention to the less precise observations), and perhaps (if we have enough data) estimate the relative precision of the assays as well. To do this in the notation we have introduced, an independent variable indicating which observations are obtained with which assay is needed: we call such an independent variable an indicator variable.

3.5.1. Use of an Indicator Variable

Let one of the data items (an element of x) be labeled ASY, and let ASY_j take the value 1 if the assay used for y_j was of the first type, and the value 0, if it was of the 2nd type. The variable, ASY is an indicator variable, and it allows us to write an additive type error model, say, as

$$y_j = f(x_j, \phi) + ASY_j \eta 1_j + (1 - ASY_j)\eta 2_j \tag{3.8}$$

Here there are two types of η's, $\eta 1$ and $\eta 2$. The first applies to the first type of assay, and has variance ω_1^2; the second applies to the second type of assay, and has a possibly different variance, ω_2^2. NONMEM permits several types of η's. Different types of η's can be correlated, and NONMEM can allow this. However, this is something we would only need to consider in the example at hand if the same blood sample were measured by both assays. We will not emphasize this possibility in this introductory guide. (This possibility also applies to random variables describing unexplained interindividual differences with population data; see Chapter 4)

When the assay is done by the first method, ASY will be unity, and (3.8) becomes

$$y_j = f(x_j, \phi) + \eta 1_j \tag{3.8a}$$

so that the variance of the error is ω_1^2. When the assay is done by the second method, *ASY* will be zero, and (3.7) becomes

$$y_j = f(x_j, \phi) + \eta 2_j \tag{3.8b}$$

so that the variance of the error is now ω_2^2. Both ω_1^2 and ω_2^2 are random effect parameters. An equivalent form of the model that can be implemented easily is

$$y_j = \begin{cases} f(x_j, \phi) + \eta 1_j, & \text{if } ASY_j \text{ is } 1 \\ f(x_j, \phi) + \eta 2_j, & \text{if } ASY_j \text{ is } 0 \end{cases} \tag{3.8c}$$

3.6. Two Different Types of Observations

The same need for separate scales for different measurements can arise when more than one type of observation is made. Suppose both plasma concentrations (C) and urine concentrations (Cu) are measured. The structural model for C_j might be (3.1), (3.3). If we assume that urine is collected between each observation of C, then the structural model for Cu_j, the drug concentration in the urine collected between time t_{j-1} and time t_j might be

$$\tilde{Cu}_j = f_o \frac{D}{Vu_j} (e^{-kt_{j-1}} - e^{-kt_j}) \tag{3.9}$$

where f_o is the fraction of drug excreted unchanged (a parameter), and Vu_j is the urine volume collected between time t_{j-1} and t_j (a data item)†. Assuming again, for sake of the example, that we want to use an additive type error model for the observations, the problem is that urine concentrations can be orders of magnitude larger than plasma concentrations, so that, while an additive error model might be appropriate for either type of observation alone, the two types of observations must have different typical error magnitudes; i.e., different variances (ω^2's).

An indicator variable can again be used. Let the indicator variable *TYP* be unity if the j^{th} observation is a C, and 0 if it is a Cu. We now need to use it for both the structural and error models, so that:

$$y = TYP_j \tilde{C}_j + (1 - TYP_j)\tilde{Cu}_j + TYP_j \eta 1_j + (1 - TYP_j)\eta 2_j \tag{3.10}$$

A little thought shows that the indicator variable selects the correct prediction ($\tilde{C}$ or $\tilde{Cu}$) and the correct error term for each observation (y).

An equivalent form of the model that can be implemented easily is

$$y = \begin{cases} \tilde{C}_j + \eta 1_j, & \text{if } TYP_j \text{ is } 1 \\ \tilde{Cu}_j + \eta 2_j, & \text{if } TYP_j \text{ is } 0 \end{cases} \tag{3.10a}$$

3.7. More Than One Indicator Variable

Of course, there could be three types of assays, or more, and similarly, more than two types of observations. One usually needs one less indicator variable than types of things to be distinguished. So, if there were three assays, one would define *ASY*1 and *ASY*2. *ASY*1 would be 1 if the assay were of the first type, and zero otherwise; *ASY*2 would be 1 if the assay were of the second type, and zero otherwise. The error model for the data

† With all PREDPP pharmacokinetic models there is an output compartment for which the total amount of drug leaving the system is computed automatically. The concentration in the urine is then obtained by setting the scaling parameter for the output compartment to Vu.

would require three types of η's, $\eta 1$, $\eta 2$, and $\eta 3$.

$$y_j = f(x_j, \phi) + ASY1_j \eta 1_j + ASY2_j \eta 2_j + (1 - ASY1_j)(1 - ASY2_j)\eta 3_j \quad (3.11)$$

Equation (3.11) results in the following:

Assay	$ASY1$	$ASY2$	Type of η	var(y_j)
1	1	0	$\eta 1$	ω_1^2
2	0	1	$\eta 2$	ω_2^2
3	0	0	$\eta 3$	ω_3^2

An equivalent form of the model that can be implemented easily is

$$y_j = \begin{cases} f(x_j, \phi) + \eta 1_j, & \text{if } ASY1_j \textit{ is } 1 \\ f(x_j, \phi) + \eta 2_j, & \text{if } ASY2_j \textit{ is } 1 \\ f(x_j, \phi) + \eta 3_j, & \text{if } ASY1_j \textit{ is } 0 \text{ and } ASY2_j \textit{ is } 0 \end{cases} \quad (3.10a)$$

3.8. The General Mixed Effects Model for an Individual

We have just seen examples of more complicated error models than the simple Additive model. We here give a mathematical form for the most general mixed effects model that is considered within the scope of this document:

$$y_j = f(x_j, \phi) + h'(x_j, \phi)\eta_j \quad (3.12)$$

where h is a vector valued function of x and parameters ϕ (where the latter is interpreted broadly to contain parameters such as p of (3.7)), and η is a vector of different different η types. The notation h' denotes vector transpose. When there is more than one η type, there will be several ω^2's, one for each type. The collection of these is denoted Ω and is labeled OMEGA in NONMEM input and output. This collection is regarded as a diagonal matrix (diagonal for now; but see Chapter 4), rather than as a vector. We will use the symbol ω_k^2 and ω_{kk} interchangeably in this text to denote the (diagonal) element of this matrix found in position k, k.

Chapter 4 - Models for Population Data

1. What This Chapter is About

In this chapter, models for data from (animal or human) populations will be discussed. These models describe observations from a number of individuals sampled from the population. The distinguishing feature of the data to which such models apply is that there is *more than one* observation from some (usually most) individuals. A population model includes the structural model of Chapter 3, but also a new model, which shall be called the parameter model, for each individual's kinetic parameters. The parameter model can have both fixed and random effects. A population model also includes the error model of Chapter 3.

2. General

Individuals differ, and the types, degrees and causes of these differences are often what we want to learn. NONMEM was designed to help us learn these things. These individual differences can be due to fixed and/or random effects, but they all manifest themselves by affecting the value of an individual's parameters, ϕ. That is, first, each individual is regarded as having his own particular value of ϕ. If the data come from $i = 1, \ldots, N$ individuals, then we may rewrite the (not completely) general mixed effects model, (3.4) for y_{ij}, the j^{th} observation from the i^{th} individual, as

$$y_{ij} = f(x_{ij}, \phi_i) + \varepsilon_{ij} \tag{4.1}$$

Eq (4.1) is now (part of) a population model because it explicitly recognizes, through the subscript, i, that the data come from distinct individuals. Note too that we have written ε, rather than η. According to NONMEM conventions, when modeling data from a *population*, the random effects in the residual errors are denoted by ε, their individual variances by σ^2, and the collection of the variances by the matrix Σ, denoted SIGMA in NONMEM input and output. We also adopt the same convention here as we did for Ω: the k^{th} diagonal element of Σ is interchangeably denoted σ_k^2 or σ_{kk}.

When dealing with population data, the symbol η is reserved for random effects influencing the vectors ϕ_i, as is now explained. We can write a general model (but not yet as general a model as we will present later) for ϕ_i:

$$\phi_i = g(z_i, \theta) + \eta_i \tag{4.2}$$

It is called the parameter model. Here, g is a structural (though non-kinetic) type model (of which examples will be given shortly), which is a function of fixed effects, z_i, and fixed effects parameters, θ. Note that since, in general, ϕ is a vector, g must be a vector-valued function, and for the same reason, η is usually a vector. This will be discussed further later. All fixed effects, whether they are part of the kinetic structural model, or are part of the parameter model, are input to NONMEM in a uniform way. For the purposes of this discussion, the symbol z is used for the particular fixed effects in g, such as the individual's height, weight, and so forth (this will be discussed further in a moment). Even though most often ϕ is regarded as time invariant, as is done in most of the discussion in this document, fixed effects can change with time, and thus kinetic parameters within ϕ can change with time. This will be discussed further in Section 3.4.2.

3. Structural Parameter Models

The symbol in (4.2) for the fixed effects parameter vector is θ, not ϕ. As mentioned in Chapter 3, we reserve the symbol ϕ, in this document, for an individual's fixed effect parameters and use the symbol θ for a vector of *population* (fixed effects and possibly random effects) parameters.

Recall the phenobarbital example of Chapter 2. For the second run, the input contained the line of code

```
TVCL = THETA(1) + THETA(3)*WT
```

Translated into the symbols we are using here, this is

$$\tilde{Cl}_i = \theta_1 + \theta_3 WT_i \tag{4.3}$$

In (4.3), θ_1 and θ_3 are the first and third elements of the parameter vector θ, and WT_i is an element of z_i (recall that this value of weight appears as a data item). The tilde over CL is meant to distinguish this typical population value of clearance from the i^{th} individual's actual value of clearance. According to this model, $\tilde{CL}_i$ will be the same for any two individuals both of whom have the same value of weight. Equation (4.3) defines an element (the one associated with clearance) of the vector-valued function g. Note that in (4.3), we use the subscript i to stress that this equation applies to the i^{th} individual, but there is no confusion when, as in the NM-TRAN input, and in the following, the subscript is omitted. It should always be understood that all variables and data items used in the parameter model definition refer to the same individual. Many different models are possible to describe the dependence of individual parameters on fixed effects. However, certain model forms are simple, easy to use, and cover most cases. An assortment of these will be discussed briefly next.

3.1. Linear Models

The simplest form that g can take, and the most common, is one that is linear in θ. An example is (4.3): all elements of θ appear as linear coefficients of terms involving data items. The data items themselves can appear nonlinearly, without affecting the linearity with respect to θ. For example, if clearance is the sum of renal and non-renal components, and renal clearance is believed to be proportional to renal function as described according to a standard formula involving the elements of z: age (AGE), serum creatinine ($SECR$), and weight (WT), then one might write

$$\tilde{CL}_{met} = \theta_1 WT \tag{4.4}$$

$$RF = WT\,\frac{1.66 - .011 AGE}{SECR} \tag{4.5a}$$

$$\tilde{CL}_{ren} = \theta_4 RF \tag{4.5b}$$

$$\tilde{Cl} = \tilde{CL}_{met} + \tilde{CL}_{ren} \tag{4.6}$$

Clearly, RF is a nonlinear function of $SECR$, for example, and so, therefore, is $\tilde{Cl}$, but $\tilde{Cl}$ is linear in θ, and (4.4 - 4.6) is still considered a linear model. (Do not worry about the non-consecutive numbering of the elements of θ; a model for $\tilde{CL}$ is being developed (an alternative to 4.3), and the missing elements θ_2 and θ_3 will appear presently.)

3.2. Multiplicative Models

Multiplicative models are linear models, but on a logarithmic scale. For example, if patients covering a very wide range of weights are studied, metabolic clearance might vary with weight, but not linearly, and a substitute for (4.4) might be

$$L\tilde{C}l_{met} = \theta_1 + \theta_2 log(WT) \tag{4.4.1}$$

$$\tilde{C}l_{met} = exp(L\tilde{C}L_{met})$$

Note that the logarithm of $\tilde{CL}_{met}$ ($L\tilde{C}L_{met}$) is linear in θ, but $\tilde{CL}_{met}$ itself is not. Of course, (4.4.1) can also be written

$$\tilde{C}l_{met} = \theta_1 WT^{\theta_2} \tag{4.4.2}$$

Models (4.4.1) and (4.4.2) are equivalent so far as $\tilde{Cl}$ is concerned, but θ_1 of (4.4.2) corresponds to exp(θ_1) of (4.4.1).

3.3. Saturation Models

A useful model for processes reaching a maximum is a hyperbolic model. For example, if a second drug, (whose steady-state plasma concentration, $Cpss_2$ is known and available in the data set), is present in some individuals and it is believed that this second drug is an inhibitor of the metabolism of the study drug, one might wish to use

$$\tilde{C}l_{met} = WT\left(\theta_1 - \frac{\theta_2 Cpss_2}{\theta_3 + Cpss_2}\right) \tag{4.4.3}$$

This model is shown in figure 4.1. The inhibition is expressed by the ratio occurring within the brackets and is a concave hyperbola, asymptoting to a maximum value equal to θ_2. It is identical in form to the familiar Michaelis-Menten model.

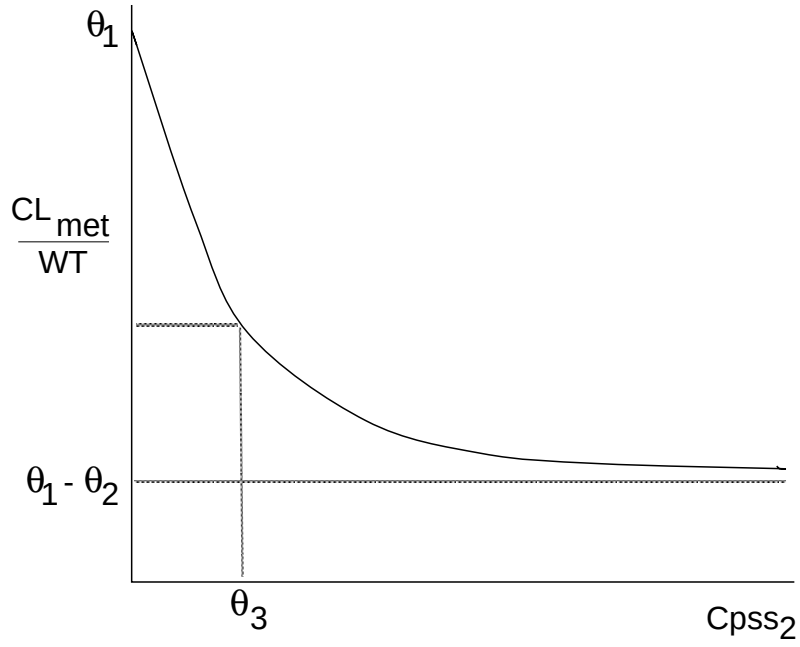

Figure 4.1. A hyperbolic model for metabolic clearance of drug on the ordinate, as inhibited by another drug at steady-state concentration $Cpss_2$ on the abscissa.

3.4. Models with Indicator Variables

Indicator variables were discussed in Chapter 3 in connection with the error model. They can be quite useful when modelling individual parameters. They are usually used in a linear model. For example, if the clinical condition, heart failure, is noted as "present" or "absent", one can define an indicator variable, *HF* which equals 0 if heart failure is absent, and 1 if it is present. If metabolic clearance is thought to be affected by heart failure, one might choose

$$\tilde{Cl}_{met} = (\theta_1 - \theta_2 HF)WT \tag{4.4.4}$$

Here, the non-heart-failure cases will have $\tilde{Cl}_{met} = \theta_1 WT$, while the heart-failure cases will have $\tilde{Cl}_{met} = (\theta_1 - \theta_2)WT$†.

3.4.1. Combinations

Given the basic building blocks of linear, multiplicative and saturation models, these can be combined in the usual algebraic ways (usually by addition) to make more complex models. For example, one could use (4.4.3), (4.5), and (4.6) as a model for $\tilde{Cl}$. A non-additive example arises if plasma and urine concentrations are both observed and (kinetic) model (3.6) is to be used for the latter. The parameter f_o, the fraction of drug excreted unchanged into the urine might be modeled as

$$f_o = \frac{\tilde{Cl}_{ren}}{\tilde{Cl}} \tag{4.7}$$

where $\tilde{Cl}_{ren}$ is given by (4.5) and $\tilde{Cl}$ by (4.6) (using any of the (4.4) variants).

3.4.2. Time Varying *z*

As mentioned in Section 2, although most of the time the data items affecting an individual's ϕ do not change over the course of his data, they occasionally do, and PREDPP can handle this. For example, if an individual had heart failure for part of his observation period, but not the rest, $\tilde{Cl}_{met}$, according to (4.4.4) should change. Or, if acute renal failure occurred during a patient's observation period, $\tilde{Cl}_{ren}$ would change, according to model (4.5).

PREDPP implements its kinetic model recursively: given the state of the system at time t_j (by state we mean the amounts of drug in all the compartments), it updates (i.e. advances) the state to that at time t_{j+1}, using the value of z (and in general, the value of x) at time t_{j+1} to compute a value of ϕ holding between times t_j and t_{j+1}. The value of z used to compute this ϕ is that value found on the data record with time t_{j+1}. So, in order to have ϕ change appropriately as z does, one places a value of z which is typical for the time period t_j to t_{j+1} on the data record associated with the time point t_{j+1}. This will not always be easy since the relevant element(s) of z may not be measured at, for example, the midpoint of the time interval (the value at the *midpoint* of the time interval is a good choice for the *typical* value for the interval). If not, one will have to use some interpolation method to arrive at the typical value. The important point is that the values of the independent variables at time t_{j+1} determine the values of the individual's parameters applying to the entire period t_j to t_{j+1}.

---

† Heart failure is expected to decrease metabolic clearance. If it does, using a minus sign in (4.4.4) allows the more pleasing result that θ_2 will be estimated as positive. The model is identical to one with a positive sign, but then θ_2 would probably be negative. If θ_2 were constrained to be non-negative, then the sign chosen in the model statement would, of course, be important.

3.5. Structural Kinetic Models

The kinetic models (i.e., the models for responses such as drug concentrations) used when performing a population analysis do not differ at all from those used for an individual analysis. One still needs a model for the relationship of y to ϕ and x, and this relationship does not depend on whether ϕ changes from individual to individual or with time within an individual.

4. Population Random Effects Models

Under NONMEM conventions, there are two levels of random effects, and η and ε are the symbols used for the vectors of first and second level random effects, respectively. With data from a single individual, only first-level random effects are needed. However, with data from a population of individuals, both first- and second-level random effects are needed. First-level effects are needed in the parameter model to help model unexplainable interindividual differences in ϕ, and second-level effects are needed in the (intraindividual) error model. For example, in (4.2) there is an element of η_i, η_i^V, that is the difference between the individual value V_i (an element of ϕ) and $\tilde{V}_i$, the typical value of V_i. This is a first-level random effect. In (4.1) ε_{ij} is the error between y_{ij} and $f(x_{ij}, \phi_i)$. This is a second-level random effect.

4.1. Models for Interindividual Errors

The difference between ϕ_i and $g(z_i, \theta)$ is called an interindividual error. It arises from a few sources: the function g may be only approximate, and/or z may be measured with error. It is regarded as a random quantity, and it may be modeled in terms of η variables. As usual, each of these variables is assumed to have mean 0 and a variance denoted by ω^2 which may be estimated. This variance describes biological population variability.

The difference between y_{ij} and $f(x_{ij}, \phi_i)$ is called an intraindividual error. It has been discussed at some length in Chapter 3. Although in that discussion about individual data, this difference was modeled in terms of η variables, in this discussion about population data, it is modeled in terms of ε variables. Each ε variable is assumed to have mean 0 and a variance denoted by σ^2 which also may be estimated.

Each pair of elements in η has a covariance, and NONMEM can also estimate this, although often we choose to assume that the covariance is zero (we made this same assumption for the different elements of η in Chapter 3, Section 3.5.1).
A covariance between two elements of η, η_k and η_m, say, is a measure of statistical association between these two random variables. Their covariance is related to their correlation, ρ_{km} ($\rho_{km} \equiv \rho_{mk}$) by

$$cov(\eta_k, \eta_m) = \rho_{km}\omega_k\omega_m \tag{4.8}$$

(Note that now that we are suppressing the subscript i on η, we may, without confusion, use the subscript position to designate elements of η.)

The variances and covariances among the elements of η are laid out in a covariance matrix, called Ω, and labeled OMEGA in NONMEM input and output. This matrix was defined in Chapter 3, Section 3.8, but some additional comment here may be helpful. If η has, for example, 3 elements, Ω has the following form:

$$\begin{matrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{21} & \omega_{22} & \omega_{23} \\ \omega_{31} & \omega_{32} & \omega_{33} \end{matrix}$$

Here, as previously, ω_{kk} is another way of writing the variance ω_k^2, and ω_{km} ($k \neq m$) is the covariance between η_k and η_m.

The elements ω_{11}, ω_{22}, ω_{33} are called the diagonal elements of the matrix. If the nondiagonal elements (the covariances) are all zero, i.e. the correlation among all pairs of η elements is zero, the matrix is called a diagonal matrix. The lower triangular elements of the matrix are the elements

$$\begin{matrix} \omega_{11} & & \\ \omega_{21} & \omega_{22} & \\ \omega_{31} & \omega_{32} & \omega_{33} \end{matrix}$$

To specify the matrix only its lower triangular elements need be given (and these are all NONMEM does give), since from (4.8) it is clear that for all k, m, $\omega_{km} = \omega_{mk}$.

4.1.1. Additive/Multiplicative Models and the Exponential Model

Frequently, the model for an interindividual error is the simple additive one (as in (4.2)), such as

$$V = \tilde{V} + \eta_2 \tag{4.9}$$

A feature of (4.9) is that the resulting units for ω^2 depend on the units of the parameter (V in this case). For example, this model was used in the theophylline problem of Chapter 2 (Figure 2.6). The final estimate of ω_2^2 is .286 (Figure 2.8). Assuming that the units of V are liters, we interpret this to mean that the standard deviation of V between individuals is .53 Liters ($.53 = \sqrt{.286}$).

Perhaps even more often, a multiplicative model equivalent to the Constant Coefficient of Variation (CCV) error model (3.5) is used, such as

$$V = \tilde{V}(1 + \eta_2) \tag{4.10}$$

This model is also referred to as the proportional error model.
A feature of (4.10) is that the resulting units for ω^2 are independent of the units of the parameter (V in this case). When this model is used in the theophylline problem instead of the additive model, so that Figure 2.6 contains the code `V=TVVD*(1+ETA(2))` instead of `V=TVVD+ETA(2)`, then NONMEM estimates ω_2^2 to be .146. We interpret this to mean that the coefficient of variation of V in the population is 38% ($.38 = \sqrt{.146}$).

The exponential model is

$$V = \tilde{V} \exp(\eta_2) \tag{4.10a}$$

During simulation, (Chapter 12, Section 4.8), the exponential and proportional models give different results. During estimation by the first-order method, the exponential model and proportional models give identical results, i.e., NONMEM cannot distinguish between them. During estimation by a conditional estimation method, the exponential and proportional models for inter-individual variability give different results. The exponential model is preferred for conditional estimation methods. (See NONMEM User's Guide Part VII, Conditional Estimation Methods.)

4.1.2. Other Models

Occasionally, a model for an individual's pharmacokinetic parameter might involve scaling an η, as in (3.6), or two or more η's as in (3.10). For example, a study might involve patients in the intensive care unit (ICU) and others on non-acute care units. It might be

reasonable to suppose that some aspects of the kinetics of ICU patients (e.g., metabolic clearance of drug) are more variable, due to unmeasured factors (e.g., hepatic function) that vary greatly among acutely ill patients. Even though the variation is, in reality, due to a potentially measurable fixed effect (hepatic function), if information on this fixed effect is not available, differences caused by it must be assigned to random factors (η). In this case, one might wish to use an indicator variable, *ICU* (which equals 1 if the patient is in the ICU, and 0, otherwise), and a model such as

$$Cl_{met} = \tilde{Cl}_{met} + (1 - ICU)\eta_1 + ICU\,\eta_2 \tag{4.11}$$

In addition to model (4.11) we might have, for example,

$$Cl_{ren} = \tilde{Cl}_{ren} + \tilde{Cl}_{ren}\eta_3 \tag{4.12a}$$

$$Cl = Cl_{ren} + Cl_{met}. \tag{4.12b}$$

Models (4.11) and (4.12) together, along with suitable models for $\tilde{Cl}_{ren}$ and $\tilde{Cl}_{met}$, form a complete model for an individual's *Cl* parameter, and involve 3 η's.

4.1.3. General Form for the Parameter Model

As we have just seen in (4.10) and in (4.11)-(4.12), an element of η need not act in a simple additive way and may act solely on an intermediate variable (e.g. Cl_{met}). Indeed, there may be more or fewer elements in η than in ϕ, the elements in η may act in nonlinear ways to influence ϕ, and one element of η may influence more than a single element of ϕ. We now give a more general form for the parameter model than (4.2) and then an example illustrating it.

The general form of the parameter model is

$$\phi_i = g(z_i, \theta, \eta_i) \tag{4.13}$$

Here, g is a very general function of fixed effects, z_i, fixed effects parameters, θ, and a vector of η's, η_i. The dimensions of the vectors ϕ_i and η_i need not be the same. An individual's kinetic parameter may change with time. As explained in Section 1.6, with NONMEM-PREDPP changes can occur only at discrete time points. Therefore, the parameter actually can be regarded as being a number of parameters, each one applying to a different time period. The parameter ϕ_i in (4.13), being a vector of all the kinetic parameters for the i^{th} individual, can be regarded as encompassing these time-interval-specific parameters.

An example utilizing this generality is provided by a model for observations of both plasma and urine drug concentrations, similar to the one presented previously. Ignoring the details of the structural part of the model, consider the following model

$$\begin{aligned} Cl_{met} &= \tilde{Cl}_{met} + \eta_1 \\ Cl_{ren} &= \tilde{Cl}_{ren} + \eta_2 \\ Cl &= Cl_{met} + Cl_{ren} \\ f_o &= \frac{Cl_{ren}}{Cl} \\ V &= \tilde{V} + \eta_3 \end{aligned} \tag{4.14}$$

In this model, $\phi = (V, Cl, f_o)$; the parameters Cl_{met} and Cl_{ren} are regarded as intermediate parameters. We have $\eta = (\eta_1, \eta_2, \eta_3)$, where both η_1 and η_2 influence both Cl (linearly) and f_o (nonlinearly).

4.2. Statistical Models for an Individual's Observations

Model (4.1) can be generalized by incorporating a model like those given in Chapter 3 for the residual errors, i.e. for the differences between the y_{ij} and $f(x_{ij}, \phi_i)$, rather than using just the simple Additive model. A particular instance of such a model may have several types of ε's, and as mentioned in Section 2, the variances of these ε's are denoted by σ^2's. With a population model these variances could change from individual to individual. With NONMEM, they are considered as constants over individuals. The ε's can covary. A covariance matrix Σ, like the Ω matrix given in Section 4.1, gives the variances and covariances of the ε's, as already discussed at the end of Chapter 3. This does not preclude the magnitudes of the errors from being affected by fixed effects. A model such as (3.8) can still be used. This is shown explicitly by the general model given in the next section.

5. The Population Mixed Effects Model

We have now presented all of the parts needed to fully define a population model. It may be useful to recap this information by stating the entire general model here:

$$y_{ij} = f(x_{ij}, \phi_i) + h'(x_{ij}, \phi_i)\varepsilon_{ij} \tag{4.15a}$$

$$\phi_i = g(x_{ij}, \theta, \eta_i) \tag{4.15b}$$

$$cov(\varepsilon_{ij}) = \Sigma; \quad cov(\eta_i) = \Omega$$

$$\varepsilon_{ij}, \varepsilon_{kl} \text{ independent for } (i, j) \neq (k, l)$$

$$\eta_i, \eta_k \text{ independent for } i \neq k$$

$$\varepsilon_{ij}, \eta_k \text{ independent for all } i, j, k,$$

where here, ε_{ij} is a vector, along with x_{ij}, ϕ_i, θ and η_i, and Σ and Ω are square matrices with dimensions equal to those of ε_{ij} and η_i.

To try to represent the relationship between all the fixed and random effects of a population model graphically, consider figure 4.2. The model corresponding to this figure is

$$y_{ij} = \frac{D}{V_i} exp[-(Cl_i/V_i)t_{ij}] + \varepsilon_{ij}$$

$$Cl_i = \theta_1 + \theta_2 RF_i + \eta_i^{Cl} \tag{4.16}$$

$$V_i = V$$

$$var(\varepsilon_{ij}) = \sigma^2; \quad var(\eta_i^{Cl}) = \omega_{Cl}^2$$

where the V_i are all equal to a constant V, i.e. there is no random interindividual variability in the volume of distribution, so that for the sake of this example, η_i is just a scalar.

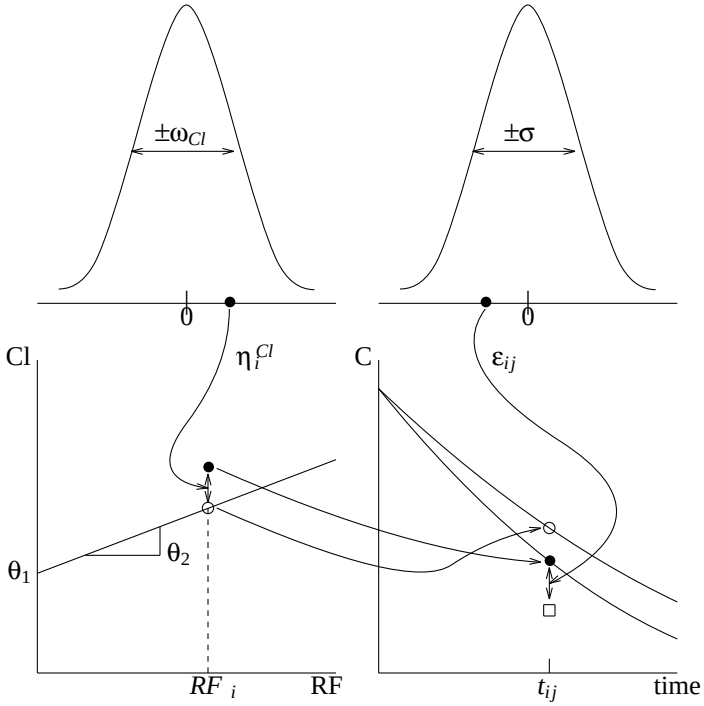

Figure 4.2. Random and fixed effects influence observation, C_{ij}, from the population point of view. Open circle, lower left, is population parameter predicted clearance, closed circle is true clearance for i^{th} individual which differs from population prediction by η_i^{Cl}, chosen randomly from a distribution (upper left) with mean 0 and SD ω_{Cl}. Similarly, lower right, the observed C at time t_{ij} (open square) differs by ε_{ij} from the true value (closed circle) by an error ε_{ij}, chosen independently from a distribution with mean 0 and SD σ_ε. The C corresponding to the population-based prediction is also shown (upper curve, open circle).

Chapter 5 - Estimates, Confidence Intervals, and Hypothesis Tests

1. What This Chapter is About

In this chapter, we discuss the fitting criterion that NONMEM uses, parameter estimates, and standard error estimates. We then discuss how to form confidence intervals for parameters and do hypothesis tests with NONMEM.

2. Model Fitting Criterion

In principle, all fitting procedures attempt to adjust the values of the parameters of the model to give a "best fit" of the predictions to the actual observations. The set of parameters that accomplish this are called the parameter estimates, and are denoted here as $\hat{\theta}$, $\hat{\Omega}$, and $\hat{\Sigma}$. Methods differ in how they define "best". The criterion that NONMEM uses is a Least Squares (LS) type criterion. The form of this criterion varies as the error model varies, and as population models with multiple random effects must be considered. We briefly discuss these various criteria next, to give the reader a feel for what NONMEM is doing. A detailed knowledge of the statistical basis for the choice of fitting criterion is not necessary either to use or interpret NONMEM fits. In this chapter, a fixed effects parameter will be denoted by a θ; the distinction between individual fixed effects parameters (ϕ) and population fixed effects parameters will not be important here.

2.1. Least Squares for Individual Data with an Additive Error Model

For the Additive error model (3.4), the Ordinary Least Squares criterion (OLS) chooses the estimate $\hat{\theta}$ so as to make the sum of squared (estimated) errors as small as possible. These estimates cause the prediction, here denoted $\tilde{y}$, to be an estimate of the mean value of y, which is intuitively appealing. The prediction is obtained by computing the value for y under the model with parameters set to their estimated values and η set to zero†.

2.2. Least Squares for Individual Data with Other Types of Error Models

The simple OLS criterion just defined becomes inefficient and is no longer the "best" one to use when the error model is other than the Additive error model. It treats all estimated errors as equally important (i.e. a reduction in the magnitude of either of two estimated errors that are of the same magnitude is equally valuable in that either reduction decreases the sum of squared errors by the same amount), and this results in parameter estimates that cause all errors to have about the same typical magnitude, as assumed under the Additive model. The CCV error model, though, says that the typical magnitude of an error varies monotonically with the magnitude of the (true) prediction of y. In principle, Weighted Least Squares (WLS) gives a fit more commensurate with the CCV or other non-Additive error model. WLS chooses $\hat{\theta}$ as that value of θ minimizing

$$O_{WLS}(\theta) = \sum_j w_j (y_j - \tilde{y}_j)^2 \tag{5.1}$$

Each w_j is a weight which, ideally, is set proportional to the inverse of the variance of y_j. In the CCV model this variance is proportional to $\tilde{y}_j^2$ (evaluated at the true value of θ). Use of such weights will down-weight the importance of estimated squared errors associated with large values of $\tilde{y}$ and promote the relative contribution of those associated with small values of $\tilde{y}$.

---

† η, not ε, since we follow the NONMEM convention and, when discussing individual type data as here, use η to denote the random effects of a single level that appear in the model, those in the residual error model.

In many cases, users can supply approximate weights, and the WLS objective function can be used as stated in (5.1). When, as with the CCV model for example, the ideal weights depend on the true values of parameters, these true values can be replaced by initial estimates, and then the WLS objective function as given in (5.1) can be minimized. Alternatively, instead of viewing O_{WLS} as a function of θ only through the estimated error's dependence on θ, it can be viewed as a function of θ through both that dependence and *also* through the ideal weights' dependence on θ. The entire function can then be minimized with respect to θ. That this creates a problem is most easily seen when the error model contains a parameter which is not itself a parameter of the structural model, but which, nonetheless, must be regarded as an element of θ. Such an error model is the Power Function model of (3.7), and the "extra" parameter is p. The WLS objective function with the reciprocal variance of y_j substituted for w_j is†

$$O^*_{WLS}(\theta) = \sum_j \left[\frac{(y_j - \tilde{y}_j)^2}{\omega^2 \tilde{y}_j^p} \right] \tag{5.2}$$

In this case if p were set to a very large number, while the other parameters in θ were only such as to make all $\tilde{y}_j > 1$, then all $\tilde{y}_j^p$ would be very large, and the summation would attain a very small value. (The value of ω^2 is irrelevant to the minimization with respect to θ.) Thus, all elements in θ other than p would be indeterminate (as long as they were such that all $\tilde{y}$ were greater than 1); a most unsatisfactory state of affairs.

There is a way to deal with this problem that preserves the spirit of least-squares fitting, and NONMEM uses it. In essence, it adds to the WLS objective function a term proportional to the sum of the logarithms of the error variances. Thus a penalty is paid for increasing the error variances without a concomitant decrease in the estimated errors themselves. This modified objective function is called the Extended Least Squares (ELS) objective function. It is minimized with respect to all parameters of the structural and error models simultaneously (in the current example, θ and ω^2, as p can be considered an element of θ). Note that the objective function may be negative. This has no particular significance.

2.3. Least Squares for Population Data

The complications arising from a population model are due entirely to the random interindividual effects occurring in the parameter model. To deal with this, NONMEM uses an approximation to the true model. The approximate model is linear in all the random effects. For this linearized model, the vector of mean values for the observations from the i^{th} individual is the vector of true predictions for these observations. These predictions are obtained from the model by setting the parameters to their true values and by setting all the elements of both ε and η to zero. In other words, these are the true predictions for the mean individual with fixed effects equal to those of the i^{th} individual. For this linearized model it is also possible to write a formula for the variance-covariance matrix of the observations from the i^{th} individual. This matrix is a function of the individual's fixed effects and the population parameters θ, Ω, and Σ. Finally, the ELS objective function discussed above is generalized to be a sum over individuals, rather than observations, and where the i^{th} term of the sum involves a squared error between a vector of observations and an associated vector of predictions, weighted by the reciprocal of the associated variance-covariance matrix for the i^{th} individual.

---

† Again, we call attention to the symbols used for the random effects parameter: the term ω^2 appears in the objective function, (5.2), not σ^2, because we are discussing individual type data, not population type data.

3. Parameter Estimates

It is useful to consider how to estimate parameters that do not appear in the model we use to fit the data, but are, instead, functions of them (e.g. the half-life parameter $t_{\frac{1}{2}} = .693/k$, when the rate constant of elimination k is the model parameter).

It is always possible, of course, to parameterize the model in the function of interest. For example, we have already seen (Chapters 2 & 3) that we may use the function (parameter) *Cl* in the one-compartment model instead of k. However, we may be interested in the values of several alternative parameterizations (e.g., we may want to know k, clearance, and half-life). Rather than rerun the same problem with several alternative parameterizations, we can use the fact that the LS estimate of a function of the parameters is given by the same function of the LS parameter estimates. Formally, if $\theta' = q(\theta)$ is the function of interest, then $\hat{\theta}'_{LS} = q(\hat{\theta}_{LS})$. E.g. Letting $\theta' = t_{\frac{1}{2}}$, $\theta = k$, and $q(\theta) = .693/\theta$, then $\hat{t}_{\frac{1}{2}} = .693/\hat{k}$.

4. Precision of Parameter Estimates

Clearly, it is almost impossible for the estimates to actually be the true values. The question is: how far are the true values from the estimates? To discuss this question, imagine replicating the entire experiment (gathering new data, but keeping x fixed) multiple times. Also, for simplicity, imagine that the model has only one scalar parameter, θ, and that its true value, θ_T is known. If, after each replication, the estimate of θ is recorded, and a histogram of these values is plotted, one might see something like figure 5.1A or 5.1B.

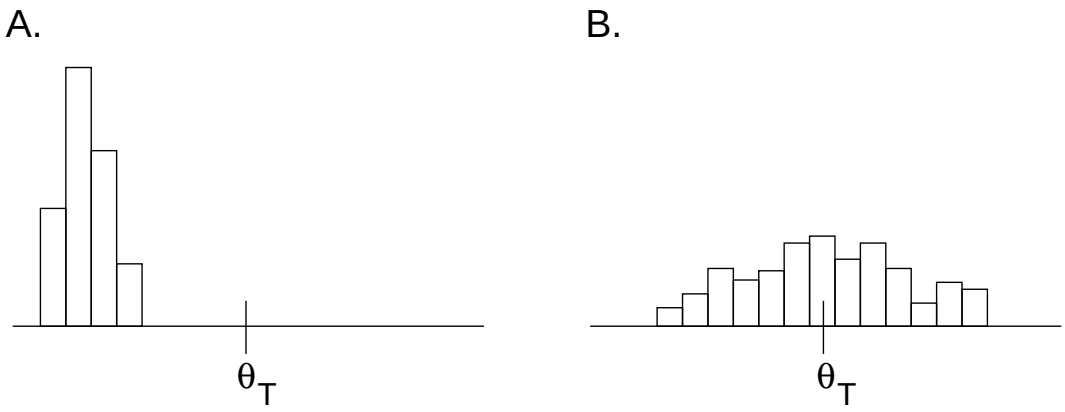

Figure 5.1. Two hypothetical histograms of estimates of a single parameter upon replication of a given experiment. Left panel (A): The estimates have small variance (spread) but are biased (the mean of the estimates differs from the true value, θ_T); Right panel: The estimates have large variance but are relatively unbiased.

The difference between the estimate and the true value, θ_T, obviously differs from replication to replication. Let this difference be called the estimation error. We cannot know the estimation error of any particular estimate (if we could, we could know the true value itself, by subtraction), but we can hope to estimate the mean error magnitude. Since errors can be positive or negative, a measure of magnitude that is unaffected by sign is desirable. This is traditionally the Mean Squared Error (*MSE*). The MSE can be factored into two parts:

$$MSE = B^2 + SE^2 \tag{5.2}$$

where B is the bias of the estimates (mean (signed) difference between the estimates and

the true value) and *SE* is the standard error of the estimates (SE^2 is the variance of the estimates). As illustrated in figure 5.1, the *MSE* can be about the same for two types of estimates while both their bias and *SE* differ. It is very hard to estimate the bias of an estimator unless the true parameter value is, in fact, known. This is not true of the *SE*: the standard deviation of the distribution of estimates of a parameter on replication is the *SE*; no knowledge of the true value of the parameter is required. In many situations, LS estimators of fixed effects parameters are unbiased, although in nonlinear problems, such as most pharmacokinetic ones, this cannot be assured.

4.1. Distribution of Parameters vs Distribution of Parameter Estimates

Figure 5.1 illustrates the distribution of parameter estimates that might result if an experiment were replicated. The bias and spread depend on the estimation method, the design of the experiment (x, which implicitly includes n) and on the true parameter values, including the variances (and covariances) of the random effects influencing y. If, for example, more observations were obtained in each experiment (more individuals in a population study), the spread of the estimates (one from each experiment) would decrease until, in the limit, if an infinite number of observations were obtained in each experiment, every estimate would be the same (equal to the true value plus the bias of the estimator). Thus, the distribution of the estimate tells us nothing about biology or measurement error, but only about the *precision* of the estimate itself.

In contrast, Ω and Σ tell us about unexplained (or random) interindividual variability (biology) or error magnitude (biology plus measurement error), not about how precisely we know these things. No matter how many observations we make, interindividual variability will remain the same size (but the variability of our estimate of its size will decrease), as will the measurement error variability of a particular instrument.

It is very important not to confuse variability (e.g., between individuals) in a model parameter with variability in the estimate of that parameter, despite the fact that the terms we use to describe both variabilities are similar. Thus an element of η, say $\eta 1$ has a *variance*, ω_{11}, while the estimate of ω_{11}, $\hat{\omega}_{11}$, also has a *variance* given by the square of the standard error for $\hat{\omega}_{11}$. Indeed, the use of the term "standard error" rather than "standard deviation" to name a measure of the spread in the distribution of the parameter *estimate* rather than in the parameter helps call attention to the distinction between these two types of things.

4.2. Confidence Interval for a Single Parameter

Acknowledging that any particular estimate from any particular experiment is unlikely to equal the true parameter value implies that we should be interested in "interval" estimates of parameters as well as (instead of?) point estimates. An interval estimate of a parameter is usually a range of values for the parameter, often centered at the point estimate, such that the range contains the true parameter value with a specified probability. The probability chosen is often 95%, in which case the interval estimate is called the 95% Confidence Interval (CI).

A CI is often based only on the parameter estimate and its *SE*. In the next sections we discuss three questions about such CIs a little further. (i) How to estimate the *SE* from a single set of data (we cannot replicate our experiment many times and construct a histogram as in figure 5.1). (ii) Given an estimate of *SE*, how to use that number to compute a (95% confidence) interval with 95% chance of containing the true parameter value. (iii) Given an estimate of *SE*, how to compute a confidence interval for a function of the parameter.

4.2.1. Estimating a Parameter's Standard Error

Remarkably, the *SE* of a parameter estimate can be estimated using only the data from a single experiment. The idea is that the data give us estimates of the variances of all random effects in our model, from which we can estimate the variability in future data (if we were to replicate the experiment). That is, the SE of the estimates on replication depends only on quantities we either know or have estimates of: the x, the number of y observed (n), and the variances of all random effects.

It is a little beyond the scope of this discussion to give the method by which NONMEM actually estimates the *SE* of a parameter estimate; however, examples of how this can be done are found in any statistical textbook on regression. NONMEM presents the estimated standard error for each parameter of the model (including the random effects parameters, Ω and Σ) as part of its output.

4.2.2. Relating the Confidence Interval to the SE

Statistical theory tells us not only how to compute the *SE* of a parameter estimate, but also that for a LS estimate (and many other kinds of estimates), the shape of the distribution of the estimates is approximately Normal if the data set is large enough. This means that we may use percentiles of the Normal distribution, to compute confidence interval bounds (when n is small, the t distribution is often used instead; this is discussed in statistics texts). In general, a 100(1-α)% confidence interval for a single parameter, θ say, is computed as $\hat{\theta} \pm Z_{1-\alpha/2}SE$. Here $Z_{1-\alpha/2}$ denotes the $1-\alpha/2$ percentile of the Normal distribution. As previously noted, α is often chosen to be .05, in which case Z is approximately 2.

4.2.3. A Confidence Interval for a Function of a Single Parameter

As discussed above, one can reparameterize the model in terms of the new parameter, and NONMEM will then estimate its standard error. If re-running the fit presents a problem, there is a simple way to compute a confidence interval for a function q of a single parameter. If the lower and upper bounds of a 100(1-α)% confidence interval for $\hat{\theta}$ are denoted b_l and b_u, respectively, then a 100(1-α)% confidence interval for $q(\hat{\theta})$ has lower and upper bounds $q(b_l)$ and $q(b_u)$, respectively. This confidence interval, however, is not necessarily centered at $q(\hat{\theta})$.

4.3. Multiple Parameters

4.3.1. Correlation of Parameter Estimates

A truly new feature introduced by multiple parameters is the phenomenon of correlation among parameter estimates. NONMEM outputs a correlation matrix for the parameter estimates. The (i, j) element of the matrix is the correlation between the ith and jth parameter estimates. A large correlation (e.g. $\geq .95$) means that the conditional distribution of the ith estimate, given a fixed value of the jth estimate, depends considerably on this fixed value. Sometimes each parameter estimate of a pair that is highly correlated has a large standard error, meaning that neither parameter can be well-estimated. This problem may be caused by data that do not distinguish among the parameters very well, while a simpler model, or a different design, or more data might permit more precise estimates of each.

As a simple example, imagine a straight line fit to just two points, both at the same value of x: neither slope nor intercept can be estimated at all as long as the other is unknown, but fixing either one (simplifying the model) determines the other. Both parameters could

be estimated by observing another point at some other value of x (more data), or, still using just 2 points, by placing these points at two different values of x (modifying the design). Thus, when standard errors are large, it is useful to examine the correlation matrix of parameter estimates to see, in particular, if some simplification of the model may help.

4.3.2. Confidence Intervals for a Function of Several Parameters

There is an approximate formula for computing a standard error, and hence a confidence interval for a function of several parameters (e.g., a confidence interval for half-life when the estimated parameters are Cl and V). It uses the standard errors of the parameter estimates and the correlations between the parameter estimates. However, discussion of this formula is beyond the scope of this introduction. If a confidence interval for a function of several parameters is desired, it is often more convenient to re-fit the data with the model reparameterized to include the function as an explicit parameter.

5. Hypothesis Testing

Before going into details, a note of caution is in order about hypothesis testing in general. The logic of hypothesis testing is that one sets up a hypothesis about a parameter's value, called the null hypothesis, and asks if the data are sufficiently in conflict with it to call it into question. If they are, one rejects the null hypothesis. However, logically, if they are not, one has simply failed to reject the null hypothesis; one does not necessarily have sufficient data to accept it. An extreme example will make this clear. Let the null hypothesis be any assertion at all about the state of nature. Gather no evidence bearing on the question. Clearly, the evidence (which is void) is insufficient to reject the null hypothesis, but just as clearly, in this case the evidence provides no support for it either.

In less extreme cases there is a way to view failure to reject as lending some support to the null hypothesis, but the matter is problematic. It is somewhat less problematic to use a confidence interval to quantify support for a null hypothesis. A null hypothesis is an assertion that a parameter's true value is found among a set of null values. A confidence interval puts reasonable bounds on the possible values of a parameter. One can then say that the evidence (the data from which the parameter estimate is derived) *does* support a null hypothesis (about the value of the parameter) if the null values are included in the interval and that the evidence fully support the null hypothesis if all nonnull values lie outside. An example will help make this clear.

Consider that mean drug clearance in adults varies linearly with the weight of the individual to a *clinically* significant degree. Formally, this can be put as a statement about a regression coefficient in a model such as

$$Cl = \theta_1 + \theta_2(WT - 70), \tag{5.3}$$

The null hypothesis might be that θ_2 is close to zero, i.e. that it is not so different from zero as to be clinically significant. To make this precise, suppose that we know that mean clearance for a 70 kg person (i.e., θ_1) is about 100 ml/min. If θ_2 were .20 ml/min/kg or less, a 50 kg increment (decrement) in weight from 70 kg would be associated with less than a 10% change in total clearance. This is clinically insignificant, so the appropriate null values for θ_2 might be 0.0 to .20, assuming, of course, that a physical lower bound for the parameter is zero. (More usually in statistical discussions a set of null values consists of a single number, e.g. 0.) If the confidence interval for θ_2 only includes null values (e.g., it is .10 to .15), one might then safely conclude that weight, if it has any effect at all, has no *clinically* significant effect, and that the data fully support the null hypothesis. If

the confidence interval includes null values and others (e.g., it is 0.0 to .60), one would conclude that there is some support for the null hypothesis, but that there is also some support for rejecting it. In this case the data are insufficient to allow outright acceptance or rejection. If the confidence interval includes no null values (e.g., it is .80 to 1.3), one would reject the null hypothesis and conclude that weight has a clinically significant (linear) effect on clearance.

For these reasons, we urge caution when performing hypothesis tests and suggest that confidence intervals are often more useful. None the less, the popularity of hypothesis tests requires that they be done, and we now describe two methods for so doing, the first somewhat more approximate and less general than the second, but easier to do.

5.1. Hypothesis Testing Using the SE

A straight-forward way to test a null hypothesis about the value of a parameter is to use a confidence interval for this purpose. In other words, if the confidence interval excludes the null values, then the null hypothesis is rejected. As described in Section 4.2.2, such a confidence interval is based on the estimated standard error. This method generalizes to a hypothesis about the values of several parameters simultaneously, but this is beyond the scope of this introduction.

5.2. Hypothesis Testing Using the Likelihood Ratio

An approach that involves the extra effort of re-fitting the data has the advantage of being less approximate than the one that uses a confidence interval based on the SE. This method is the so-called Likelihood Ratio Test.

The basic idea is to compare directly the goodness of fit (as indicated by the minimum value of the extended least squares objective function) obtained between using a model in which the parameter is fixed to the hypothesized value (the *reduced* model) and a model in which the parameter must be estimated (the *full* model).

5.2.1. Definition — Full/Reduced Models

A model is a reduced model of a full model if it is identical to the full model except that one or more parameters of the latter have been fixed to hypothesized values (usually 0). Consider the examples:

E.g. #1. Valid Full/Reduced Pair:

Full model: $\tilde{Cl} = \theta_1 + \theta_2 WT$

Reduced model: $\tilde{Cl} = \theta_1$

E.g. #2. Invalid Full/Reduced Pair:

Full model: $\tilde{Cl} = \theta_1 WT$

Reduced model: $\tilde{Cl} = \theta_1$

In example #1, fixing θ_2 to 0 produces the reduced model, while in example #2, no parameter of the full model can be fixed to a particular value to yield the "reduced" model. It will always be true that if the models are set up correctly, the number of parameters that must be estimated will be greater in the full model than in the reduced model. Note that this is not so for example #2.

5.2.2. Reduced/Full Models Express the Null/Alternative Hypotheses

The reduced model expresses the *null hypothesis*; the full model expresses an *alternative hypothesis*. In example #1 above, the null hypothesis is "typical value of clearance is

independent of weight", and the alternative is "typical value of clearance depends linearly on weight."

Note an important point here: the alternative hypothesis represents a *particular* alternative, and the likelihood ratio test using it will most sensitively reject the null hypothesis *only when* this particular alternative holds. If the full model were that "the typical value of clearance is inversely proportional to weight" (so that as weight increases, the typical value of clearance decreases, a situation which rarely holds), the likelihood ratio test using the alternative we have stated would not be particularly sensitive to rejecting the null hypothesis, and we might fail to do so. In contrast, we might succeed in rejecting the null hypothesis if we used some other alternative model closer to the truth.

5.2.3. The Likelihood Ratio Test

Part of the NONMEM output is the "Minimum Value of the Objective Function" (see Chapter 2). Denote this by l. If NONMEM's approximate model were the true model, then l would be minus twice the maximum logarithm of the likelihood of the data (for those readers unfamiliar with likelihoods, and curious as to what they are, we suggest consulting a statistics textbook). Statistical theory tells us that the difference in minus twice the maximum log likelihoods between a full and reduced model can be referenced to a known distribution. Thus, to perform the Likelihood Ratio Test, one proceeds as follows.

Let l_f be the minimum value of the objective function from the fit to the full model, and let l_r be the corresponding quantity from the fit to the reduced model. Fit both models separately yielding l_f and l_r, and form the statistic,

$$C^2 = l_r - l_f$$

This statistic is approximately distributed chi-square (χ^2) with q degree of freedom, where q is the number of parameters whose values are fixed in the reduced model. For an α-level test, compare C^2 to $\chi^2_{1-\alpha}(q)$, the 100(1-α) percentile of the χ^2 distribution.

In particular, when exactly one parameter of the full model is fixed in the reduced model, a decrease of 3.84 in the minimum value of the objective function is significant at $p < .05$.

If NONMEM's approximate model (linear in the random effects) were the true model, and in addition, f were linear in the fixed effects, then C^2/q would be (approximately) distributed according to the F distribution with q, and $n - p$ degrees of freedom ($F(q, n-p)$). Since $qF(q, n-p)$ is equal to $\chi^2(q)$ only when n is "large", and is greater otherwise, it is more conservative to reference C^2 to $qF(q, n-p)$ in all instances, even when f is nonlinear.

6. Choosing Among Models

An idea related to hypothesis testing is this: when faced with alternative explanations (models) for some data, how does one use the data to determine which model(s) is (are) most plausible? When one of the models is a reduced sub-model of the other, and there is some *a priori* reason to prefer the reduced model to the alternative, then the Likelihood Ratio test can be used to test whether this a priori preference must be rejected (at the α level). However, when one gives the matter some thought, there is usually little *objective* reason to prefer one model over another on a priori grounds. For example, although possibly more convenient, a monoexponential model is, if anything, less likely on biological grounds than a biexponential.

Not only may there not be a clear *a priori* probability favoring one contending model over another, but the two models may not form a full and reduced model pair. In such circumstances, one must rely on some goodness-of-fit criterion to distinguish between the models. Consider choosing between just two models (the ideas to be discussed readily generalize to more than two), denoted model A and model B. If the number of free parameters in model A (p_A) is the same as that of B (p_B), then here is a reasonable criterion: favor the model with the better fit. Note that there is no p value associated with this statement; no hypothesis is being tested.

Unfortunately, if $p_A \neq p_B$ one cannot simply compare l_A and l_B and choose the one with the smaller value. The reason is best understood when A and B are a full and reduced model pair. The full model will *always* fit the data better (i.e., have a smaller l) as it has more free parameters to adjust its shape to the data. While the same is not always true for any pair of non-linear models with different numbers of parameters, it is often true: the model with the greater number of parameters will fit a given data set better than the model with fewer parameters. Yet the larger (more parameters) model may not really be better; it may, in fact, fit an entirely new data set worse than the simpler model if its better fit to the original data was simply because it exploited the flexibility of its extra parameter(s) to better fit some random aspect of that data.

Based on the above intuitive argument, it seems that one should penalize the larger model in some way before comparing the likelihoods. This intuition is formally realized in the Akaike Information Criterion (AIC) which says that one should compute $AIC = l_A - l_B + 2(p_A - p_B)$, and choose model B if AIC is >0, and model A if AIC is <0. The second term penalizes model A if $p_A > p_B$, and vice versa. When $p_A = p_B$, the AIC reduces to the comparison of l_A and l_B described previously.

Chapter 6 - Data Sets, $DATA and $INPUT Records, and the Data Preprocessor

1. What This Chapter is About

This chapter tells how to create data for analysis by NONMEM-PREDPP. It tells how to describe the data using $DATA and $INPUT records. The requirements for formatting the data for NONMEM-PREDPP are somewhat more stringent than are the requirements for formatting the data for NM-TRAN. The Data Preprocessor is a component of NM-TRAN which modifies the data so that it becomes formatted appropriately for NONMEM-PREDPP.

2. Data Sets for NONMEM

2.1. Data Records

A data set for NONMEM analysis consists of a series of records ("lines" in the terminology of editing programs). Each record must consist of a fixed number of data items and each must have the same format. Figure 6.1 shows how such a data set may be pictured. In data base terminology, this is a "flat" structure.

	Data item #1	Data item #2	Data item #3	...	Data item #n
Record #1					
Record #2					
Record #3					
.					
.					

Figure 6.1. A NONMEM input data set. Each data record is a row; each type of data item is in a different column.

NONMEM imposes no limit on the number of records in the data set. It does not (nor does PREDPP or NM-TRAN) sort the data records before processing them, so the data records must already be in the correct sequence. NONMEM itself cannot be instructed to delete or drop records from the data set, but see the DROP and IGNORE options of the $DATA record, below.

2.2. Data Items

NONMEM reads records from the data set with a FORTRAN FORMAT specification, and so each data item must occupy a fixed number of character positions. Data items are always numbers. However, if no particular number is appropriate for a given data item on a given record, the data item is called a null data item; it may be given the numerical value 0 or the nonnumerical value ".", or left blank. Zero's were used in the first two lines of the Theophylline example of Chapter 2, which appeared as follows:

```
2     320.       0.        0.
2       0.        .27      1.71
```

The Data Preprocessor allows each value in the data set to occupy only as many character positions as it needs, so long as the data items are separated by blanks (spaces) or commas. Tab characters may also be used as separators if they are stored as explicit characters, e.g., ASCII 011, although this is platform-dependent and should be tested carefully. When there are no commas or tabs, the value "." or 0 *must* be used to hold the place of a null data item. The two lines above could have been entered as follows:

```
2 320. 0. 0.
2,,.27,1.71
```

(Note the use in the second line of adjacent commas ",," to denote a null data item.)

The contents of the data items must be purely numeric; i.e., character values such as Y, N, M or F may not be recorded. Instead, numeric codes such as 0 or 1 must be used.

With NONMEM VI, the number of data items per data record is given by constant PD in file SIZES. The default value is 20. With NONMEM 7.1, the default value is 50. With NONMEM 7.2, there is no limit on the number of data items per data record. If the value in SIZES is not sufficient, a larger value may be specified on the $SIZES record.

2.3. Clinical Data and Data Conversion

Clinical data often has a "hierarchical" file structure, with (say) two record formats: a patient record, containing fixed information about a patient (ID number, sex, age, prior history of smoking or drug use, etc), followed by one or more visit records, containing doses and physical observations during the course of the study. Visit records may not even contain the same number of items as patient records, nor have the same format. The hierarchy is shown schematically in figure 6.2.

```
Patient record
  Visit record
  Visit record
  ...
Patient record
  Visit record
  Visit record
  ...
```

Figure 6.2. A hierarchical data file. Patient and visit records have different formats.

NONMEM cannot accept such data. For NONMEM, the (fixed) information on the "patient" record must be copied onto every "visit" record, and the "patient" records must be eliminated. This is the user's responsibility and is typically done in a one-time data conversion step using the system editor and/or a specially written computer program. If an individual's data is to be deleted because he did not complete the study or had an adverse outcome, it should be done at this time. In addition, numeric codes should be substituted for alphabetic codes. Clinical data sometimes includes multi-digit, non-consecutive patient identification numbers drawn from some patient identification system. Such patient identification numbers can be used with NONMEM as the identification data item described in Section 6.2. However, it is preferable to append to each patient's data records numbers from the sequence 1, 2, 3, ..., for use as the ID data item. This will make it easier to read a scatterplot which includes ID along one of the axes (e.g., residual vs ID).

When there is a large amount of data, we strongly suggest that a small amount of data (from one or two individuals) be prepared for NONMEM-PREDPP analysis and a run in

which only tables and scatterplots are output be made to check that the data is correctly prepared before a great deal of labor is expended on the remainder.

3. Data Sets for PREDPP

When PREDPP is used with NONMEM, the data must meet additional requirements. First, PREDPP is concerned with time-ordered events such as dose events, which introduce drug into the system at particular times, and observation events, which report observations taken at particular times. PREDPP insists that these events be recorded on separate records. That is, dosing information cannot be recorded on the same record as an observed value. Second, PREDPP requires that the time of each event be recorded on each data record, and that the physical sequence of the data records be the same as their sequence in time. (E.g., if a dose event immediately precedes an observation event in time, then the dose event record must immediately precede the observation event record.) Again, neither PREDPP nor the Data Preprocessor will physically sort or resequence the data records.

4. The $DATA Record

The $DATA record describes the characteristics of the external data file to be processed by NONMEM. NONMEM is not a data base management system and does not store a data set between runs; once a file has been prepared for NONMEM, it must be re-read each time it is to be analyzed. The first character string appearing after $DATA is the name of the file containing the data. Since it is to be used in a FORTRAN OPEN statement, this name may not include embedded special characters such as slashes (/ or \), commas, semi-colons, parentheses, equal signs or spaces unless it is surrounded by single quotes ’ or double quotes ". The filename may contain 80 characters. (If a file is to be opened by NONMEM rather than by NM-TRAN, the filename may not contain embedded spaces, and may contain at most 71 characters.) A FORTRAN format specification suitable to read the data may follow the file name; this is optional and can be supplied by the Data Preprocessor. The choice is discussed more fully in Section 10.4 of this chapter.

Certain options may be specified if desired. Among these are:

RECORDS=n

> This tells the number of records to be read from the data file. If omitted, the records are read to the end-of-file or to a NONMEM FINISH record (Users Guide II). The RECORDS option may be used to limit NONMEM processing to the initial portion of the file and is useful during the early stages of debugging.

RECORDS=label

> "Label" is a data item label. The data records for the problem will start at the place where the file is positioned before data records are read and include all contiguous data records having the same value for the data item. In particular, the ID label may be used (or alternatively, the option may be coded RECORDS=IR, RECORDS=INDREC, or RECORDS=INDIVIDUALRECORD) to obtain the data for a single individual.

IR,INDREC,INDIVIDUALRECORD.

NOREWIND|REWIND

> With the first problem specification in a control stream, the file is positioned at its initial point so that the first record in the file is used. The options REWIND and NOREWIND apply only with a $DATA record in a subsequent problem

specification.
REWIND: Reposition the file at the start.
NOREWIND: Leave the file at its current position so that the next record in the file is read. Used when the $DATA record with the previous problem specification included the RECORDS option so that NM-TRAN did not read to a physical end-of-file. This is the default.

LRECL=n

This tells the length of the physical data records. It is required if the operating system associates a fixed physical record length with every disk file *and* considers it a fatal I/O error if a READ command requests more characters than the records contain. If this is true of your operating system, the operating system will issue an error message when you first run NM-TRAN without the LRECL option in the $DATA record.

WIDE

This requests that the NONMEM data set produced by NM-TRAN always contain single-line records, and that these records always include at least one space between data items. Such a data set can be further processed by other programs. (The default is NOWIDE, in which case NM-TRAN limits the records to 80 characters by creating multi-line records and/or eliminating spaces between data items if necessary.) It may not be used if a FORTRAN format specification is present. It also provides an extra character for relative times computed by the Data Preprocessor.

NULL=c

This requests that the NONMEM data set produced by NM-TRAN contain the character c in place of null data items. For example, NULL=0 requests that all null data items be replaced by 0. The syntax NULL='c' and NULL="c" is also permitted. The default is NULL=' '. It may not be used if a FORTRAN format specification is present.

IGNORE=c

This instructs NM-TRAN to ignore data records having character c in the first character position ("column 1") of the record. This allows the use of "comment" records in the NM-TRAN data set. The syntax IGNORE='c' and IGNORE="c" is also permitted. It may be used even if a FORTRAN format specification is present.
The character @ has a special meaning. It signifies that any data record containing an alphabetic character (or special characters @ or #) as its first non-blank character (not just in column 1) should be ignored. Alphabetic characters are the letters A-Z and a-z. Thus, a table file produced by NONMEM in an earlier run can be used as an NM-TRAN data set. Any header lines included in this table can be dropped by specifying IGNORE=@.

When the IGNORE option is omitted, any records containing the character # in column 1 are ignored.

IGNORE=(list), ACCEPT=(list)

This form of the IGNORE option allows records to be dropped based on the values of data items. For example,

```
IGNORE=(GEN.EQ.1,AGE.GT.60).
```

Records having GEN equal to 1 or AGE greater than 60 are dropped. All others are accepted. The ACCEPT option allows records to be accepted based on the values of data items. FORTRAN logical operators .EQ., .NE., .GT., .GE., .LT., .LE. may be used, as well as FORTRAN 90 logical operators ==, /=, >, >=, <, <=.

Special operators .NEN. and .EQN. request that character strings be converted to numeric prior to being compared (nm73). See Guide VIII for more information.

LAST20=nn

"nn" is a 2 digit number that specifies the highest 2-digit year that is assumed to be in the 21st. century, i.e., that represents 20nn rather than 19nn. See Section 10.1.5 below.

TRANSLATE

The translate option must be followed by parentheses enclosing a list of one or more translate specifications. For example,

```
$DATA filename TRANSLATE(TIME/24,II/24)
```

Translate specification `TIME/24` causes the value of TIME to be divided by 24, whether or not day-time translation occurs (i.e., whether or not relative times are being computed). This has the effect of changing the unit of TIME from hours to days. Similarly, translate specification `II/24` causes the value of II (interdose interval) to be divided by 24 whether or not ":" appears in any II value. With NONMEM 7.3, any value may be given for dividing time and II values, and any precision may be requested. See Section 10.1.4 below.

5. The $INPUT Record

This record describes how many data items there are on each data record, the order of the data items, and tells what the labels of the data items are.

5.1. Data Item Labels

A data item label is one to four letters (A-Z) or numerals (0-9). With NONMEM 7 a label consists of 1-24 letters (A-Z), numerals (0-9), and the character '_'. (The length 24 is specified by constant SD in SIZES)
The first character must be a letter. These labels are the ones which will be used in other records (such as $PK or $SCATTERPLOT), and will appear in NONMEM's output. The order of the data items on the data records is not important, but must be the same on all data records in the data set. In both the examples of Chapter 2, the ID data items happened to be the first ones in the data records, and the DV data items happened to be the last ones. This order was arbitrary.

5.2. Reserved Labels and Synonyms

Certain data item labels are reserved in that they identify data items which are recognized specifically by NONMEM, PREDPP, or NM-TRAN. The data items they label are themselves called NONMEM, PREDPP, or NM-TRAN data items, respectively.

- Reserved NONMEM data item labels are: ID, L2, DV, and MDV. They are discussed in Section 6 of this chapter and in Section 4.2 of Chapter 12. Additional reserved NONMEM data item labels are: MRG_, RAW_, and RPT_. See Guide VIII for a discussion of these items.
- Reserved PREDPP data item labels are: TIME, EVID, AMT, RATE, SS, II, ADDL, CMT, PCMT, CALL, and CONT. They are discussed in Section 7 of this chapter and in Section 2.4 of Chapter 12. With NONMEM 7.2, additional reserved PREDPP data items are the extra EVID labels, XVID1, XVID2, XVID3, XVID4, and XVID5. See Guide VIII for a discussion of these items.

- Reserved NM-TRAN data item labels are: DATE, DAT1, DAT2, DAT3, and L1. DATE, DAT1, DAT2, and DAT3 are discussed in Section 10.1 of this chapter; L1 is discussed in Section 4.2 of Chapter 12.

If you do not want to use the reserved label, you can supply two labels: the reserved label and a "synonym". Either label can be used in subsequent records, but only the synonym will appear in NONMEM output. For example,

```
$INPUT PNO=ID,CONC=DV,DOSE=AMT,WT,....
```

The first three data items are given the labels PNO, CONC, and DOSE. These labels are synonyms for the NONMEM data items ID and DV and for the PREDPP data item AMT. The last data item is given the label WT and is not a reserved data item; it is an example of fixed effect ("concomitant") data *x*

When $PK and $ERROR records are present, certain labels may not be used at all as data item labels. These are: the labels for the basic and additional PK parameters for the pharmacokinetic model, as listed in Appendices 1 and 2 (e.g., for ADVAN1 and TRANS2: CL, V, S1, S2, F1, F0), and specific labels for the arguments of the PK and ERROR subroutines: IDEF, IREV, N, GG, IRGG, HH, and G.

5.3. Dropping Data Items via DROP

If no format specification is included on the $INPUT record, then another synonym, DROP, may be used with any data item. DROP may be used as a synonym more than once. It identifies data items to be dropped (i.e. eliminated) from the NM-TRAN data set by the Data Preprocessor while constructing the NONMEM data set. This provides a way to limit the number of data items in the NONMEM data set and to eliminate non-numeric data items.

6. NONMEM Data Items

6.1. DV Data Item

There must always be a Dependent Variable data item labeled DV. This is a value of an observation. There can be only one DV data item per data record. The position of the DV data item (and the ones described below) is not important. However, its position must be the same on all records.

6.2. ID Data Item for Population Data

When the data is from a population, NONMEM expects the Identification data item, labeled ID, and expects the data to be organized into two or more "individual records". An individual record is a group of contiguous data records having the same value for the ID data item and presumably containing data from the same individual. ID data item values need not be consecutive, increasing, unique, nor begin with 1. E.g., 3, 5, 6, 10, 3, etc. is a possible sequence of ID values. Note the two instances of 3 as ID data item values. As long as these two instances are separated by different ID data item values (e.g. 5, 6, 10), they represent different individuals.

6.3. MDV data item

If there are records in an input data set which do not contain values of observations, then NONMEM needs to be informed of this fact. This is done using the Missing Dependent Variable data item labeled MDV. The values of MDV are:

0 The DV data item of the data record contains a value of an observation. The record is referred to as an observation record.

1 The DV data item of the data record does *not* contain a value of an observation.

NONMEM 7 limits the number of observation records per individual record to 250. To change this limit, see Users Guide III. With NONMEM 7.3, there is no limit on the number of observation records.

When PREDPP is used, the Data Preprocessor is able to recognize which records contain observed values and which do not, and it can supply the MDV data item if it is not already present in the data set, i.e. if the label MDV does not occur in the $INPUT record. (When PREDPP is not used, the Data Preprocessor cannot do this.)

7. PREDPP Data Items

7.1. TIME Data Item

PREDPP will in general need the
Time data item, labeled TIME. With NONMEM 7.4, the value of TIME may be negative. With earlier versions of NONMEM, the value of TIME must be non-negative. Within an individual record, values of TIME may not decrease. (Exceptions exist for reset and reset-dose events; see Section 7.3.) The units are optional (e.g., minutes or hours), but should be consistent with other units used in the problem. The TIME of the first event record may be zero or non-zero. (If non-zero, then PREDPP in effect subtracts this value from all other TIME values within the same individual record, so that PREDPP always works with relative time values.) The Data Preprocessor permits TIME to be expressed as clock time (e.g., 8:30, representing the time, half-past 8 o'clock). Such times are converted by the Data Preprocessor into relative times. Details are given in Section 10.1 below.

7.2. AMT, RATE, SS, II: Dose-related Data Items

Doses are described using one or more of these four data items, depending on the kind of dose. A detailed discussion of these data items and of dose records in general is deferred to Section 8 below.

7.3. EVID Data Item

When PREDPP is used, all data records are also called event records. Every event record must contain an Event Identification data item identifying the kind of event described by the record, and labeled EVID. The values of EVID and the five kinds of event records are:

0 Observation event. This record contains an observed value (in the DV data item). Dose-related data items such as RATE and AMT must be 0.

1 Dose event. This record describes a dose. The contents of the DV data item are ignored.

2 Other event. This record is used for a variety of purposes. It can be used to obtain a predicted value at a point in time at which no actual observation or dose event took place; it can be used to turn a compartment off or on at a point in time; it can be used to mark a time at which a change in a physiological data item (e.g. weight) occurs (as well as give the new value of the data item). Dose-related data items must be 0. The contents of the DV data item are ignored.

3 Reset event. This record is used to reset the kinetic system at some point in time, without actually starting a new individual record: time is set to *whatever* time appears in the event record, the amounts in each compartment are set to zero, prior doses are cancelled, and the on/off status of each compartment is set to its initial status. It is in all other respects identical to an other event type record. It is typically used within an individual record, when the individual had a course of drug treatment, followed by a wash-out period, followed by another course of drug treatment. It should appear prior to the start of the second course.

4 Reset-dose event. This record combines EVID types 3 (reset) and 1 (dose). First the system is reset, and then a dose is introduced. It is in all other respects identical to an ordinary dose event type record.

If only dose and observation event records are present in the NM-TRAN data set, and if EVID is not already present in the data set (i.e. EVID does not appear in the INPUT record), then EVID will be supplied automatically by the Data Preprocessor. This is what was done in the examples of Chapter 2. If other or reset type event records are present in the data set, then the $INPUT record must include the EVID data item, and the data set must include the appropriate values for EVID on *all* the data records.

7.4. CMT and PCMT Data Items

The Compartment data item (CMT) and Prediction Compartment data item (PCMT) are similar. Both contain the number of a compartment in the model. (Compartments and compartment numbers are discussed in Chapter 7 and Appendix 1, as are default compartments. It may help to look at Chapter 7 and Appendix 1 at this time.) If CMT or PCMT is not defined in the data set (i.e., not listed in the $INPUT record), or has the value 0 on a given event record, the appropriate default compartment is used, except as noted below. This is what was done in the examples of Chapter 2. The meaning of the two data items depends on the particular kind of event record.

- Observation event: CMT specifies the compartment from which the predicted value of the observation is obtained. PCMT is ignored. When CMT specifies the output compartment, it is allowed to have a negative sign (e.g., with the One-compartment model, CMT may be -2). This signals that *after* the prediction is computed the output compartment is to be turned off, i.e. the amount in the compartment is to be set to zero. The amount remains zero until the compartment is subsequently turned on. This is quite useful with urine observations; see Section 9 below.†
- Dose event: CMT specifies the compartment into which the dose is introduced. The compartment is turned on if it was previously off. PCMT specifies the compartment for which a predicted observation is computed. This predicted value is not associated with an observation, but it can be useful because it will appear in tables or scatterplots.
- Other event: A positive value of CMT specifies that the compartment is to be turned on if it is off. A negative value of CMT specifies that the compartment is to be turned off if it is on. (If CMT is 0, no compartment is turned on or off.) PCMT is the same as for dose events.
- Reset event: CMT is ignored. PCMT is the same as for dose events.
- Reset-dose event: CMT and PCMT are the same as for dose events.

† This is also permitted with output-type compartments; see Chapter 12, Section 2.8.

7.5. CALL Data Item

The Call data item (CALL) is used to force a call to either or both of the PK and ERROR subroutines with the event record when such a call would not normally occur. A call to the PK or ERROR subroutine causes the code specified by the $PK or $ERROR records, respectively, to be executed with the event record. This is discussed in Chapters 7 ($PK) and 8 ($ERROR).) When not defined in the data set, CALL is assumed to be 0 always. The values are:

0 No forced call; PREDPP takes its normal action.

1 Force a call to ERROR.

2 Force a call to PK.

3 Force a call to both PK and ERROR.

10 Force a call to ADVAN9. May be combined with other values. E.g., the value 12 means "Force a call to PK and to ADVAN9".

8. Describing Doses to PREDPP

Doses are described using one or more of the data items discussed below. A detailed discussion of the actual kinds of doses that PREDPP recognizes follows in Section 8.2, including a precise definition of what is meant by the term "steady-state dose" (Section 8.2.3). A data item that is not needed to describe the kinds of doses used in the study need not be defined in the data set; it will in effect always have the value 0. Only AMT (Dose amount) was used in the examples of Chapter 2, for example. The values of dose-related data items should be 0 for non-dose events and for those dose events to which they are not relevant.

8.1. Dose-related Data Items

AMT data item

The Amount data item (AMT) gives the amount of a bolus dose or of an infusion of finite duration. This amount should be a positive number.

RATE data item

The Rate data item (RATE) gives the rate of an infusion. This rate should be a positive number. (Negative values are discussed in Chapter 12, Section 2.3.)

SS data item

The Steady-state data item (SS) can take four values.

0 This record does not describe a steady-state dose.

1 This record describes a steady-state dose. If this is not the first event record for the individual, then the system is first reset as if by a reset event record (except that the on/off status of the compartments is unchanged from what it was prior to the event record and the time on the event record must not be less than the time on the previous event record). The compartment amounts are then computed using steady-state kinetic formulas.

2 This record describes a steady-state dose. No reset of the kinetic system occurs. Compartment amounts are computed using steady-state kinetic formulas and are then added to the amounts already present at the event time. The use of SS=2 will be discussed further in Section 8.2.7, below.

3 This record describes a steady-state dose. It is exactly like a steady-state dose with SS data item = 1, except that existing compartment amounts and derivatives are

retained and used as initial estimates. The computed steady-state levels replace these compartment amounts and derivatives. This value of SS may be specified only with SS6 and SS9 (the General Nonlinear Models).

II data item

The Interdose Interval data item labeled II gives the time between implied doses (see Section 8.2.3 and Chapter 12, Section 2.4). For a steady-state infusion, it should be 0. For other steady-state doses, it should be a positive number whose units are the same as the TIME data item.

8.2. Different Kinds of Doses

Any of the doses described here may be introduced into any compartment of the model except the output compartment. Examples are given below that are fragments of data records, identifying the data items of interest and showing their contents on the dose record. The units of various data items are presumed to be appropriate for some actual data.

8.2.1. Instantaneous Bolus Doses

All the examples in Chapter 2 involve instantaneous bolus doses, which we shall refer to simply as bolus doses. (There is also such a thing as a "zero-order bolus dose", see Chapter 12, Section 2.3.) These are dose records having AMT>0, RATE=0 and SS=0. (Recall that if RATE and SS are not defined on the $INPUT record, they are effectively 0.) If the $PK record computes a bioavailability fraction parameter for the compartment into which the dose is introduced, then the contents of the AMT data item is multiplied by the current value of this parameter before the amount is added to the compartment. A bolus dose enters the dose compartment immediately; the predicted (scaled) amount in the dose compartment, if displayed in a table or scatterplot, will include the dose.
Example:

```
TIME AMT
4.   10.
```

This is a dose of 10 to be added to the default dose compartment at time 4.

A bolus dose to the central compartment might be interpreted as an IV bolus dose; to the depot it might be an oral tablet; to a peripheral compartment it might be an intra-muscular injection.

8.2.2. Infusions

Infusions are doses having AMT>0 and RATE>0. The duration of the infusion is computed by PREDPP by dividing the AMT by the RATE. As with bolus doses, AMT is first multiplied by the bioavailability parameter for the dose compartment, if any. There is no explicit "end of infusion" record. Drug amounts in the system cannot be affected in a detectable way at the time an infusion begins by any drug introduced by the infusion; the predicted (scaled) amount in the dose compartment, if displayed in a table or scatterplot, will not include the dose. Infusions may overlap. That is, subsequent dose records may start new infusions before old ones have finished. It is not an error if an infusion's duration is so large as to extend beyond the time of the last event record for the individual; the remainder of the drug is ignored. A reset or reset-dose event, or a steady-state dose event with SS=1, will also terminate any infusions in progress.
Example:

```
TIME AMT RATE
4.   10.  2.
```

The duration of the infusion will be computed as 10./2., and so the infusion, which begins at time 4, will terminate at time 9. (=4.+5.).

An infusion to the central compartment might be interpreted as an IV infusion; to the depot it might be a sustained release tablet; to a peripheral compartment it might be an implant or skin patch which releases drug at a known constant rate. It is possible for NONMEM-PREDPP to estimate the input rate of a constant-rate drug delivery system (see Chapter 12, Section 2.3).

8.2.3. Steady-State Doses

A steady-state dose can be regarded as the last one of a series of doses just like the one specified in the dose event record, which have been given at a regular interdose interval since time $-\infty$, and such that they have led to a steady-state periodic pattern of drug amounts in the system by the time this last dose has been administered. The doses of similar kind that precede it are called implied doses, because their existence is not described by separate dose records in the data set, but rather is implied by the description of the single steady-state dose. By stipulating that a dose is a steady-state dose, the user instructs PREDPP to update the drug amounts in the system at the time the dose is given by using steady-state kinetic formulas. This can take less computational time than using separate dose records to describe the implied doses and using transient kinetic formulas to advance the system from one dose time to the next (as well as requiring fewer dose records). The formulas used to compute the steady-state amounts at the time the steady-state dose is introduced use the values of the basic and additional pharmacokinetic parameters in effect at this time; any values in effect at earlier times are ignored. Moreover, when using a steady-state dose, the user is assuming that under reasonable values of the pharmacokinetic parameters, steady-state is in fact effectively reached by the time the dose is introduced; PREDPP does not check this assumption. The output compartment must be off when a steady state dose record is encountered in the data set.

(The Model Event Time (MTIME) feature described in Chapter 12 does not apply during steady-state computations. The Absorption lag (ALAG) feature described in Chapter 12 does apply. See Guide VI, Chapter V, Notes 3 and 4.)

8.2.4. Steady-State with Multiple Bolus Doses

These are dose events having AMT>0, RATE=0, SS=1, and II>0. The II data item (interdose interval) tells how many time units apart the doses were given. As with non-steady-state bolus doses, AMT is first multiplied by the bioavailability parameter for the dose compartment, if any.

Figure 6.3 shows how drug levels vary with time. The concentration-time profiles over each interdose interval look the same since, in principle, there is an *infinite* number of implied doses.

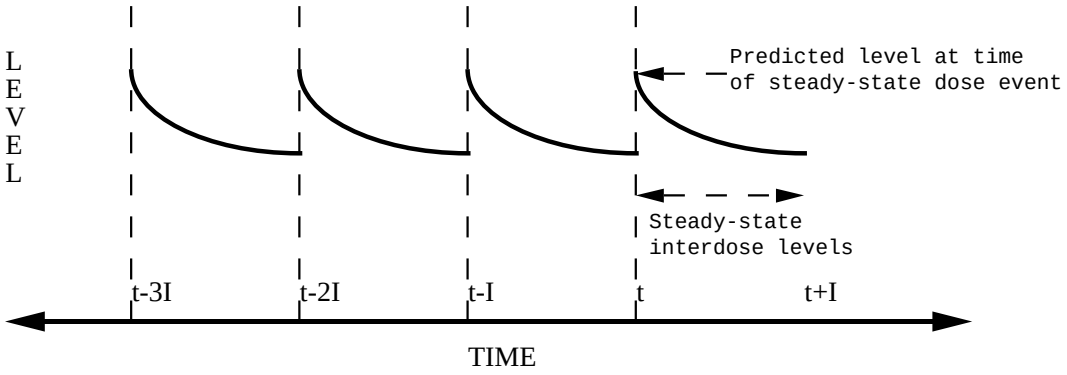

Figure 6.3. Steady-state with multiple bolus doses. The dose is given at time t. The interdose interval is I. Steady-state levels can be predicted between times t and t+I.

Example:

```
  TIME AMT SS II
    8  10.  1 12
```

Here, an infinite number of bolus doses, 10 units each, are assumed to have been given 12 hours apart, with the last of these given at time 8AM, at which time steady-state is reached. The fact that TIME is 8 has no effect on the computed amounts, but is important in relation to the records that follow. Steady-state levels can be predicted at any time between the time on the dose record (8) and the end of the succeeding interdose interval (12) (provided there are no further doses introduced *during* this interval). If another (steady-state or *non-steady-state*) dose just like the steady-state one is introduced at time 20, then predictions in the interdose interval following this time will also be steady-state levels.

8.2.5. Steady-State with Multiple Infusions

These are dose events having AMT>0, RATE>0, SS=1, and II>0. Each such event describes the last of a series of regularly spaced infusions, all of the same amount and rate. As with a non-steady-state infusion, the duration of each infusion is given by AMT/RATE. The bioavailability fraction applies to each infusion of the series.

Figure 6.4 shows how drug levels vary with time. The concentration-time profiles over each interdose interval look the same since, in principle, there is an *infinite* number of implied doses.

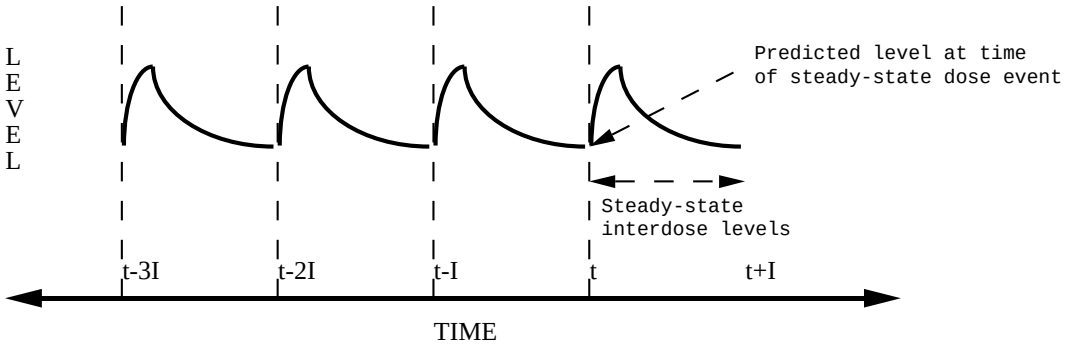

Figure 6.4. Steady-state with multiple infusions. The dose is given at time t. The interdose interval is I. Steady-state levels can be predicted between times t and t+I.

Example:

```
  TIME AMT RATE SS II
   16  10.  5.   1  6
```

Here, infusions, each 10 units and of duration 2 (=10/5), are assumed to have been given 6 hours apart, with the last of these started at time 4PM, at which time steady-state is reached. The daily dose times were 4 AM, 10 AM, 4 PM, and 10 PM. Again, the value of TIME has no effect on the computed amounts but is important in relation to the records that follow. Steady-state levels can be predicted between times 16 (4 PM) and 22 (10 PM) (provided there are no further doses introduced *during* this interval).

8.2.6. Steady-State with Constant Infusion

These are dose events having AMT=0, RATE>0, SS=1, and II=0. Such an event consists of infusion with the stated rate, starting at time $-\infty$, and *ending* at the time on the dose event record. Bioavailability fractions do not apply to these doses.
Figure 6.5 shows how drug levels vary with time.

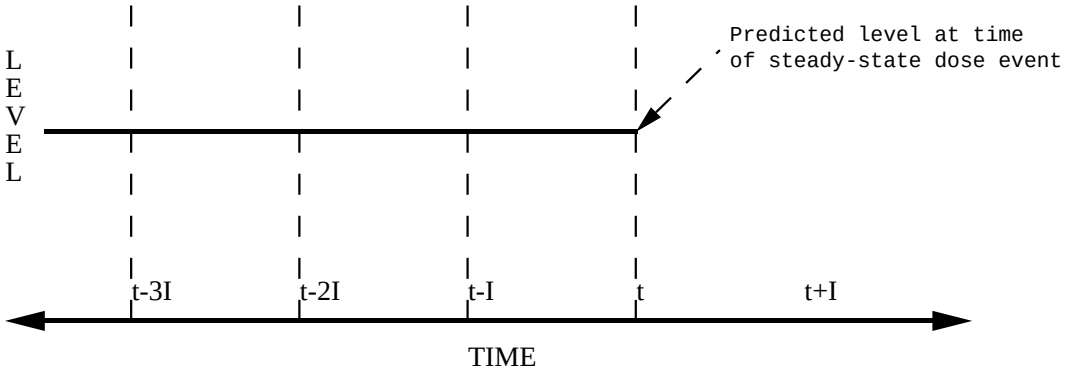

Figure 6.5. Steady-state with constant infusion. Steady-state level can be predicted only at time t.

Example:

```
  TIME RATE SS
   16   2.   1
```

Here, a steady-state infusion at rate 2 is specified as ending at 4 PM. A steady-state level can be predicted only at this time.

8.2.7. Multiple Steady-State Doses

Doses with SS=2 are exactly like doses with SS=1. Doses with SS=2 are similar to non-steady-state doses in that compartment amounts are computed in two steps. First, compartment amounts are computed at the time on the dose event record based on the prior dosing history of the system. Second, steady-state amounts are computed from the dosing information on the record and added to the existing compartment amounts. Thus, if the kinetics are linear, this results in an application of the superposition principle wherein the amounts in the system resulting from doses described by dose event records preceding the time of the steady-state dose are superposed on the (steady-state) amounts in the system resulting from the steady-state dose and the implied doses.

As with any steady-state dose, the steady-state amounts are obtained using the values of the pharmacokinetic parameters computed from the information on the steady-state dose record. In the case that SS=2, though, if these values differ from those computed from the information on the previous dose record(s), then the compartment amounts at the time in the steady-state dose record are not truly steady-state amounts, and the computed steady-state levels are not valid predictions. PREDPP will not detect this error. We emphasize that superposition is only valid with linear kinetic systems; all the kinetic systems (ADVANs) discussed in this text are linear.

SS=2 records can be used to achieve the specification of complicated dosing regimens. For example, Figure 6.6 shows how drug levels vary with time when two different doses are alternated. In this illustration, two steady-state doses are specified, each with interdose interval I and with time between the two steady-state doses equal to I/2. Even more complex patterns are possible.

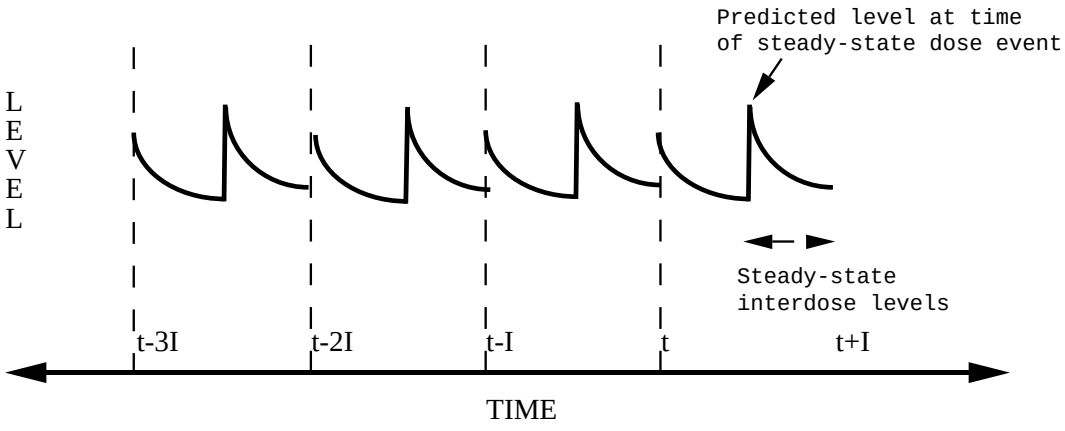

Figure 6.6. Multiple steady-state doses. Two separate steady-state doses are given. As pictured, they are each bolus doses, but they do not have to be. The first dose event record is at time t. The second dose event record is at time t+I/2. They each have interdose interval I. Steady-state levels can be predicted between times t+I/2 and and t+I.

Example:

```
  TIME AMT SS II
    8  10.  1 24
   20  15.  2 24
```

This describes the following dosing regimen: a dose of 10 units every morning at 8 AM and a dose of 15 units every evening at 8 PM (20 hours is 12 hours past 8). Note that steady-state is not truly established until *after* the second dose record; any observation

events interposed between the two dose records will reflect only the first steady-state dosage (i.e., 10 units every 8 AM). Another way to achieve the same steady-state is by the following:

Example:

```
TIME AMT SS II
 20  10.  1 12
 20   5.  2 24
```

This describes doses of 10 units every 12 hours, the last of which is given at 8 PM (i.e. at 8 AM and 8 PM daily), plus additional doses of 5 units at 8 PM daily. In both examples, the steady-state levels can be predicted from time 20 hours to time 32 hours.

8.2.8. Combining Non-Steady-State Doses with Steady-State Doses

Non-steady-state dose records may appear before, among, or after steady-state dose records. Such a dose record may occur *before* a steady-state dose record to reflect a transient dose given among a series of regular doses leading to steady-state, but which is not a part of this series. E.g., a patient who has been maintained at steady-state takes an extra dose by mistake shortly before his appointment. A non-steady-state dose record may occur *after* a steady-state dose record in order to continue the pattern of doses beyond the steady-state dose. Ordinarily, steady-state levels can only be predicted between t_1, the time on the steady-state dose record, and t_2, the sum of t_1 and the interdose interval. If it is not only necessary to compute a steady-state prediction between t_1 and t_2, but also after t_2, then there must also occur one or more non-steady-state dose records at t_2, $t_2 + I$, etc. with doses just like the steady-state dose. (The "additional doses" data item, labeled ADDL, is especially useful for this purpose; see Chapter 12, Section 2.4.)

Example:

```
TIME AMT SS II
  8  10.  1 24
 20  15.  2 24
 32  10.  0  0
 44  15.  0  0
```

Here, the last two records continue the steady-state pattern of the first two. Steady-state levels may be predicted between times 20 and 56.

Similarly, a steady-state constant infusion may be extended with a non-steady-state infusion. In the example below, steady-state levels can be predicted from time 0 to time 100.

```
TIME RATE  AMT SS
  0   30    0.  1
  0   30 3000.  0
```

9. The Output Compartment: Urine Collections and Observations

In this section we show how urine collections and observations of urine concentration, C_u, can be described. The first-time reader may prefer to return to this section after reading Section 4.3.3 of Chapter 7. As an example, consider the one-compartment model with first-order absorption (ADVAN2). The sequence of events is:

6:00 AM A bolus dose of 100 is given.
8:00 AM A urine collection is started.
9:30 AM C_u and urine volume (UVOL) are measured and a new collection is started.
11:45 AM C_p, C_u, and urine volume are measured.

The $INPUT record is:

```
$INPUT ID TIME EVID UVOL DV CMT AMT
```

The data records appear as follows:

```
 ID  TIME EVID UVOL  DV   CMT  AMT
  1  6.00   1    0   0      1  100
  1  8.00   2    0   0      3    0
  1  9.50   0   75  .058   -3    0
  1  9.50   2    0   0      3    0
  1 11.75   0  100  .067   -3    0
  1 11.75   0  100  5.80    2    0
```

Notice that urine collections start with an other type event record (EVID=2) whose CMT contains the number of the output compartment, the effect of which is to turn this compartment on at 8AM, i.e. to begin accounting for the amount of drug appearing in this compartment from 8AM. Because other type event records are included, the EVID data item *must* be present in the data. The CMT data item must be present in all event records since it is needed to refer to the output compartment in at least one record. Care must be taken to use correct values for the CMT data item; default values used when this data item is not present are not relevant in this case. The DV value on the observation record at 9:30 is the measured C_u. Because the value of CMT is negative, the output compartment is also turned off at 9:30. Since the collection is to continue, the compartment must be explicitly turned on again (the fourth record). Note that UVOL is recorded on both observation records at time 11:45. Strictly speaking, it need only be recorded on the second (C_u observation). This point is discussed further in Chapter 7, Section 4.3.3. See also Chapter 12, Section 7, for a modification to this data file for output-type compartments.

10. The Data Preprocessor

This section discusses in more detail the ways in which the Data Preprocessor can modify data, and discusses when a format specification should be included in the $DATA record.

10.1. Day-time Translation

10.1.1. TIME Data Item

Sometimes the data contains clock times hh:mm (e.g., the time 1:30 PM is recorded as 13:30). With NONMEM 7.3, clock times may also include the seconds hh:mm:ss. These times must be converted to decimal-valued times (e.g., 13.5). The Data Preprocessor can perform this task when it is processing unformatted data. Within an individual record, the Data Preprocessor replaces the TIME value in the first data record with 0, and then replaces subsequent records' TIME values with the relative time (i.e., the number of hours elapsed since the first record). (The TIME value is also reset to 0 on a reset (EVID=3) or reset-dose (EVID=4) record.) Here is an example of relative time calculation:

```
Contents of original file:     Contents of new file:
 ID    TIME                     ID    TIME
  1    9:15                      1    0.00
  1    9:30                      1    0.25
  1    10                        1    0.75
  1    14:40                     1    5.42
  1    32.5                      1   23.25
  2    8                         2    0.00
  2    8.0                       2    0.00
  2    44:50                     2   36.83
  2    58                        2   50.00
```

The presence of the colon ":" in the TIME data item of at least one record of the data causes the Data Preprocessor to convert all the TIME values to elapsed values. Elapsed times are also called relative times. Note that recorded data (lines 5, 8, and 9 of the original file) spanned more than one day. The user had to add 24 to the TIME values on each day subsequent to the first to communicate the correct times to the Data Preprocessor.

10.1.2. DATE Data Item

Here is another way the above data could have been recorded, using a data item called DATE whose value is 1 for the first day, 2 for the second day, and so on. This allows TIME values to be recorded more naturally using values in the range 0-24.

```
Contents of original file:     Contents of new file:
 ID    DATE     TIME            ID    DATE    TIME
  1       1     9:15             1       1    0.00
  1       1     9:30             1       1    0.25
  1       1     10               1       1    0.75
  1       1     14:40            1       1    5.42
  1       2     8.5              1       2   23.25
  2       1     8                2       1    0.00
  2       1     8.0              2       1    0.00
  2       2     20:50            2       2   36.83
  2       3     10               2       3   50.00
```

The DATE data item is of significance only to the Data Preprocessor; NONMEM-PREDPP will not make use of it. Even if there are no ":" characters among the TIME values, the existence of a DATE data item will cause the Data Preprocessor to replace TIME values by relative times.

10.1.3. Calendar Dates

The Date data item (DATE) can also be used to record calendar dates in month-day-year format. Any alphabetic character (e.g., / or -) can be used to separate the components. Here is a third way the same example could be recorded:

```
Contents of original file:          Contents of new file:
 ID     DATE=DROP     TIME           ID      TIME
  1     10-1-86       9:15            1      0.00
  1     10-1-86       9:30            1      0.25
  1     10-1-86       10              1      0.75
  1     10-1-86       14:40           1      5.42
  1     10-2-86       8.5             1     23.25
  2     10-12         8               2      0.00
  2     10-12         8.0             2      0.00
  2     10-13         20:50           2     36.83
  2     10-14         10              2     50.00
```

This example illustrates two features. First, when calendar dates are used, the DATE item should be specified as "DATE=DROP", so that the data item is omitted from the new data file (see Section 5.3). Otherwise, the alphabetic characters which separate the components will cause read errors when NONMEM reads the data. Second, the year value is optional; only month and date were actually needed. (Within a single individual record, however, either all dates should specify a year, or none should.)

Data labels DAT1, DAT2, and DAT3 are also recognized by the Data Preprocessor and can be used instead of DATE. The label given to the Date data item describes its format:

DATE	month day year
DAT1	day month year
DAT2	year month day
DAT3	year day month

If only one of the three components is present, it is assumed to be the day†. If only two components are present, they are assumed to be month and day (with DATE and DAT2) or day and month (with DAT1 and DAT3). The year may be omitted or given as 1, 2, 3, or 4 digits.

10.1.4. Converting Hours to Days and More General Conversions

The units of the relative TIME values resulting from the Data Preprocessor's day-time translation are hours. If the correct units for relative time should be days, then the TRANSLATE option of the $DATA record may be used to request that hours to be converted to days. For example,

```
$DATA filename TRANSLATE(TIME/24)
```

or

```
$DATA filename TRANSLATE(TIME/24.000)
```

With the former, values of TIME have two significant digits, e.g., xxxx.xx. With the latter, they have three significant digits, e.g., xxxx.xxx.

With NONMEM 7.3, more general conversions are possible. Any value may be given for dividing time and II values, and any precision may be requested. Examples are:

```
$DATA filename TRANSLATE(TIME/1.0000)
```

or

```
$DATA filename TRANSLATE(TIME/1/4)
```

for formatting times in FDATA with 4 digits to the right of the decimal. Another example is

† In this case *only*, the Date data item may be zero or negative. Day -1 means one day prior to day 0.

```
$DATA filename TRANSLATE(II/0.01/6)
```

which divides II values by 0.01, and writes 6 digits to the right of the decimal for the II data item. See Guide VIII for more information.

10.1.5. The Year 2000 - LAST20

The user may supply 4 digit years starting (e.g.) "19" and "20", and such dates are processed correctly. (Three digit years "000"-"999" are permitted, but would represent exactly those years, and should not normally be used.) If the year is omitted, it is assumed to be a non-leap year. A problem arises when the year supplied, but has only 1 or 2 digits. Such years are assumed by default to be in the 1900's. If this is not a correct assumption, two errors may be made by the Data Preprocessor when computing relative times. First, 1900 was not a leap year, but 2000 is a leap year. Hence, if consecutive dates in a data file are 02-28-00 and 03-01-00 (signifying February 28 and March 1, 2000), an elapsed time of 24 hours, rather than 48 hours, is computed. Second, if consecutive dates have years 99 and 00, the computed elapsed time is negative and an error message is generated.

With NONMEM V and later versions there is a constant LAST20. The value of LAST20 is a 2 digit number nn that specifies the highest 2-digit year that is assumed to be in the 21st. century, i.e., to represent 20nn rather than 19nn. For example, with `LAST20=50`, then 1 and 2 digit years are interpreted as follows:

00-50 represents 2000-2050
51-99 represents 1951-1999

The elapsed time between 02-28-00 and 03-01-00 is calculated to be 48 hours, and the elapsed time calculated between the years 99 (1999) and 00 (2000) is positive.

There are two ways that a value for LAST20 can be specified.

First, when NM-TRAN is installed, a value is given to constant LAST20 in TrGlobal.f90 (in the resource directory): DATA LAST20
The default value of this constant in the distribution medium is 50. Please ask your system support department if they modified the LAST20 constant when NM-TRAN was installed.

Regardless of what value was assigned to the LAST20 constant in TrGlobal.f90, there is an option LAST20 of the $DATA record that may be used to specify the value of the constant for the current run. For example:

```
$DATA filename LAST20=50
```

This insures that all 1 and 2 digit years are interpreted as above (2000-2050; 1951-1999).

10.1.6. Leap Year Warning - LYWARN

There may be two circumstances such that 1 or 2 digit years are recorded as 00, 01, ... (equivalently, 0, 1, ...). First, these may represent the years 2000, 2001, etc. Or, they may represent years 0, 1, etc., of a study. Suppose the latter is the case, and that none of the years of the study was a leap year. Then if LAST20 is set to a value greater than -1, the year 0 is assumed incorrectly to be the leap year 2000, and elapsed times may be computed incorrectly. The Data Preprocessor issues a warning message under the following circumstances:

1) The year is recorded as "00" or "0",
2) The value of LAST20 is greater than -1 by default (so that the year is understood to be 2000), and

3) The LAST20 option of the $DATA record was not used to modify LAST20 for this run.

The warning message is as follows:

```
(DATA WARNING  3) RECORD     3, DATA ITEM  3: 01-01-00
THE YEAR IS ASSUMED TO BE 2000 (A LEAP YEAR). IF THIS IS INCORRECT, USE
$DATA'S LAST20 OPTION TO OVERRIDE THE DEFAULT VALUE OF LAST20 IN NMTRAN'S
ABLOCK, OR CHANGE THE DEFAULT: 50
```

Suppose these warning messages are appropriate, that is, year "00" (or "0") should not be a leap year. The LAST20 option of the $DATA record may be used to specify that such years are in the 1900's for the current data set:

```
$DATA filename LAST20=-1
```

A constant LYWARN is defined in NM-TRAN's ABLOCK module. The default value of LYWARN is 1 ("data warning message 3 enabled"). If the value of LYWARN is set to 0 ("data warning message 3 disabled") and NM-TRAN is recompiled, then the warning message is suppressed for all runs.

10.2. Interdose Interval (II) Conversion

When the input data is unformatted and PREDPP is being used, the Interdose Interval (II) data item is checked for values containing a colon (:). Any such value is assumed to signal a clock time hh:mm. The minutes portion is converted to a decimal number containing as many decimal places as digits in the original. E.g., the value ":30" is replaced by ".50". This conversion is performed whether or not day-time translation is also being done.

10.3. Data Items Generated by the Data Preprocessor

When the data is from a single individual, the Data Preprocessor will almost always generate an ID data item‡. It does this whether or not PREDPP is used. This is done because, when the data is from a single individual, the ID data item must take on very special non-constant values for NONMEM. The generated ID data item is given the label ".ID." (i.e., ID surrounded by dots). If this data item is to be shown in any NONMEM output (e.g., in a table), it must be referred to on subsequent NM-TRAN records by this label.

When PREDPP is used, the Data Preprocessor will always generate the required EVID data item if it is not already listed on the $INPUT record. (This was discussed in Section 7.3 above.)

When PREDPP is used, the Data Preprocessor will always generate the MDV (Missing Dependent Variable) data item if it is not already listed on the $INPUT record.

These data items are generated by the Data Preprocessor whether or not a format specification was coded on the $INPUT record. They are appended to the end of each data record.

10.4. When Must a Format Specification be Included or Omitted?

When coding the $DATA record, you will need to decide whether to include a FORTRAN format specification describing the data file or to omit it and let the Data Preprocessor construct it. Here are some guidelines in making this decision.

---

‡ Section 4.2 of Chapter 12 discusses the L1 data item, which is used to prevent NM-TRAN from generating an Identification data item for individual data.

A format specification is *required* when:

Some data values are left blank on some data records, without having the value 0 or . (or a pair of commas) to hold the place of the missing value.

Some data values are adjacent on some data records; they are not separated by a space or a comma.

The data records span multiple physical records; that is, the character / is needed in the format specification. (The Data Preprocessor may generate such a format specification for the NONMEM data set; we are speaking here of the NM-TRAN data set.) The NOOPEN option of $DATA is used.

A format specification should *not* be present when:

The $INPUT record includes DROP as a data item label or synonym.

Day-time translation is desired.

II conversion is desired.

Commas are used to separate the data items.

The data values are not lined up into columns with uniform width, so that no format specification can be written to describe the file.

The IGNORE/ACCEPT option of $DATA is used to drop records from the data set.

Many data files do not fall in either category. A format specification is optional for such files.

NM-TRAN performs more checks on the data file when there is no format specification. Some features of NM-TRAN are the same with or without a format specification.

Comment records may be used.
NM-TRAN appends EVID, MDV, .ID., as needed.
NM-TRAN checks for blank records, and the BLANKOK option of $DATA may be used.
NM-TRAN gives a warning for unusual characters in the data file.
NM-TRAN counts records and supplies the count in FCON.

10.5. Skipping Data Items

It is always possible to omit (skip) data items that are not of interest for a given NONMEM run. When a format specification is coded, two things must be done: first, replace the data item's specification by an "X" specification (e.g., replace F8.0 by 8X) and second, delete the data item's label from the $INPUT record. When no format specification is coded, all that need be done is to replace the data item's label in the $INPUT record by DROP (or include DROP as a synonym).

Chapter 7 - $SUBROUTINE Record and $PK Record

1. What This Chapter is About

This chapter tells how to write a $SUBROUTINE record and how to write a simple $PK record for both individual and population data. This chapter is meant to be read in parallel with Chapters 3 and 4.

2. The $SUBROUTINE Record

The $SUBROUTINE record describes which pharmacokinetic model is to be used. Recall that NONMEM calls a subroutine named PRED to compute the predicted value. The user must choose to use his own PRED subroutine or to use the PREDPP package. In this text it is assumed that the PREDPP package is chosen.

2.1. Choosing an ADVAN Subroutine: Standard Pharmacokinetic Models

The PREDPP Library includes subroutines which are pre-preprogrammed, each for a specific pharmacokinetic model. They are:
ADVAN1 (One Compartment Linear Model)
ADVAN2 (One Compartment Linear Model with First Order Absorption)
ADVAN3 (Two Compartment Linear Model)
ADVAN4 (Two Compartment Linear Model with First Order Absorption)
ADVAN10 (One Compartment Model with Michaelis-Menten Elimination)
ADVAN11 (Three Compartment Linear Model)
ADVAN12 (Three Compartment Linear Model with First Order Absorption)
PREDPP calls only one subroutine, ADVAN; the different names above are external names distinguishing different instances of the ADVAN routine in the PREDPP Library. The name 'ADVAN' is used because the routine advances (i.e. updates the state of) the kinetic system from one point in time to the next. There are additional ADVAN routines in the Library which implement more general types of pharmacokinetic models; see Chapter 12, Section 2.2. Each of the ADVAN's can be used for either individual or population data. The (external) name of the ADVAN to be used is coded on the $SUBROUTINE record; this also implies that PREDPP is to be used. As an example, the following record specifies the One Compartment Linear Model:

```
$SUBROUTINE ADVAN1
```

The ADVAN's are described in Appendix 1. They share certain features.

1. The compartments are numbered. These numbers are used in two places. First, they are used in the CMT and PCMT data items to describe specific compartments. Second, the compartment number n is part of the name of PK parameters such as compartment scale (Sn), as discussed below.
2. Each model has a default observation compartment, which for each of the above ADVAN's happens to be the central compartment. If an event record contains an observation (i.e. is an observation event record), the prediction associated with that record will be the scaled drug amount in this compartment, unless the CMT data item on the record specifies differently. The prediction associated with a non-observational event record will again be the scaled drug amount in this compartment, unless the PCMT data item on the record specifies differently.
3. Each model has a default dose compartment. Unless specified differently by the CMT data item, it is understood that a dose is input into this compartment. With

ADVAN1, ADVAN3, and ADVAN10, this is the central compartment. With ADVAN2 and ADVAN4, a drug depot compartment is part of the model and is the default dose compartment. In these cases, if a dose is to go directly into the central compartment, its compartment number (2) must be present in the CMT data item of the dose record. Note that it is never *required* that there be doses into the depot compartment. In a study involving mixed oral and IV doses, for example, some patients may receive only IV doses. All dose event records for such patients will have the value 2 in the CMT data item.

4. Each model has an output compartment. The amount of drug in this compartment is the accumulated amount of drug eliminated from the system and typically represents the amount of drug which accumulates in the urine. This compartment is special. It may not receive a dose. It is initially off, and it remains off (so that the amount therein remains zero) until it is explicitly turned on by an other type event record which has the output compartment's number in the CMT data item. It is computed by "mass balance", as follows. Between any two points in time, it increases by an amount equal to the amount of drug in the other compartments at the first point in time, plus the amount added via doses between the two time points, less the amount remaining in the other compartments at the second point in time. (This difference is multiplied by an output fraction (F0) parameter, if F0 is computed by the PK routine.) The output compartment can be turned off (i.e. its amount reset to zero). If the compartment is interpreted as a urine compartment, this is equivalent to "emptying" the compartment. This is done by putting the *negative* of its number in the CMT data item of an other type or observation event record.†

   On event records, the output compartment is referred to by the compartment number given in Appendix 1. A PK parameter which refers to the output compartment may use either this number or 0 (zero). Thus, F0 and F2 both denote the output fraction for ADVAN1; similarly, S0 and S4 both denote the scale for ADVAN4's output compartment. SC denotes the scale for any ADVAN's central compartment.

5. Each model has a set of basic (required) pharmacokinetic (PK) parameters, which are the microconstants used to compute the amounts of drug via the kinetic equations for the model. Each one also has a set of additional (optional) PK parameters, including compartment scales (Sn), bioavailability fractions (Fn), and output fraction (F0). Compartment scales are typically used to convert amounts to concentrations, but they also can be used for other purposes. Bioavailability fractions multiply dose amounts. The output fraction is described above.

6. Each model's basic and additional pharmacokinetic parameters must be computed for it by a subroutine named PK. The error model must be described by a subroutine named ERROR. $PK and $ERROR abbreviated code provide an easy way to specify the essential computations that must occur in these subroutines.

2.2. Choosing a TRANS Subroutine: Alternative Parameterizations

As discussed in Chapter 3, we may prefer to use pharmacokinetic parameters in our PK routine other than the microconstants used by PREDPP. Appendix 2 shows several commonly-used parameterizations. The PREDPP package includes a family of subroutines called TRANS routines which are pre-programmed to translate (reparameterize) from these commonly used parameterizations to the ones expected by PREDPP. Appendix 2

† This is also permitted with output-type compartments; see Chapter 12, Section 2.8.

also gives the TRANS routine for each alternative parameterization. As with ADVAN, TRANS is the name of the routine. The names given in Appendix 2 are instances of external subroutine names used in the PREDPP Library. The first member of the family, TRANS1, simply translates a set of microconstants into these same microconstants and must be included in the NONMEM load module in lieu of the others when the $PK abbreviated code computes the microconstants.

The user must describe on the $SUBROUTINE record which TRANS routine is to be used. For example, the following record requests the One Compartment Linear Model parameterized (in the PK routine) in terms of clearance and volume.

```
$SUBROUTINE ADVAN1,TRANS2
```

When a TRANS other than TRANS1 is used, only the alternative parameters listed in column 1 need be assigned values in the $PK abbreviated code. In this example, these are CL, V, and KA.

Note that TRANS1 is the default. That is, if no TRANS routine is listed on the $SUBROUTINE record, it is assumed that TRANS1 is intended. This is the case in the examples of Chapter 2. Alternative parameterizations using TRANS1 are discussed later in this chapter in Section 4.2.

3. $PK Abbreviated Code

$PK abbreviated code consists of a block of $PK statements, one per line, which look much like FORTRAN statements. In fact, they are a subset of FORTRAN: simple assignment statements, certain kinds of conditional (IF) statements, and certain kinds of CALL, WRITE, PRINT, RETURN, OPEN, CLOSE, REWIND statements. The $PK abbreviated code must be preceded by a record containing the characters "$PK". This record and the abbreviated code constitute the $PK record.

$PK statements must include assignment statements giving a value to every basic PK parameter for the given ADVAN and TRANS combination, as listed in Appendix 1 (when TRANS1 is used) or column 1 of Appendix 2 (when a different TRANS is used). They may also include assignment statements giving values to one or more of the additional PK parameters.

3.1. Syntax

We assume the readers of this document are familiar with writing FORTRAN assignment and conditional statements. If not, the examples in this and the following chapter should give adequate guidance. FORTRAN statement numbers are not used, and the statements may start in any column. As with all NM-TRAN records, blank lines are permitted, and all text following a semi-colon (;) is ignored and may be used for comments. FORTRAN 95 continuation lines are permitted. An ampersand (&) is used at the end of a line to be continued.

The statements are built using the following: elements of the THETA array (e.g., THETA(1)); constants; names of input data items appearing on the $INPUT record; names of previously-assigned variables; FORTRAN library functions SQRT, LOG, LOG10, EXP, SIN, COS, ABS, TAN, ASIN, ACOS, ATAN, INT, MIN, MAX, and MOD; NONMEM functions GAMLN and PHI†; arithmetic operators +, -, *, /, **; and arithmetical and logical expressions using all of the above. In addition, statements may include representations for random variables such as ETA(1) and EPS(3). Input data

† PHI gives the value of the cumulative distribution function. GAMLN gives an accurate evaluation of the logarithm of the gamma function.

items have the values appearing on the current event record, and thus these values may change from one event record to the next. A user-defined variable name follows the usual FORTRAN rules (1-6 letters and digits, starting with a letter) and may not be subscripted. It is defined ("declared") by being assigned a value (i.e., by appearing to the left of = in an assignment statement).

Nested parentheses and nested IF statements are allowed. A pair of parentheses enclosing a subscript may be nested within another pair of parentheses. All subscripts must be constants (e.g. THETA(1)). The statements are evaluated sequentially, in the order in which they appear.

All variables, constants, and expressions are evaluated using floating-point (not integer) arithmetic. Single or double precision function names and constants may be used interchangeably.

3.2. When are $PK Statements Evaluated?

$PK statements are normally evaluated with every event record for both population and individual data. This enables the amounts in the compartments to be updated from event time to event time using current values of the data items. This may be more frequent than is necessary. In the theophylline example of Chapter 2, no data item is used in the $PK statements. In the phenobarbital example, the data item used, WT, is constant within any individual's data. In these cases, it is sufficient, and it can save noticeable amounts of run time, to evaluate the $PK statements once per individual record. PREDPP can be instructed that the set of event records with which the $PK statements are evaluated are to be limited in some way (see Chapter 12, Section 2.7). The CALL data item can be used to force the statements to be evaluated with any event records.

Certain advanced forms of dosing (additional and lagged doses; see Chapter 12, Sections 2.4 and 2.5) introduce doses at times which do not necessarily coincide with any event record. PREDPP does not normally evaluate the $PK statements at such times, but can be instructed to do so (See Chapter 12, Section 2.6). Model event time parameters can be used to instruct PREDPP to evaluate the $PK statements at specified times (See Chapter 12, Section 2.7).

3.3. Time Varying PK parameters

The state of the kinetic system at a given event time is obtained using PK parameter values computed with the data items on the event record. Using these parameter values the system is advanced to the event time from the last event time. Population models sometimes use data items which change value within individual records and thus give rise to PK parameters whose values change within individual records. In Chapter 4, Section 3.1.6, it is pointed out that it is desirable for the value of such a data item on the event record to be that value holding at the midpoint of the interval between the current event time and the last previous event time, since the system is advanced over this interval using the PK parameters determined with this value.

If the data item changes too rapidly for this value to fairly represent the data item over the entire time period, it is possible to subdivide the interval into smaller intervals. Event records with EVID=2 (other type event records) can be introduced for this purpose. For example, between two consecutive event records r_j and r_{j+1}, with event times t_j and t_{j+1}, one might introduce two new other type event records R_1 and R_2, with event times T_1 and T_2, into the data set. The value of the data item in R_1 will be used to compute the PK parameters used to advance the system over the interval t_j to T_1 and should be the value of the data item holding at the midpoint of this interval. Similarly, the value of the data

item in R_2 will be used to compute the PK parameters used to advance the system over the interval T_1 to T_2 and should be the value of the data item holding at the midpoint of this interval, and so on.

4. $PK Statements for Individual Data

4.1. Basic and Additional Parameters

With individual data, the parameters to be estimated are (usually) the individual's PK parameters, and therefore, elements of θ should be associated with these PK parameters. (NONMEM estimates the elements of the θ vector.) By an individual's PK parameters, we mean here the basic PK parameters and, possibly, some additional PK parameters (e.g. a bioavailability fraction, or volume of distribution when the latter is not a basic PK parameter). To illustrate, in the theophylline example of Chapter 1 there occur these $PK statements

```
$PK
KA=THETA(1)
K=THETA(2)
```

The parameters KA and K are the basic PK parameters for ADVAN2 and TRANS1 (the default TRANS routine). They are used to compute the amounts in the compartments. Typically, however, the observations are concentrations. A scale parameter is used to convert the amount into a concentration. Thus, in the theophylline example we see two additional $PK statements:

```
V=THETA(3)
S2=V
```

Here, V is a user-defined variable standing for the volume of distribution of the central compartment. It is neither a basic nor additional parameter. The parameter S2 is the scale parameter for the central compartment; upon dividing the amount in that compartment by S2, the concentration results. (An observation is usually predicted by an amount for a compartment divided by that compartment's scale parameter). In fact, these two statements could be replaced by the single statement

```
S2=THETA(3)
```

However, it may be helpful to the user to distinguish in his code between the calculation of the central volume itself and the calculation of the scale parameter.

There is no particular need for certain elements of θ to be associated with certain PK parameters. In the above example, the roles of θ_1 and θ_2 could have been reversed. NONMEM's θ vector may contain more or fewer elements than there are PK parameters, depending on how these parameters are modeled.

PK parameters must be explicitly modeled, usually in terms of parameters to be estimated and user-defined data items; the user communicates this model with the $PK statements. If a certain parameter's value is known a priori (say, S2 has the known value 500), there are several ways the value can be incorporated into the $PK statements. The following examples show how it can be done via a constant, via a fixed element of θ, and via a (differently-named) data item:

1.
```
S2=500
```

2.
```
$THETA  .6  9.  (500 FIXED)
$PK
S2=THETA(3)
```

Here, rather than be estimated, θ_3 is constrained to the value 500. This is discussed in Chapter 9.

3.
```
$INPUT ... VOL ..
$PK
S2=VOL
```
Here, VOL is assumed to have the value 500 on the data records. When the data is from a population, this third technique allows a unique value of VOL to be supplied for each individual.

4.2. Alternative Parameterizations using $PK Statements

It is possible to use an alternative parameterization while still using TRANS1. The reparameterization is performed within the $PK statements by explicitly computing the microconstants from the alternative parameters. Such "reparameterization" statements are given in column 2 of Appendix 2. They must follow the assignment statements that give the alternative parameters their values, as in the phenobarbital example of Chapter 2.

The advantage of using $PK statements to reparameterize, rather than using a TRANS subroutine, is that the NONMEM-PREDPP load module will then always consist of the same set of subroutines for a given choice of ADVAN, which simplifies the job of creating and running it. It will also run slightly faster. We assume in this document that this approach is taken.

Other parameterizations are possible besides the ones in Appendix 2. For example, with ADVAN1 and TRANS1, one might code:
```
CL=THETA(1)
K=THETA(2)
V=CL/K
S1=V
```
The ability to express a large variety of modeling possibilities with NONMEM-PREDPP provides great freedom and flexibility, but as always with flexible modeling capability, certain pitfalls arise. Suppose, for example, that with a one compartment system the compartment amount, rather than the concentration is observed. With ADVAN1 and TRANS1 the statements
```
CL=THETA(1)
V=THETA(2)
K=CL/V
```
will lead to difficulty because only the ratio of θ_1 to θ_2 affects the amount in the compartment, and therefore, the data do not allow θ_1 and θ_2 to be separately estimated. The statements should read:
```
K=THETA(1)
```
It is important to remember that only those elements of θ which affect the predictions of observations will be estimated by NONMEM. Here is some problematic code using ADVAN1 with TRANS1:
```
K=THETA(1)
V=THETA(2)
CL=THETA(3)
S1=V
```
Once again, NONMEM will be unable to produce separate estimates of all elements of θ. The kinetics of a simple one compartment system cannot be determined by three independent parameters. With TRANS1, PREDPP itself does not "know" about the relationship K=CL/V which defines a dependency among the parameters. Indeed, the parameters CL and V are both regarded as user-defined variables. The value of θ_3 has no effect on the prediction. Were it not for the fact that S1 is set equal to V, the value of θ_2 would have no

effect on the prediction either. With TRANS2 this code is also incorrect for essentially the same reason. Here, K is regarded as a user-defined variable, and the relationship CL=K*V is not "known" to PREDPP. (PREDPP does know that CL/V is the rate constant of elimination, but it does not recognize the variable K as denoting this rate constant, and θ_1 has no effect on the prediction.)

4.3. Scale Parameters

Scale parameters are mentioned in Section 2.1. Predicted compartment amounts are divided by them and are thus converted to predicted concentrations. They are only needed for those compartments whose concentrations are directly observed. With ADVAN3, for example, the peripheral compartment's scale S2 does not need to be computed explicitly if there are no observation events giving measured values of concentrations in the peripheral compartment. Predicted values for this compartment may still be plotted against time, for example, but these values need not be scaled drug amounts; the (unscaled) amount alone is sufficient to show the shape of the curve. (The various volume parameters shown in Appendix 2 must be modeled when they are used as basic parameters, but they need not be assigned as values to compartment scale parameters.) Any scale parameter which is not modeled by $PK statements is assumed to be 1 (i.e., predicted values are always amounts).

4.3.1. Scaling by a Known Constant

In Chapter 3, Section 2.2.1, the units of V were changed from kiloliters to liters using the model:

$$S = V/1000$$

This can be coded in a $PK statement similar to the way it appears here, except that the compartment number must be specified:

```
S1=V/1000
```

Basic PK parameters may also be rescaled in this manner.

4.3.2. Scaling by a Parameter: Conditional Statements and Indicator Variables

In Chapter 3, Section 2.2.2, the following model appeared:

$$S = \begin{cases} V, & \text{if } \textit{assay is } 1 \\ hV, & \text{if } \textit{assay is } 2 \end{cases}$$

There are two ways this can be coded in $PK statements. The *assay* data item can be tested directly, or an indicator variable can be used. An indicator variable is a variable whose value is 0 or 1. It may be identified with an input data item, or it may be a user-defined variable in the $PK statements. For example, suppose variable ASY is to be used as an indicator variable. If some input data item is given value 1 when assay 1 was used and value 0 when assay 2 was used, then this data item could simply be named ASY on the $INPUT record. Suppose, however, that the assay number itself (1 or 2) was recorded in the data and that we have named the data item ANUM on the $INPUT record. We must compute the user-defined variable ASY for use as an indicator variable. There are two ways this can be done: using a logical IF and using a block IF.

1.
```
ASY=1
IF (ANUM.EQ.2) ASY=0
```
Here, ASY is "provisionally" given the value 1. The value is changed to 0 if the data indicates assay 2.

2.
```
IF (ANUM.EQ.1) THEN
ASY=1
ELSE
ASY=0
ENDIF
```

The choice between these forms of IF is purely a matter of style. Now let us assume that the compartment to be scaled is compartment 2, and that h is to be identified with θ_5. The parameter S2 can now be coded unconditionally:

```
S2=ASY*V+(1-ASY)*THETA(5)*V
```

Alternatively, ANUM can be tested and ASY avoided altogether:

1.
```
S2=V
IF (ANUM.EQ.2) S2=THETA(5)*V
```

2.
```
IF (ANUM.EQ.1) THEN
S2=V
ELSE
S2=THETA(5)*V
ENDIF
```

4.3.3. Scaling by a Data Item

If observations of urine concentration C_u are included in the data (see e.g. Chapter 6, Section 9), it is necessary to provide urine volume as a scale for the output compartment. Presumably, this volume varies between urine observations and is recorded in the data records. Suppose this data item is called UVOL in the $INPUT record. (The name given to the data item has no special significance; any name could be chosen.) An additional $PK statement is necessary:

```
S0=UVOL
```

UVOL need be recorded on only those observation events to which it applies, although it does no harm to record it on other event records. For example, it may well happen that both plasma and urine responses are measured at the same time, so that there are two observation event records with the same value of TIME, one for each compartment observed at that time. As described in Section 3.2 above, $PK statements are normally evaluated with every event record. Consider, for example, the sample data below. Assume that the Central compartment is compartment 2 and the output compartment is compartment 3. (Note the use of -3 to signify that compartment 3 is to be turned off after the observation time. The compartment will remain off until the time another urine collection begins, as indicated with an other type record; see Chapter 6, Section 7.4). Either 1. or 2. will produce the correct value of S0:

1. Record UVOL on the event record to which it applies. The order of the records does not matter.

```
TIME UVOL DV    CMT
10.  0    5.80    2
10.  100  .067   -3
```

2. Record UVOL on all event records having the same value of TIME. The order of the records does not matter.

```
TIME UVOL DV    CMT
10.  100  5.80    2
10.  100  .067   -3
```

The following will not produce the correct value of S0 unless PREDPP is instructed to evaluate the $PK statements only once for each distinct value of TIME:

```
TIME UVOL DV    CMT
10.  100  5.80   2
10.  0    .067  -3
```

4.4. Bioavailability Fraction Parameters

PK parameters of the form Fn, where n is the number of a compartment into which a dose may be introduced, are bioavailability fractions. If a dose record specifies a dose for compartment n, the dose amount given on the event record is multiplied by the value of Fn computed from the $PK statements evaluated with this record, and this product is the dose amount introduced into the system. For example, F1 multiplies the amount of dose which is to be added to compartment 1. Any Fn which is not computed by $PK statements is assumed to be 1 (i.e., the dose is 100% available).

As an example, suppose two different preparations of the same drug are administered, and it is assumed that they differ only in their bioavailability. The indicator variable (or data item) PREP has value 1 for the first preparation and 0 for the second. The ratio of the bioavailability of the second preparation to that of the first preparation is identified with θ_6. Usually, the method of drug administration permits this ratio to be estimated, but not the separate bioavailabilities. Without loss of generality, the bioavailability of the first preparation can be taken to be 1. Assuming the drug enters compartment 1 of the model, there are three ways this can be coded:

1.
```
F1=PREP+(1-PREP)*THETA(6)
```

2.
```
F1=1
IF (PREP.EQ.0) F1=THETA(6)
```

3.
```
IF (PREP.EQ.0) THEN
F1=THETA(6)
ELSE
F1=1
ENDIF
```

Again, the choice is a matter of style.

Once a dose is introduced into the dose compartment, it begins to distribute into the other compartments. Whether or not the original dose was 100% available, it is assumed that none of the dose appearing in the dose compartment, and in other compartments after the dose is introduced, is further reduced due to bioavailability effects. PREDPP cannot model "bioavailability effects" between compartments.

4.5. Output Fraction

The Output Fraction parameter, F0, is an optional additional PK parameter of every model. As discussed in Section 2 above, every model contains an output compartment. If this compartment has been turned on prior to the advance from time t_{j-1} to time t_j, then the amount of drug lost from the system during this interval via elimination is multiplied by F0 and added to the prior contents of the output compartment. If the $PK statements do not include an assignment statement giving a value to F0, it is taken to be 1 (i.e., 100% of the drug excreted goes to the output compartment). In model (4.7), an example of the use of F0 is given. Assuming that the variables CLREN (renal clearance) and CL (total clearance) have been calculated with earlier $PK statements, the statement

```
F0=CLREN/CL
```

can be used to compute F0.

5. $PK Statements for Population Data

With population data, the structural models for the PK parameters tend to be more complicated than with individual data. In addition, the influence of interindividual random effects needs to be described. These will involve differences in the $PK statements, but the same $SUBROUTINE record and the same ADVAN and TRANS subroutines are used, and the same general requirements and examples of the earlier sections of this chapter still mostly apply. In this section, the models of Chapter 4 are implemented via $PK statements. Many of these models could be implemented in a variety of ways; an experienced programmer may prefer to code them differently.

With population data, we must distinguish between the typical value of a PK parameter in the population and the value of that parameter for a given individual, the individual's value. The typical value is computed by a structural model involving only fixed effects. We have chosen to denote it with the use of a tilde: e.g. $\tilde{Cl}$. The individual's value is computed by a model including random interindividual effects (represented by random variables) and is denoted without a tilde: e.g., Cl. There is no tilde character in the FORTRAN character set, and with NM-TRAN we do not need to distinguish typical and individual values. However, for purposes of clarity, in all the examples which follow we will include the letters TV (Typical Value) at the start of those variable names which we think of as having a tilde (e.g., TVCL). This is a matter of style.

5.1. Structural Part of Parameter Models

In models such as (4.3), the subscript i indicates that the model applies to the i^{th} individual. As noted in Chapter 4, the subscript is not needed and, indeed, is not used in $PK statements.

5.1.1. Linear Models

Models (4.4), (4.5a), (4.5b) and (4.6) can be coded as they appear. Assuming that WT, AGE, and SECR are input data items or have been calculated with earlier $PK statements, the code is:

```
TVCLM=THETA(1)*WT
RF=WT*(1.66-.011*AGE)/SECR
TVCLR=THETA(4)*RF
TVCL=TVCLM+TVCLR
```

5.1.2. Multiplicative Models

Model (4.4.1) can be coded as follows:

```
TVLCLM=THETA(1)+THETA(2)*LOG(WT)
TVCLM=EXP(TVLCLM)
```

Model (4.4.2) can also be coded as it appears:

```
TVCLM=THETA(1)*WT**THETA(2)
```

5.1.3. Saturation Models

Model (4.4.3) presents a problem. Subscripted variables that can appear in $PK statements are few; naturally, they include THETA and (as seen below) ETA. The variable CPSS cannot be subscripted, and a variable name such as CPSS2 (rather than CPSS(2)) must be used for $Cpss_2$. The model can be coded exactly as it appears:

```
TVCLM=WT*(THETA(1)-THETA(2)*CPSS2/(THETA(3)+CPSS2))
```

5.1.4. Models with Indicator Variables

When dealing with typical values, indicator variables (0/1 variables) can be used interchangeably with conditional (IF) statements, as we have already seen. Model (4.4.4) can be coded in a variety of ways, two of which are:

1.
```
TVCLM=(THETA(1)-THETA(2)*HF)*WT
```

2.
```
IF (HF.EQ.0) THEN
TVCLM=THETA(1)*WT
ELSE
TVCLM=(THETA(1)-THETA(2))*WT
ENDIF
```

5.2. Population Random Effect Models

Random variables η are used in the models for interindividual errors. (With population models, random variables ε are used in the models for intraindividual errors; see Chapter 4, Section 2.) In $PK statements they are denoted by ETA(1), ETA(2), etc. Even if there is only once such variable it must still be subscripted. It is the presence of one or more such variables that indicates to NM-TRAN that the data is from a population. Just as there is no particular need for certain θ elements to be identified with certain PK parameters, there is no particular need for certain θ elements to be associated with certain η variables, and any association need not be one-to-one. The following models are both valid:

1.
```
CL=THETA(1)+ETA(1)
V=THETA(2)+ETA(2)
```

2.
```
CL=THETA(1)+ETA(2)
V=THETA(2)+ETA(1)
```

However, it will be easier to keep things straight if the first model is used.

Here are three different ways of coding a model for an individual's value of clearance:

1.
```
TVCL=THETA(1)
CL=TVCL+ETA(1)
```

2.
```
CL=THETA(1)
CL=CL+ETA(1)
```

3.
```
CL=THETA(1)+ETA(1)
```

We prefer the first way because it clearly distinguishes the model for the typical value from the model for the individual's value. With any of the three ways for coding the model the typical value of the parameter can be computed as follows: The η variables are set to 0, and the parameter is computed. Any variable whose value depends on η variables is called a random variable.

Random variables are called true-value variables in the first edition of this guide. This is because, in principle, a random variable can assume an individual's true value under the model. Such a variable is in contrast to a variable which assumes only a typical value for the population.

An individual's true value is never actually known, although an estimate of it can be obtained. See Chapter 12, Sections 4.11-4.13.

5.3. Models for Interindividual Errors

Here we show how to express the most commonly used models for interindividual errors with $PK statements. In addition, all the error models described in Chapter 8 may also be used in $PK statements.

5.3.1. Additive/Multiplicative Models

This is the error model of (4.9):

```
K=TVK+ETA(1)
```

This is the error model of (4.10):

```
K=TVK*(1+ETA(1))
```

This model can also be coded as:

```
K=TVK+TVK*ETA(1)
```

Here, the variable TVK has been "multiplied through". The choice is a matter of style.

5.3.2. Other Models

The model (4.11) may be coded as written.

```
CLM=TVCLM+(1-ICU)*ETA(1)+ICU*ETA(2)
```

It may also be coded with an IF statement.

```
IF (ICU.EQ.0) THEN
CLM=TVCLM+ETA(1)
ELSE
CLM=TVCLM+ETA(2)
ENDIF
```

The choice is a matter of style.

Note that, under the parameterizations given in Appendices 1 and 2, CLM is neither a basic nor an additional PK parameter, yet its model involves an η variable. This is legitimate: any variable can be defined in terms of an η variable. However, just as with θ's, the values assigned to the η variables must somehow affect the predictions of observations. Otherwise, the variance of some η variable cannot be estimated, and consequently, none of the variances of these variables can be estimated. Presumably, within the $PK statements, CLM is used to compute CL, and (either within the $PK statements or within the TRANS routine) CL is used to compute K.

5.4. Restrictions on Random Variables

This section discusses the use of random variables in some depth and may be skipped by the casual reader. The remarks here apply to all random variables: both the ETA variables of this chapter and the ERR/EPS variables of Chapter 8.

In general, ETA variables can be used like any other variables.

Any variable whose value is affected by an ETA variable is a random variable, whether the ETA variable occurs explicitly in the defining expression for the random variable or whether another random variable occurs in this expression. For example, consider the following:

```
TVCLM=THETA(2)*WT
CLM=TVCLM+ETA(2)
RF=WT*(1.66-.011*AGE)/SECR
TVCLR=THETA(4)*RF
CLR=TVCLR+ETA(1)
CL=CLM+CLR
```

CL is a random variable, because it is computed from random variables. It depends on both η_1 and η_2.

Random variables may be changed and may be assigned conditionally, subject to the following restrictions.

A random variable may not appear anywhere within a nested if structure.

A random variable defined in the $PK block may not be redefined in the $ERROR block.

As an example of the first restriction, suppose in the model (4.11) it is also believed that, for ICU patients, age affects CLM. The following code expresses the model, but is not permitted:

```
IF (ICU.EQ.1) THEN
   IF (AGE.GE.50) THEN
    TVCLM=THETA(1)
   ELSE
    TVCLM=THETA(2)
   ENDIF
   CLM=TVCLM+ETA(1)
ELSE
   TVCLM=THETA(3)
   CLM=TVCLM+ETA(2)
ENDIF
```

An alternate code follows, in which the calculation of TVCLM (which involves a nested IF) precedes the calculation of CLM (which does not require a nested IF). This code is permitted.

```
IF (ICU.EQ.1) THEN
   IF (AGE.GT.50) THEN
      TVCL=THETA(1)
   ELSE
      TVCL=THETA(2)
   ENDIF
ELSE
   TVCL=THETA(3)
ENDIF
IF (ICU.EQ.1) THEN
  CL=TVCL+ETA(1)
ELSE
  CL=TVCL+ETA(2)
ENDIF
```

Indentations are used in the above code for clarity, but have no affect on NM-TRAN's processing of the abbreviated code.

Chapter 8 - $ERROR Record

1. What This Chapter is About

This chapter tells how to write a simple $ERROR record for PREDPP. This chapter is meant to be read in parallel with Chapters 3 and 4.

2. $ERROR Abbreviated Code

$ERROR abbreviated code consists of a block of $ERROR statements, one per line. The $ERROR abbreviated code must be preceded by a record containing the characters "$ERROR". This record and the abbreviated code constitute the $ERROR record.

$ERROR statements describe the error model for PREDPP. These statements are very similar for individual data and for population data. In fact, by making use of variables called ERR variables, the $ERROR statements are identical for both kinds of data.

2.1. Syntax

The syntax of a $ERROR record is very similar to that of a $PK record. Certain differences will be mentioned here.

There must be an assignment statement giving a value to a special (reserved) variable Y. Y is a random variable representing the random variable y (the modeled observation). Y is usually defined in terms of a special (reserved) variable F, which represents the prediction for Y. When the data are from a population, F is a random variable. With individual data, ETA variables may be used in the definition of Y. With population data, EPS variables may be used in the definition of Y. There are also special random variables called ERR variables. The variable ERR(I) is the same as ETA(I) or EPS(I), depending on whether the data are individual or population, respectively. For the purpose of giving a general discussion, applying equally to both individual and population data, ERR will be used in all the examples in this chapter. (It is also useful to use ERR in $ERROR statements as a practical matter. Sometimes the same data is analyzed from both the population and the individual point of view. By using ERR variables, changes to the NM-TRAN input file are minimized.) An ERR variable (as with ETA and EPS variables) must always include a subscript (e.g., ERR(1)), even when there is only one such variable in the model.

Variables computed within $PK statements may be used in $ERROR statements, but not vice versa.

2.2. When are $ERROR Statements Evaluated?

$ERROR statements are normally evaluated with every event record. This may be more frequent than is necessary. PREDPP can be instructed that the set of event records with which the $ERROR statements are evaluated is to be limited to only observation events, once per individual record, or once per problem. Such limitation does not apply to the Simulation Step (Chapter 12, Section 4.8). With the additive (3.4) and constant coefficient of variation (3.5) error models, and with the exponential error model, NM-TRAN instructs PREDPP to evaluate the $ERROR statements only once per problem. Again, the CALL data item can be used to force evaluation of the $ERROR statements with any event records.

3. Error Models

The following sections show how the error models of Chapter 3 are expressed using $ERROR statements.

3.1. The Additive Error Model

This is the error model (3.4):

```
Y=F+ERR(1)
```

Both examples in Chapter 2 use this error model.

3.2. The Constant Coefficient of Variation and Exponential Models

This is the CCV error model (3.5):

```
Y=F*(1+ERR(1))
```

This error model can also be coded as:

```
Y=F+F*ERR(1)
```

Here, the variable F has been "multiplied through". The choice is a matter of style.

This is the exponential error model (3.5a):

```
Y=F*EXP(ERR(1))
```

When the $ERROR statements consist solely of one of these statements (in any of the forms), the output from PREDPP will include the message:

```
ERROR IN LOG Y IS MODELED
```

This is done because during data analysis NONMEM cannot distinguish between the CCV error model $y = \tilde{f} + \tilde{f}\varepsilon$ and the exponential error model $y = \tilde{f}\exp(\varepsilon)$, for which $\log(y) = \log(\tilde{f}) + \varepsilon$†. By using the latter model and modelling the error in $\log(y)$ rather than in y, NM-TRAN enables PREDPP to achieve an improvement in run time.

3.3. Combined Additive and CCV Error Model

This is the error model (3.6):

```
Y=F+F*ERR(1)+ERR(2)
```

3.4. The Power Model

This is the error model (3.7):

```
Y=F+F**P*ERR(1)
```

The variable P must be assigned a value before its use above. P is typically identified with an element of θ so that it can be estimated in the fitting process. Let us assume that θ_4 is chosen for this purpose. Then an alternative coding is:

```
Y=F+F**THETA(4)*ERR(1)
```

3.5. Two Different Types of Measurements

We have already seen how an indicator variable, e.g., ASY, can be used in $PK statements for a variety of purposes. The same technique is used in $ERROR statements. Consider model (3.8) where the variable ASY has the value 1 or 0, corresponding to assay 1 or assay 2. ASY is a data record item. Then the error model (3.8) is coded:

```
Y=F+ASY*ERR(1)+(1-ASY)*ERR(2)
```

This model can also be coded in several ways, the choice of which is a matter of style.

1)
```
IF (ASY.EQ.1) THEN
Y=F+ERR(1)
```

† During Simulation, NONMEM does distinguish between the CCV and exponential error models.

```
      ELSE
      Y=F+ERR(2)
      ENDIF
```

2)
```
      IF (ASY.EQ.1) Y=F+ERR(1)
      IF (ASY.NE.1) Y=F+ERR(2)
```

3)
```
      Y=F+ERR(2)
      IF (ASY.EQ.1) Y=F+ERR(1)
```

3.6. Two Different Types of Observations

In Chapter 3, Section 3.6, an example is given in which there are two kinds of observations, plasma (C) and urine (C_u). With PREDPP, measurements from different compartments of the model are recorded in the DV data item of different observation event records. The CMT data item identifies the compartment from which the prediction associated with the event record is to be obtained. When the $ERROR statements are evaluated for a given event record, the variable F contains the prediction from the compartment specified for that event record. All that need be done is to select the correct error model, depending on the compartment. Suppose, for example, that ADVAN2 is used, so that the central compartment is compartment 2 and the output (urine) compartment is compartment 3. Then the error model (3.10) can be coded:

```
TYP=0
IF (CMT.EQ.2) TYP=1
Y=F+TYP*ERR(1)+(1-TYP)*ERR(2)
```

This model can also be coded in several ways, one of which is shown here:

```
IF (CMT.EQ.2) THEN
Y=F+ERR(1)
ELSE
Y=F+ERR(2)
ENDIF
```

3.7. More than One Indicator Variable

In Chapter 3, Section 3.7, an example is given in which there are three kinds of observations. Suppose that there are two data items, ASY1 and ASY2. ASY1 is 1 if assay 1 is used and 0 otherwise. ASY2 is 1 if assay 2 is used and 0 otherwise. This is the error model (3.11):

```
Y=F+ASY1*ERR(1)+ASY2*ERR(2)+(1-ASY1)*(1-ASY2)*ERR(3)
```

This model can also be coded in several ways, one of which is shown here:

```
Y=F+ERR(3)
IF (ASY1.EQ.1) Y=F+ERR(1)
IF (ASY2.EQ.1) Y=F+ERR(2)
```

Chapter 9 - Additional NM-TRAN Records

1. What This Chapter is About

This chapter tells how to give initial estimates to NONMEM's parameters ($THETA, $OMEGA, $SIGMA records); how to tell NONMEM what tasks to perform ($ESTIMATION, $COVARIANCE records); and how to tell NONMEM what additional output to produce ($TABLE, $SCATTERPLOT records).

2. Providing Initial Estimates For θ: The $THETA Record

This record provides an initial estimate (and, optionally, provides lower and upper bounds) for every element of NONMEM's θ vector.

2.1. Providing Initial Estimates For Elements Of θ

The $THETA record contains a list of values, separated by spaces or commas, which are the initial estimates for the θ's used in the $PK and $ERROR statements. The position of a value in the list corresponds to its position (subscript) in the θ vector. For example, consider the following statement:

```
$THETA 1.7 .102 29.
```

This says that the initial estimate for θ_1 is 1.7, the initial estimate for θ_2 is .102, and the initial estimate for θ_3 is 29. Some users of NONMEM prefer to code each value on a separate line so that they can include comments to themselves describing the significance of the θ's. The above record could have been coded as follows:

```
$THETA  1.7  ;  RATE CONSTANT OF ABSORPTION
        .102 ;  RATE CONSTANT OF ELIMINATION
        19.  ;  VOLUME OF DISTRIBUTION
```

This is a matter of style.

2.2. Providing Constraints for Elements of θ

When NONMEM is told to estimate the parameters (Section 4.1, the Estimation Step, below), it varies the elements of θ to find values which cause the model to fit the observations best. The values on the $THETA record are the initial estimates of θ for this search. When only an initial estimate is provided, NONMEM is free to chose any positive or negative value for that element of θ. We then say that the θ element is *unconstrained*, which means that its lower bound (lower limit) is $-\infty$ and its upper bound (upper limit) is $+\infty$. When finite bounds are desired, the initial estimate and its bounds must be enclosed in parentheses and specified in the order (lower, initial, upper). When the upper bound needn't be finite, the initial estimate and its lower bound are enclosed in parentheses and specified in the order (lower, initial). Note that when no estimation is performed, upper and lower bounds have no effect.

In the theophylline example of Chapter 2, for example, negative θ values are physiologically impossible. Each θ element was given a lower bound of 0, which constrained it to be non-negative:

```
$THETA  (0, 1.7)  (0, .102)  (0, 29.)
```

It is possible to mix constrained and unconstrained θs; this was done in Chapter 2, figure 2.12:

```
$THETA  (0,.0027) (0,.70) .0018  .5
```

An upper bound of $+\infty$ may be stated explicitly using the value `1000000` or the word `INFINITY`. Similarly, a lower bound of $-\infty$ may be stated explicitly as `-1000000` or `-INFINITY`.

2.3. Fixing Elements of θ

When estimation is performed, it is sometimes desirable to hold one or more elements of θ to a constant value. One example is when a full model is reduced to a simpler model, as discussed in Chapter 5, Section 2.1; usually this is done by holding some θ element to 0. In fact, the value 0 may not be used as an initial estimate for an element of θ unless this element is fixed to this value. A θ element is held constant by inserting the word FIXED *after* the initial estimate. For example, the following statement allows θ_1 and θ_3 to vary, but holds θ_2 to the value .102:

```
$THETA 1.7 .102 FIXED  29.
```

Parentheses may be used to make the statement easier to read:

```
$THETA 1.7 (.102 FIXED) 29.
```

If the lower, initial, and upper values for an element of θ are identical, the element of θ is understood to be fixed, even if the word FIXED does not appear.

2.4. How to Obtain Initial Estimates for θ

When estimating parameters, good initial estimates for θ are sometimes important. Poor initial estimates may occasionally cause the NONMEM run to take excessive amounts of computer time, to produce parameter estimates that are not physiologically reasonable, or to fail to produce any parameter estimates at all. For some drugs and models, initial estimates for θ can be obtained from published literature describing prior studies with the drug. For some studies, very accurate values may have been obtained by prior runs with NONMEM or other regression programs. Highly accurate values should be perturbed (modified) by about 10% before being used as initial estimates in a NONMEM run. (Initial estimates that are too close to what may be the actual final estimates will cause problems in a NONMEM run; see Chapter 13.) Sometimes, however, there is little guidance in choosing initial estimates for some elements of θ.

One approach with population data, where there is a reasonable amount of data for each individual, is as follows. It is often easier to guess at appropriate parameter values for individual data than for population data. So, first estimate each individual's parameter values using only the data from the individual. The mean values of the individuals' parameter estimates can then be used as the initial parameter estimates in the population analysis. Results from individual runs can also be used to obtain initial estimates for elements of Ω and Σ; see below.

Another approach is simply to let NONMEM find an initial estimate. NONMEM has an automatic strategy for so doing; see Chapter 12, Section 4.4.

3. Providing Initial Estimates for Ω and Σ: the $OMEGA and $SIGMA Records

Recall that Ω and Σ are variance/covariance matrices for the following random variables:

Individual Model

Ω (OMEGA) for η (Random Intraindividual Variability)

Population Model

Ω (OMEGA) for η (Random Interindividual Variability)

Σ (SIGMA) for ε (Random Intraindividual Variability)

In all the examples in this document, Ω and Σ are *diagonal* matrices, in which covariance elements such as ω_{12} (which is $cov(\eta_1, \eta_2)$) are assumed to be zero. NONMEM also allows full variance/covariance matrices; this is beyond the scope of this text, but see Chapter 12, Section 4.1.

Initial estimates for the variances must be provided to NONMEM via the $OMEGA and $SIGMA records. Initial estimates of *all model parameters* (θ, Ω, and Σ) must be provided even if estimation is not requested. $OMEGA and $SIGMA records each contain a list of values, separated by spaces or commas, which are the estimates for the corresponding variances. As in the $THETA record, the position of a value in the list corresponds to the position (subscript) of the corresponding variance (along the diagonal) in the matrix.

3.1. $OMEGA Record With Individual Data

With individual data, η variables are used in $ERROR records, where they are called either ERR or ETA. For example, in the theophylline problem of Chapter 2 (figure 2.1) there appear the records:

```
$ERROR
Y=F+ERR(1)
$OMEGA 1.2
```

Here, ERR(1) corresponds to η_1, and the initial estimate for its variance is 1.2: i.e., $\Omega_{11} = \omega_1^2 = var(\eta_1) = 1.2$.

3.2. $OMEGA Record With Population Data

With population data, η variables are used in $PK statements. For example, in the phenobarbital problem of Chapter 2 (figure 2.6) there appear the lines:

```
CL=TVCL+ETA(1)
V=TVVD+ETA(2)
$OMEGA  .0000055, .04
```

The $OMEGA record says that the initial estimate for the variance of η_1 is $5.5x10^{-6}$, and of η_2 is .04: i.e., $\Omega_{11} = \omega_1^2 = var(\eta_1) = 5.5x10^{-6}$ and $\Omega_{22} = \omega_2^2 = var(\eta_2) = .04$. Some users of NONMEM prefer to code each value on a separate line so that they can include comments:

```
$OMEGA  .0000005  ; VARIANCE IN CL
        .04       ; VARIANCE IN V
```

3.3. The $SIGMA Record

This record is used only with population data, and is similar to the $OMEGA record. It gives the initial estimates of the variances of the ε variables used in the $ERROR statements, where they are called either ERR or EPS. For example, in Figure 2.6, there also appears the records:

```
$ERROR
Y=F+ERR(1)
$SIGMA  25
```

Here, ERR(1) corresponds to ε_1, and the initial estimate for its variance is 25: i.e., $\Sigma_{11} = \sigma_1^2 = var(\varepsilon_1) = 25$.

3.4. Fixing Elements of Ω or Σ

It is sometimes desirable to hold one or more elements of Ω or Σ to constant value(s). In the population example of Chapter 2 it is possible to ignore interindividual variability in

CL by fixing η_1 to 0†. The variance of an η or ε variable is held constant by inserting the word FIXED *after* the initial estimate:

```
$OMEGA  0 FIXED .0225
```

Parentheses may be used to make the statement easier to read:

```
$OMEGA  (0 FIXED) .0225
```

As with θ, the value 0 may not be used as an initial estimate for any element of Ω or Σ unless the element is fixed to this value.

3.5. How to Obtain Initial Estimates for Ω and Σ

The initial estimates for the variances will depend on the particular (interindividual and/or intraindividual) error models chosen. The estimates do not have to be particularly accurate, although values which are much too small can cause difficulties for NONMEM. In general, it is better to over-estimate the variances rather than to under-estimate them. As with initial estimates for θ, initial estimates can sometimes be obtained from published literature or from prior runs with NONMEM or other regression programs.

Initial estimates can also be obtained by an approach which we illustrate with examples for both intraindividual and interindividual error models. The standard deviation of a physiological quantity is generally some fraction r of its typical value t: $\sigma_y = rt$.

For the additive model:

$$y = f + \varepsilon$$

$$\sigma_y = \sigma_\varepsilon = rt$$

$$var_\varepsilon = \sigma_\varepsilon^2 = (rt)^2 = r^2t^2$$

Some ambiguity exists about what we mean by "the typical value" of y. For the purpose of obtaining an initial estimate of the variance, we need not be too particular about this. For the theophylline example (Figure 2.1), we may choose the mean of the observed values as the typical value. This value is approximately 5.4. Assuming 20% error, i.e. $r = .2$, then $\sigma_\varepsilon^2 = (.2x5.4)^2 = 1.2$. Similarly, in the first phenobarbital example (Figure 2.6), the mean of the observations is approximately 25. Again assuming 20% error, then $r = .2$, and $\sigma_\varepsilon^2 = (.2x25)^2 = 25$. For that same example, the typical value of CL was estimated according to the model for the parameter: `TVCL=THETA(1)`. We used the initial estimate of θ_1, .0047, as the typical value of CL, and assumed 50% error: $\Omega_{11} = (.5x.0047)^2 = 5.5x10^{-6}$. The model for V is `TVVD=THETA(2)`. Again, we used the initial estimate of θ_2, .99, as the typical value of V, but assumed 20% error: $\Omega_{22} = (.2x.99)^2 = .04$. Note finally that in the second phenobarbital example (Figure 2.12), we used as initial estimates of variance the final estimates obtained from the first example (understanding that these estimates could be somewhat large due to some of the variability being explained in this example by a systematic influence of weight).

For the constant coefficient of variation model:

$$y = f + f\varepsilon$$

$$\sigma_y = f\sigma_\varepsilon = rt$$

† One could also re-write the $PK statements to eliminate ETA(1) in the model for CL, which also requires that ETA(2) in the model for V be re-numbered as ETA(1). It is easier to modify only $OMEGA.

$$var_\varepsilon = \sigma_\varepsilon^2 = \frac{r^2 t^2}{f^2}$$

If we identify t with the value of f (whatever it may be), we have:

$$var_\varepsilon = r^2$$

In other words, using the CCV model, we do not need to estimate the typical value of the variable. For example, assuming 20% error, $var_\varepsilon = .2^2 = .04$.

As with θ, it is possible for NONMEM itself to obtain initial estimates of Ω and Σ automatically; see Chapter 12, Section 4.4.

4. Specifying Optional Tasks

Two main tasks of NONMEM, the Estimation Step and the Covariance Step, are optional and must be specifically requested by including the $ESTIMATION and $COVARIANCE records. To skip the Estimation Step, simply omit the $ESTIMATION record. To skip the Covariance Step, simply omit the $COVARIANCE record.

In every run NONMEM computes and prints the value of the objective function and the final parameter estimates. The values printed are based on the final parameter estimates if the Estimation Step is requested, and are based on the initial estimates if it is not.

4.1. Requesting the Estimation Step: the $ESTIMATION Record

In the Estimation Step, NONMEM performs a search to obtain those values of θ, Ω, and (for population studies) Σ which give the lowest value of the objective function. The output of this step is the pages whose titles are "MONITORING OF SEARCH:", "MINIMUM VALUE OF OBJECTIVE FUNCTION", and "FINAL PARAMETER ESTIMATE". This step is requested by the presence of the following statement:

```
$ESTIMATION
```

There are several options, which are described in the NONMEM Users Guide, Part IV. The most frequently used ones are as follows.

METHOD=0

NONMEM always sets etas to 0 during the computation of the objective function. Also called the "first order (FO) method." This is the default. It may also be coded as METHOD=ZERO.

METHOD=1

NONMEM uses conditional estimates for the etas during the computation of the objective function. METHOD=1 is also called the "first order conditional estimation (FOCE) method." It may also be coded as METHOD=CONDITIONAL. When the option INTERACTION is also present, the method is called the "FOCE with INTERACTION method". It is recommended for continuous variables unless the data are very sparse. These methods are discussed in Guide VII, Conditional Estimation Methods.

SIGDIGITS=n

By default, the search continues until the estimates of all elements of θ, Ω, and Σ have been determined to at least 3 significant figures. Because only 3 significant digits are used to print parameter estimates in the output, and for other reasons as well, this amount of accuracy is often sufficient. However, the SIGDIGITS option can be used to request a different number (n) of significant digits.

MAXEVAL=n

The Estimation Step always runs with a specific limit on the number of objective function evaluations allowed during the search, as a protection against infinite loops and excessively long runs. The default maximum is computed according to the number of parameters being estimated. The MAXEVALS option can be used to request a different number (n) for the maximum number of function evaluations.

PRINT=n

As the Estimation Step progresses, by default it prints intermediate output summarizing the progress of the search. The search proceeds in stages, called iterations. At the end of certain iterations a summarization is printed which consists of the values of the objective function, its gradient vector with respect to the parameters, and the parameter values themselves. By default, this summarization is only printed for the first and last iterations. The PRINT option can be used to request a number (n) such that starting from the first iteration, only n-1 iterations are skipped before another summarization is printed†.

An example of the use of these options is:

```
$EST SIG=6,MAX=900,PRI=5
```

In addition to the first and last iterations, summarizations are printed every 5th iteration. Notice that abbreviations of the record and option names were used; this is a matter of style.

4.2. Requesting the Covariance Step: the $COVARIANCE Record

In the Covariance Step, NONMEM obtains information on the precision of the parameter estimates obtained in the Estimation Step. The output of this step are pages with titles "STANDARD ERROR OF ESTIMATE", "COVARIANCE MATRIX OF ESTIMATE", "CORRELATION MATRIX OF ESTIMATE", and "INVERSE COVARIANCE MATRIX OF ESTIMATE". This step is requested by the presence of the following record:

```
$COVARIANCE
```

There are several options, which are discussed in NONMEM Users Guide, Part IV. The Covariance Step cannot be requested by itself; the Estimation Step must precede it‡.

5. Specifying Optional Output

$TABLE and $SCATTERPLOT records are used to request NONMEM steps which generate additional output. If one of these records is omitted, NONMEM does not produce the corresponding additional output. Tables and scatterplots are generated *after* all other tasks have been performed. This affects the values displayed for PRED, RES, and WRES. If the Estimation Step is *not* run, then the *initial* estimates of the parameters are used to compute these quantities. If the Estimation Step *is* run, then the *final* parameter estimates are used. Residuals (RES) and weighted residuals (WRES) are defined in Chapter 11, Section 4.4.2.

The UNCONDITIONAL option of the $TABLE and $SCATTERPLOT records requests that output of this type be generated even if the Estimation Step terminates unsuccessfully, and is the default. The CONDITIONAL option of these records requests that output of this type be generated only if the Estimation Step terminates successfully.

---

† The PRINT option can also be used to suppress intermediate printout altogether, but this should usually not be done because the information is often of value. See Chapter 10, Section 4.

‡ The Estimation Step may be omitted when the run is continued from a prior run using a Model Specification input file; see Chapter 12, Section 4.3, and Chapter 13, Section 3.2.

5.1. Requesting the Table Step: the $TABLE Record

The values of DV, PRED, RES, and WRES are always printed in every table. Other data items to be printed should be listed on the record. The data items are printed in the order in which they are listed. This does not have to be the same order as in the data file, nor does every data item have to be listed. For example, the following record appears in Chapter 2, figure 2.12:

```
$TABLE ID TIME AMT WT APGR
```

Figure 10.10 in Chapter 10 shows a portion of the resulting output.

It is possible for the lines of a table to be sorted into a different order than that of the original input file; this is discussed in the NONMEM Users Guide, Part IV.

More than one table can be printed. A separate $TABLE record is used to request each one.

5.2. Requesting Scatterplots: the $SCATTERPLOT Record

Chapter 2 contained many examples of $SCATTERPLOT records and the resulting output. Here, for example, are the records from figure 2.6:

```
$SCATTERPLOT   PRED VS DV  UNIT
$SCATTERPLOT   RES  VS WT
```

The word UNIT requests a unit-slope line, which is the line PRED=DV. Figures 2.10 and 2.11 show the resulting output.

Similarly, the word ORD0 can be used to request a zero line on the ordinate axis.

It is possible to generate several scatterplots with a single record, by using a list of data item names:

```
$SCATTERPLOT  (PRED,RES,WRES)  VS WT
```

This produces three scatterplots, and has the same effect as the three records:

```
$SCATTERPLOT  PRED  VS WT
$SCATTERPLOT   RES  VS WT
$SCATTERPLOT  WRES  VS WT
```

Sometimes it is desirable to partition a scatterplot into a number of separate scatterplots. For example, if the data contain both plasma and urine observations, it would be better to separate the scatterplot of PRED vs. DV into one scatterplot where the DV values are the plasma observations, and another one where the DV values are the urine observations. To do this, it is necessary to specify a partitioning data item, in this case, the CMT data item, which gives the compartment number of the observation. The following record could be used.

```
$SCATTERPLOT PRED VS DV BY CMT UNIT
```

This will produce separate scatterplots: one with plasma observations (CMT=1 if ADVAN1 is used), and one with urine observations (CMT=2 if ADVAN1 is used).

Two partitioning items can also be specified:

```
$SCATTERPLOT PRED VS DV BY CMT SEX UNIT
```

One scatterplot is produced for each unique *combination* of values of the two partitioning data items.

6. Placement and Order of Records

Two main rules control the placement and order of records within a NM-TRAN control file:

> The $INPUT record must appear *before* any records which contain data item names ($PK, $ERROR, $TABLE, $SCATTERPLOT)

The $SUBROUTINE, $PK, and $ERROR records should appear in the indicated order, but do not have to be consecutive.

The records $DATA, $THETA, $OMEGA, $SIGMA, $ESTIMATION, $COVARIANCE, $TABLE, and $SCATTERPLOT can be placed anywhere among the control records, in any order. However, NONMEM always performs its tasks in a fixed order:

Estimation Step
Covariance Step
Table Step
Scatterplot Step

Thus, even if the $TABLE record precedes the $ESTIMATION record, the values of PRED, RES, and WRES in the table will be based on the final parameter estimates.

7. INCLUDE records

One or more records of the form

```
INCLUDE filename n
```

may appear anywhere among the NM-TRAN control records. The characters INCLUDE may be upper- or lower-case. "n" is an optional integer, and gives the number of copies (default is 1).

NM-TRAN opens the named file and reads it to end-of-file. The contents of the named file may be any portion of an NM-TRAN control stream, e.g., NM-TRAN control records and/or abbreviated code. After reaching end-of-file, if the number of copies is greater than 1, NM-TRAN rewinds the file and re-reads it the specified number of times. After reaching end-of-file on the final (or only) copy, NM-TRAN resumes reading the original control stream after the include record.

There may be more than one INCLUDE record, but they may not be nested. That is, an included file may not contain INCLUDE records.

For example,

```
$PROBLEM Model "a" with data set 27, proportional error
INCLUDE data27.def
INCLUDE modela.def
$ERROR Y=F+F*ERR(1)
$THETA 1.3 4
$OMEGA .04
$SIGMA 1
$ESTIMATION
```

The file data27.def contains the $INPUT and $DATA records.
The file modela.def contains the $SUBROUTINE record and $PK block.

Chapter 10 - Reading the Output

1. What This Chapter is About

This chapter describes NONMEM's output in detail. Each page of a NONMEM-PREDPP output file is shown and discussed.

The input file to NM-TRAN is that of figure 2.12, which is reproduced here as figure 10.1 for convenience.

```
 1    $PROBLEM PHENOBARB WITH WEIGHT IN MODELS FOR CL AND V
 2    $INPUT   ID TIME AMT WT APGR DV
 3    $DATA    INDATA
 4    $SUBROUTINE ADVAN1
 5    $PK
 6       TVCL=THETA(1)+THETA(3)*WT
 7       CL=TVCL+ETA(1)
 8       TVVD=THETA(2)+THETA(4)*WT
 9       V=TVVD+ETA(2)
10                          ; THE FOLLOWING ARE REQUIRED BY PREDPP
11       K=CL/V
12       S1=V
13    $ERROR
14       Y=F+ERR(1)
15    $THETA  (0,.0027) (0,.70) .0018  .5
16    $OMEGA  .000007, .3
17    $SIGMA   8
18    $ESTIMATION PRINT=5
19    $COVARIANCE
20    $TABLE ID TIME AMT WT APGR DV
21    $SCATTER   PRED VS DV  UNIT
22    $SCATTER   RES  VS WT
```

Figure 10.1. The NM-TRAN input file (same as figure 2.12). The line numbers on the left are not actually part of the file.

2. NONMEM Describes its Inputs

The first page of NONMEM's output is shown in figure 10.2. In this page, NONMEM repeats ("echos") the instructions it was given in the control file and describes the data file. The first page of the output should be checked carefully. Problems in a NONMEM run can often be traced to errors in the problem specification. For example, always check that the initial parameter estimates were entered correctly.

```
 1 NONLINEAR MIXED EFFECTS MODEL PROGRAM (NONMEM)    DOUBLE PRECISION NONMEM    VERSION IV LEVEL 1.0
 2 DEVELOPED AND PROGRAMMED BY STUART BEAL AND LEWIS SHEINER
 3
 4 PROBLEM NO.  1
 5 PHENOBARB WITH WEIGHT IN MODELS FOR CL AND V
 6
 7 DATA CHECKOUT RUN:              NO
 8 DATA SET LOCATED ON UNIT NO.:    2
 9 THIS UNIT TO BE REWOUND:        NO
10 NO. OF DATA RECS IN DATA SET:  744
11 NO. OF DATA ITEMS IN DATA SET:   8
12 ID DATA ITEM IS DATA ITEM NO.:   1
13 DEP VARIABLE IS DATA ITEM NO.:   6
14 MDV DATA ITEM IS DATA ITEM NO.:  8
15
16 INDICES PASSED TO SUBROUTINE PRED ARE:
17  7  2  3  0  0  0  0  0  0
18  0  0
19
20 LABELS FOR DATA ITEMS ARE:
21   ID    TIME     AMT      WT    APGR      DV    EVID     MDV
22
23 FORMAT FOR DATA IS:
24 (6E6.0,2F2.0)
25
26 TOT. NO. OF OBS RECS:     155
27 TOT. NO. OF INDIVIDUALS:   59
28
29 LENGTH OF THETA:  4
30
31 OMEGA HAS SIMPLE DIAGONAL FORM WITH DIMENSION:  2
32
33 SIGMA HAS SIMPLE DIAGONAL FORM WITH DIMENSION:  1
34
35 INITIAL ESTIMATE OF THETA:
36 LOWER BOUND    INITIAL EST    UPPER BOUND
37  0.0000E+00     0.2700E-02     0.1000E+07
38  0.0000E+00     0.7000E+00     0.1000E+07
39 -0.1000E+07     0.1800E-02     0.1000E+07
40 -0.1000E+07     0.5000E+00     0.1000E+07
41
42 INITIAL ESTIMATE OF OMEGA:
43 0.7000E-05
44 0.0000E+00   0.3000E+00
45
46 INITIAL ESTIMATE OF SIGMA:
47 0.8000E+01
48
49 ESTIMATION STEP OMITTED:           NO
50 NO. OF FUNCT. EVALS. ALLOWED:     360
51 NO. OF SIG. FIGURES REQUIRED:       3
52 INTERMEDIATE PRINTOUT:            YES
53 MSF OUTPUT:                        NO
54
55 COVARIANCE STEP OMITTED:    NO
56 EIGENVLS. PRINTED:    NO
57 SPECIAL COMPUTATION:  NO
58
59 TABLES STEP OMITTED:    NO
60 NO. OF TABLES:       1
61 TABLES PRINTED:    YES
62
63 USER-CHOSEN DATA ITEMS FOR TABLE  1,
64 IN THE ORDER THEY WILL APPEAR IN THE TABLE, ARE:
65   ID    TIME     AMT      WT    APGR
66
67 SCATTERPLOT STEP OMITTED:    NO
68 NO. OF PAIRS OF ITEMS GENERATING
69         FAMILIES OF SCATTERPLOTS:  2
70
71 ITEMS TO BE SCATTERED ARE:      DV    PRED
72     UNIT SLOPE LINE INCLUDED
73 ITEMS TO BE SCATTERED ARE:      WT    RES
```

Figure 10.2. The first page of the output report. The line numbers on the left are not actually part of the report.

Line 5 is an identification line for the output report. The contents of the $PROBLEM record are shown here.

Line 7 indicates that this is not a data checkout run. (Data checkout mode is discussed in Chapter 12 Section 4.10.) Lines 8 through 27 describe the input data file. Lines 10 and 11 describe the numbers of rows and columns in the input file, as shown in figure 6.1. Specifically, line 10 shows how many data records were read according to the FORTRAN format specification given in line 24. Line 11 describes the number of data items per record, which is the number of data items listed in the $INPUT record, less any that were

dropped by the Data Preprocessor, plus any that it added (see Chapter 6). Lines 12, 13, and 14 describe the locations of those data items of interest to NONMEM itself (i.e. NONMEM data items). Lines 16 through 18 are discussed in Section 3. Line 21 gives the labels for all the data items. The first six labels are those of the data items specified in the $INPUT record and the next two (EVID, MDV) are those of two data items added to the data set by the Data Preprocessor. (NONMEM itself supplies labels PRED, RES, and WRES for the prediction, residual, and weighted residual data items.) In the terminology of Chapter 4 (e.g. (4.15a)), ID, TIME, AMT, WT, and APGR are the elements of x; DV is y; PRED is f (evaluated for the typical individual in the population). Line 24 shows the format used to read each data record. In this example, the format was generated by the Data Preprocessor and describes the data file after processing by the Data Preprocessor.† Line 26 gives the number of observation records. Line 27 gives the number of individual records; that is, one less than the number of times that the ID data item changed value.

Lines 29 through 47 describe the contents of the $THETA, $OMEGA and $SIGMA records. First, the number of elements of θ, Ω and Σ are given (lines 29, 31 and 33), then their initial estimates are displayed. In lines 38-41, notice the values 0.1000e+07 and -0.1000e+07. These are NONMEM's way of expressing the values $+\infty$ and $-\infty$; i.e., of describing θs which are unbounded on one or both sides. Another FORTRAN system may display these numbers differently (e.g., 1.0000e+06), but the absolute value will always be 1,000,000. In lines 43 and 44, notice that the variances from the $OMEGA record appear along the diagonal of the Ω matrix, and that the off-diagonal element $cov(\eta_1, \eta_2)$ is zero. Line 31 states that NONMEM understands Ω to be diagonal; the off-diagonal element(s) are automatically fixed at zero.

The remaining lines of figure 10.2 describe the tasks that NONMEM will perform. Lines 49 through 53 describe the $ESTIMATION record. Lines 50 through 53 show the defaults (set by NM-TRAN) for various options, all of which could have been specified explicitly on the $ESTIMATION record. In line 50 for example, NONMEM displays the maximum number of times it will evaluate the objective function during the Estimation Step (this number can be slightly exceeded). The value 360 was supplied by NM-TRAN. It is a function of the sizes of θ, Ω, and Σ. Line 51 displays the desired number of significant digits in the final parameter estimate; the value 3 is the default number requested by NM-TRAN.

Lines 55 through 59 describe the $COVARIANCE record, giving the default options chosen by NM-TRAN.

Lines 59 through 61 describe the $TABLE record. Lines 67 through 73 describe the $SCATTERPLOT records.

3. PREDPP Describes Its Inputs

The next two pages are produced by PREDPP and will not appear if $PRED statements (or a user-written PRED subroutine) are used. PREDPP uses these pages to repeat ("echo") the instructions it was given in the control file, and to identify the ADVAN and TRANS routines chosen by the user. The first page of PREDPP's output is shown in figure 10.3.

In its first page of output, PREDPP describes the features of the pharmacokinetic model and its parameterization encoded into the ADVAN and TRANS routines specified on the $SUBROUTINE record. The information displayed here includes the kind of information summarized in Appendices 1 and 2. In the particular output of Figure 10.3 no

† When a format specification is supplied on the $DATA record, and no data items are dropped or added by the Data Preprocessor, the original format specification is used unchanged and appears here.

```
 1 DOUBLE PRECISION PRED   VERSION III LEVEL 1.0
 2
 3 ONE COMPARTMENT MODEL (ADVAN1)
 4
 5 MAXIMUM NO. OF BASIC PK PARAMETERS:   2
 6
 7 BASIC PK PARAMETERS (AFTER TRANSLATION):
 8   ELIMINATION RATE (K) IS BASIC PK PARAMETER NO.:  1
 9
10
11 COMPARTMENT ATTRIBUTES
12 COMPT. NO.   FUNCTION   INITIAL    ON/OFF      DOSE      DEFAULT    DEFAULT
13                         STATUS     ALLOWED    ALLOWED    FOR DOSE   FOR OBS.
14    1         CENTRAL      ON         NO         YES        YES        YES
15    2         OUTPUT       OFF        YES        NO         NO         NO
```

Figure 10.3. The first page of PREDPP's output. The line numbers on the left are not actually part of the report.

information concerning an alternate parameterization appears because TRANS1 was specified. The information concerning basic parameters and compartments is displayed in a format similar to that used in NONMEM Users Guide, Part VI, which is the complete reference for PREDPP.

Lines 5 and 8 describe the basic PK parameters, which in this example is the single microconstant K. If a translator other than TRANS1 had been requested, an additional line would appear describing the translation. E.g., with TRANS2, this line would read:

```
TRANSLATOR WILL CONVERT PARAMETERS CLEARANCE (CL) AND VOLUME (V) to K
```

Lines 10 through 14 describe the compartment attributes. Even though the output compartment is never turned on by the data of this example, its attributes are described here because it is part of the model.

The information presented so far describes the model for computing drug amounts. For a given choice of ADVAN and TRANS, the contents of this page are completely fixed. PREDPP's second page of output describes user choices related to the given ADVAN routine, including choices for the scale parameters (and thus, to the model for computing concentrations). This page is shown in figure 10.4.

```
 1 ADDITIONAL PK PARAMETERS - ASSIGNMENT OF ROWS IN GG
 2 COMPT. NO.                             INDICES
 3              SCALE      BIOAVAIL.   ZERO-ORDER  ZERO-ORDER  ABSORB
 4                         FRACTION    RATE        DURATION    LAG
 5    1           3           *           *           *           *
 6    2           *           -           -           -           -
 7             - PARAMETER IS NOT ALLOWED FOR THIS MODEL
 8             * PARAMETER IS NOT SUPPLIED BY PK SUBROUTINE;
 9               WILL DEFAULT TO ONE IF APPLICABLE
10
11 DATA ITEM INDICES USED BY PRED ARE:
12   EVENT ID DATA ITEM IS DATA ITEM NO.:      7
13   TIME DATA ITEM IS DATA ITEM NO.:          2
14   DOSE AMOUNT DATA ITEM IS DATA ITEM NO.:   3
15
16
17 PK SUBROUTINE CALLED WITH EVERY EVENT RECORD.
18 PK SUBROUTINE NOT CALLED AT ADDITIONAL DOSE OR LAGGED DOSE TIMES.
19
20 DURING SIMULATION, ERROR SUBROUTINE CALLED WITH EVERY EVENT RECORD.
21 OTHERWISE, ERROR SUBROUTINE CALLED ONCE IN THIS PROBLEM.
```

Figure 10.4. The second page of PREDPP's output. The line numbers on the left are not actually part of the report.

Lines 2 through 9 describe the additional PK parameters that are computed by the $PK statements (or PK subroutine). In line 5, the position marked with "3" corresponds to the scale parameter for compartment number 1. Thus, we know that the $PK statements contained an assignment statement for S1. From the prior page we can see that compartment number 1 is the central compartment. The value "3" is a row number within GG, an array used for communication between PREDPP and the PK subroutine. With the use of NM-TRAN and $PK statements, row numbers are of no interest to the user. With a user-written PK subroutine, it is important to check their correctness. Positions marked with "*"

correspond to additional PK parameters that are allowed by the model but that are not assigned a value by $PK statements; an example is F1, the bioavailability fraction for compartment 1. Positions marked with "-" correspond to additional parameters that may not be computed; for instance, dose-related parameters are not allowed for the output compartment, because (as shown on the preceding page) this compartment cannot receive doses.

Lines 11 through 14 describe the locations in the input data record of those data items of interest to PREDPP (PREDPP data items). (NM-TRAN causes the locations of these data items in the data set to be passed by NONMEM to PREDPP, as indicated in lines 15 through 17 of figure 10.2. NONMEM is not concerned with the significance of these data items.) Note that data item 7, Event ID, was appended by the Data Preprocessor.

Line 17 reflects the fact that, by default, $PK statements are evaluated with every event record†. Lagged and additional doses are discussed in Chapter 12, Sections 2.4 and 2.5. They are not used in this example.

Line 21 reflects the fact that the $ERROR statements describe the simple error model (3.4). This model uses no data items and no elements of θ whatsoever (directly or indirectly). NM-TRAN has instructed PREDPP that the $ERROR statements need be evaluated only once at the beginning of the problem. Line 20 indicates that, should the Simulation Step be implemented, PREDPP will disregard this limitation and evaluate the $ERROR statements with every event record, so that randomly-generated values of intra-individual error can be applied at every observation event. (This example does not involve simulation, but the PK and ERROR routines which implement the $PK and $ERROR statements are capable of supporting all NONMEM tasks, including simulation.)

Finally, note that the $PK and $ERROR models (figure 10.1, lines 5-14) are not documented in the NONMEM-PREDPP output. It is a good idea to attach a printed copy of the NM-TRAN input records to the corresponding NONMEM output. MS/DOS batch file nmfe73.bat and Unix C-shell script nmfe73 (supplied with NONMEM) do this automatically.

4. Diagnostic Output from the Estimation Step

The next page of output, figure 10.5, is produced during the running of the Estimation Step.

4.1. Intermediate Output from the Estimation Step

Lines 1 through 42 are referred to as the intermediate output. Lines 4 through 7 give numbers summarizing the 0-th iteration, which are based on the initial parameter estimates. Line 4 shows the initial value of the objective function. The value following "NO. OF FUNC. EVALS." is the number of objective function evaluations which were needed during the iteration. Line 5 gives the cumulative number of function evaluations including this and all prior iteration summaries.
Line 6 gives the unconstrained parameter (UCP) estimates. The search is carried out in a different parameter space. The parameters are transformed to unconstrained parameters (UCP). In the transformation process a scaling occurs so that the initial estimate of each of the UCP is 0.1. Thus, in line 6, all parameters are .1 at the 0-th iteration. Parameters

† In this example, the $PK statements (lines 5 through 12 of the input file, figure 10.1) involve only WT, which is constant for each individual. It is possible to limit the event records with which the $PK statements are evaluated to the first event record of each individual, in order to reduce run time. This decision is left to the user.

```
 1 MONITORING OF SEARCH:
 2
 3
 4 ITERATION NO.:    0     OBJECTIVE VALUE:  0.6757E+03     NO. OF FUNC. EVALS.: 8
 5 CUMULATIVE NO. OF FUNC. EVALS.:    8
 6 PARAMETER:  0.1000E+00  0.1000E+00  0.1000E+00  0.1000E+00  0.1000E+00  0.1000E+00
 7 GRADIENT:  -0.7986E+03 -0.1594E+04 -0.4294E+03 -0.1000E+04  0.1542E+03  0.5269E+03  0.9128E+02
 8
 9 ITERATION NO.:    5     OBJECTIVE VALUE:  0.6502E+03     NO. OF FUNC. EVALS.:10
10 CUMULATIVE NO. OF FUNC. EVALS.:   58
11 PARAMETER:  0.8878E-01  0.1003E+00  0.2055E+00  0.1296E+00  0.6695E-01  0.7822E-01  0.1071E+00
12 GRADIENT:   0.1060E+04  0.2567E+04  0.3675E+03  0.8472E+03 -0.1807E+03 -0.5093E+03  0.9841E+02
13
14 ITERATION NO.:   10     OBJECTIVE VALUE:  0.6153E+03     NO. OF FUNC. EVALS.: 9
15 CUMULATIVE NO. OF FUNC. EVALS.:  107
16 PARAMETER:  0.5008E-01  0.6626E-01  0.2425E+00  0.1663E+00 -0.6718E-01  0.6382E-01  0.1004E+00
17 GRADIENT:   0.9732E+02  0.3034E+03  0.3185E+02  0.1228E+03 -0.1162E+03  0.1252E+03  0.6450E+02
18
19 ITERATION NO.:   15     OBJECTIVE VALUE:  0.6108E+03     NO. OF FUNC. EVALS.: 9
20 CUMULATIVE NO. OF FUNC. EVALS.:  152
21 PARAMETER:  0.4235E-01  0.4508E-01  0.2462E+00  0.1831E+00 -0.5721E-01  0.5237E-01  0.1008E+00
22 GRADIENT:   0.3989E+02  0.7394E+02 -0.1782E+01  0.8527E+02 -0.9309E+02  0.1867E+02 -0.1773E+02
23
24 ITERATION NO.:   20     OBJECTIVE VALUE:  0.6095E+03     NO. OF FUNC. EVALS.: 9
25 CUMULATIVE NO. OF FUNC. EVALS.:  197
26 PARAMETER:  0.1927E-01  0.3153E-01  0.2615E+00  0.1898E+00 -0.4458E-01  0.4904E-01  0.1047E+00
27 GRADIENT:   0.1609E+02 -0.3621E+02  0.5228E+01  0.9614E+00 -0.1740E+02  0.1329E+02  0.3111E+01
28
29 ITERATION NO.:   25     OBJECTIVE VALUE:  0.6091E+03     NO. OF FUNC. EVALS.: 9
30 CUMULATIVE NO. OF FUNC. EVALS.:  242
31 PARAMETER:  0.2389E-02  0.4171E-01  0.2652E+00  0.1833E+00 -0.4413E-01  0.4998E-01  0.1043E+00
32 GRADIENT:   0.2273E+01 -0.5333E+01  0.3914E+01 -0.5397E+01  0.1271E+01  0.2610E+01  0.3584E+00
33
34 ITERATION NO.:   30     OBJECTIVE VALUE:  0.6091E+03     NO. OF FUNC. EVALS.:16
35 CUMULATIVE NO. OF FUNC. EVALS.:  299
36 PARAMETER: -0.1278E-03  0.4166E-01  0.2650E+00  0.1835E+00 -0.4414E-01  0.5003E-01  0.1043E+00
37 GRADIENT:  -0.1120E+00 -0.9411E+00 -0.3719E+00 -0.2540E+01 -0.5135E-01  0.1420E+00 -0.9524E-01
38
39 ITERATION NO.:   32     OBJECTIVE VALUE:  0.6091E+03     NO. OF FUNC. EVALS.: 0
40 CUMULATIVE NO. OF FUNC. EVALS.:  315
41 PARAMETER: -0.7284E-05  0.4150E-01  0.2650E+00  0.1836E+00 -0.4411E-01  0.5003E-01  0.1043E+00
42 GRADIENT:  -0.6416E-02  0.9336E-01  0.4548E-01  0.4826E-01  0.1263E-02  0.9652E-01  0.4629E-01
43
44 MINIMIZATION SUCCESSFUL
45 NO. OF FUNCTION EVALUATIONS USED:  315
46 NO. OF SIG. DIGITS IN FINAL EST.:  3.9
```

Figure 10.5. The output from the Estimation Step. The line numbers on the left are not actually part of the report.

are printed in the following order: elements of θ, elements of Ω, elements of Σ. In this example, reading from left to right, the parameters are θ_1, θ_2, θ_3, θ_4, Ω_{11}, Ω_{22}, and Σ_{11}.

Two points should be noted. First, fixed parameters do not appear in the list. Therefore, the off-diagonal element Ω_{12}, which is effectively fixed to 0, does not appear. Second, when off-diagonal elements of Ω are being estimated, then as many additional UCP's appear as there are off-diagonal elements of Ω being estimated. However, a 1-1 correspondence between each of the elements of Ω and an UCP does not exist. The same is true for elements of Σ and the UCP's for Σ when off-diagonal elements of Σ are estimated.

With NONMEM 7, the parameter estimates are also displayed in their natural (unscaled) space. These lines are identified as NPARAMETR and precede the PARAMETER lines, which display the UCP values.

Line 7 shows the gradient for each parameter, which may be thought of as the partial derivative of the objective function with respect to that parameter.

The Estimation Step proceeds in a series of stages called iterations. In this example, intermediate printout is produced for each of every 5 iterations, as well as for the 0-th and final iterations, for which intermediate printout is always printed by default. This printout consists of the same four lines as for the 0-th iteration, but using the parameters estimates obtained at the end of the iteration.

In lines 4, 9, 14, 19, 24, 29, 34, and 39, observe that the objective function drops quickly at first, and then more slowly. After iteration number 25, there is no change above the

fourth significant digit.

In lines 6, 11, 16, 21, 26, 31, 36, and 41, observe that each parameter also changes rapidly at first and then more slowly as it converges to its final value. (The first parameter, θ_1, is an exception. It is clearly approaching a very small value close to its lower bound, 0. In Chapter 12, we shall see that both θ_1 and θ_2 are best fixed at 0.)

Finally, in lines 7, 12, 17, 22, 27, 32, 37, and 42, observe that the gradients also approach 0, another sign that a minimum of the objective function has been located.

The values computed for the gradients are very sensitive to differences in computer arithmetic and precision. If a given NONMEM run is repeated on a different computer, or on the same computer with different machine precision or a different FORTRAN compiler, it is likely that the gradients will be different. This will cause the search to follow a different path to the minimum, so that lines 4 through 42 may be quite different. However, each final estimate of a UCP should always be the same to the number of requested significant digits. (Minor differences may also be observed in the output of the Covariance Step, below; this output is also sensitive to computational differences.)

4.2. Summary Output from the Estimation Step

Lines 44, 45 and 46 are always printed, even when intermediate printout is suppressed. Line 44, "MINIMIZATION SUCCESSFUL", signifies that the search appears to have located a minimum of the objective function. Before one can be certain that a minimum has been located, or one which corresponds to a reasonable parameter estimate (there can be a number of "local minima"), the final parameter estimates must be examined in their (untransformed) state; see Section 5 below. The Estimation Step is not always successful. Chapter 13 discusses two other messages that sometimes appear instead of line 44.

In line 45, note that the number of function evaluations used, 315, is a total value and includes all iterations (not just those for which intermediate printout was displayed). This is under the limit of 360 supplied by NM-TRAN (figure 10.2, line 57).

The number of significant digits in the final estimate is given in line 34 as 3.9. This can be interpreted as meaning that no (transformed) *parameter estimate* is actually determined to less than 3.9 significant digits. More specifically, when the UCP estimates were compared between the last two iterations, none differed in the first (almost) 4 significant figures *including* leading zeros after the decimal point. Note that the final θ_1 UCP estimate is -0.7284E-05, and so the 7284 are not significant digits at all! Because NONMEM displays only 3 significant digits in the printed parameter estimates, and for other reasons as well, by default NM-TRAN requests only 3 significant digits. However, more significance can be requested, as was discussed in Chapter 9, Section 4.1.

5. Minimum Value of the Objective Function and Final Parameter Estimates

The next two pages in the NONMEM output are produced whether or not the Estimation Step was implemented and, if it was, whether or not the search terminated successfully. They give the values of the objective function and the parameter estimates, using the final parameter estimates if the Estimation Step was implemented (whether or not the search terminated successfully), and using the initial parameter estimates otherwise. These pages have already been shown in Chapter 2, figure 2.13. Even when the minimization routine is successful in locating a minimum of the objective function, the final (untransformed) parameter estimates must be carefully checked. Is any parameter's final estimate physiologically unreasonable? Is any parameter's final estimate near its upper or lower constraint? If either answer is yes, the model, the constraints on θ's, or the data may be

incorrect; see Chapter 11.

Sometimes the final estimates do not match anticipated values, e.g., values obtained by some other system of analysis. Additional refinement of the model may be needed, as discussed in Chapter 11. However, the discrepancy may well be traceable to an error in model specification, such as an error in specifying a compartment's scale. Along with the Estimation Step, it is important to obtain a scatterplot of PRED vs DV and make sure the unit slope line is visible. See Chapter 13, Section 4.4.

6. Output from the Covariance Step

Figures 10.6 through 10.7 show the output of the Covariance Step, which was requested via the $COVARIANCE record. Figure 10.6 has already been displayed as figure 2.14, but is included here for completeness. This page displays the standard errors of the parameter estimates. Standard errors are discussed extensively in Chapters 5 and 11. A detailed discussion of the remaining three pages, containing the covariance, correlation, and inverse covariance matrices, is beyond the scope of this text. Note, however, the use of the notation ".....". Each sequence of dots denotes a value (such as the standard error in the estimate of Ω_{12}) that is 0 by definition, rather than due to a computation.

```
 1 ********************************************************************************************************************************
 2 ********************                                                                                           ************
 3 ********************                          STANDARD ERROR OF ESTIMATE                                       ************
 4 ********************                                                                                           ************
 5 ********************************************************************************************************************************
 6
 7
 8
 9 THETA - VECTOR OF FIXED EFFECTS  **********************
10
11
12            TH 1      TH 2      TH 3      TH 4
13
14         9.49E-11  1.46E-01  2.24E-04  1.13E-01
15
16
17
18 OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********
19
20
21            ETA1      ETA2
22
23 ETA1    7.24E-07
24
25 ETA2    .........  3.63E-02
26
27
28
29 SIGMA - COV MATRIX FOR RANDOM EFFECTS - EPSILONS  ****
30
31
32            EPS1
33
34 EPS1    1.71E+00
```

Figure 10.6. Standard error of the estimate. The line numbers on the left are not actually part of the report.

```
 1 ************************************************************************************************************************
 2 ********************                                                                                ************
 3 ********************                          COVARIANCE MATRIX OF ESTIMATE                         ************
 4 ********************                                                                                ************
 5 ************************************************************************************************************************
 6
 7
 8            TH 1      TH 2      TH 3      TH 4      OM11      OM12      OM22      SG11
 9
10 TH 1    9.02E-21
11
12 TH 2    3.93E-12  2.14E-02
13
14 TH 3   -5.23E-15 -1.45E-05  5.00E-08
15
16 TH 4   -3.69E-12 -1.57E-02  1.04E-05  1.27E-02
17
18 OM11   -1.11E-17  2.13E-08 -6.39E-12 -1.52E-08  5.24E-13
19
20 OM12   ......... ......... ......... ......... ......... .........
21
22 OM22   -1.79E-14  4.40E-04 -5.30E-07  5.58E-04  6.27E-10 .........  1.32E-03
23
24 SG11    1.04E-11 -5.69E-02  1.12E-04  4.45E-02 -3.74E-07 ......... -1.03E-02  2.92E+00
```

Figure 10.7. Covariance matrix of the estimate. The line numbers on the left are not actually part of the report.

```
 1 ************************************************************************************************************************
 2 ********************                                                                                ************
 3 ********************                          CORRELATION MATRIX OF ESTIMATE                        ************
 4 ********************                                                                                ************
 5 ************************************************************************************************************************
 6
 7
 8            TH 1      TH 2      TH 3      TH 4      OM11      OM12      OM22      SG11
 9
10 TH 1    1.00E+00
11
12 TH 2    2.83E-01  1.00E+00
13
14 TH 3   -2.46E-01 -4.44E-01  1.00E+00
15
16 TH 4   -3.45E-01 -9.53E-01  4.13E-01  1.00E+00
17
18 OM11   -1.61E-01  2.01E-01 -3.95E-02 -1.86E-01  1.00E+00
19
20 OM12   ......... ......... ......... ......... ......... .........
21
22 OM22   -5.21E-03  8.29E-02 -6.53E-02  1.37E-01  2.39E-02 .........  1.00E+00
23
24 SG11    6.44E-02 -2.28E-01  2.94E-01  2.31E-01 -3.02E-01 ......... -1.66E-01  1.00E+00
```

Figure 10.8. Correlation matrix of the estimate. The line numbers on the left are not actually part of the report.

```
 1 ************************************************************************************************************************
 2 ********************                                                                                ************
 3 ********************                  INVERSE COVARIANCE MATRIX OF ESTIMATE                         ************
 4 ********************                                                                                ************
 5 ************************************************************************************************************************
 6
 7
 8            TH 1      TH 2      TH 3      TH 4      OM11      OM12      OM22      SG11
 9
10 TH 1    1.56E+20
11
12 TH 2    1.46E+11  1.25E+03
13
14 TH 3    1.35E+13  4.42E+04  2.80E+07
15
16 TH 4    2.32E+11  1.63E+03  3.98E+04  2.23E+03
17
18 OM11    3.04E+15 -3.96E+06 -7.46E+08 -1.76E+06  2.26E+12
19
20 OM12   ......... ......... ......... ......... ......... .........
21
22 OM22   -1.56E+11 -1.14E+03 -2.82E+04 -1.55E+03  2.82E+06 .........  1.86E+03
23
24 SG11   -1.93E+09 -7.14E+00 -1.06E+03 -1.03E+01  2.67E+05 .........  9.91E+00  4.78E-01
```

Figure 10.9. Inverse covariance matrix of the estimate. The line numbers on the left are not actually part of the report.

7. Additional Output: Tables and Scatterplots

The use of $TABLE and $SCATTERPLOT records to request tables and scatterplots is discussed in Chapter 9.

7.1. Output from the Table Step

The first 12 lines of the table produced by the $TABLE record are shown in figure 10.10. This is the data for the first individual.

```
 1 TABLE NO.  1
 2
 3
 4
 5 LINE NO.     ID      TIME       AMT        WT      APGR        DV      PRED      RES       WRES
 6
 7    1    1.00E+00  0.00E+00  2.50E+01  1.40E+00  7.00E+00  0.00E+00  1.78E+01  0.00E+00  0.00E+00
 8
 9    2    1.00E+00  2.00E+00  0.00E+00  1.40E+00  7.00E+00  1.73E+01  1.76E+01 -3.14E-01 -2.92E-01
10
11    3    1.00E+00  1.25E+01  3.50E+00  1.40E+00  7.00E+00  0.00E+00  1.92E+01  0.00E+00  0.00E+00
12
13    4    1.00E+00  2.45E+01  3.50E+00  1.40E+00  7.00E+00  0.00E+00  2.07E+01  0.00E+00  0.00E+00
14
15    5    1.00E+00  3.70E+01  3.50E+00  1.40E+00  7.00E+00  0.00E+00  2.20E+01  0.00E+00  0.00E+00
16
17    6    1.00E+00  4.80E+01  3.50E+00  1.40E+00  7.00E+00  0.00E+00  2.33E+01  0.00E+00  0.00E+00
18
19    7    1.00E+00  6.05E+01  3.50E+00  1.40E+00  7.00E+00  0.00E+00  2.45E+01  0.00E+00  0.00E+00
20
21    8    1.00E+00  7.25E+01  3.50E+00  1.40E+00  7.00E+00  0.00E+00  2.56E+01  0.00E+00  0.00E+00
22
23    9    1.00E+00  8.53E+01  3.50E+00  1.40E+00  7.00E+00  0.00E+00  2.66E+01  0.00E+00  0.00E+00
24
25   10    1.00E+00  9.65E+01  3.50E+00  1.40E+00  7.00E+00  0.00E+00  2.77E+01  0.00E+00  0.00E+00
26
27   11    1.00E+00  1.08E+02  3.50E+00  1.40E+00  7.00E+00  0.00E+00  2.87E+01  0.00E+00  0.00E+00
28
29   12    1.00E+00  1.12E+02  0.00E+00  1.40E+00  7.00E+00  3.10E+01  2.81E+01  2.88E+00  6.88E-01
```

Figure 10.10. A portion of a NONMEM table. The line numbers on the left are not actually part of the report.

Each row in the table corresponds to a record of the input file, and the rows appear in the same order as do the corresponding records of the input data file. Note that the values of RES and WRES are always shown as zero for non-observation records†, whereas a (possibly) nonzero value of PRED is printed for every record.

If there are more than 900 data records, separate tables are produced for groups of 900 records. The last table contains the remaining records. If the rows of the table are sorted, each group of records is sorted separately. When the input data file is large, the table will require many pages to print. Therefore, the $TABLE record should be omitted unless needed for diagnostic purposes (such as when initially checking a new data set or model).

7.2. Output from the Scatterplot Step

Many examples of scatterplots are present in Chapters 2 and 11. They are not reproduced here. Whereas all the records in the input data file correspond to rows of a table, this is not true of a scatterplot that includes one or more of the items RES, WRES, and DV. When one of these three is being plotted, then only observation records contribute points to the scatterplot†. In figure 2.5, there are exactly 10 points "*", corresponding to the 10 observation records in figure 2.2; the dose record does not contribute a point.

NONMEM displays only the first 900 records of the appropriate type in a scatterplot. This limit applies before any partitioning. For example, in a plot of DV VS ID, the first 900 observation records are displayed; in a plot of WT vs ID, the first 900 records of the data file are displayed. Additional scatterplots can be requested, showing additional points, using options "FROM=" and "TO=" of the $SCATTERPLOT record. See NONMEM Users Guide, Part IV.

---

† Strictly speaking, RES and WRES are always zero for records having MDV=1. With PREDPP, this is the same thing.

† Strictly speaking, it is only the records having MDV=0 that contribute points. With PREDPP, this is the same thing.

Chapter 11 - Model Building

1. What This Chapter is About

In this chapter, the simple phenobarbital example begun in Chapter 2 will be continued to illustrate how NONMEM is used to build a model for population data. The topic of model building, diagnosis and verification is a large one. This chapter can only give a very abbreviated example.

2. The Stages of Model Building

To analyze a population data set and build a model for it, one must proceed in logical stages. There are five stages, and their relationship to one another is presented diagrammatically in figure 11.1. One begins by checking the data. One then tries to find an adequate model incorporating the fixed effects; then an adequate model incorporating the random effects and describing random inter- and intra-individual variability. After a reasonably complete model is found, attempts are made to refine it, and finally, if desired, the various parts of the models (which often, in effect, simply assert the existence of certain relationships between independent variables and the dependent variable) can be subject to formal hypothesis tests, as described in Chapter 5. (However, it is well known by statisticians that formal hypothesis testing undertaken *after* model building is just an approximation for the type of hypothesis testing described in textbooks, which assumes that the model is the correct model).

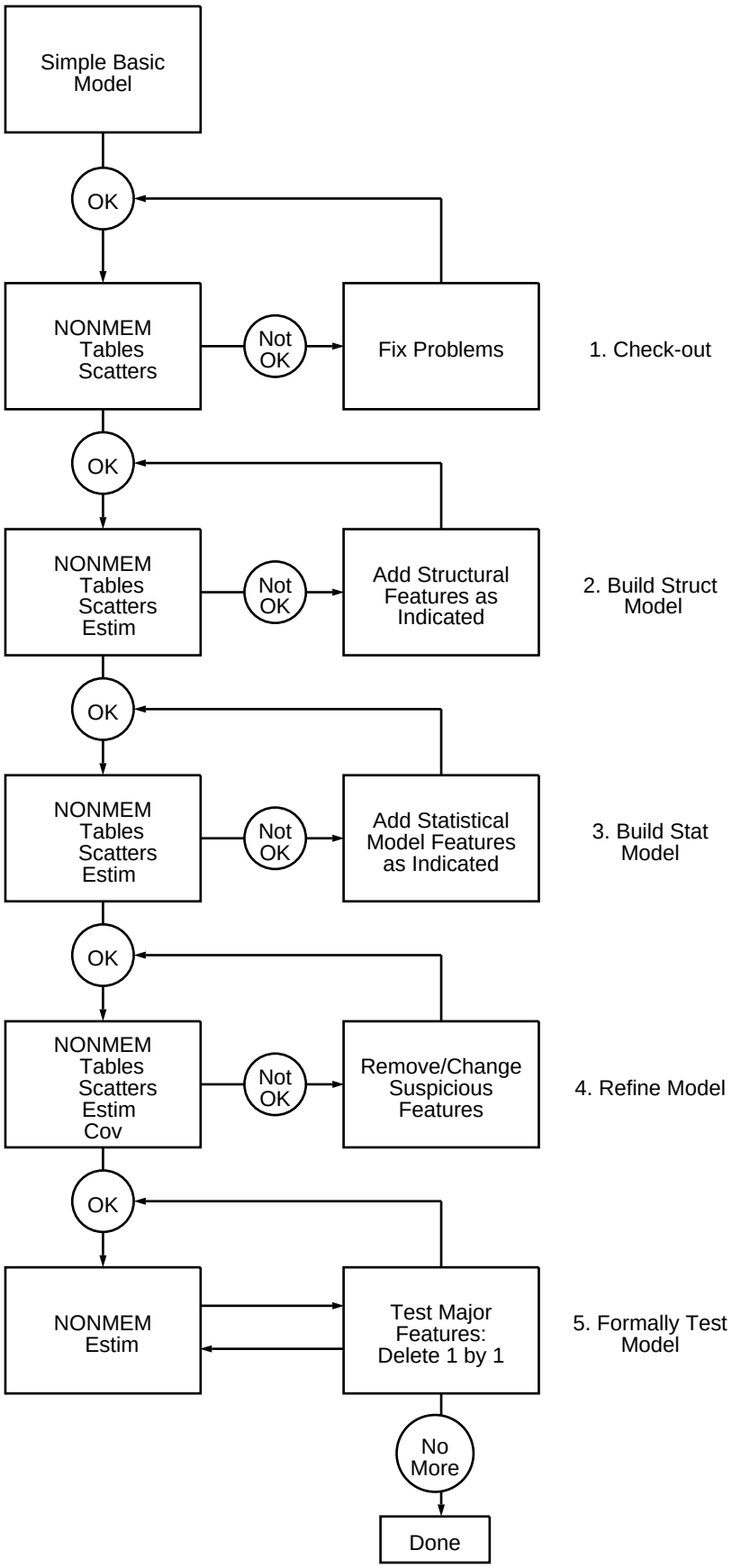
Simple Basic Model
OK
NONMEM Tables Scatters
Not OK
Fix Problems
1. Check-out
OK
NONMEM Tables Scatters Estim
Not OK
Add Structural Features as Indicated
2. Build Struct Model
OK
NONMEM Tables Scatters Estim
Not OK
Add Statistical Model Features as Indicated
3. Build Stat Model
OK
NONMEM Tables Scatters Estim Cov
Not OK
Remove/Change Suspicious Features
4. Refine Model
OK
NONMEM Estim
Test Major Features: Delete 1 by 1
5. Formally Test Model
No More
Done

Figure 11.1. Stages in model building.

3. Check-out — Index Plots

The goal of this stage is to assure that the data are correct. There is no point to trying to model the data when gross errors are present. Most gross errors are encoding errors that cause certain values to be different from the intended value by a considerable amount (for example, a misplaced decimal point changes a value by a factor of 10), so that graphical display of the data is usually adequate to detect these. No numerical or statistical approaches are needed. Indeed, they are not usually useful, even for more subtle errors, as such errors cannot easily be detected by any means (how is a 10% error to be distinguished from inter- or intra-individual variability?).

To detect gross errors, then, one makes scatterplots of different data item types vs individuals' identification numbers (i.e. the ID data item, or, if the values of this data item are arbitrary, another data item that identifies patients using sequential integer values; call this the sequence data item: SEQ). Such plots (of one data item versus ID or SEQ) are called here index plots, and are quite useful for revealing the structure of the data, as will be noted below, as well as for finding gross errors.

If NONMEM is used to make index plots, it will also be useful to implement the Tables Step, so that if a problem is noted in a scatterplot, one can refer to the table to try to find the datum that might account for the problem. To run NONMEM some model must be specified, even if all that is desired is an index plot. In such case, it makes little difference what model is used. It is easiest and useful to (i) start with a simple ADVAN that is likely to provide at least a roughly satisfactory fit, (ii) set each PK parameter to a (different) element of θ, (iii) use only one η variable, modifying the scale parameter only, and one ε variable, and (iv) use roughly reasonable fixed initial estimates.

For the phenobarbital example, one might use ADVAN1 with $K = \theta_1$, $V = \theta_2 + \theta_2\eta_1$, and $y = f + f\varepsilon_1$. Initial estimates might be: K = .0057 hr $^{-1}$ (half-life = 5 days, a typical value for adults); V= 1.44 L. (the first patient has a concentration of 17.3 mg/L some few hours after an initial loading dose of 25 mg; 1.44 times 17.3 = 25); ω = .25 (50% variability); σ^2 = .04 (20% variability).

Figures 11.2 and 11.3 show index plots that might be seen in a check-out run (gross errors have been added). In figure 11.2, DV is plotted vs ID (here ID and SEQ are the same), and a gross error occurring at about patient #13 is seen (an observation of about 24 mcg/ml was erroneously recorded as 240 mcg/ml). In figure 11.3, AMT is plotted vs ID, and patient #3 appears to have a grossly erroneous value (again, a decimal point error; a dose of 18 was misrecorded as 180).

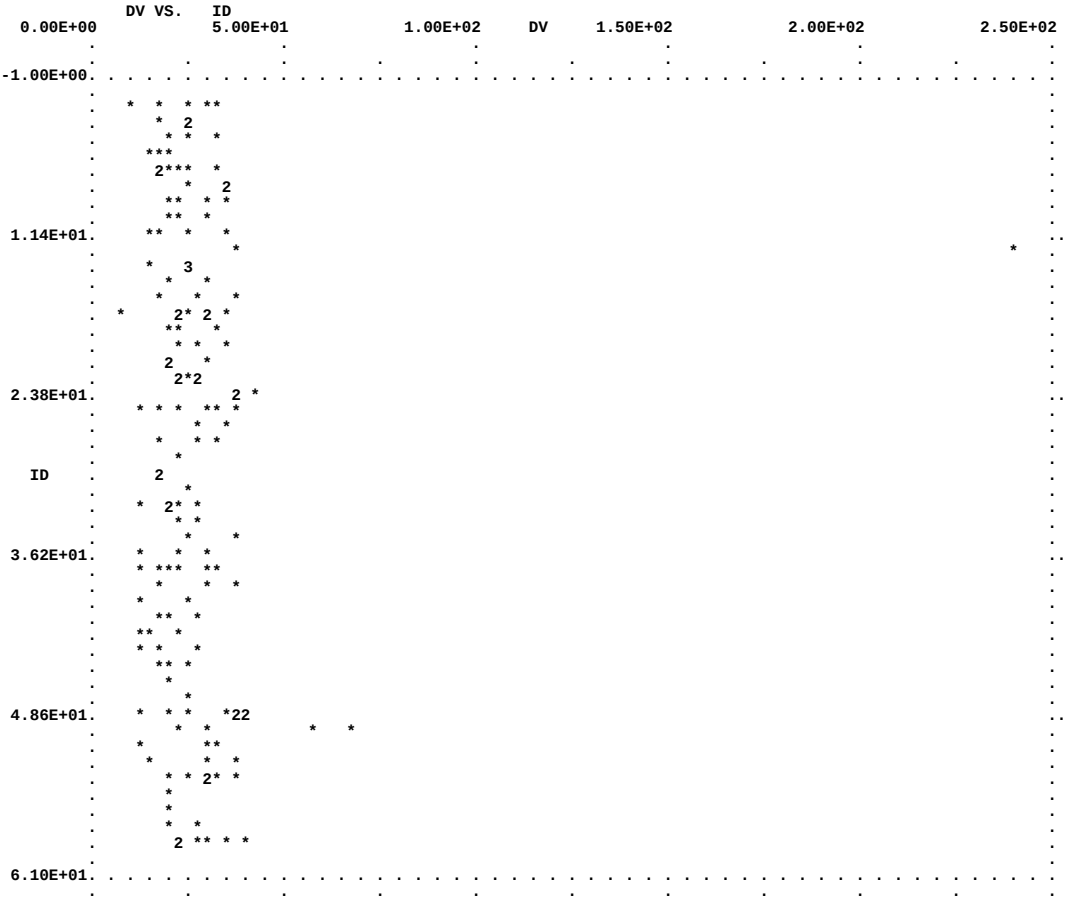

Figure 11.2. A scatterplot of the dependent variable, DV, vs the patient's ID number (a type of index plot). Note the outlier at about ID = 13.

Actually, figure 11.3 reveals a considerable amount about the data structure (this will be seen better in figure 11.4, below, when the outlier has been removed). Many points lie along the line AMT = 0, where one sees integers 5, 3, 3, 6, etc, as one proceeds along the ID axis, each integer indicating the corresponding number of points over-plotted at that location. They correspond to the observation records, since the doses on these records are all zero. Thus one can see how many observations each individual contributes (other type records would also plot at AMT=0, however). Proceeding to the next highest "line" of doses (where many points over-plot for each patient), one "sees" the event records giving the maintenance dose amount since this amount stays constant within a individual (many maintenance doses were given per individual), and this amount is approximately the same across individuals. Last, at the highest doses (except for the outlier), one has mostly single points. These are the loading doses. There is occasional over-plotting of loading-dose points. These points represent overlapping patient ID numbers (at the resolution of the NONMEM plot), not multiple loading doses to the same patient.

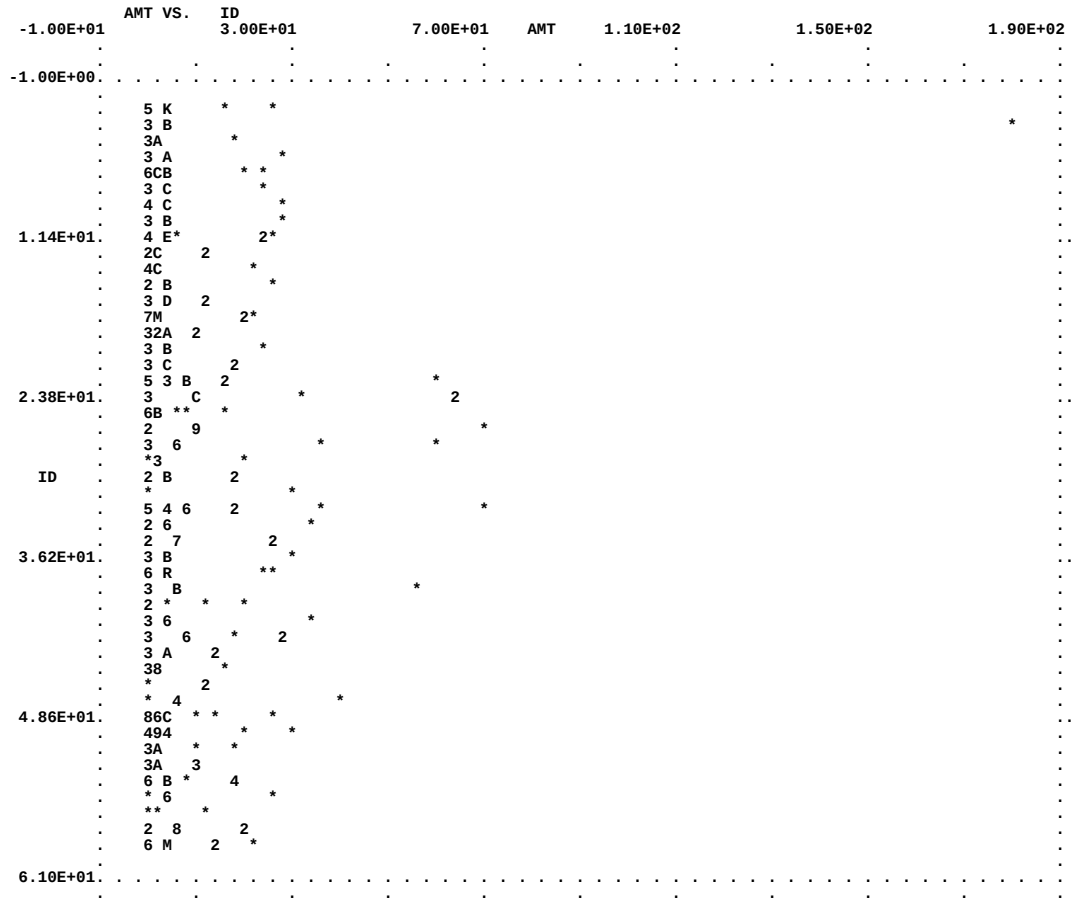

Figure 11.3. An index plot of the independent variable, dose amount (AMT). Note the outlier at about ID = 3.

Figure 11.4 replots the same data as figure 11.3 but with the misrecorded values of the data items corrected. Figures 11.5 and 11.6 show the index plots for the other two data items of interest to this data analysis: weight (WT) and Apgar score (APGR).

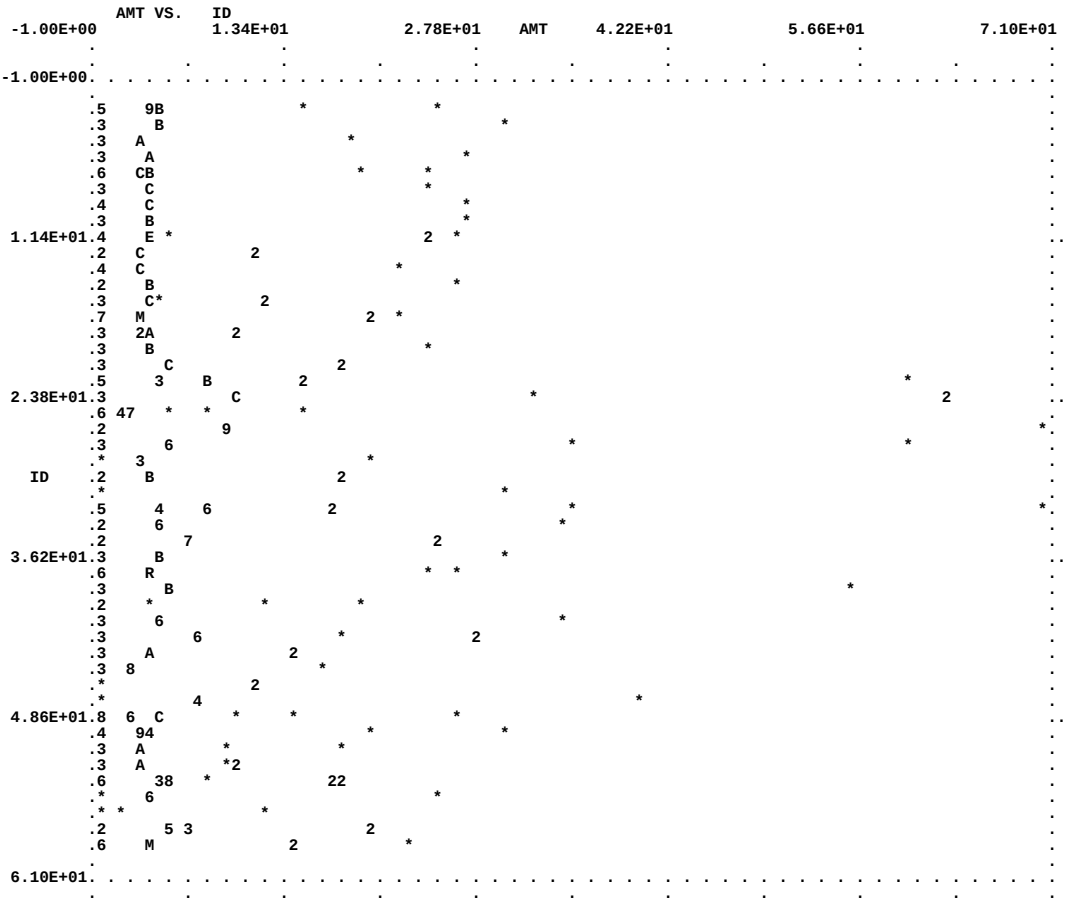

Figure 11.4. See figure 11.3; the outlier has been corrected.

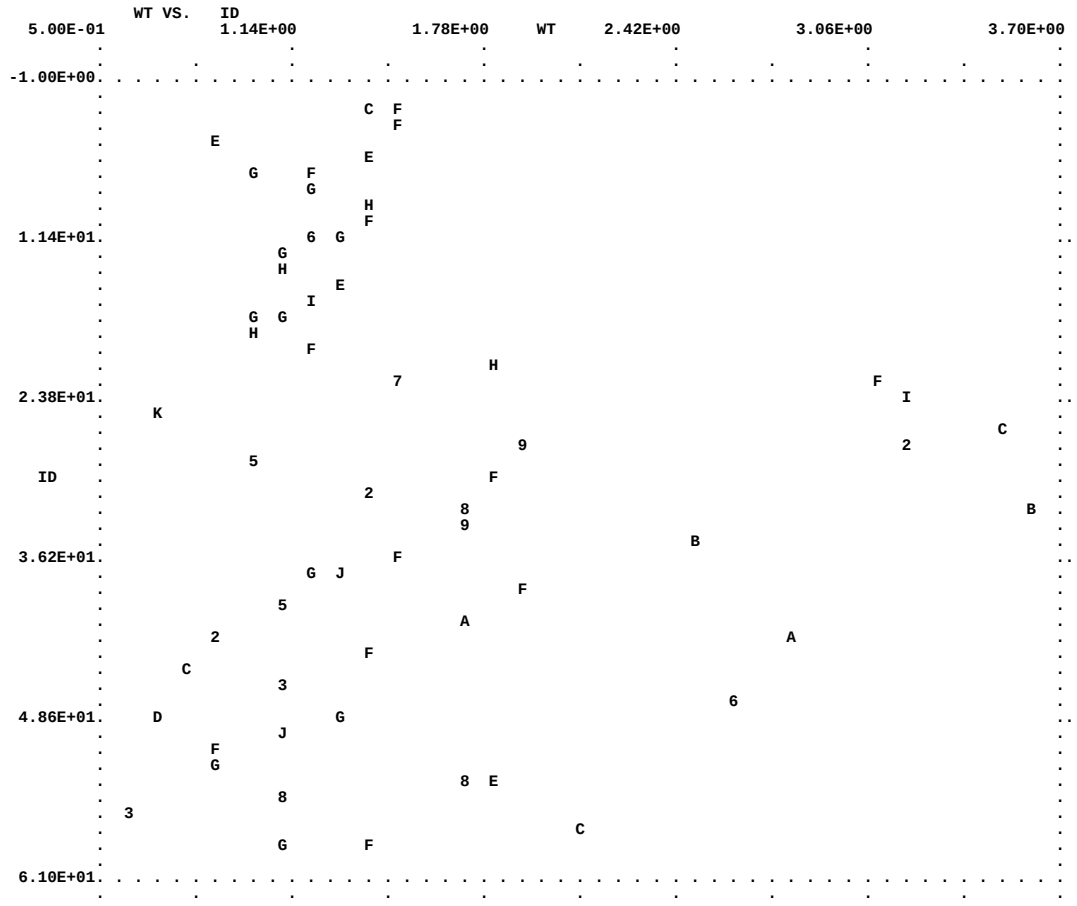

Figure 11.5. Index plot for weight (WT)

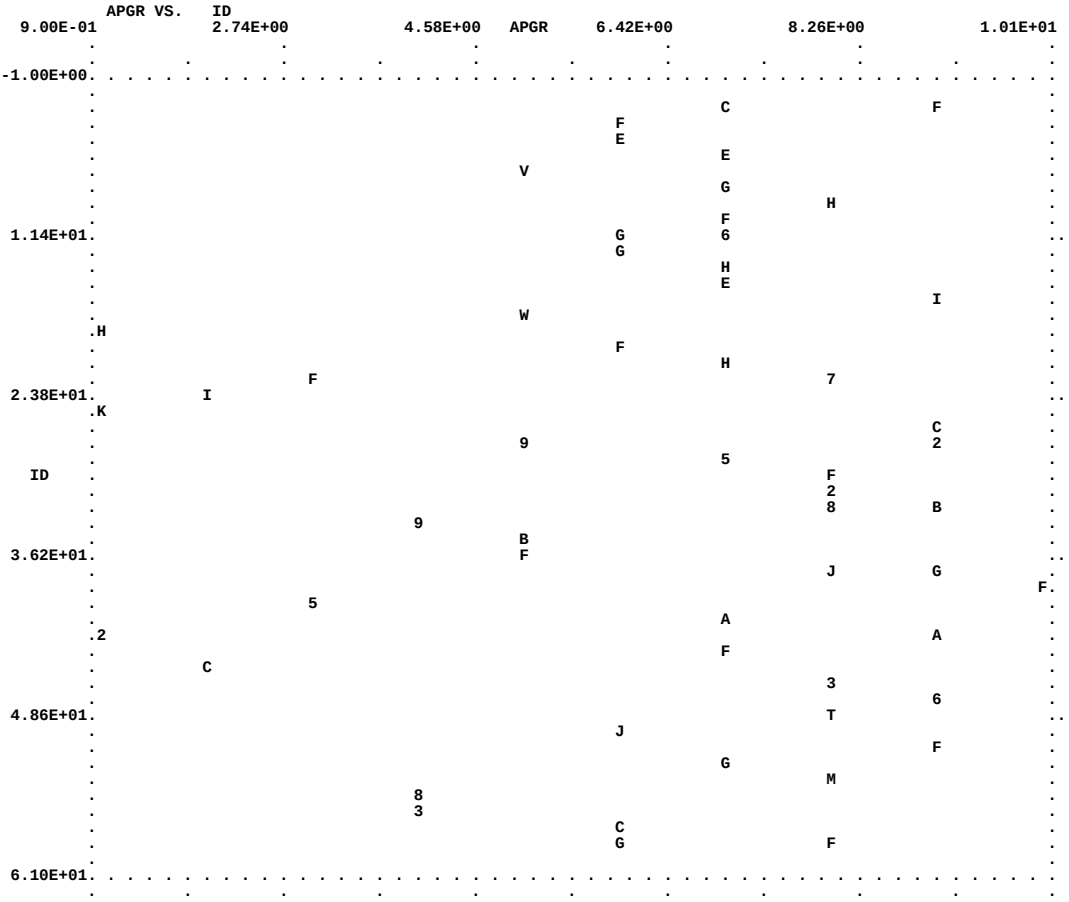

Figure 11.6. Index plot for Apgar score (APGR)

These plots will be useful in the next stage of model building.

4. Building the Structural Part of the Model

One must first consider the choice of the structural kinetic model. For the phenobarbital data, a monoexponential kinetic model has been chosen. Presumably, the basic structural kinetics are already known well enough for this well studied drug, and it is not necessary to explore the fits of other possible structural kinetic models to the data at hand. Rather, in this section we focus on the newer task to most users of NONMEM, the task of building the structural part of the model for the PK parameters.

4.1. A General Approach

It is generally advisable to start from the simplest reasonable model, and proceed toward greater complexity, stopping whenever further additions fail to improve the model fit. Thus, one needs several types of tools: (i) those to choose a "minimal" model, (ii) those to indicate what part of a current model needs to be altered or elaborated (called model diagnosis or model criticism), and (iii) those to judge whether an alteration or elaboration has led to an improved model.

With such tools, one proceeds step by step from the minimal model, running NONMEM and using the diagnostic tools at each step to suggest a single addition for the next step. The process will terminate when the judgement tools indicate no improvement by any of

the additions suggested by the diagnostic tools, or when the diagnostic tools fail to suggest any more additions.

The NONMEM runs at this stage, since there will be many of them, should be made as short as possible. To do so, only the estimation, table and scatterplot features need be used; the Covariance Step need not, in general, be run.

4.2. The Minimal Model

As suggested above, the minimal model involves the simplest pharmacokinetic model (ADVAN) likely to fit the data, and the simplest possible structural PK parameter model: each parameter is simply identified with a separate element of θ.

At this stage, the statistical model should also be very simple. Only one, or at most two η variables should be defined. These will usually affect (first) the scale parameter (which itself, is often a volume of distribution parameter) and (second) some other parameter influencing the overall kinetics. Since the overall kinetics exhibited in the data will usually be dominated by elimination, the second η should usually modify the rate constant of elimination or clearance. However, some (kinetic) data sets are dominated by absorption or distribution, and in such cases, the second η should probably modify the parameter most affecting these processes. A single ε should usually suffice. Both inter- and intra-individual errors can conveniently be modeled as proportional, so that the determination of initial estimates of variances is made easier, and all such estimates are on the same scale, but this is a matter of taste. The model for the phenobarbital data defined in Chapter 2 (figure 2.6) almost fulfills the spirit of these restrictions. However, the inter- and intra-individual error models there are additive, rather than proportional. The minimal model used on the phenobarbital data in this chapter is therefore a modified version of that used in Chapter 2. It is:

$$CL = \theta_1(1 + \eta_1) \tag{11.1a}$$

$$V = \theta_2(1 + \eta_2) \tag{11.1b}$$

$$y = F(1 + \varepsilon_1) \tag{11.1c}$$

In (11.1), it is understood that $S = V$, and that F is the prediction of y from ADVAN1 using CL and V. A control file to NM-TRAN that specifies this model, and instructs NONMEM to produce the desired output is:

```
$PROBLEM PHENOBARB  SIMPLE MODEL (#1)
$INPUT   ID TIME AMT WT APGR DV
$DATA    PHENO
$SUBROUTINE   ADVAN1
$PK
     TVCL=THETA(1)
     CL=TVCL*(1+ETA(1))
     TVVD=THETA(2)
     V=TVVD*(1+ETA(2))
     K=CL/V
     S1=V
$ERROR
     Y=F*(1+ERR(1))
$THETAS  (0,.0105)  (0,1.05)
$OMEGAS  .25   .25
$SIGMAS  .04
$ESTIMATION
$TABLE          ID TIME AMT WT APGR
$SCATTERPLOT    PRED VS DV UNIT
$SCATTERPLOT    RES  VS (PRED,WT,APGR)
$SCATTERPLOT    WRES VS (PRED,WT,APGR)
```

4.3. Use of Constraints

It is important to realize that constraints on elements of θ or Ω may be part of a model.

For example, constraining clearance to be positive is a modelling choice. One might implement this constraint in NONMEM using a lower bound on the $THETA record, and this would assure that the estimate of clearance will be positive. It may not be necessary to do this; even without the lower bound, the data might clearly force the estimate to be positive.

Often, however, analysts will constrain the range of a parameter in the belief that doing so will shorten computing time or stabilize the search for the minimum of the objective function. While this benefit may be gained, the data may force the parameter estimate to the constraint boundary even though this boundary may not, in fact, represent a true modeling choice. In this case the proper action is to relax the constraint and rerun the problem. To do otherwise, and leave the parameter estimate to be the boundary value, implies that at the outset the user assumes that the parameter must be within the boundary and elevates the constraint to the status of a modeling choice. If an estimate lies on a boundary, NONMEM will print a warning message (along with the standard message regarding the status of the termination of the Estimation Step). The reader is cautioned to look for such a message, and in general, it is a good idea to check the values of the final estimates against the boundary values. Alternatively, the implementation of constraints that are not intended to represent modeling choices might be used cautiously and only if they really seem necessary to stabilize a search.

4.4. Diagnostic Tools

4.4.1. Plot of DV vs PRED

Most useful diagnostic tools are graphical. For an overall sense of the fit, a useful diagnostic plot is DV vs PRED. When there are substantial and systematic deviations from the line of identity, this plot suggests that there are problems with the fit, but it does not suggest what exactly these problems might be or what to do about them. This plot for the fit of the phenobarbital data to model (11.1) is seen in figure 11.7.

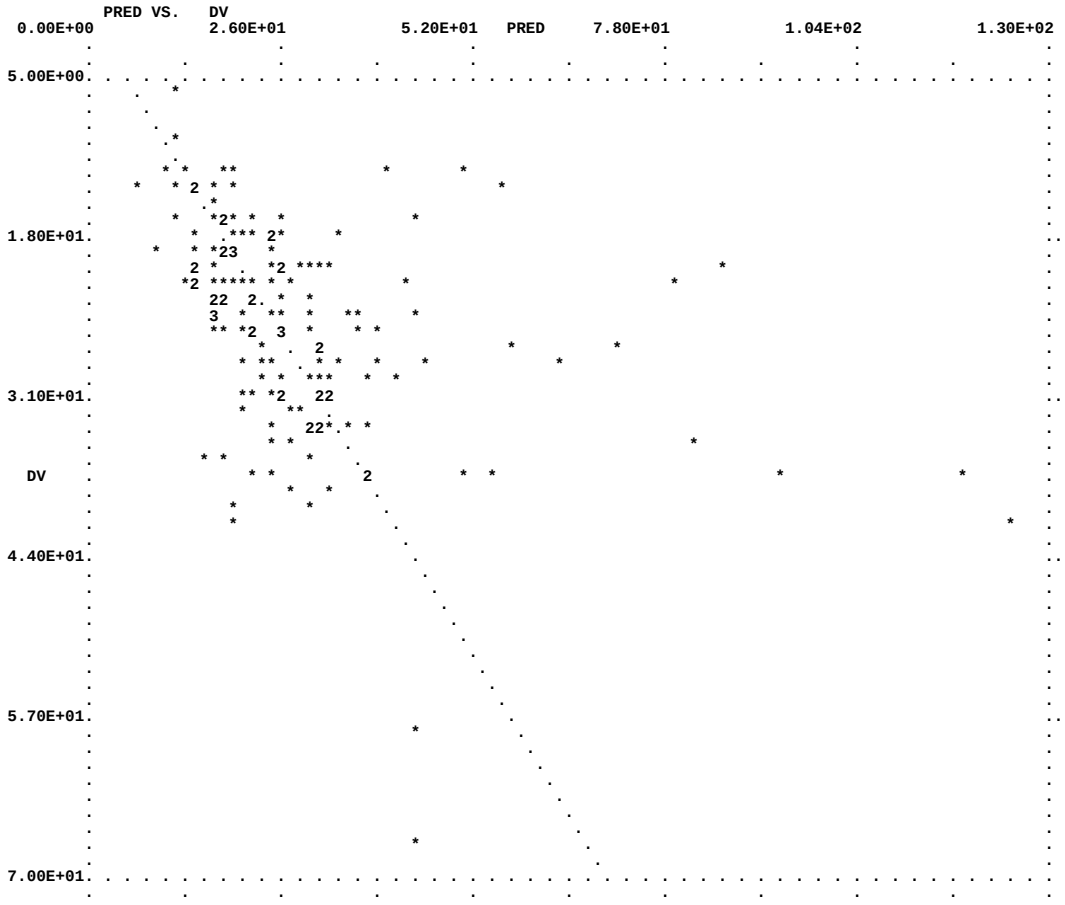

Figure 11.7. Predictions from fit of model (11.1) to phenobarbital data vs observations themselves. The line of identity (...) shows where the points should, ideally, fall.

Figure 11.7 reveals that there is a group of points where the observation is much lower than the prediction. To begin to determine why this is so, it will be useful to look at residual plots. Such plots are the basis of the most important of the diagnostic tools.

4.4.2. Residual Plots

As mentioned in Chapter 2, a residual is the difference between an observation and its prediction. The prediction in this case (the same prediction as denoted by PRED) is the population prediction, i.e. the prediction for the typical individual having the given values for all the concomitant variables.

With population data, weighted residuals are often more informative than (plain) residuals. The weighted residuals for an individual are formed by transforming the individual's residuals so that under the population model, and assuming the true values of the population parameters are given by the estimates of those parameters, all weighted residuals

have unit variance and are uncorrelated. Weighted residuals are more informative for several reasons. First, since they have unit variance, or what is the same, unit standard deviation, "large" weighted residuals are those with absolute values greater than 3 or so. Second, loosely speaking, although plain residuals remove the structural model from the data, allowing one to see what part of the data is not (yet) modeled, they do not remove the statistical model (formally, they are still correlated). Weighted residuals have both models 'removed' so any pattern in these is definitely not accounted for by the current model. This provides a more secure basis for future model building choices.

4.4.2.1. Index Plots of Residuals

Figure 11.8 is an index plot of residuals, which is a useful plot when combined with index plots of other data items. One can look for an association between unusual residuals and values of another data item. E.g. Are the largest discrepancies between model and data associated with certain (possibly extreme) values of the data item?

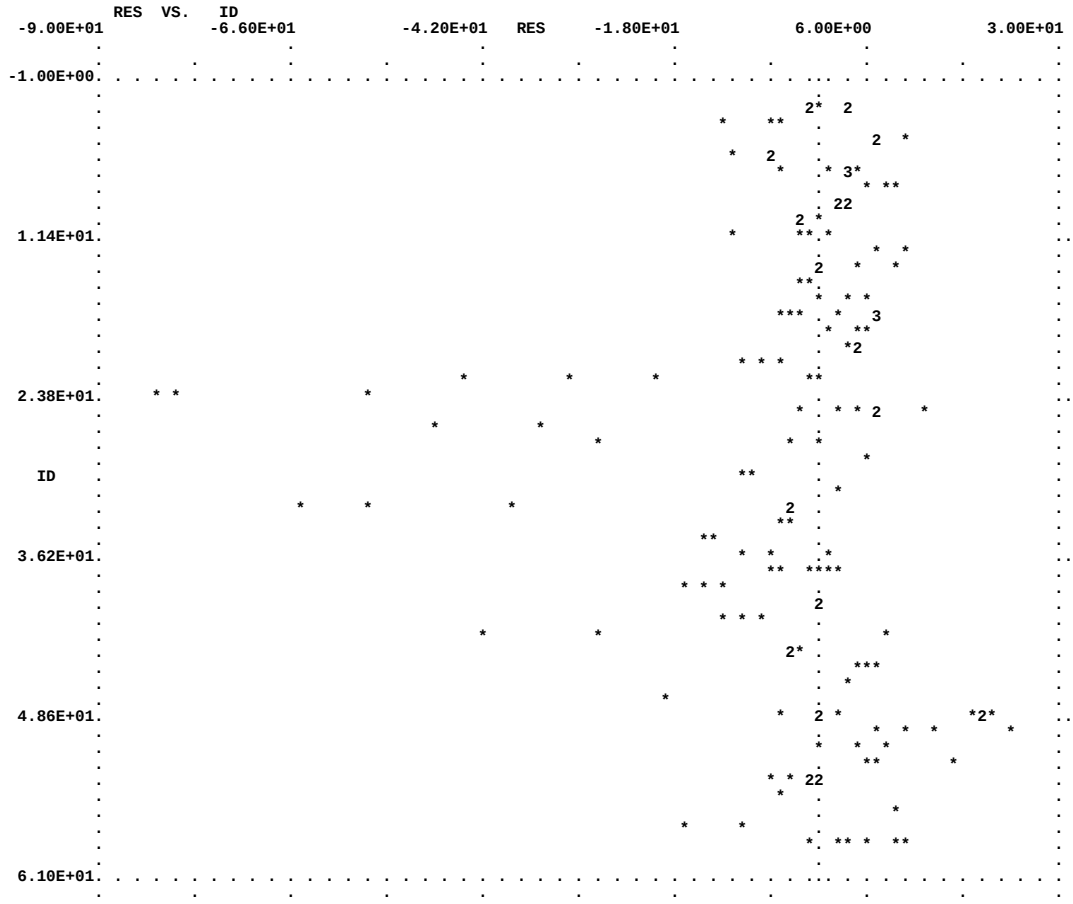

Figure 11.8. Index plot of residuals from fit of model (11.1) to phenobarbital data.

In the phenobarbital example, this is clearly so: The large negative residuals (i.e., predictions greater than observations), first noted in figure 11.7, are here seen to be associated with patients 22 to 32 or thereabouts. In figures 11.4 and 11.5 it is clear that these same patients are those who received the highest doses and who weigh the most. An obvious explanation, then, of the over-predictions is that they are in the patients who weigh the most, and because weight is not in the model, neither volume nor clearance is adjusted to

be larger in such individuals, so that predictions are strictly proportional to dose alone and may be too large for these heavier patients.

4.4.2.2. Plot of WRES vs Independent Variable

Another way to see the association between weight (or dose) and the large residuals is to plot the residual against weight, say. Figure 11.9 is this plot, but where, for reasons already discussed, weighted residuals, rather than plain residuals, are used.

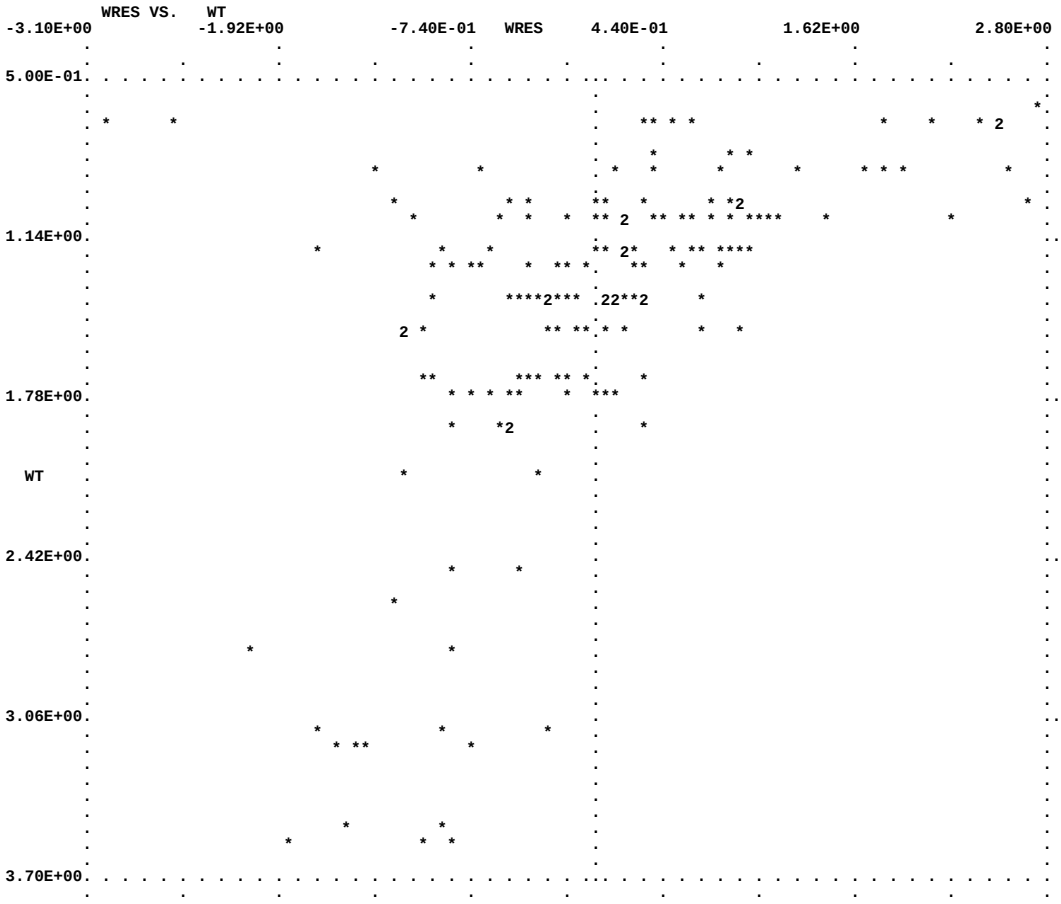

Figure 11.9. Plot of weighted residuals vs weight for fit of minimal model to phenobarbital data.

It is clear from figure 11.9, in a way that is particularly compelling, that it is precisely those individuals with the largest weight whose residuals are large and negative. This type of residual plot, where (weighted) residuals are plotted against some independent variable, is the single most useful diagnostic tool.

Systematic patterns of weighted residuals, then, suggest possible model improvements. For an independent variable that already appears in the model, such a pattern may suggest that the way in which it enters the model is incorrect; e.g., it might appear as having a linear influence on a PK parameter, and a curvilinear influence might be better, or it might affect additional PK parameters, beyond those it affects in the current model. An example of this will be seen shortly. For a variable that does not yet appear in the model, as in figure 11.9, such a pattern suggests that the element should appear.

Before examining what happens if patient weight is added to the model, a caution about residual plots is in order. Neither residuals, nor weighted residuals, should ever be

plotted against the observations themselves. Such a plot will always show a correlation, spuriously suggesting a problem with the model. This is most easily appreciated by considering the simple model that gives rise to the constant-valued prediction given by the the mean of the observations. All positive residuals (observations greater than prediction) must then be associated with observations greater than the mean, while all negative residuals must be associated with observations less than the mean. Clearly, then, the residuals plotted against the observations must show a line with positive slope. This same type of association, although to a lesser degree, holds true, even in less extreme cases. The phenomenon is illustrated in figure 11.10.

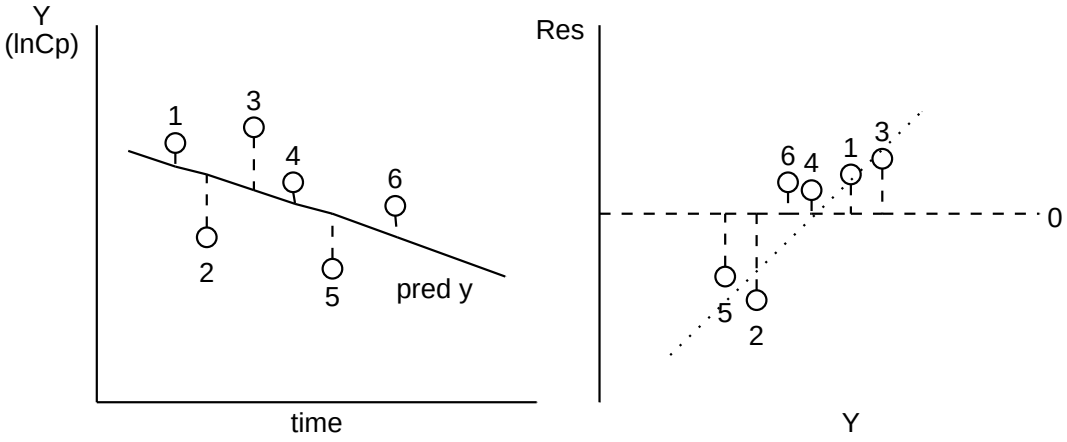

Figure 11.10. Residuals always correlate with the observations themselves; the more so, the less the model explains the data!

4.5. Judging Goodness of Fit

A more complex model is acceptable only if the complexity can be justified by some significant improvement in the fit. To evaluate whether this has been accomplished, several measures should be examined; no single measure suffices.

In the phenobarbital example, based on the finding in figure 11.9, a modified model is suggested. This model, (11.2), has (11.2b)=(11.1b), and (11.2c)=(11.1c), but

$$Cl = (\theta_1 + \theta_3 WT)(1 + \eta_1) \qquad (11.2a)$$

which is a full model relative to the reduced model (11.1a), whence (11.2) is a full model relative to the reduced model (11.1).

The model-defining statements to NM-TRAN ($PK and $ERROR) now become:

```
$PK
      TVCL=THETA(1)+THETA(3)*WT
      CL=TVCL*(1+ETA(1))
      TVVD=THETA(2)
      V=TVVD*(1+ETA(2))
      K=CL/V
      S1=V
$ERROR
      Y=F*(1+ERR(1))
```

We now examine some measures of goodness of fit, and see how (11.2) fares relative to (11.1).

4.5.1. A Global Measure — Change in the Objective Function

A global measure of goodness of fit is, of course, the objective function value based on the final parameter estimates, which, in the case of NONMEM, is minus twice the log likelihood of the data (see Chapter 5, Section 5.2.3). As noted in Chapter 5, if the new model differs from the previous model only by the addition of some new variable(s) (so that the two models form a full/reduced model pair), then the difference in objective function values has a known (approximate) statistical distribution. More informally, during model-building, a fall in objective function value of 4 when a single new parameter is introduced (and no old ones are eliminated) indicates that the new model has substantially improved the overall goodness of fit. Model (11.2) shows a decrease in objective function of 35.2 relative to (11.1), indicating considerable improvement.

4.5.2. Decrease in Unexplained Variability

The purpose of adding independent variables to the model is usually to explain kinetic differences among individuals. This means that prior to adding a variable, such differences were not "explained" by the model, and hence were part of random interindividual variability, although these differences could also have been reflected as a part of random intraindividual variability. Accordingly, elaboration of the model should be accompanied by a decrease in the estimates of the variances in Ω and/or Σ.

The estimates of ω^2_{CL}, ω^2_V, and σ^2 from the fit to Model (11.2) are .057, .12, and .0196, corresponding to coefficients of variation of 24%, 35%, and 14%, respectively. The corresponding values from the fit to (11.1) are .25 (CV=50%), .14 (CV=37%), and .016 (CV=13%), so that a considerable reduction in the variance of clearance is seen.

4.5.3. Improvement in Plots

The last, and most useful, evidence confirming the value of elaborating a model is to find that the pattern(s) in the PRED vs DV and weighted residual plot(s) that suggested the need for the addition have now disappeared. Indeed, when a model is relatively complete, all weighted residual plots should show no pattern: the "unexplained" part of the data should have become featureless random noise.

Figures 11.11 and 11.12 correspond to 11.7 and 11.9, but are from the fit to model (11.2).

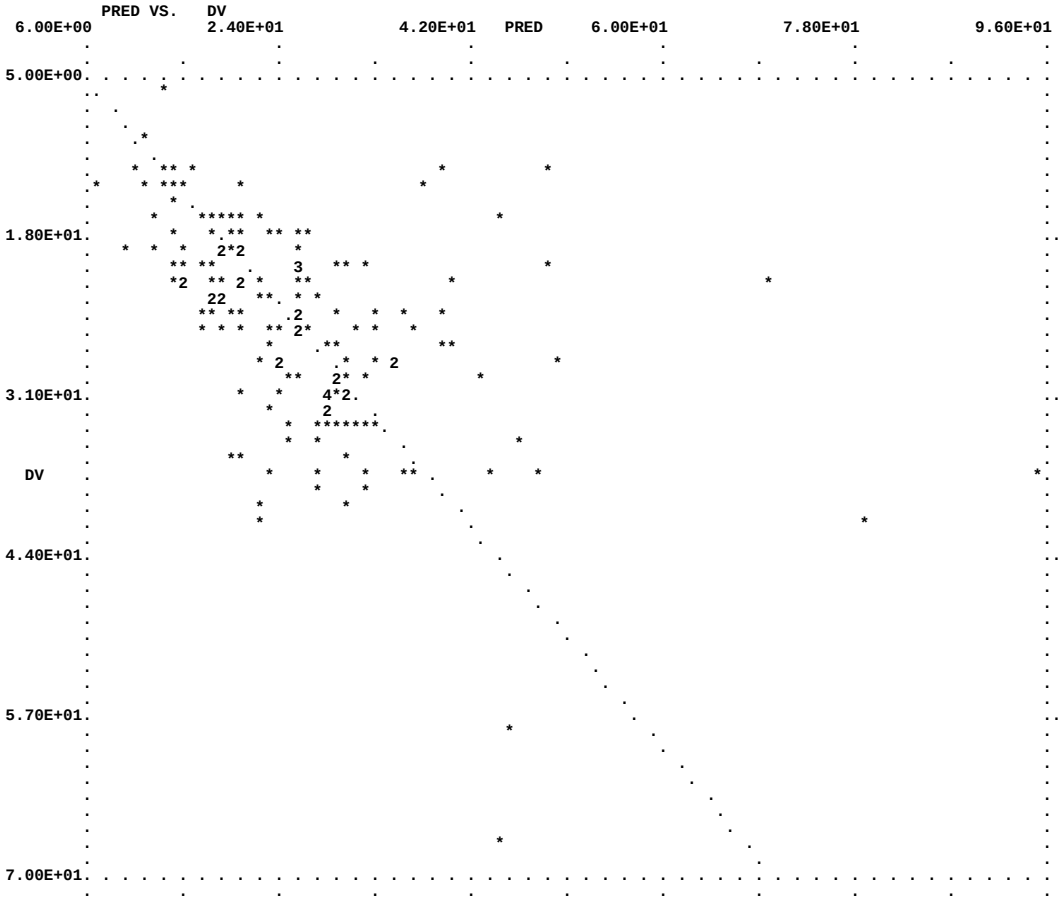

Figure 11.11. Predictions from fit of model (11.2) to phenobarbital data vs observations.

Compared to figures 11.7 and 11.9, figures 11.11 and 11.12 indicate an improvement in that the *number* of large negative residuals is clearly reduced in both plots.

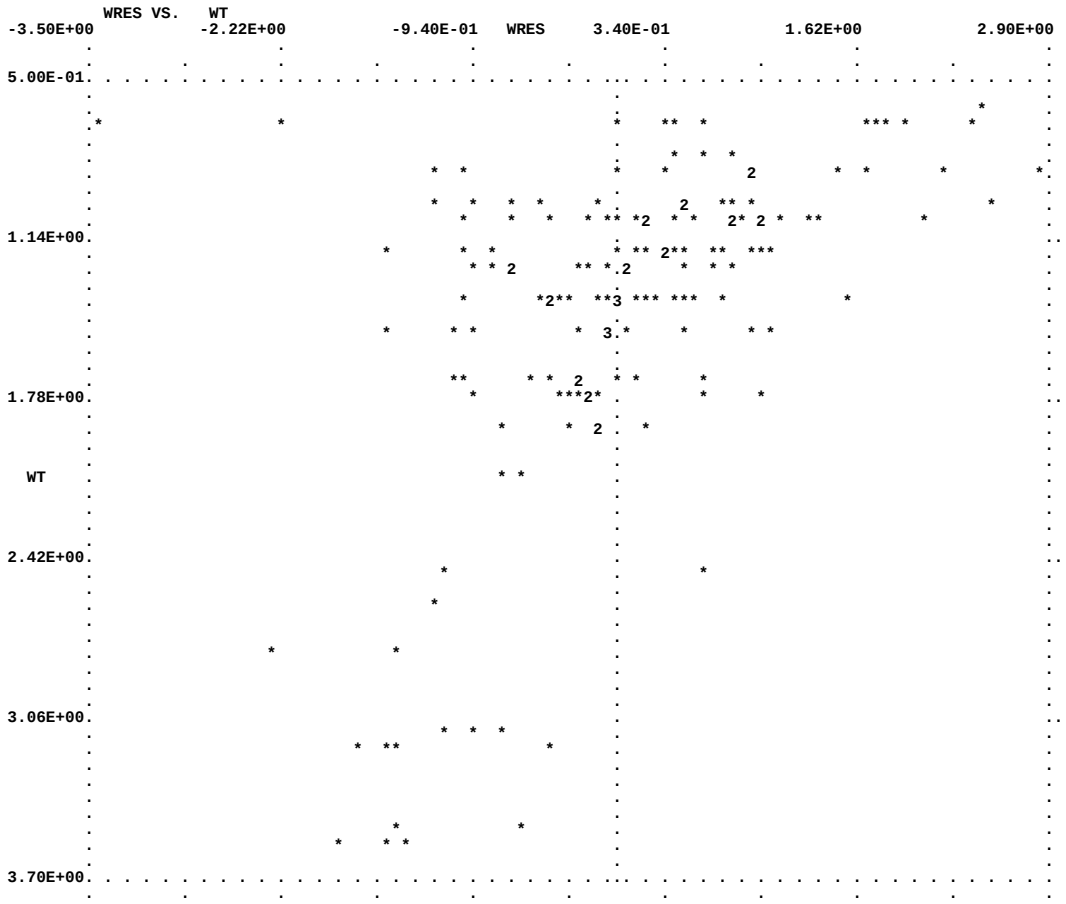

Figure 11.12. Plot of weighted residuals vs weight for fit of model (11.2) to phenobarbital data.

4.6. Using the Tools: Further Improvement

4.6.1. An Additional Effect of WT

While all of the above suggests that model (11.2) is superior to model (11.1), figure 11.12 shows a persistent linear relationship between weight and residuals. Thus, the addition of weight to the model for clearance does not fully exploit the information in the variable, weight. An obvious modification to model (11.2) that might deal with this is to have weight affect V as well as CL. Accordingly, define model (11.3) such that (11.3a)=(11.2a), (11.3c)=(11.2c), but

$$V = (\theta_2 + \theta_4 WT)(1 + \eta_2) \qquad (11.3b)$$

The model-defining portion of the control stream now becomes:

```
$PK
      TVCL=THETA(1)+THETA(3)*WT
      CL=TVCL*(1+ETA(1))
      TVVD=THETA(2)+THETA(4)*WT
      V=TVVD*(1+ETA(2))
      K=CL/V
      S1=V
$ERROR
      Y=F*(1+ERR(1))
```

When model (11.3) is fit to the data, the objective function decreases fully 126 relative to model (11.2). Moreover, the estimates of ω^2_{CL}, ω^2_V, and σ^2 are now .050 (CV=22%), .028 (CV=17%), and .011 (CV=10%), indicating a further substantial decrease in unexplained variation. The plots corresponding to 11.7/11.11 and 11.9/11.12 are shown as figures 11.13 and 11.14.

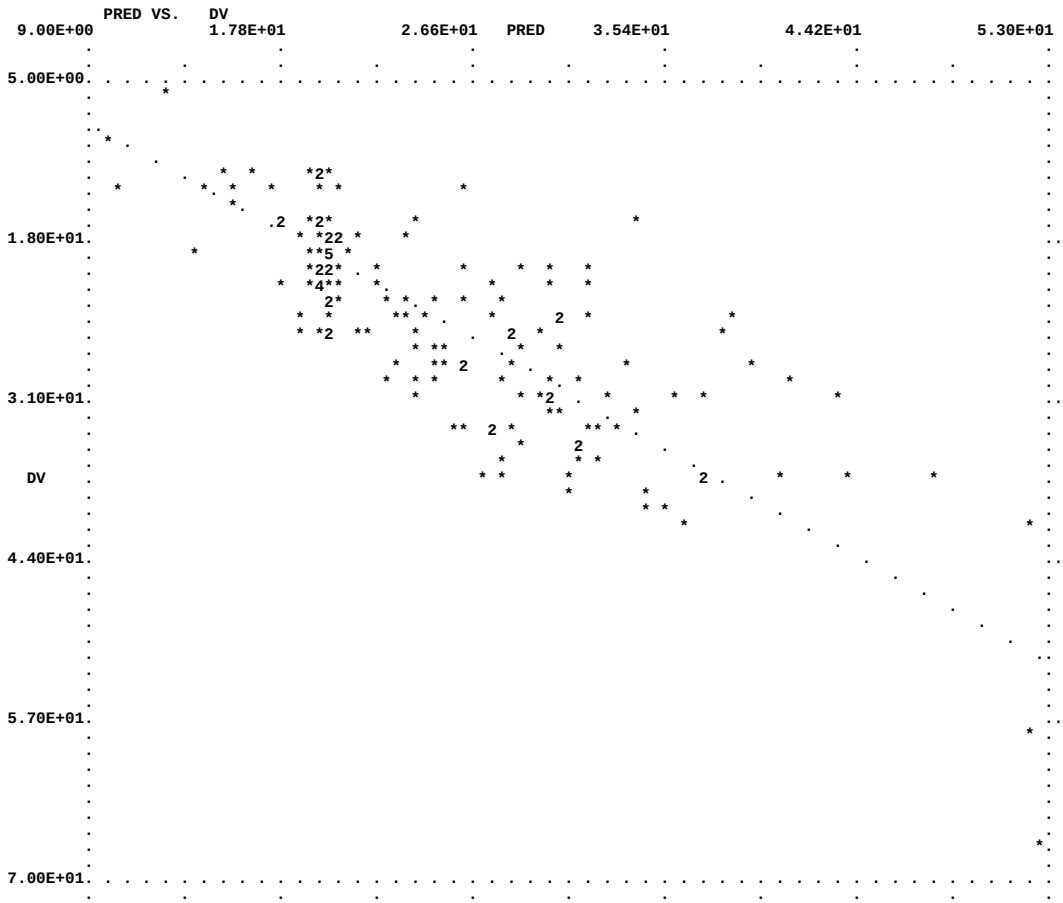

Figure 11.13. Predictions from fit of model (11.3) to phenobarbital data vs observations.

Now there are no obvious discrepancies, and the plot of weighted residuals vs *WT* shows no pattern, so that it is likely that no further use of weight in the model is required.

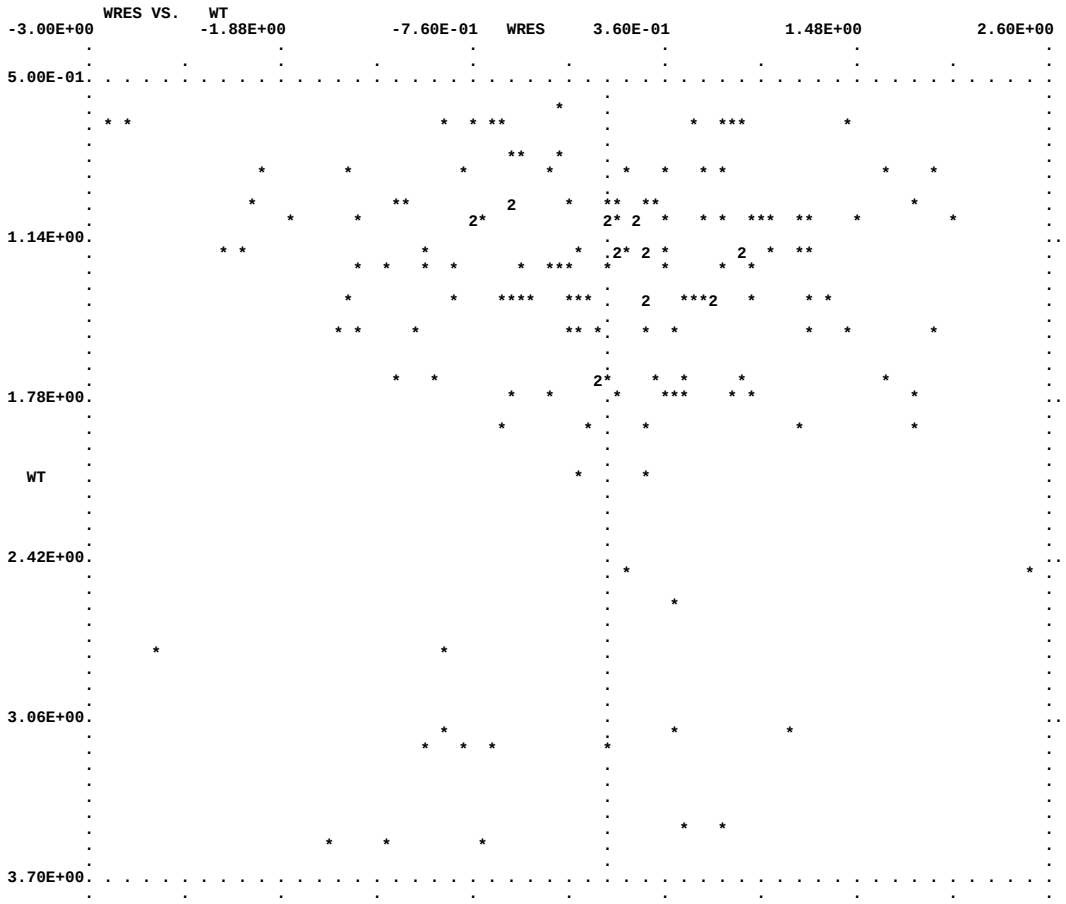

Figure 11.14. Plot of weighted residuals vs weight for fit of model (11.3) to phenobarbital data.

4.6.2. The Effect of APGR

The structural model building stage is not over until all available independent variables have been examined for influence, and there is one additional variable, the Apgar score, that has not yet been seriously considered. A plot of the weighted residuals from the fit to model (11.3) vs *APGR* is shown in figure 11.15.

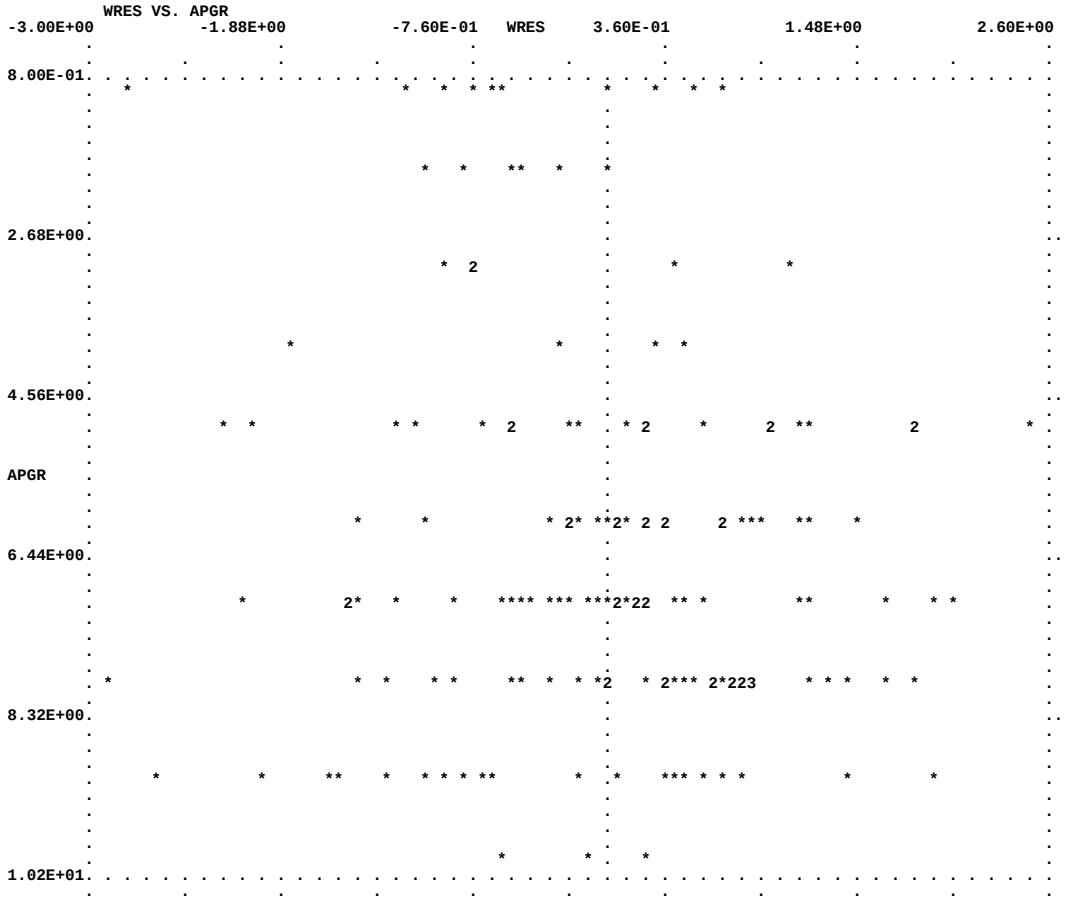

Figure 11.15. Plot of weighted residuals vs Apgar score for fit of model (11.3) to phenobarbital data.

There is a weak suggestion from figure 11.15 that for *APGR* less than 3, the weighted residuals tend to be negative. Accordingly, a new model (11.4) can be proposed, which is identical to (11.3) except that

$$V = \left\{ \begin{array}{ll} \theta_2 + \theta_4 WT, & \text{if APGR} > 2 \\ (\theta_2 + \theta_4 WT)\theta_5, & \text{if APGR} \le 2 \end{array} \right\}(1 + \eta_2) \qquad (11.4b)$$

The relevant statements for NM-TRAN now become:

```
$PK
      TVCL=THETA(1)+THETA(3)*WT
      CL=TVCL*(1+ETA(1))
      TVVD=THETA(2)+THETA(4)*WT
      IF (APGR.LE.2) TVVD=TVVD*THETA(5)
      V=TVVD*(1+ETA(2))
      K=CL/V
      S1=V
$ERROR
      Y=F*(1+ERR(1))
```

When this model is fit to the data, θ_5 is estimated to be 1.18, implying that indeed, the volumes of distribution for infants with Apgar scores less than 3 are typically 18% higher than those of infants (of the same weight) with higher Apgar scores. The measures of improvement are now marginal, however: the objective function decreases only 3.7, and the decreases in the variances are all less than 10% of their previous values, with the variance of ε actually increasing a few percent. Inspection of figure 11.6 suggests a reason for this: note that only 5 distinct individuals (separate symbols) have Apgar scores less than 3. There is simply not very much information about babies with low Apgar scores in this data set.

For completeness, figure 11.16 corresponds to figure 11.5, but using model (11.4), and now shows no distinct pattern.

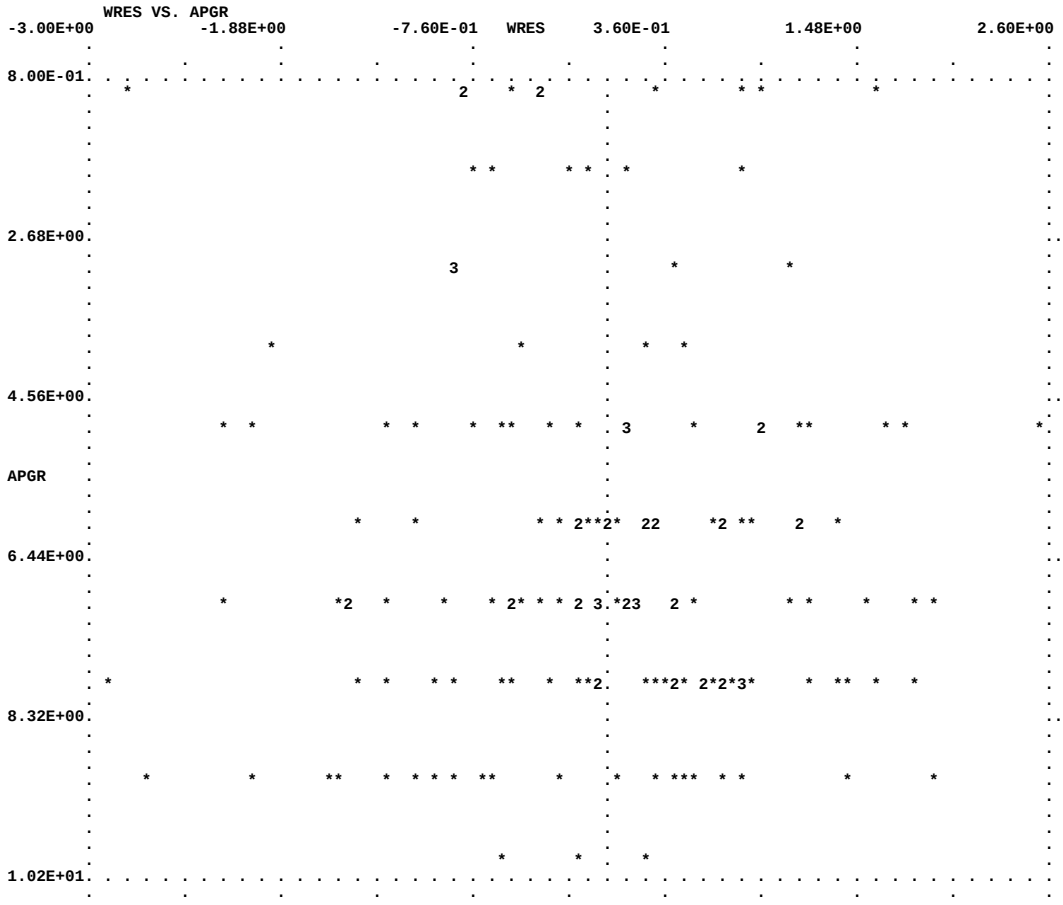

Figure 11.16. Plot of weighted residuals vs Apgar score for fit of model (11.4) to phenobarbital data.

The stage of building the structural model is now complete.

5. Building the Statistical Model

5.1. Judging Among Alternatives

NONMEM can provide estimates of the η variables for each individual (see Chapter 12, Sections 4.11-4.13). Plots of the estimated interindividual differences, which can be regarded as interindividual residuals, can be obtained. Plots of these residuals (associated with a particular PK parameter) versus the values of an independent variable provide

further help in building the structural part of the parameter model. Moreover, a plot of these residuals versus the typical values of the parameter (whose values depend on covariates) also provides help in building the model for interindividual differences themselves. For example, if interindividual differences are modeled with the additive model, but the plot shows a linear trend (in the boundaries enveloping the residuals), this suggests that a proportional model be tried.

Lastly, help can be provided in the selection of a model for intraindividual error. Predictions of concentrations, and hence residuals, based on estimates of individuals' ηs, can be computed, and the residuals can be plotted versus the predictions. (This requires advanced techniques.) Again, if intraindividual errors are modeled with the additive model, but the plot shows a linear trend in the envelope, this suggests that a proportional model be tried.

The statistical model is usually of less interest than the structural model, so that frequently all that is sought is an adequate model, not necessarily the correct one, nor does one care whether the estimates of the random effects parameters (the elements of Ω and Σ) are particularly precise. Sometimes, however, the variability in the random effects is of genuine primary interest. In such cases more attention must be paid to building the random effects model. This, however, may not be easy because it is an unfortunate, but unavoidable fact, that a great deal more data is needed to estimate random effects parameters with a given precision than is needed to estimate fixed effect parameters with comparable precision.

The tools used to elaborate the statistical model are similar to those used for elaborating the structural model: alternative models are assessed using (available) residual plots, especially ones like those just discussed, and relative changes in the objective function.

5.1.1. Unexplained Variability

When the statistical model is developed, a new η variable may be added, or an old η variable used differently. Then differences in the ωs between models cannot really be used to judge the benefit of the addition, and this evaluation tool becomes less useful. On the other hand, an addition of an η might be confirmed by a reduction in the estimates of the variances in Σ, the variances of the random components in the model for residual error. However, there is one sure sign that too many η's are in the model: NONMEM may estimate one or more of the ωs to be zero, or very nearly zero. This can be disconcerting, particularly if the η variable is the only such variable affecting Volume, for instance, as then this estimate seems to suggest that with respect to Volume, there is no interindividual variability in the population whatsoever! The result should not be interpreted this way, however. Rather, assuming the η affecting Volume is the one most recently added, it indicates that given the previous statistical model, no *additional* variability *needs* to be ascribed to volume to explain all the variability seen. The data cannot support such an elaborate statistical model, and a simpler model, such as the previous one, must be used.

5.1.2. Residual Plots

The most important residual plot is now a plot of the weighted residuals against predictions, where a pattern in the shape of the outer envelope of points can indicate deficiencies in the statistical model (recall that a distinct pattern in the local "average" residual vs the predictions would indicate a defect in the structural model). This can be illustrated using the phenobarbital data. Figure 11.17 shows the plot of weighted residuals vs predictions for model (11.4), and figure 11.18 shows the same plot for a modified model, (11.5), identical to (11.4) except for

$$y = F + \varepsilon_1 \qquad (11.5c)$$

Only the $ERROR statements of the control stream change:

```
$ERROR
      Y=F+ERR(1)
```

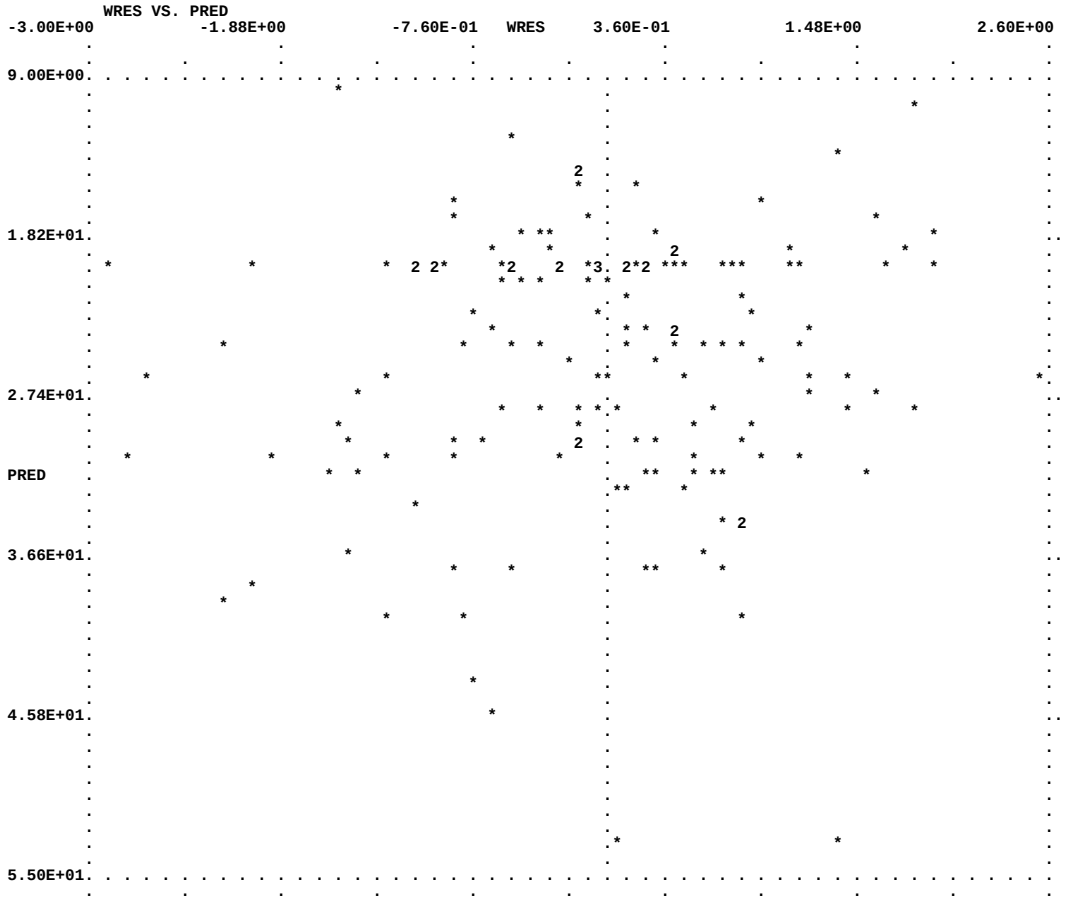

Figure 11.17. Plot of weighted residuals vs predictions for fit of model (11.4) to phenobarbital data (proportional intraindividual error).

Although the plots do not differ greatly, there is a small suggestion in figure 11.18 that the envelope of weighted residuals is somewhat V-shaped with the apex of the V at PRED=0 (but which does not show on the plot), while in figure 11.17 the weighted residuals seem more homogeneous, and their magnitude seems less dependent on that of the predictions. That this impression is valid is suggested also by the increase in objective function of 7.6 in going from model (11.4) to (11.5).

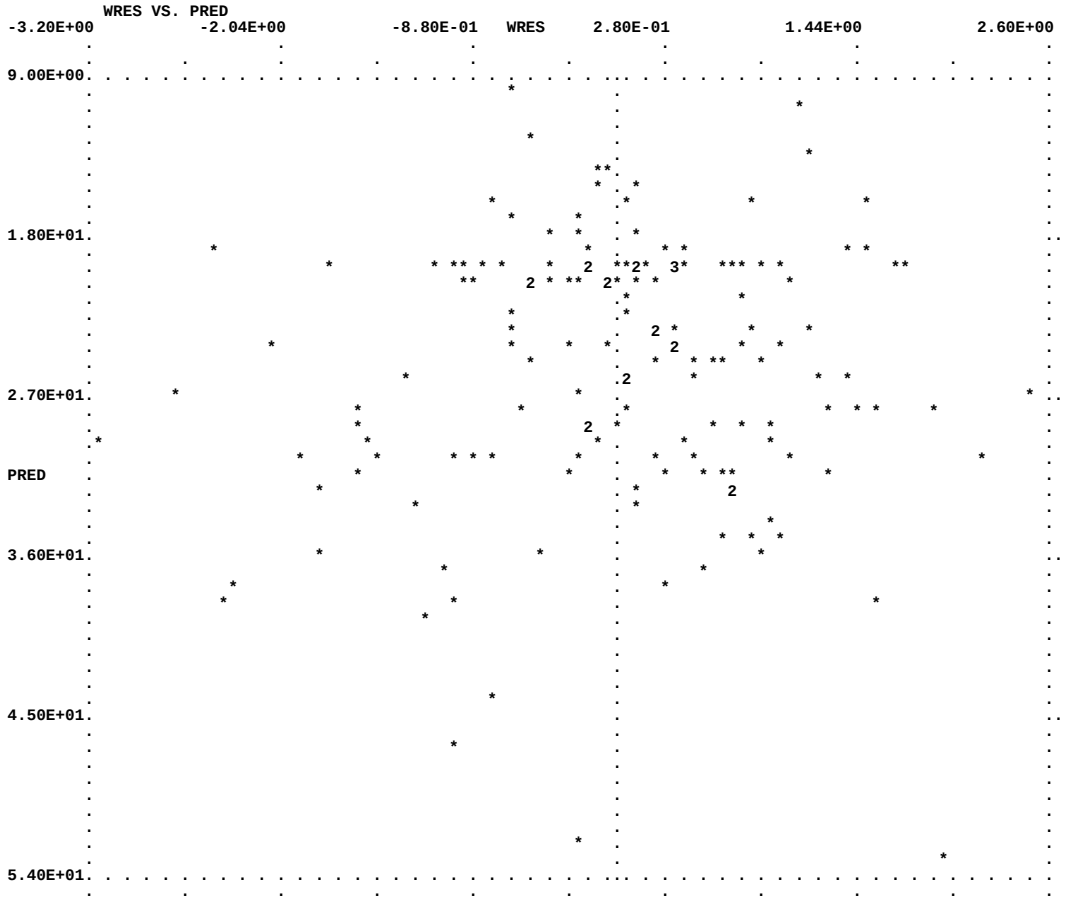

Figure 11.18. Plot of weighted residuals vs predictions for fit of model (11.5) to phenobarbital data (additive intraindividual error).

6. Refine Model

The goal of this stage is to check whether the model is as parsimonious as reasonable, since if it is not, certain important parameters may not be estimated with as good precision as can be achieved. Although up to this stage, one tries to avoid adding parts to the model which are not well supported by the data, it is nonetheless possible that a part added at one stage may seem unnecessary after adding another part at a later stage. Perhaps, for example, weight affects V, and V and CL are correlated in the population (independent of weight), but first the influence of weight on CL is examined, and later its influence on V is examined (this was the order illustrated above). Then initially, weight might appear to influence CL, although this influence might only derive from the correlation between the two PK parameters. Later, after the influence of weight on V is a part of the model, the influence of weight on CL might disappear. One wants to check this possibility and, if indicated, eliminate the influence of weight on CL from the model at the stage now being described. The basic technique at this stage, is to run the Covariance Step with the best model thus far, and look for parameters with confidence intervals that include the parameter's null value, i.e., the value that causes the parameter to be effectively deleted from the model. A null value is usually zero (for a parameter quantifying an additive portion of the model), and sometimes unity (for a parameter quantifying a multiplicative part of the model). If such parameters are found, then one at a time, each

can be set to its null value and the consequences examined as discussed above for the earlier model building stages.

Figures 11.19 and 11.20 show two pages of NONMEM output from a run fitting model (11.4) to the phenobarbital data, and implementing the Covariance Step. Figure 11.19 shows the final parameter estimates, and 11.20, their standard errors.

```
********************************************************************************************************************************
********************                                                                                        ********************
********************                                  FINAL PARAMETER ESTIMATE                              ********************
********************                                                                                        ********************
********************************************************************************************************************************


 THETA - VECTOR OF FIXED EFFECTS  **********************

            TH 1      TH 2      TH 3      TH 4      TH 5
         7.50E-05  2.66E-02  4.62E-03  9.53E-01  1.18E+00


 OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********

            ETA1      ETA2
 ETA1     4.59E-02
 ETA2     0.00E+00  2.63E-02


 SIGMA - COV MATRIX FOR RANDOM EFFECTS - EPSILONS  ****

            EPS1
 EPS1     1.10E-02
```

Figure 11.19. Parameter estimates from fit of model (11.4) to phenobarbital data.

```
********************************************************************************************************************************
********************                                                                                        ********************
********************                              STANDARD ERROR OF ESTIMATE                                ********************
********************                                                                                        ********************
********************************************************************************************************************************


 THETA - VECTOR OF FIXED EFFECTS  **********************

            TH 1      TH 2      TH 3      TH 4      TH 5
         9.59E-04  9.24E-02  7.33E-04  7.48E-02  8.36E-02


 OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********

            ETA1      ETA2
 ETA1     2.25E-02
 ETA2      .........  7.08E-03


 SIGMA - COV MATRIX FOR RANDOM EFFECTS - EPSILONS  ****

            EPS1
 EPS1     2.01E-03
```

Figure 11.20. Standard errors of parameter estimates from fit of model (11.4) to phenobarbital data.

6.1. Use of Standard Errors and Confidence Intervals

The null values of θ_1 through θ_4 are zero, while the null value of θ_5 is unity. Using the numbers from the figures, it is easily seen that for θ_1 and θ_2, the parameter estimate minus the null value is a fraction of one standard error, and hence θ_1 and θ_2 may not be

different from their null values.

As indicated in Chapter 5 (Section 4.2.2), an approximate (two-sided) 95% confidence interval for a parameter estimate is

$$\hat{\theta} \pm Z_{.975} SE$$

where $Z_{.975}$ is the 97.5 percentile of the normal distribution (≈ 2) and SE is the standard error of the parameter estimate. Therefore, for θ_5, a 95% confidence interval is given by $1.18 \pm (2)(.0836)$, which is 1.01 - 1.35. This range only barely misses including the null value, unity, indicating, as did the marginal change in the objective function associated with going from (11.3) to (11.4), that one cannot be very sure of the influence of Apgar score on volume.

Finally, note the magnitudes of the standard errors of the other parameters' estimates. For θ_3 it is 16% of the estimate (i.e., the CV of the estimation error is 16%), for θ_4 it is 7.8%, while for the 2 elements of Ω it is 49% and 27%. This pattern is typical: the precision of the fixed effect parameter estimates is considerably greater than that of the random effects parameter estimates, except when the number of individuals sampled is enormous.

6.2. A Model Refinement

Based on the observation that θ_1 and θ_2 may be equal to their null values, these parameters are next set to their null values, defining a new (and final) model,

$$Cl = \theta_3 WT(1 + \eta_1) \tag{11.6a}$$

$$V = \left\{ \begin{array}{ll} \theta_4 WT, & \text{if APGR} > 2 \\ (\theta_4 WT)\theta_5, & \text{if APGR} \leq 2 \end{array} \right\}(1 + \eta_2) \tag{11.6b}$$

$$y = F(1 + \varepsilon_1) \tag{11.6c}$$

which is communicated to NM-TRAN without changing the $PK or $ERROR statements, but simply by fixing the values of θ_1 and θ_2 to 0, using the FIXED option in the $THETA record:

```
$PK
      TVCL=THETA(1)+THETA(3)*WT
      CL=TVCL*(1+ETA(1))
      TVVD=THETA(2)+THETA(4)*WT
      IF (APGR.LE.2) TVVD=TVVD*THETA(5)
      V=TVVD*(1+ETA(2))
      K=CL/V
      S1=V
$ERROR
      Y=F*(1+ERR(1))
$THETAS  (0 FIXED)  (0 FIXED)  (0,.0018) (0,.43) 1.0
```

When this model is fit to the data, the objective function increases only .12 (a trivial change). However, now the CV's of the estimation errors in θ_3 and θ_4 are 4.4% and 2.5% respectively.

This is the main point of this section on model refinement; deletion of imprecisely estimated parameters can improve the precision of other parameter estimates. This is related to the correlation between parameter estimation errors, mentioned in Chapter 5. With little data from patients who weigh virtually nothing, θ_1 and θ_2, the values of CL and V for such patients, are not well estimated (regardless of the fact that one might rationally model the values of these parameters to be 0), and so their parameter estimates are largely dependent on the estimates of the slope parameters θ_3, θ_4, and θ_5. The correlation between the estimates of θ_1 and θ_3 is -.96, and that between θ_2 and θ_4 is -.95. Of course, since slope itself can only be well determined when the intercept is well determined, the parameter estimates of θ_3, θ_4, and θ_5 themselves largely depend on the estimates of θ_1 and θ_2; correlations are symmetric. In other words, neither type of parameter (intercept or slope) is very precisely estimated since the estimate of each depends on the value assigned to the other. But if one of the parameters can be eliminated from the model (i.e., rationally assigned a fixed value), then the other can be more precisely determined.

7. Testing the Model

This step is undertaken when it is desirable to assign p-values to the hypothesis test of one or more parameter values against null values. The procedure is as follows: Beginning with the final model resulting from all previous steps, each parameter to be tested is set, in turn, to its null value, and the reduced model is fit to the data (only the Estimation Step need be run; no tables, graphs or covariance output are necessary). A likelihood ratio test is done using the difference in minimum objective function values obtained with both the full (original) and reduced models. In doing this one must be careful that in the Estimation Step with a reduced model, no parameter other than the one under test (and those which are already constrained to fixed values under the full model) be constrained to a fixed value.

Chapter 12 - Brief Descriptions of Other Features

1. What This Chapter is About

This chapter briefly describes a variety of features of PREDPP and NONMEM that are somewhat advanced for this text but are of interest to most users of NONMEM. References are given to other documents where additional information can be found. Section 2 is concerned with PREDPP, Section 3 is concerned with user-written PREDs, and Section 4 describes general NONMEM features. Section 5 contains an example that includes several of the advanced features. Descriptions of NM-TRAN control records in Section 4 have been augmented with sections headed "More about ...". These contain additional details, plus new options for NONMEM 7.3. Section 6 is new for NONMEM 7.3. It contains a supplemental list of features through NONMEM 7.4, including features from previous releases that are not otherwise discussed in this guide.

Note that wherever $PK, $ERROR, $DES, $AES, $MODEL, $MIX, $INFN, $TOL and $PRED statements are referred to below, user-written subroutines PK, ERROR, DES, AES, MODEL, MIX, INFN, TOL and PRED can be used instead.

2. Advanced Features of PREDPP

2.1. Pharmacodynamic Modeling Using the $ERROR Record

$ERROR statements may modify the value of F, the scaled drug concentration. They may also introduce new θ and η variables. This allows pharmacodynamic modeling to be performed using PREDPP. Such models occur when a study involves measurement of a drug effect, such as blood pressure. A proposed model might relate the predicted effect to a pharmacokinetic quantity such as plasma level. PREDPP can be used to model C_p as is usual, and the predicted effect can be computed in the $ERROR statements.

For example, suppose that a modified version of the phenobarbital data of Chapter 2 includes observations of some drug effect (in this case, perhaps a measure of the degree of sedation) but none of the concentration observations. The dose event records are the same as those of the earlier example. Suppose that the drug concentrations from each individual have been used to estimate that individual's K and V parameters, and that these estimates are now included on every event record for the individual. Finally, suppose that the proposed structural model for the effect, E, is an "E-max" model:

$$E = E_{\max} \frac{C_p}{C_{50} + C_p}$$

where here C_p is understood to mean the prediction of an individual's drug concentration in the plasma, and $E_{\max}$ and C_{50} are PD (pharmacodynamic parameters) modeled as

$$E_{\max} = \theta_1 + \eta_1$$

$$C_{50} = \theta_2 + \eta_2$$

To fit this data we can use the control statements of figure 12.1. To obtain initial parameter estimates, let us assume that the following is observable in the data. The average value of all effect measurements is about 50. Across individuals, the average value of the largest effect measurement within each individual's data is about 100, and the average value of the individual's observed concentration at about half this largest measurement is about 20. (This is seen when concentration measurements and effect measurements are

examined together.) Let us also assume 20% random interindividual variability in E_{max} and C_{50} and 4% intraindividual variability in the observation. From this we obtain initial estimates of 100 and 20 for θ_1 and θ_2, $(100\times.2)^2$ for Ω_{11}, $(20\times.2)^2$ for Ω_{22}, and $(50\times.04)^2$ for Σ.

This example is examined again in in Section 3.2, which shows the use of $PRED statements, and in Section 5, which shows how observed concentrations and effects can be fit simultaneously.

References: Users Guide VI (PREDPP) IV.B.2

```
$PROBLEM PHARMACODYNAMIC MODEL USING $ERROR STATEMENTS
$INPUT   ID TIME AMT INDK INDV DV
$DATA    EFFDATA
$SUBROUTINE ADVAN1
$PK
   K=INDK
   V=INDV
   S1=V
$ERROR
 EMAX=THETA(1)+ETA(1)
 C50=THETA(2)+ETA(2)
 E=EMAX*F/(C50+F)
 Y=E+ERR(1)

$THETA   100  20
$OMEGA   400  16
$SIGMA   4
$ESTIMATION
```

Figure 12.1. The input to NONMEM-PREDPP for analysis of effect observations.

2.2. Other Pharmacokinetic Models: ADVAN5-9, ADVAN13-15

Appendix 1 lists ADVAN routines for the most commonly-used pharmacokinetic models. Other ADVAN routines are:
ADVAN5 (General Linear)
ADVAN6 (General Nonlinear)
ADVAN7 (General Linear with Real Eigenvalues)
ADVAN8 (General Nonlinear Kinetics with Stiff Equations)
ADVAN9 (General Nonlinear Kinetics with Equilibrium Compartments)
ADVAN13 (General Nonlinear Kinetics With Stiff/Nonstiff Equations using LSODA)(nm71)
ADVAN14 (General Nonlinear Kinetics With Stiff/Nonstiff Equations using CVODES)(nm74)
ADVAN15 (General Nonlinear Kinetics with Equilibrium Compartments using IDAS)(nm74)

With the general methods the user defines a model of up to 999 compartments using special options of the $MODEL record. For a linear model (ADVAN5 and ADVAN7), it is sufficient to specify (directed) compartmental connections and to compute their rate constant parameters with $PK statements. ADVAN 5 and 7 make use of numerical approximations to the matrix exponential. For a nonlinear model (ADVAN6, ADVAN8, ADVAN9, ADVAN13, ADVAN14, ADVAN15), differential equations must be supplied to govern the kinetics, via $DES statements. It is possible to specify initial conditions for the differential equations using the I_SS (Initial Steady State) feature; Reserved variable ISSMOD may be used.

For ADVAN9 and ADVAN15, algebraic equations may also be supplied via $AES statements.

The use of the term 'nonlinear' with ADVAN 6, 8, 9, 13, 14, and 15 only indicates that a system of any type of first-order differential equations is allowed; such equations could be linear or non-linear.

In all cases, the basic features of PREDPP described in Chapter 7 are still available, such as the ability to introduce doses of any kind to any compartment of the model. It should be noted that the general ADVAN routines are relatively slow. For example, when a general method is used for a model identical to that of an analytic method (ADVAN1 through ADVAN4 or ADVAN10 through ADVAN12) the run time increases, usually by an order of magnitude.

Some ADVAN and SS routines must be told the number of accurate digits that are required in the computation of drug amounts, i.e., the relative tolerance. They may also be told the absolute tolerance. With some ADVAN and SS, the tolerances may be specified for each compartment. They may be specified by $SUBROUTINES record options TOL, ATOL, SSTOL, SSATOL; by the corresponding options of the $TOL record; or by user-written subroutine TOL (which may also specify tolerances by NONMEM Step). Option TOL (relative tolerance) may also be specified on the $COVARIANCE record. Option ATOL (absolute tolerance) may also be specified on $ESTIMATION and $COVARIANCE records.

See Guide NONMEM 7, "Controlling the Accuracy of the Gradient Evaluation and Individual Objective Function Evaluation"

With ADVAN9, ADVAN13, ADVAN14, and ADVAN15, reserved variable MXSTEP may be used to set the number of integration steps.

With $AES, $AESINIT statements are also required. If there is no TIME data item, $AESINIT may specify a calling protocol for the AES subroutine. (See 2.7 below for a discussion of calling protocols.)

`CALLFL=-1`:
Call ADVAN9 and ADVAN15 and AES with every event record (default)

`CALLFL=1`:
Call ADVAN9 and ADVAN15 and AES once per individual record.

Equivalent calling protocol phrases are:

```
(EVERY EVENT)
(ONCE PER IR)
```

References: Users Guide VI (PREDPP) VI, VII
References: Users Guide IV (NM-TRAN) V.C.3, 4, 7-10

2.3. Zero-Order Bolus Doses

Instantaneous bolus doses, which have AMT>0 and RATE=0, are described in Chapter 6. Such doses appear instantaneously in the dose compartment. Zero-order bolus doses are doses that enter the dose compartment via a zero-order process (in the same manner as do infusions) except that the rate or duration of the process is computed with $PK statements. When the RATE data item has the value -1, then the $PK statements must include an assignment statement for an additional PK parameter, Rn (the "modeled rate for compartment n"), whose value gives the rate of entry of the drug during the interval of time between the last event record and the current one. There is a different such parameter for every compartment receiving a zero-order bolus dose. When the RATE data item has the

value -2, then the $PK statements must include an assignment statement for an additional PK parameter, Dn (the "modeled duration for compartment n"), whose value at the time of the dose event gives the duration time of the dose. The rate and duration parameters can be modeled like any other PK parameters; in particular, the assignment statements can involve θ's which are to be estimated. These parameters can be used to model the drug release rate or dissolution time of a tablet or capsule.

Steady-state levels involving zero-order bolus doses can be computed.

Steady-state with constant infusion was described in Chapter 6. Steady-state infusions may also have modeled rates (i.e., the RATE data item may be -1).

References: Users Guide VI (PREDPP) III.F.3, F.4

2.4. The Additional Dose Data Item: ADDL

ADDL is a dose-related data item that is used to request that a given number of additional doses, just like the dose specified on the event record, be added to the system at a regular time interval, starting from the time on the event record. PREDPP itself adds these doses at the appropriate future times; no actual dose event record is generated by the Data Preprocessor or by PREDPP. A positive integer value in ADDL specifies how many additional doses (i.e., in addition to that already specified in the event record) are to be given, and the value in the II (interdose interval) data item (which is required) specifies the time interval between doses.

ADDL may be non-zero on a steady-state dose event record (except for steady-state infusions), in which case additional doses are given, maintaining the dosing regimen into the future. Non-steady-state kinetic formulas are used to advance the system between each additional dose. Reserved variables DOSTIM (the time of a lagged dose or additional dose to which the system is being advanced) and DOSREC (the dose record corresponding to the dose entering at DOSTIM) may be used.
See also Section 2.6 below.

References: Users Guide VI (PREDPP) V.K

2.5. Lagged doses: the ALAG Parameter

PREDPP permits an additional PK parameter called an absorption lag time. One such parameter can be defined for each compartment and applies to all doses to that compartment. It gives the amount of time that a dose is held as a "pending" dose. When the absorption lag time has expired, the dose is input into the system. In effect, the value of the absorption lag time parameter is added to the value of the TIME data item on the dose event record. With NM-TRAN, recognized names for absorption lag time parameters have the form ALAGn, where n is the compartment number. Reserved variables DOSTIM (the time of a lagged dose or additional dose to which the system is being advanced) and DOSREC (the dose record corresponding to the dose entering at DOSTIM) may be used.
See also Section 2.6 below.

See Guide VI, Chapter V, Note 3 for the effect of ALAGn with Steady-State doses.

References: Users Guide VI (PREDPP) III.F.6
References: Users Guide IV (NM-TRAN) V.C.5

2.6. Model Event Times: MTIME

Model event times MTIME(i) are additional PK parameters defined in the PK routine or $PK block. A model event time is not associated with any compartment, but, like an

absorption lag time, defines a time to which the system is advanced. When the time is reached, indicator variables are set and a call to PK is made. At this call (and/or subsequent to this call) PK or DES or AES or ERROR can use the indicator variables to change some aspect of the system, e.g., a term in a differential equation, or the rate of an infusion. Reserved variables MNEXT, MPAST, MNOW, MTDIFF may be used.

MTIME does not apply to Steady-State doses. See Guide VI, Chapter V, Note 4.

2.7. Controlling Calls to PK and ERROR

In order to evaluate the $PK and $ERROR statements, PREDPP calls the PK and ERROR subroutines. By default, the subroutines are called with every event record. PREDPP may be instructed to limit calls to certain event records in order to save the computing time involved with unnecessary calls (e.g. when the PK parameters do not vary from event record to event record within an individual). It is also possible to cause the PK subroutine to be called at times which do not correspond to any actual event record.

Using NM-TRAN, calls to PK are controlled by the presence of one of the following pseudo-statements, at the start of the $PK block:

`CALLFL=-2`:
: call with every event record, at additional and lagged dose times, and at modeled event times.

`CALLFL=-1`:
: call with every event record (default).

`CALLFL=0`:
: call with the first event record of each individual record and with new values of TIME.

`CALLFL=1`:
: call once per individual record.

A calling protocol phrase may be used instead of a pseudo-statement. A calling protocol phrase may use upper- or lower-case characters. It must be enclosed in parentheses. NM-TRAN can understand minor variations in the wording. E.g., the word "CALL" and prepositions such as WITH can be omitted. Here are calling protocol phrases equivalent to the above four pseudo-statements, respectively.

```
(CALL WITH NON-EVENT TIMES)
(CALL WITH EVERY EVENT RECORD)
(CALL WITH FIRST EVENT RECORD AND NEW TIME)
(CALL ONCE PER INDIVIDUAL RECORD)
```

The choice `CALLFL=-2 (CALL WITH NON-EVENT TIMES)` is intended to be used when PK parameters Dn and/or Fn apply to additional or lagged doses *and* the model for these parameters depends on some time-varying concomitant variable such as type of drug preparation or patient weight. By default, the values of the PK parameters which apply to the dose are those values computed by PK with the first event record having a value of TIME greater than the time at which the dose actually enters the system (the additional or lagged dose time). However, if PREDPP is instructed to also call PK at the additional or lagged dose time, then the values of the PK parameters are those values computed at these special calls. At such calls, PK has available to it information from the initiating dose event record itself, and information from the two event records whose TIME values bracket the additional or lagged dose time. Along with `CALLFL=-2` in the $PK block, the NM-TRAN $BIND record may be useful; see Users Guide IV.

Using NM-TRAN, calls to ERROR are controlled by the presence of one of the following pseudo-statements at the start of the $ERROR block:

`CALLFL=-1`:
call with every event record (default).

`CALLFL=0`: call with observation events only.

`CALLFL=1`: call once per individual record.

A calling protocol phrase may be used instead of a pseudo-statement. As in the $PK block, the calling protocol phrase may use upper- or lower-case characters and must be enclosed in parentheses.

Here are calling protocol phrases equivalent to the above three pseudo-statements, respectively.

```
(CALL WITH EVERY EVENT RECORD)
(CALL WITH OBSERVATION EVENTS)
(CALL ONCE PER INDIVIDUAL RECORD)
```

NM-TRAN automatically instructs PREDPP to limit calls to ERROR to once per *problem* for the simple error models discussed in Chapter 8, Sections 3.1 and 3.2:

```
Y=F+ERR(1)
Y=F+F*ERR(1)
Y=F*(1+ERR(1))
Y=F*EXP(ERR(1))
```

During the Simulation Step, PREDPP ignores any limitation and calls the ERROR subroutine with every event record.

Even when calls to PK and/or ERROR are limited, the CALL input data item can be used to force additional calls for specific event records as needed.

References: Users Guide VI (PREDPP) III.B.2, III.H, IV.C, V.J
References: Users Guide IV (NM-TRAN) V.C.5, C.6

2.8. Output-Type Compartments

With all versions of PREDPP output-type compartments may be defined using the $MODEL record. Suppose there is a compartment named METABURI (for metabolite in urine). If it is to be an output-type compartment, it must defined as follows:

```
$MODEL COMP=(METABURI,NODOSE,INITIALOFF)
```

The compartment is initally off, may be turned on and off, and may not receive a dose. Just as with the default output compartment, CMT may be negative on an observation record, allowing the observation to be obtained, and then the compartment turned off, with a single record. There may be more than one such compartment, in addition to the default output compartment. An output-type compartment must be turned on with an other-type event record in order to start accumulating drug. An output-type is not computed by mass-balance, but must instead be computed explictly by the ADVAN routine, e.g., using a differential equation when a general non-linear model is used.

For an example, see Chapter 6, Section 9.

```
 ID  TIME EVID UVOL  DV   CMT  AMT
  1  9.50   0   75  .058   -3    0
```

With ADVAN2, compartment 3 is the default compartment for output, and the observation at TIME 9.50 may have CMT=-3. But suppose a general linear or non-linear model

is used (ADVAN5,6,7,8,9,13) and there are more than 3 user-defined compartments.

If the $MODEL statement describes the 3d. compartment as simply

```
$MODEL COMP=NAME
```

then the default compartment attributes apply (initial on, off/on allowed, dose allowed) and the compartment is not an output-type compartment. PREDPP produces this error message for data record 3:

```
SPECIFIED COMPARTMENT MAY NOT BE TURNED OFF WITH AN OBSERVATION RECORD
```

There are two ways to avoid this error message. First, it is always possible (for any compartment that may be turned off, even the output compartment) to use two records instead of one, e.g., first the observation, then a record with EVID=2 that turns off the compartment:

```
  1  9.50   0   75  .058    3    0
  1  9.50   2    0   0     -3    0
```

Alternately, it is possible to leave the data as-is, and change the $MODEL statement so that compartment 3 is an output-type compartment.

2.9. Transgeneration of Input Data: the INFN Subroutine

NONMEM may be used to modify the data records before any computations are performed and also after all computations have been performed. This is referred to as transgeneration of the data. Transgeneration at the beginning of a problem can be used, for example, to change weight-normalized doses to unnormalized doses. PREDPP allows the user to supply a subroutine called INFN or a $INFN block of abbreviated code ("initialization/finalization") in which transgeneration can be performed. (The PREDPP library includes a default INFN subroutine which does nothing.)

The NONMEM PASS subroutine is used for transgeneration. $INFN and $PRED code may use the following statements to process each record of the data set. ICALL values may be 0, 1 or 3, for run initializaton, problem initialization, and problem finalization, respectively.

```
IF (ICALL == 3) THEN
DOWHILE(DATA)
  ...
ENDDO
ENDDO
```

Reserved variable PASSRC may be of interest.

References: Users Guide VI (PREDPP) VI.A

3. User-written PRED Subroutines

It is not necessary to use PREDPP with NONMEM. Either $PRED statements or a user-written PRED subroutine may be used in place of PREDPP to supply NONMEM with predicted values for the DV data item according to some (not necessarily pharmacokinetic) model. An example using $PRED statements is given here. A special caveat applies to user-written PRED subroutines that are recursive: see 4.6 below.

References: Users Guide I (Basic) C.2

3.1. Required Data Items

The only required data items when PREDPP is not used are the NONMEM data items DV, MDV, and ID. When PREDPP is used, the Data Preprocessor is able to recognize

which records contain observed values and which do not, and it supplies the MDV data item if it is not already present in the data file. When PREDPP is not used, the Data Preprocessor cannot do this. The input data file must already contain the MDV data item if it is needed, i.e., if the DV item of some data record does not contain a value of an actual observation.

If $PRED statements are used, they must calculate a variable called Y, using input data items and NONMEM's θ, η, and (for population models) ε vectors in the calculation.

References: Users Guide I (Basic) B.1
References: Users Guide IV (NM-TRAN) III.B.8

3.2. An Example of $PRED Statements: Pharmacodynamic Modeling

The syntax of $PRED statements is essentially the same as discussed for $PK and $ERROR statements. $PRED statements can be used for simple pharmacokinetic and pharmacodynamic models. In figure 12.1 above an example was given of pharmacodynamic modeling using $ERROR statements. Suppose that in that example, drug concentration is always measured at the same time as drug effect. Suppose too, that rather than input the individuals' values of K and V and use them to compute a predicted drug concentration for the individual, the observed drug concentration itself is used in the Emax model. This means that the the observed concentrations are again incorporated into the data, but now as values of an independent variable, rather than as the DV data item. This also means that a pharmacokinetic model is not needed, and therefore, PREDPP is not needed either. Figure 12.2 shows the control stream for this new example.

```
$PROBLEM A SIMPLE PHARMACODYNAMIC MODEL
$INPUT   ID TIME CP DV
$DATA    EFFDATA
$PRED
 EMAX=THETA(1)+ETA(1)
 C50=THETA(2)+ETA(2)
 E=EMAX*CP/(C50+CP)
 Y=E+ERR(1)
$THETA   100   20
$OMEGA   400   16
$SIGMA   4
$ESTIMATION
```

Figure 12.2. The input to NONMEM including $PRED statements for analysis of effect data.

4. Advanced Features of NONMEM

4.1. Full Covariance Matrices: $OMEGA BLOCK and $SIGMA BLOCK

In the examples of Chapter 2 and 9, there appeared statements such as:

```
 $OMEGA  .0000055, .04
```

This is an example of the specification of initial parameter estimates for a variance-covariance Ω matrix which is constrained to be *diagonal*. Initial estimates are given for the variances of η_1 and of η_2. The covariance between η_1 and η_2 is constrained to be 0, i.e., $\omega_{12} = cov(\eta_1, \eta_2) = 0$. Another way of writing this statement is:

```
 $OMEGA DIAGONAL(2)  .0000055, .04
```

The option `DIAGONAL(2)` states explicitly that the block contains two ηs and that it has diagonal form.

If the data supports the possibility that η_1 and η_2 covary with each other, it may be useful to model Ω as being unconstrained and allow NONMEM to estimate the covariance. A special form of the $OMEGA record is used, in which initial values are supplied for both variances and the covariance. For example:

```
$OMEGA BLOCK(2)  .0000055, .0000001, .04
```

The option `BLOCK(2)` states that there are two η variables in the block, and that covariance is to be estimated. The new element is $\omega_{12} = \omega_{21} = cov(\eta_1, \eta_2) = cov(\eta_2, \eta_1) = 1 \times 10^{-7}$.

$OMEGA BLOCK is used for both population and individual studies, i.e., it is the same whether η is used in the first case in a model for residual error or is used in the second case in a model for random interindividual error. In a population study, if there is more than one ε variable, and the model allows these variables to covary, then $SIGMA BLOCK is used in a similar manner.

The initial estimates of even more complicated Ω and Σ matrices may be given using multiple $OMEGA and $SIGMA records. For example, the initial estimates of a mixture of correlated and uncorrelated random variables may given. Also, in this context (as with the simple form of the $OMEGA and $SIGMA records described in Chapter 9, Section 3) variances-covariances may be constrained to fixed values by means of the FIXED option. Finally, some variances-covariances may be constrained to equal others by means of the BLOCK SAME option.

The ability to fix all variances-covariances in both Ω and Σ allows Bayesian estimates to be obtained of the pharmacokinetic parameters of a single individual, based on the individual's data and a prior population distribution for the parameters.

References: Users Guide IV (NM-TRAN) III.B.10

4.1.1. More About $OMEGA and $SIGMA

Initial estimates of a block of $OMEGA or $SIGMA must be positive definite unless the entire block is fixed to 0.

If initial estimates of a block of $OMEGA or $SIGMA is not positive definite because of rounding errors, a value will be added to the diagonal elements to make it positive definite. A message in the NONMEM report file will indicate if this was done. (nm73).

Additional options include:
`VARIANCE` (initial estimates of diagonal elements are variances (default))
`STANDARD` or `SD` (initial estimates of diagonal elements are standard deviations)
`COVARIANCE` (initial elements of off-diagonal elements are covariances (default))
`CORRELATION` (initial elements of off-diagonal elements are correlation)
`CHOLESKY` (the block is specified in its Cholesky form)

NONMEM converts all initial estimates to variance and covariances. The values desplayed in the NONMEM report and in the raw and additional output files are always variances and covariances.

If the initial estimate of $OMEGA or $SIGMA has band-symmetric form, NONMEM will be constrained to retain this form (nm7).

Special value of $OMEGA elements for unconstrained etas: If all diagonal elements of $OMEGA are "1.0E+06 FIXED" this indicates that, in a multi-subject data set, each subject's data is to be analyzed as individual data. This is described by NONMEM as
`ANALYSIS TYPE: POPULATION WITH UNCONSTRAINED ETAS`(nm73)

Short-cuts may be used for entering repeated information.

BLOCK SAME(m) option
: A count m may be included. With `$OMEGA BLOCK(n) SAME(m)` the `$OMEGA BLOCK(n) SAME` record is repeated m times. Similarly for $SIGMA records (nm73).

$THETA, $OMEGA, $SIGMA Repeated values
: When specifying initial estimates, a repeated value can be coded using notation (...)xn. E.g., $OMEGA (2)x4 can be used in place of $OMEGA 2 2 2 2. Simliarly for $SIGMA and $THETA.

$OMEGA,$SIGMA VALUES option
: If initial estimates of all diagonal elements of $OMEGA or $SIGMA are the same, and initial estimates of all off-diagonal elements are the same, they can be specified simply as $OMEGA BLOCK(n)VALUES(diag,odiag).

Informative record names for `$OMEGA` and `$SIGMA` may be used to make it easier place the records in the control stream.

`$OMEGAP` specifies omega priors
`$OMEGAPD` specifies degrees of freedom (or dispersion factor) for omega priors

They are identical to `$OMEGA` records, but understood to specify prior information for NWPRI. They may be placed anywhere in the control stream, whereas the same records without "P" or "PD" would have to be in a specific location.

Informative record names `$SIGMAP` and `$SIGMAPD` may be used similarly.

4.2. Grouping Related Observations: The L1 and L2 Data Items

The $ERROR statements for a problem may sometimes involve more than one random variable. For example, there may be two types of observations. One type may be an observation from one compartment of a PK system, or with one assay or preparation, and another type may be an observation from a different compartment or with a different assay or preparation. The model for the two types of observations would typically involve at least two ε variables (e.g. (3.8)). If all observations are made at sufficiently separated times, there may be little reason to be concerned about correlation between the two random errors. However, if the two types of observations are taken at the same or very close to the same time, it is possible that correlation will exist; whatever circumstance has influenced one observation to be different from the predicted level may also have some influence on the other observation. In this case a covariance between the two ε variables should be allowed, as described above in Section 4.1. Then the two types of observations at the same time point are regarded as two elements of a multivariate observation.

In the case of population data, there exists a NONMEM data item, L2, which is used to identify the elements of a multivariate observation. In effect, L2 acts in a similar way as ID, but grouping observations *within* individual records.

In the case of individual data, the ID data item already serves this purpose: it forms groups of observations whose η variables may be correlated. Thus, in the input data file, the ID data item should be the same for those observations which may have correlated ηs. However, for individual data, the Data Preprocessor normally replaces the ID data item with a new set of values which describe every observation as being independent of the others. To prevent the Data Preprocessor from doing this, L1 should be included in the $INPUT record as the name or synonym for the user-supplied ID data item.

Auto-correlation: The values of epsilons used in the intraindividual model may be correlated across the observations contained in the L2 record. Auto-correlation may be part of both Simulation and Estimation. The CORRL2 reserved variable may be used.

References: Users Guide IV (NM-TRAN) II.C.4, III.B.2
References: Users Guide II (Supplemental) D.3

4.3. Continuing a NONMEM Run: MSFO and MSFI

The `MSFO` (Model Specification Output File) option of the $ESTIMATION record instructs NONMEM to write a Model Specification File (MSF). It is created when NONMEM writes the first iteration summary to the intermediate output file, and is re-written when every subsequent iteration summary is written. This file can then be read in a subsequent NONMEM run using a $MSFI (Model Specification File Input) record. This file has much of the information about the model used in the previous run, thus the name "Model Specification File". It also contains all the information that allows the Estimation Step from the previous run (which may have terminated, for example, due to the number of function evaluations exceeding its limit or a computer crash or some other externally-caused interruption of the NONMEM run) to be continued in the subsequent run. There are a number of benefits to using a MSF. First, what might be a long Estimation Step (due to a very lengthy search) can be split over a series of runs, each with a limited number of function evaluations. Any run which terminates prematurely due to computer failure can be restarted from the MSF output in the previous run. (This provides a "checkpoint/restart" capability.) The progress made in the Estimation Step can also be evaluated between runs, and a decision made as to whether it is worth continuing a search which is consuming excessive amounts of computer time. Second, the Covariance, Tables, and Scatterplot Steps can be performed in later runs, each using the MSF from the final run with the Estimation Step. It is advisable to perform the Covariance Step only after satisfactory results have been obtained from the Estimation Step. Third, when NONMEM writes to the MSF, it also writes iteration summaries to the intermediate printout file (INTER). These iteration summaries are in the original parameterization (nm72).

Options are described in Guide VIII. These include NORESCALE, ONLYREAD , and NPOPETAS (nmvi). (NPOPETAS gives information to NM-TRAN rather than NONMEM.) The VERSION option allows NONMEM to read MSF files generated by previous versions of NONMEM (nm74). The NOMSFTEST option tells NONMEM to turn off strict MSFI error testing (nm74).

Option NEW allows analysis to continue, or to allow an analysis on a new data set, resuming from the final parameters of the MSF file. (nm74)

References: Users Guide I (Basic) C.4.4
References: Users Guide IV (NM-TRAN) III.B.6, B.12
References: Introduction to NONMEM 7

4.4. NONMEM Can Obtain Initial Estimates for θ, Ω, Σ

NONMEM can be directed to obtain initial estimates for one or more elements of θ, Ω, or Σ. This is done in a separate Initial Estimates Step. For an element of θ, omit the initial estimate but include lower and upper bounds, e.g., (1, ,50) in the $THETA record. (The NUMBERPOINTS option may be used to control the number of points in θ space examined by NONMEM during the search for initial estimates of θ.) For a block of Ω or Σ, omit all initial estimates on the $OMEGA BLOCK (or DIAGONAL) record, or $SIGMA BLOCK (or DIAGONAL) record, respectively.

Note that when $PK and $ERROR statements are present but the $OMEGA and/or $SIGMA records are absent, NONMEM will be directed to obtain initial estimates for the variances of the random variables in question, assuming the diagonal form of the matrix.

References: Users Guide IV (NM-TRAN) III.B.9-11

4.5. Improving Parameter Estimates: REPEAT and RESCALE

The Estimation Step can be immediately repeated after the search has terminated successfully, by including the `REPEAT` option on the $ESTIMATION record. This can improve the accuracy of the parameter estimates when one or more initial estimates are wrong by a few orders of magnitude. The final estimates from the first implementation of the Estimation Step are used as the initial estimates of the second implementation, and thus the scaling used with the STP is different from that with the first implementation, allowing fewer leading zeros after the decimal point in the STP. When the Estimation Step is continued by means of a Model Specification File, similar rescaling can be requested using the `RESCALE` option of the $MSFI record.

References: Users Guide IV (NM-TRAN) III.B.12, B.14
References: Users Guide II (Supplemental) F

4.6. The Covariance Step: R^{-1}, S^{-1}, Special Computation

The Covariance Step, which computes standard errors of the parameter estimates, first computes a covariance matrix of the parameter estimates. (This is not the same as the Ω or Σ matrix). It is possible to request that this covariance matrix be computed in one of three different ways: either as R^{-1}, S^{-1}, or $R^{-1}SR^{-1}$ (the default), where R and S are two matrices from statistical theory, the Hessian and Cross-Product Gradient matrices, respectively. Options `MATRIX=R` and `MATRIX=S` of the $COVARIANCE record are used to request the R^{-1} and S^{-1} matrices, respectively. The Covariance Step can produce additional output. When the default covariance matrix is used, R^{-1} and/or S^{-1} can be printed. This is requested by options `PRINT=R` and/or `PRINT=S`. Eigenvalues are be printed if requested by option `PRINT=E`. Multiple `PRINT` options can be specified.

A special computation is *required* when the data are from a single individual and a recursive PRED is used. A recursive PRED is one which stores the results of certain computations using the values from one event record, and uses these results in later computations with the values from a later event record. PREDPP advances the kinetic system from one time point to the next and therefore is an example of a recursive PRED. When PREDPP is used and the data is from a single individual, NM-TRAN automatically requests the special computation. When a recursive user-written PRED is used and the data are from a single individual, the `SPECIAL` option of the $COVARIANCE record *must* be used.

The `CONDITIONAL` option of the $COVARIANCE record requests that the Covariance Step be implemented only if Estimation Step terminates successfully, and is the default. The `UNCONDITIONAL` option can be used to request that it be implemented no matter how the Estimation Step terminates.

References: Users Guide IV (NM-TRAN) III.B.15
References: Users Guide II (Supplemental) D.2.5

4.6.1. More About $COVARIANCE

Other options of interest:

`COMPRESS` (affects how the Covariance matrices are displayed in the NONMEM report)
`NOSLOW|SLOW` (SLOW Requests a slower method of computation)
`SIGL|SIGLO` (affects how computations are done in the Covariance Step)
`RESUME` (allows the Covariance Step to resume from a MSF)
`NOFCOV` (turns of the Covariance Step for Estimation steps using the classical methods)

The $ESTIMATION record option NOCOV may be used to turn off the Covariance Step following a particular Estimation step, and to turn it back on again.

See Section 6.8 for more about $COV.

4.7. Multiple Problems in a Single NONMEM Run

NONMEM can implement more than one problem in a single run. That is, the input control stream can contain more than one $PROBLEM record, each followed by its own set of problem specification statements. This feature can be useful in a variety of situations. A series of what otherwise would be separate runs, each analyzing a single individual's data within a population data file, can be performed conveniently without building separate data files for each individual. Also, more than one data set can be analyzed using the same model and the same problem specification. Multiple problems are also useful with NONMEM's Simulation Step, described below.

Note that abbreviated code such as $PK and $ERROR statements cannot appear after the first problem. If the $DATA record is omitted or the filename is specified as * on a $DATA record in a problem subsequent to the first, the previous data set is re-used.

With multiple problems, the following NONMEM reserved variables are of interest:
NPROB,IPROB

A sequence of problems may be defined to be a superproblem by means of the NM-TRAN $SUPER record, and NONMEM may also be directed to repeat them a specific number of times.

With superproblems, the following NONMEM reserved variables are of interest:
S1NUM S2NUM S1NIT S2NIT S1IT S2IT
SKIP_ variable for Superproblem termination

References: Users Guide IV (NM-TRAN) III.B.1

4.8. Simulation Using NONMEM: The $SIMULATION Record

The term simulation refers to the generation of data points according to some model. A simple form of simulation is performed when the Estimation Step is omitted but the Table Step is implemented. The PRED column of the table contains predictions based on the information in the data records and the initial estimates of θ, under the model specified in the PRED (PREDPP) subroutine. Random variables η and ε (if any) have no effect on the predictions and may be omitted. If the only purpose of the run is to obtain simulated values, and these variables are present, it is best (but not required) that their variances be fixed to 0. NONMEM does not compute the objective function in this circumstance, which has certain advantages.

NONMEM can also perform a Simulation Step, in which another type of simulation is performed. In the Simulation Step, each value of the DV data item of each record with MDV=0 is replaced by a simulated observation generated from the model, but including statistical variability†. The PRED (PREDPP) routine uses η and ε values that are

† During the Simulation Step, values of F computed by PRED or PREDPP for records having MDV=1 are irrelevant and are ignored by NONMEM.

supplied by NONMEM according to user-specified random distributions (e.g., with variances given by the initial estimates of Ω and Σ). If Ω and Σ matrices are fixed to zero, for example, the simulated values are the same as the predictions described above.

If the data are then displayed by the Table Step, the DV column for records with MDV=0 contains the simulated observations obtained from the Simulation Step. For records having MDV=1, the DV column contains whatever was in the original data record. The PRED column of the table contains predictions as described above. If the Estimation Step was not implemented, the values of θ used for these predictions are the initial values. If the Estimation Step was implemented, the values of θ used for the predictions in the PRED column are the final parameter estimates. Note that the observations that are fit during the search are the simulated values obtained by the Simulation Step.

Often data are simulated using the Simulation Step, then analyzed using one or more other steps (e.g. Estimation and Covariance Steps), and this process is repeated a fixed number of times, using the same model. The Simulation Step accommodates this easily with the notion of a NONMEM subproblem, whereby these steps are repeated within the same NONMEM problem. However, on occasion it can be useful to have multiple problems (see Section 4.7), where one problem implements the Simulation Step, and the subsequent problem implements other steps. For example, this is one way to obtain different initial parameter estimates for the Estimation Step than for the Simulation Step.

The `ONLYSIMULATION` option causes NONMEM to suppress evaluation of the objective function. With this option, PRED-defined variables displayed in tables and scatterplots (see Section 4.13) are simulated values, i.e., use simulated ηs and initial θs, and weighted residual values in tables and scatterplots are always 0.

References: Users Guide IV (NM-TRAN) III.B.13
References: Users Guide VI (PREDPP) III.E.2, L.1 , IV.B.1-2, C, G.1

4.8.1. More About $SIMULATION

With simulation, subroutines SIMETA and SIMEPS are used.

With simulation and subproblems, the data set for each subproblem after the first is the same data set used by the previous subproblem, and includes any changes (transgeneration) made by the previous subproblem. With nm74, the REWIND option of $SIMULATION may be used to request that the original data set be used for all sub-problems. (If transgeneration is performed on the data set by $INFN when ICALL=1, the resulting data set is considered to be the original data set.)

See Section 6 for a discussion of the BOOTSTRAP and STRAT (stratification) features of simulation, and also parallelization during simulation.

The following NONMEM reserved variables are of interest during simulation:
IREP, NREP

NONMEM subroutine RANDOM may be used in abbreviated code to obtain numbers from a random source (nmiv, nm7).

The $SIMULATION record has other options, including:

a random seed and options `NEW`, `NORMAL`, `UNIFORM`, or `PARAMETRIC` for each of several random sources;
`TRUE=INITIAL`, `TRUE=FINAL`, or `TRUE=PRIOR`, to specify what the "true parameter values" for the Simulation should be;
`PREDICTION` or `NOPREDICTION` to specify whether the Y (or F) variable or the DV variable is set to the prediction;

REQUESTFIRST or REQUESTSECOND to specify if any eta partials are to be computed.

NONMEM can use the BOOTSTRAP method for simulations. With BOOTSTRAP, other options are possible:

REPLACE or NOREPLACE
STRAT or STRATF.

$SIM NOSUPRESET feature allows the simulation seeds not to be reset with each iteration of a super-problem.

4.9. Files for Subsequent Processing: the $TABLE Record

NONMEM can write the data for a table to an external formatted file, as requested by the FILE option of the $TABLE record. Other computer programs can read these files. Such programs can perform further analysis or provide improved graphical displays. These files normally contain header lines similar to those in a printed table, but the header lines can be suppressed entirely or in part by means of the NOHEADER, ONEHEADER, ONEHEADERALL, ONEHEADERPERFILE options. NOTITLE (suppresses the table titles) and NOLABEL (supresses column labels) may be used.

Tables may be written to the same or to different table files.

References: Users Guide IV (NM-TRAN) III.B.16

4.9.1. More about $TABLE and $SCATTER

Some options may be used only with a table file.

Options NOFORWARD and FORWARD control whether a table file which is used with multiple problems is positioned at the start of the file or forwarded to the end of the file.

Option NOPRINT may be used suppress the table in the NONMEM report, or PRINT to include it as ususal. A printed table is limited to 8 items but a non-printed table file may have an ulimited number of items (controlled by PDT in $SIZES with default 500).

FORMAT supplies an alternate format for every numeric item in a table file (the default is s1PE11.4). An alternate name for this option is DELIM.
RFORMAT supplies an alternate format for the full numeric record of a file.
LFORMAT supplies an alternate format for the full label record in a file.

Other options can be used with both printed tables and table files.

FIRSTONLY (include only the first data record from each individual record)
LASTONLY (include only the last data record from each individual record)
FIRSSTLASTONLY (include only the first and last data record from each individual record)
BY (sort records in the table)
NOAPPEND (suppress items DV, PRED, RES, WRES)
APPEND (list items DV, PRED, RES, WRES; this is the default)

With a $SCATTER record, additional options are:

FIRSTONLY (include only the first data record from each individual record)
OBSONLY (include only the observation records, having MDV=0)

The option ABS0 is similar to ORD0 described in Chapter 9, but adds a line zero line on the abscissa axis of the scatterplots.

Many additional diagnostic and reserved variables may be listed in tables and scatters; see 6.3 below.

With the Monte-Carlo generated diagnostics, new options of the $TABLE record may be used. Note that if these options affect the values of the weighted residual, the scatterplots will also be affected.

```
ESAMPLE=n1
WRESCHOL
SEED=n2
RANMETHOD=[n|S|m]
```

4.10. Data Checkout Mode

NONMEM's data checkout mode is intended for preliminary display of data without the use of a model. In data checkout mode, the PRED routine is not called. Predictions, the objective function, residuals, and weighted residuals are not computed. Only the Table and Scatterplot Steps can be implemented in the problem. With NM-TRAN, this mode is requested by coding the option `CHECKOUT` on the $DATA record. A $SUBROUTINES record and abbreviated code are required, but they have no effect and need only be syntactically correct.

References: Users Guide IV (NM-TRAN) III.B.6

4.11. Obtaining Individual Parameter Estimates - Conditional Estimates of ηs

With population data, NONMEM can obtain estimates of individual-specific true values of η from any given set of values of θ, Ω, Σ, and the individual's data. These are called conditional estimates of η. When the conditional estimates are obtained after estimation is carried out by the First-Order method, they are referred to as "posthoc" estimates. With NM-TRAN, they are requested by the option `POSTHOC` on the $ESTIMATION record.

References: Users Guide IV (NM-TRAN) III.B.14

4.12. Population Conditional Estimation Methods

NONMEM can obtain conditional estimates of η variables as part of the computation of population parameter estimates. These are called conditional estimation methods. With NM-TRAN, such methods are requested by including the option `METHOD=CONDITIONAL` (or `METHOD=1`) on the $ESTIMATION record. (The option `METHOD=ZERO`, or `METHOD=0`, requests the conventional First-Order method and is the default.) There are two conditional estimation methods. If NONMEM uses only first-order approximations, this is the First-Order Conditional Estimation Method. This has one variation, interaction, which takes into account η-ε interaction and is requested by the additional option `INTERACTION` on the $ESTIMATION record. If NONMEM uses a certain second-order approximation, this is the Laplacian method, which is requested by the additional option `LAPLACIAN` on the $ESTIMATION record. Interaction may be specified with any method, including the Laplacian method.

Note that this usage of the term CONDITIONAL is different from the usage on the $SCATTERPLOT, $TABLE, and $COVARIANCE records, in which it refers to the circumstances under which the step in question is implemented.

Option `CENTERING` requests that the average conditional estimates of each eta be constrained to be close to 0.

References: Users Guide IV (NM-TRAN) III.B.14

4.13. Displaying PRED-Defined Variables and Conditional Estimates of ηs

NONMEM can display PRED-defined variables in table and scatterplots. With NM-TRAN, any variable appearing on the left-hand side of an assignment statement in abbreviated code can be displayed by listing it in a $TABLE or $SCATTER record. If the data are population, NONMEM can also display conditional estimates of η. With NM-TRAN, variables ETA(1), ETA(2), etc., can be simply listed in $TABLE and $SCATTER records. When conditional estimation is not performed, the values displayed are zero. Displayed values of PRED-defined random variables will use conditional estimates of η if they have been obtained, otherwise they will be typical values. This feature is available with PREDPP, as well as with user-written PRED routines. For example, the following records could replace the $ESTIMATION record in Figure 12.2:

```
$ESTIMATION POSTHOC
$TABLE ETA(1) EMAX
```

The $ABBREVIATED record can be used to limit the number of variables available for display when the number is excessive.

References: Guide III (Installation) V.2.4
References: Guide IV (NM-TRAN) III.B.16-17
References: Guide VI (PREDPP) III.J, IV.E

4.14. Mixture Models

A mixture model is a model that explicitly assumes that the population consists of two or more sub-populations, each having its own model. For example, with two sub-populations, one might assume that some fraction p of the population has one set of typical values of the PK parameters, and the remaining fraction 1-p has another set of typical values. Both sets of typical values and the mixing fraction p may be estimated. For each individual, NONMEM also computes an estimate of the number of the subpopulation of which the individual is a member. The user must supply a FORTRAN subroutine called MIX or a $MIX block of abbreviated code to compute the fractions p and 1-p.

Reserved variables NSPOP, P, MIXNUM, MIXEST, MIXP and MIXPT can be used in abbreviated code. Reserved variable TEMPLT may be used.

References: Users Guide VI (PREDPP) III.L.2

4.15. PRED Error Return Codes and Error Messages in File PRDERR

A PRED routine can return a PRED error return code (1 or 2) to NONMEM, indicating that it is unable to compute a prediction for a given data record with the current values of θ's and η's. For example, PREDPP returns error return code 1 when a basic or additional PK parameter has a value that is physically impossible (e.g., a scale parameter which is zero or negative). Error return codes can also be specified by the user in user-written code or in abbreviated code using the EXIT statement. One reason for doing this is to constrain parameters in order to avoid floating point machine interrupts. The PRED error recovery option determines what action NONMEM will take. With NM-TRAN, the PRED error recovery option is either `ABORT` (which is the default) or `NOABORT`, and is specified on the $ESTIMATION and $THETA records.

If an error return code is returned during the Simulation, Covariance, Table or Scatterplot Step, or during computation of the initial value of the objective function, NONMEM will abort. If the error return code is returned during the Estimation or Initial Estimates Step, NONMEM will try to avoid those values of θ and η for which the error occurs. If they cannot be avoided, NONMEM's actions depend on the error return code value, as

follows:

1 If NOABORT is specified on $ESTIM or $THETA, try to avoid the current values of θ and η. If ABORT is specified on $ESTIM or $THETA, then abort.

2 Abort in all cases.

NOABORTFIRST may be specified on $THETA (nmvi) Same as NOABORT option, but also applies to the first value of the theta vector that is tried.

NOHABORT may be specified on $ESTIM (nm7).

PRED routines may optionally provide text accompanying the error return code. NONMEM writes all text associated with error return codes to a file, PRDERR. The contents of this file should always be carefully reviewed.

References: Users Guide III (Installation) III.2.1.1
References: Users Guide IV (NM-TRAN) IV.A, IV.C.5-6
References: Users Guide VI (PREDPP) III.K, IV.F

4.16. User-Written Subroutines

Although most NONMEM applications can be accomplished using NM-TRAN abbreviated code, there are cases in which user-written FORTRAN subroutines are needed. The $SUBROUTINES record allows the user to specify the names of user-written routines that are needed in the NONMEM load module. A user may choose to write his own PRED, PK, ERROR, INFN, MODEL, DES, or AES subroutine. Some subroutines that are distributed with NONMEM are dummy, or "stub" routines, that do nothing. Of these, subroutines CCONTR, CONTR, CRIT, PRIOR, USMETA, SPTWO, MIX can be replaced to obtain an objective function different from the default. NONMEM subroutine MIX must be replaced for mixture models. The names of all such routines are specified using the identically named options of the $SUBROUTINES record, e.g., PRED=subname, CONTR=subname, etc. User-written routines may call other FORTRAN subroutines, which can be specified for inclusion in the load module using the option OTHER=subname.

With user-written CONTR routines, the NM-TRAN $CONTR record may be useful.

THETAI and THETAR are stubs that may be replaced to transform initial and final theta values. The $THETAI and $THETAR records described in Section 6 can be used to generate the replacement code in FSUBS.

References: Users Guide IV (NM-TRAN) III.B.4, B.6

4.17. PRIOR

The PRIOR subroutine and $PRIOR record allows a Bayesian penalty function to be added to the NONMEM objective function. This serves as a constraint on the estimates of THETA, OMEGA, and SIGMA and thus as a way for stable estimates to be obtained with insufficient data.

NONMEM subroutines that may be used are NWPRI and TNPRI (nmvi). With NWPRI, informatively-named $THETAP, $OMEGAP, $SIGMAP records can be used to provide prior information (nm73).

The option NOPRIOR of the $ESTIMATION record controls whether or not the prior information is used for a given Estimation Step.

References: Introduction to NONMEM Version VI

5. Observations of Two Different Types

An NM-TRAN control stream is shown in Figure 12.3, for the analysis of a data set which contains observations of two different types. A fragment of the data set, shown in Figure 12.4, contains the data for one individual. This example illustrates how concentration and effect data can be fit simultaneously, and includes many of the advanced features described in this chapter, such as pharmacodynamic modeling in the $ERROR statements, correlation between elements of Σ, and the L2 data item.

Suppose that the data set for the phenobarbital example of Chapter 2 is modified to include both concentration and effect observations, and that a data item called TYPE is used to distinguish between them. When TYPE is 1, DV contains an effect measurement. When TYPE is 2, DV contains a concentration. The $PK statements are the same as those of Figure 2.12. The $ERROR statements are the same as those of Figure 12.1, except that the elements of θ and η are renumbered to follow those used in the $PK statements. The (random) variable Y1 is assigned the same value as Y in the $ERROR statements of Figure 12.1 The (random) variable Y2 is assigned the same value as Y in the $ERROR statements of Figure 2.12, except that ε_2 is used rather than ε_1.

The input data file contains observations of both types which were made at the same time value. The event records therefore include the L2 data item. Figure 12.4, like Figure 2.7, shows the data for the first individual, but includes TYPE and L2 data items and effect observations. Note that the L2 data item has a different value for each multivariate observation within the individual record. (The values 1 and 2 are chosen arbitrarily and may be re-used for the L2 data items in the next individual's data, if desired.)

The $THETA, $OMEGA, and $SIGMA records contain the values shown in Figures 2.12 and 12.1 and one other value, 2.8, for the covariance $\Sigma_{12} = cov(\varepsilon_1, \varepsilon_2)$. The estimate 2.8 is chosen so that the correlation is, arbitrarily, .5 ($2.8 = \Sigma_{12} = (\Sigma_{11}\Sigma_{22})^{\frac{1}{2}}\ corr = (8 \times 4)^{\frac{1}{2}}\ .5$).

```
$PROBLEM COMBINED PK/PD MODEL
$INPUT   ID TIME AMT WT APGR DV TYPE L2
$DATA    COMBDATA
$SUBROUTINE ADVAN1
$PK
   TVCL=THETA(1)+THETA(3)*WT
   CL=TVCL+ETA(1)
   TVVD=THETA(2)+THETA(4)*WT
   V=TVVD+ETA(2)
                     ; THE FOLLOWING ARE REQUIRED BY PREDPP
   K=CL/V
   S1=V
$ERROR
 EMAX=THETA(5)+ETA(3)
 C50=THETA(6)+ETA(4)
 E=EMAX*F/(C50+F)
 Y1=E+ERR(1)
 Y2=F+ERR(2)
 Q=1
 IF (TYPE.EQ.2) Q=0
 Y=Q*Y1+(1-Q)*Y2
$THETA  (0,.0027) (0,.70) .0018 .5 100 20
$OMEGA  .000007 .3 400 16
$SIGMA BLOCK(2) 4 2.8 8
$ESTIMATION
```

Figure 12.3. The input to NONMEM-PREDPP for analysis of the population phenobarbital data, including both concentration and effect observations.

```
1    0.    25.0    1.4      7      .   2   0
1    2.0      .    1.4      7    6.0   1   1
1    2.0      .    1.4      7   17.3   2   1
1   12.5    3.5    1.4      7      .   2   0
1   24.5    3.5    1.4      7      .   2   1
1   37.0    3.5    1.4      7      .   2   0
1   48.0    3.5    1.4      7      .   2   1
1   60.5    3.5    1.4      7      .   2   0
1   72.5    3.5    1.4      7      .   2   1
1   85.3    3.5    1.4      7      .   2   0
1   96.5    3.5    1.4      7      .   2   1
1  108.5    3.5    1.4      7      .   2   0
1  112.5      .    1.4      7    8.0   1   2
1  112.5      .    1.4      7   31.0   2   2
```

Figure 12.4. The first individual's phenobarbital data, including both concentration and effect observations.

The above $ERROR statements can be coded more simply.

```
$ERROR
 EMAX=THETA(5)+ETA(3)
 C50=THETA(6)+ETA(4)
 E=EMAX*F/(C50+F)
 IF (TYPE.EQ.2) THEN
 Y=F+ERR(2)
 ELSE
 Y=E+ERR(1)
 ENDIF
```

Figure 12.5. Alternate $ERROR statements

6. Supplemental List of Features through NONMEM 7.4

With NONMEM 7 there are many new features, including new Estimation Methods. This section lists features of NONMEM, PREDPP, and NM-TRAN that are not discussed elsewhere in this guide. The version of NONMEM in which each feature appears is listed. The user should consult other guides for details.

6.1. NONMEM Features

Odd-Type Data (nmv)

Non-continuous observed responses ("odd-type data") can be analyzed. $ESTIMATION options LIKELIHOOD or -2LL must be used. Y is set to a (conditional) likelihood.
Reserved variable F_FLAG may be used (nmvi).

New methods of Estimation

METHOD=HYBRID with option ZERO (nmv)
STIELTJES with options GRID, REPEAT1, REPEAT2, ZERO (nmvi)
ITS Iterative Two Stage (nm7)
Expectation-Maximization (EM) and Monte Carlo Bayesian (nm7)

Expectation feature (nmv)

This feature uses the NONMEM marginal (MRG_) data item. MRG_ identifies records for which NONMEM computes and displays marginal quantities (expectations) Expectations are computed when ICALL=5.

Raw data average feature (nmv)

This feature uses the NONMEM raw-data (RAW_) data item. RAW_ identifies template records for which NONMEM computes and displays raw-data averages. Raw data averages are computed when ICALL=6. Reserved variables TEMPLT and the $OMIT record may be used (nmvi). The NONMEM utility routine RANDOM may be used to obtain numbers from different random sources.

Non-parametric analysis methods (nmvi)

The $NONPARAMETRIC record is used to request the Non-parametric method of analysis. Options include:
MARGINALS or ETAS, MSFO=filename, RECOMPUTE, EXPAND, NPSUPP=n or NPSUPPE=n

SORT option of $ESTIMATION (nmvi)

With classical methods, individual contribution to the objective function and gradients may be sorted before they are summed, so that smaller numbers are summed before larger numbers.

Reserved Variables YLO/YUP (nmvi)

During the analysis an interval is defined in which (or outside of which) an observation is conditioned to exist. Reserved variable PR_Y is also of interest.

Reserved Variables CTLO/CTUP (nmvi)

An observation may be the event that the value of a normally distributed variable falls in a given interval. Reserved variable PR_CT is also of interest.

NONMEM Repetition feature (nmvi)

This features uses reserved variables RPTI,RPTO,RPTON,PRDFL. An alternate way is to use the RPT_ data item.

MU Modeling (MU Referencing) (nm7)

MU_i variables may be used in Abbreviated code with EM methods of Estimation.

NM-TRAN checks the use of MU_i variables, unless option *NOCHECKMU* of the $ABBR record is used (nm73). Thetas may be input and reported in their natural domain, even when used as logs (e.g., linear MU referencing) using $THETAI and $THETAR records (nm73).

New method of setting initial values of thetas, omegas, and sigmas. (nm72)

See CHAIN option of $ESTIMATION and $CHAIN.

Multiple Estimation steps (nm72)

If the $ESTIMATION record is present more than once within a problem, then each subsquent record requests a separate Estimation Step rather than providing more options for a single Estimation Step.

BOOTSTRAP method (nm73)

BOOTSTRAP may be specified with $NONPARAMETRIC and $SIMULATION records. This requests that a bootstrap sample be used. Options STRAT and STRATF may be used for stratification. With $SIMULATION, options REPLACE or NOREPLACE may be used. An example is given of bootrapping single subject data (nm74).

More than 2 levels of mixed effects (nm73)

Increased number of mixed effects levels. Random effects across groups of individuals, such as clinical site, can be modeled. The $LEVEL record is used.

Alternate method (POPULATION WITH UNCONSTRAINED ETAS) for single-subject analysis (nm73)

All the subjects may be analyzed together, but with $OMEGA diagonal values fixed to a special value 1.0E+06.

New values of MDV (nm73)

MDV may be set to 100, 101. Such records are ignored during Estimation. Reserved variables MDVI1, MDVI2, MDVI3 may also be used; they are defined in include file nonmem_reserved_general.

Initial Estimates for ETAs feature (nm73)

By default, the initial value used for ETA's in the Estimation Step search is 0. The $ETAS and $PHIS records provide user-supplied initial estimates.

Tranformations of THETA values (nm73)

$THETAI transforms the initial values in the $THETA and $THETAP records. $THETAR transforms the final theta values for the NONMEM report and additional output files. May be used with MU Modeling.

Constraints on model parameters (nm73)

Additional algorithmic constraints may be imposed upon model parameters by use of the subroutine CONSTRAINT. Option CONSTRAIN of the $ESTIMATION record and the $ANNEAL record may be used to give information to the subroutine. This feature is available only for the EM and Bayesian algorithms.

Additional Reserved Variables

The descriptions of the following reserved variables can be found in Introduction to NONMEM 7 MXSTEP FIRSTEM MDVRES NPDE_MODE DV_LOQ CDF_L

6.2. Miscellaneous Features

Interactive control of NONMEM (nm7)

A NONMEM run can now be controlled to some extent from the console by issuing certain control characters.

Dynamic Memory Allocation (nm72)

No need to recompile NONMEM or NM-TRAN for large problems. Most arrays are sized automatically. If necessary, the $SIZES record may be used. E.g., the default maximum number of data items per data record is 50, but may be increased by specifying a larger value for PD; the maximum number of items per table is 500, but may be increased by specifying a larger value for PDT.

Parallel Computing (nm72)

Parallel Computing is requested using the nmfe option -parafile and specified using .pnm files. The options PARAFILE of the $ESTIMATION and $COVARIANCE records may also be used. With nm74, Option FPARAFILE of the $ESTIMATION record controls parallelization for final eta (EBE) computation. Option PARAFILE of the $TABLE controls parallelization for weighted residual computation. Option PARAFILE of the $SIMULATION record may also be used.

6.3. Changes to NONMEM Outputs

Reports include Covariance and Correlation Matrices for OMEGA and SIGMA (nm72)

Reports include ETABAR, SE, N, P VAL (nm7)

Option ETABARCHECK of the $ESTIMATION record may be used.

Reports include ETAshrink, EBVshrink, EPSshrink (nm7)

Eta shrinkage evaluation using empirical Bayes variances (EBVs, or conditional mean variances) is reported. The ETASTYPE option of the $ESTIMATION record and the ETASXI reserved variable in abbreviated code may be used to control which etas from which subjects are included.

Reports include tag labels: #METH etc. (nm7)

Raw and additional output files: root.ext, root.cov, root.xml, etc. (nm7)

These files provide numerical results in a columnar format. $ESTIMATION record option ORDER may be used to control the order of theta, omega, sigma in these files. $ESTIMATION record option NUMDER may be used to request files with numerical and analytic eta derivatives: root.fgh, root.agh (nm73)

Tables and Scatters may request NONMEM-generated items

Elements of G and H (e.g., G11, H11) and elements of ETA (nmvi)

A range of etas using the format ETAS(k:n) may be requested (nm73).
Number lists or a syntax flexible (TO, :, BY) may be used(nm74). Examples are ETAS(1 TO 10 by 3), ETAS(1,5,12,4).

OBJI (Objective function values for each individual) (nm72)

Additional statistical diagnostic items (nm7, nm73)

In addition to the PRED, RES, and WRES items, the following may be listed.

PREDI, RESI, WRESI
CPRED, CRES, CWRES
CPREDI, CRESI, CWRESI
CIPRED, CIRES,CIWRES
CIPREDI, CIRESI,CIWRESI
NIPRED, NIRES, NIWRES

IPRD, IRS,IWRS
EPRED, ERES, EWRES

Monte-Carlo generated diagnostics are not linearized approximations like the other diagnostic types. These include

ECWRES
EIPRED, EIRES,EIWRES
NPDE Monte-Carlo generated normalized probability distribution error) (nm71)
NPD correlated value of NPDE (nm72)

With FIXEDETAS=(list), the specified etas are treated as if they are fixed effects when NONMEM evaluates population diagnostics during the $TABLE step.(nm74)

The EXCLUDE_BY option can be used to exclude records from the table or scatter. (nm74).

The VARCALC option asks NONMEM to report standard errors (xxx_SE) in the tables for PREDPP and user-defined items. (nm74)

A reserved variable of interest when evaluating residuals and weighted residuals is MDVRES which may be set in PRED to cause NONMEM to treat an observation as missing during the computation of residuals and weighted residuals. (nm73)

6.4. PREDPP

New PREDPP data items in $INPUT: XVID1 XVID2 XVID3 XVID4 XVID5 (nm72)
Special values of EVID allow repeated observation records, e.g., for Stochastic differential equations.

CMT and PCMT values 100,1000
Specification of the default compartment for output (nm, nm73)

Compartment Amounts A(i), TSTATE (nmvi)

A_0 (compartment initialization) (nmvi)
May be used with any ADVAN. A_0FLG

I_SS (Initial Steady State) for general non-linear models (nmvi2.0)
It is possible to specify initial conditions for the differential equations using the I_SS (Initial Steady State) feature. Reserved variable ISSMOD may be used.

DES array: COMPACT vs. FULL for general non-linear models (nmv)

ISFINL reserved variable with AES and DES (nmvi)
Allows the abbreviated code to take special action on the final call to AES and DES for an integration interval. TIME data item may be negative.

6.5. NM-TRAN

6.5.1. General Features

Case-insensitivity (nm72)
Both lower and upper case may be used in the NM-TRAN control file.

Continuation and line length (nm73)
Any line may be continued with "&" and may be 67000 characters long.

Warning messages (nmv)
The numbers of warning messages of various types may be controlled using the

$WARNING record.

6.5.2. Data Preprocessor

$DATA TRANSLATE (nmv, nm73)
: Allows TIME and II values to be rescaled, with specified number of decimal points.

ill-formed data files (nmvi)
: NM-TRAN is better able to handle a data file whose final line does not terminate correctly.

tabs in data files (nmvi)

^M in data files (nmvi)
: NM-TRAN can read data files in which tabs are present, and whose lines end with ^M.

$DATA BLANKOK (nmvi)
: NM-TRAN will not allow blank lines in a data file unless the BLANKOK option is used.

Larger data files (nmvi)
: The RECORDS=n option of $DATA may specify a number as large as 99999999.

MISDAT Missing Data Indicator (nm74)
: MISDAT specifies a numerical value indicating a missing data value in the data set, which is displayed on $TABLE table outputs, but is safely interpreted as 0 by other steps of NONMEM.

6.5.3. Abbreviated Code

FORTRAN 90/95 syntax may be used.
: For example, logical expressions may be written using symbols ==,>, instead of .EQ., .GT., etc.

Increased number of THETA, ETA, EPS (nm72)
: Subscripts of THETA, ETA, EPS may be as large as 999.

$ABBR record: COMRES, COMSAV
: Creates variables that are saved between nonmem passes. NONMEM Reserved variables COM, COMACT are used.

$ABBR record: DERIV2 (nmIV), NOFASTDER(nm72) DERIV1 (nm74)
: Affects generated code in FSUBS. See also NOFIRSTDERCODE reserved variable in abbreviated code.

$ABBR REPLACE (nm73)
: Any character string may be replaced. This allows for symbolic reference to thetas, etas, and epsilons. Replacement with selection by data item and parameter is permitted.
With nm74, the syntax is more flexible. Symbolic labels for eta may be used in the $TABLE record. Symbolic label substitutions will appear in the NONMEM report file and $TABLE outputs. $ESTIMATION record option NOSUB may be used to control label substitution in the NONMEM report file. $TABLE record option NOSUB may be used to control label substitution in $TABLE files. $SCATTER record option NOSUB may be used to control label substitution in Scatterplots.

$ABBR DECLARE (nm73)
: Allows integer variables and array (subscripted) variables to be used in Abbreviated code.

Recursive abbreviated code (nmvi)
: Allows a random variable to retain the value from the previous data record instead of being set to zero. May be used to implement recursive kinetics in $PRED.

User-supplied functions FUNCA,FUNCB,FUNCC and VECTRA,VECTRB,VECTRC (nmvi)
: FUNCA etc. are reserved names for user-supplied functions They may have scalar or vector-valued arguments. VECTRA etc. are reserved names for vectors used as arguments. When functions are used in abbreviated code, the eta derivatives of the arguments are computed correctly. Any vector may be used with any function. With NONMEM 7, there are more reserved functions and vectors.

$ABBR FUNCTION, $ABBR VECTOR (nm74)
: In NONMEM 7.4 the $ABBR FUNCTION option and $ABBR VECTOR option allows user-defined function names and user-defined argument vector names.

PROTECT functions (nm74)
: Versions of FORTRAN functions are available that protect against domain violations, divide by zero, and floating point overflows. For example, PLOG is the protective code routine that performs the LOG operation. The $ABBR PROTECT record causes NM-TRAN to automatically replace FORTRAN functions in abbreviated code with the protective functions.

WRITE/PRINT statements
: Character strings, format specification, Array options FULL vs. DIAG

DO WHILE, DO WHILE(DATA) statements
: Looping; transgeneration.

Include files for NONMEM_RESERVED variables (nm73)
: If the name of an include file starts with NONMEM_RESERVED, it may contain definitions of variables that will be parsed by NM-TRAN for use in abbreviated code.

6.5.4. Reserved Variables in Abbreviated Code

Here is a partial list of reserved variables that are not mentioned elsewhere in this guide.

ICALL
: NONMEM reserved variable. Tells PRED when NONMEM is doing Run initialization, Problem initialization, Estimation, Problem finalization, Simulation, Expectiation, Data Average. (nmv)

NEWIND
: NONMEM reserved variable. Tells PRED when data from a new individual record is starting.

NIREC, NDREC (nmvi)

FIRSTOBS, LASTOBS (nm74)
: NONMEM reserved variables. Input data file record counters. NONMEM 7.4 provides additional record counters such as FIRSTOBS, LASTOBS, etc. in file nonmem_reserved_general.

LIREC NINDR INDR1 INDR2 reserved variables (nmvi2.0)
: NONMEM reserved variables. Descriptive of the individual record.

MSEC, MFIRST, IFIRSTEM
: NONMEM reserved variables. Tells PRED which derivatives to compute.

THETAFR, OMEGA, SIGMA, SETHET, SETHETR, SEOMEG, SESIGM (nmvi,nm74)
: NONMEM reserved variables. The current values of OMEGA, SIGMA, et. al., may be used in abbreviated code.

IIDX,CNTID (nmvi)
: NONMEM reserved variables. Individual contribution to the objective function.

PRED_,RES_,WRES_, and other variables
: Variables with similar names and the same values as statistical diagnostic items PRED_, RES_, WRES_, CPRED_, CRES__, CWRES, etc., may be used on the right in $PRED and $ERROR blocks (nm7)

NONMEM_reserved_general (nm73)
: This is a file in the util directory with declarations for many additional reserved variables.

6.6. Utility Routines

This is a list of utility programs found in the util directory.

nmfe74
: The nmfe shell script has many new options, including options for parallel computing.

finedata
: Augments an NM-TRAN data file to incorporate additional, non-observation, time values spaced at regular increments.

nmtemplate
: Performs variable substitution on appropriately tagged control stream template files, and produces new control stream files. Compare with the $ABBR REPLACE feature, above.

table_quant
: Transforms results in raw output file of $COV SIRAMPLE step, places into a table file with frequencies and cumulative values

table_resample
: Resamples from raw output file of $COV SIRAMPLE step, using the WEIGHT information

table_compare
: Compares the numerical values between two table files produced by the $TABLE record.

table_to_xml
: Converts additional output table files produced by NONMEM to XML Formatted files.

xml_compare
: Compares the contents of two NONMEM report XML files.

doexpand
: Expand an NM-TRAN control stream file that has been annotated with DOE (which stands for DO expand) and ENDDOE (which stands for ENDDO expand)

directives.

ddexpand
: Expands a control stream file by adding equations for time-delay differntial equation problems.

neff
: Performs effective sample statistics on population parameters in raw output file genereated by Bayesian or NUTS analysis

neffi
: Performs effective sample statistics from individual parameters genereated by Bayesian or NUTS analysis

6.7. All Options for $ESTIMATION

This section lists all options of the $ESTIMATION record. Some are discussed earlier in this guide and are listed here for completion. Some options are only appropriate with specific estimation methods. For more information, see the $ESTIMATION help item.

-2LL (nmv)
: Y evaluated in $ERROR or $PRED is intepreted as -2 times log likelihood

ATOL (nm72)
: Absolute tolerance adjustment for ADVAN9 and ADVAN13

AUTO (nm73)
: Have NONMEM determine optimal settings for certain EM/Bayes options

CALPHA (nm7)
: alpha error rate for Monte Carlo EM and Bayes convergence

CENTERING (nmv)
: Impose centering of average empirical Bayes estimates (EBEs) about zero (FOCE).

CINTERVAL (nm7)
: Correlation iteration interval for Monte Carlo EM and Bayes convergence

CITER/CNSAMP (nm7)
: Number of iteration samples to use for Monte Carlo EM and Bayes convergence

CONDITIONAL (nmiv)
: Assess objective function around each subject's (conditional) etas during Estimation (FOCE/Laplace)

CONSTRAIN (nm72)
: Impose algorithmic constraints on thetas through CONSTRAINT subroutine (EM/BAYES)

CTYPE (nm7)
: Select convergence criterion

DERCONT (nm73)
: Correct for derivative continuity in change of objective function with theta (SAEM/IMP)

DF (nm71)
: degrees of freedom of t-distribution of sampling density for IMP and IMPMAP

DFS (nm73)
: degrees of freedom for simulating initial SIGMAS (CHAIN only)

EONLY (nm71)
: Expectation step only, no advancement of thetas or sigmas for EM methods. With

nm74, also affects individual conditional means/modes and conditional variances, and approximate variances.

ETABARCHECK (nmvi)
p-value of ETABAR (mean EBEs) tests similarity to ETABAR of a previous problem

ETADER (nm73)
Select alternative finite difference methods for eta derivatives

ETASAMPLES (nm74)
Generates posterior density samples of etas

ETASTYPE (nm73)
Determine whether non-influential etas should be included in ETABAR/Shrinkage statistics

FAST (nm74)
Uses analatical derivatives of thetas and sigmas to speed up FOCE analyses

FILE (nm71)
specify alternative name for raw ouptut file containing fixed effects parameters progress

FNLETA (nm72)
Determine how final etas are obtained for table outputs

FORMAT/DELIM (nm71)
specify alternative numerical format for output files.

FPARAFILE (nm74)
Turn ON or OFF parallelization of final etas evaluation.

GRD (nm71)
Specify gradient behavior of THETAS and SIGMAS for EM/BAYES methods

GRDQ (nm74)
Gradient quick option, specifying what number of fraction of importance samples generate should be used for gradient evaluation of non-mu modeled parameters

GRID (nmvi)
Set up search grid pattern for Stieltjes method

HYBRID (nmv)
Use conditional etas except for those etas listed in ZERO option (hybrid of FOCE and FO)

IACCEPT (nm71)
Acceptance/rejection ratio or proposal density coverage for EM/BAYES

IACCEPTL (nm74)
Scale a second multi-variate normal density, to cover long tails in the posterior density.

INTERACTION (nmiv)
Assess residual variance (epsilon terms) using conditional (non-zero) etas.

ISAMPEND (nm73)
Maximum value for ISAMPLE

ISAMPLE (nm71)
Number of Monte Carlo ETA samples to collect for each subject

ISAMPLE_M1 (nm71)
Number of ETA samplings to test in the OMEGA space (SAEM/BAYES)

ISAMPLE_M1A (nm72)
Number of ETA samplings to test using ETA samples of other subjects (SAEM/BAYES)

ISAMPLE_M2 (nm71)
Number of multi-variate ETA vector samplings to test in the local space (SAEM/BAYES)

ISAMPLE_M3 (nm71)
Number of uni-variate ETA samplings to test in the local space (SAEM/BAYES)

ISCALE_MAX (nm72)
Maximum factor to expand prospoal density for ETA sampling (SAEM/BAYES/IMP/IMPMAP)

ISCALE_MIN (nm72)
Minimum factor to scale prospoal density for ETA sampling (SAEM/BAYES/IMP/IMPMAP)

KAPPA (nm74)
Specify power term to be used in average acumulating samples for mass matrix production for NUTS analysis

KNUTHSUMOFF (nm74)
Turn off precision retaining KnuthSUm algorithm when summing individual OFVs to produce total OFV.

LAPLACE (nmiv)
2nd Order conditional estimation method

LEVWT (nm74)
Specify how to weigh subjects in nested random levels ($LEVEL) problem

LIKE (nmv)
Y evaluated in $ERROR or $PRED is interpreted as likelihood

LNTWOPI (nm74)
Add the N*log(2pi) term to the objective function

MADAPT (nm74)
Specify how the mass matrix is updated during a NUTS analysis

MAPCOV (nm74)
MAPCOV=1 is the default.

MAPINTER (nm72)
Iteration interval at which to use MAP estimates for proposal density (IMP)

MAPITER (nm72)
Number of first set of iterations at which to use MAP estimates for proposal density (IMP)

MASSRESET (nm74)
Initialize mass matrix accumulation, or borrow from previous estimation.

MAXEVAL (nmiv)
Maximum number of function evaluations (FO/FOCE/FOCEI/Laplace)

MCETA (nm73)
Number of Monte Carlo samples to assess best starting eta vector for MAP

estimation

METHOD (nmiv)
Specify method of estimation

MSFO (nmiv)
File name for containing estimation information to use in subsequent analyses

MUM (nm71)
Turn on or off MU-referencing for EM/BAYES analysis

NBURN (nm71)
Number of burn-in iterations for SAEM/BAYES methods

NITER/NSAMPLE (nm71)
Number of iterations for EM/BAYES methods

NOABORT (nmiv)
Have NONMEM Recover from numerical errors during estimation

NOCOV (nm73)
Do not evaluate covaruiance step for particular estimation step

NOHABORT (nm73)
Have NONMEM recover from all numerical errors during estimation (stronger than NOABORT)

NOLABEL (nm71)
Do not print column names in additional output files

NOOMEGABOUNDTEST (nmvi)
Do not limit how much OMEGA elements may change in an estimation (FO/FOCE/Laplace)

NOSIGMABOUNDTEST (nmvi)
Do not limit how much SIGMA elements may change in an estimation (FO/FOCE/Laplace)

NOTHETABOUNDTEST (nmvi)
Do not limit how much THETA parameters change in an estimation (FO/FOCE/Laplace)

NOSUB (nm74)
Turn off substitution of variable labels in Table headers.

NOTITLE (nm71)
Do not print title (header) in additional output files

NONINFETA (nm73)
Determine how NONMEM treats etas that do not influence the subject's data likelihood

NOPRIOR (nm71)
Turn on or off the contribution of the prior information

NSIG (nmiv)
number of signficant digits for convergence criterion (classical methods, ITS)

NUMDER (nm73)
Output numerical and/or analytical ETA derivatives

NUMERICAL (nmv)
Use finite difference method for 2nd derivative ETAS in MAP estimation (Laplace, ITS, MAP, IMPMAP)

NUTS_BASE (nm74)
Specify number of iterations for stage II of warmup process of NUTS analysis

NUTS_DELTA (nm74)
Sample acceptance rate for NUTS analysis

NUTS_EPARAM (nm74)
Specify parameterization for individual parameters/etas in NUTS analysis

NUTS_GAMMA (nm74)
Gamma factor for NUTS algorithm

NUTS_INIT (nm74)
Specify number of iterations for stage I of warmup process of NUTS analysis

NUTS_MASS (nm74)
Specify whether mass matrix should be full, diagonal, block-diagonal, etc.

NUTS_MAXDEPTH (nm74)
Sets the maximum number of total branchings to try in the NUTS algorithm in the search for the next decorrelated sample

NUTS_OPARAM (nm74)
Specify parameterization for Omegas in NUTS analysis

NUTS_REG (nm74)
Specify diagonal dominance algorithm for mass matrix in NUTS analysis.

NUTS_SPARAM (nm74)
Specify parameterization for Sigmas in NUTS analysis

NUTS_STEPINTER (nm74)
An initial step size is calculated every NUTS_STEPINTER iterations.

NUTS_STEPITER (nm74)
An initial step size is calculated for the first NUTS_STEPITER iterations.

NUTS_TERM (nm74)
Specify number of iterations for stage III of warmup process of NUTS analysis

NUTS_TEST (nm74)
Specify acceptance/rejection algorithm in NUTS algorithm

NUTS_TRANSFORM (nm74)
Specify whether estimation parameters or momentum parameters are to be transformed in NUTS algorithm.

OACCEPT (nm7)
Select acceptance/rejection ratio for Metroplis-Hastings algorithm of finding OMEGAS (BAYES)

OLKJDF (nm74)
Set degrees of freedom for LKJ correlation for Omegas

OLNTWOPI (nm74)
Include log(2pi) degrees of freedom from eta density portion of objective function

OMEGABOUNDTEST (nm74)
Limit how much OMEGA elements may change in an estimation (FO/FOCE/Laplace)

OMITTED (nmiv)
Omit estimation

OPTMAP (nm73)
Select optimization method for MAP estimation

ORDER (nm72)
Select ordering of fixed effects parameters in raw output file

OSAMPLE_M1 (nm71)
Number of samples for Metroplis-Hastings global search of finding OMEGAS (BAYES)

OSAMPLE_M2 (nm71)
Number of samples for Metroplis-Hastings local search of finding OMEGAS (BAYES)

OVARF (nm74)
The weight to STD prior to the log sqrt OMEGA diagonal elements

PACCEPT (nm71)
Select acceptance/rejection ratio for Metroplis-Hastings algorithm of finding THETAS/SIGMAS (BAYES)

PARAFILE (nm72)
Specify new parallization file for estimation, or turn ON/OFF parallelization

PARAFPRINT (nm74)
Print iteration interval for parallelization log file

PHITYPE (nm74)
have .phi file contain conditional neans phis or etas

POSTHOC (nmiv)
Assess EBEs for each subject after FO estimation

PREDICTION (nmv)
Determines how Y or F is interpreted with simulation

PRINT (nmiv)
Iteration print interval

PSAMPLE_M1 (nm71)
Number of samples for Metroplis-Hastings (MH) global search of finding THETAS/SIGMAS (BAYES)

PSAMPLE_M2 (nm71)
Number of samples for MH local multi-variate search of finding THETAS/SIGMAS (BAYES)

PSAMPLE_M3 (nm71)
Number of samples for MH local uni-variate search of finding THETAS/SIGMAS (BAYES)

PSCALE_MIN (nm73)
Minimum factor to expand prospoal density for MH sampling of THETAS/SIGMAS(BAYES)

PSCALE_MAX (nm73)
Maximum factor to scale prospoal density for MH sampling of THETAS/SIGMAS(BAYES)

RANMETHOD (nm72)
Select random number generator and behavior for Monte Carlo EM and BAYES methods

REPEAT (nmiv)
repeat estimation starting at final parameters from first loop (FO/FOCE/Laplace)

REPEAT1 (nmvi)
repeat first stage of Stieltjes estimation

REPEAT2 (nmvi)
repeat second stage of Stieltjes estimation

SADDLE_HESS (nm74)
Selects type of Hessian to be used for Saddle reset process

SADDLE_RESET (nm74)
Set the number of times a saddle_reset is performed

SEED (nm7)
Select starting seed for Monte Carlo EM and Bayes methods

SIGL (nm7)
Significant digits of individual objective function assessment

SIGLO (nm72)
Significant digits to assess ETAS in MAP estimation

SLOW (nmvi)
Use slow method of advancing fixed effects parameters

SIGMABOUNDTEST (nmvi)
Limit how much SIGMA elements may change in an estimation (FO/FOCE/Laplace)

SLKJDF (nm74)
Set degrees of freedom for LKJ correlation for Sigmas

SORT (nmvi)
Sort individual objective function values before summing into total objective function

STDOBJ (nm73)
Stochastic standard deviation tolerance of objective function to determine best ISAMPLE for IMP/IMPMAP

STIELTJES (nmvi)
Higher order assessment of objective function

SVARF (nm74)
The weight to STD prior to the log sqrt Sigma diagonal elements

THETABOUNDTEST (nmvi)
Limit how much THETA parameters change in an estimation (FO/FOCE/Laplace)

THIN (nm74)
Sample intervals to be recorded in the raw output file for Bayesian analysis

TTDF (nm74)
Set t-distribution degrees of freedom for priors to Thetas

ZERO (nmv)
List of etas for which conditional etas are not to be used in HYBRID method

References: Introduction to NONMEM 7

6.8. All Options for $COVARIANCE

This section lists all options of the $COVARIANCE record. Some are discussed earlier in this guide and are listed here for completion. For more information, see the $COVARIANCE help item.

ATOL (nm73)
: Asolute tolerance for differential equation problems

CHOLROFF (nm74)
: Have R matrix evaluated according to earlier versions of NONMEM.

COMPRESS (nmv)
: Covariance Step arrays are printed in compressed format regardless of dimension size of covariance of estimates

CONDITIONAL (nmiv)
: Evaluate covaraince step only if estimation successful

FAST (nm74)
: Uses analatical derivatives of thetas and sigmas to speed up covariance step

FILE (nm74)
: Select file name of raw output file for SIR sampling

FORMAT (nm74)
: Select format of numbers to be written to raw output file during SIR sampling

FPOSDEF (nm74)
: Force positive definiteness on R matrix after Preconditioning

IACCEPT (nm74)
: Acceptance rate (sampler expansion) during SIR importance sampling

IACCEPTL (nm74)
: Acceptance rate of the secondary sampler during SIR Importance sampling

KNUTHSUMOFF (nm74)
: Turn off precision retaining KnuthSUm algorithm when summing individual OFVs to produce total OFV.

MATRIX (nmiv)
: Select type of Information matrix to be evalauted during Covariance step

NOFCOV (nm72)
: Turn off covariance estimation for FOCE method

NOSLOW (nm72)
: Use analytical derivatives of Omegas to evaluate gradients during covariance step

PARAFPRINT (nm74)
: Print iteration interval for parallelization log file during covariance step

PFCOND (nm74)
: Force predonditioning even if Rmatrix is positive definite during covariance step

PRECOND (nm74)
: Set number of preconditioning cycles to perform during covaraince step

PRECONDS (nm74)
: Select whether preconditioning should be done on Thetas, Omegas, and/or Sigmas of R matrix portion in covariance step

PRETYPE (nm74)
Select the R matrix corrector type when preconditioning during the variance-covariance step.

PRINT (nmiv)
Select to Print out additional matrices and items (E=eigenvalues, R=R matrix, S=S matrix)

RANMETHOD (nm74)
Select randomozation method for SIR sampling

RESUME (nm73)
Collect intermediate information to resume covariance step if interrupted

SIGL (nm71)
Significant digits of individual objective function assessment during covariance step

SIGLO (nm72)
Significant digits to assess ETAS in MAP estimation during covariance step

SIRCENTER (nm74)
Where the sampling (proposal) density is to be centered during SIR sampling

SIRDF (nm74)
Degrees of freedom of t-distribution sampler used during SIR sampling

SIRNITER (nm74)
The number of times to perform SIR sampling

SIRPRINT (nm74)
Set the console print iterations interval during SIR sampling of covariance step

SIRSAMPLE (nm74)
Number of random samples to generate during SIR sampling of covariance step.

SIRTHBND (nm74)
Determines whether R and S matrix are evaluated in uncosntrained or constrained domain for thetas during SIR sampling

SLOW (nmvi)
Have Omega gradients evaluated numerically

SPECIAL (nmiv)
The special computation will be used in the Covariance Step with a recursive PRED subroutine.

THBND (nm74)
Determines whether R and S matrix are evaluated in uncosntrained or constrained domain for thetas during main covariance step

TOL (nm72)
Selects relative tolerance for differential equation integration during covariance step

UNCONDITIONAL (nmiv)
Evaluate covaraince step whether or not estimation successful

References: Introduction to NONMEM 7

Chapter 13 - Errors in NONMEM Runs

1. What This Chapter is About

This chapter describes error messages that can appear in NONMEM's output and discusses some possible causes and remedies. It is not encyclopedic; only selected messages are discussed. NM-TRAN messages are meant to be self-explanatory, as are many PREDPP messages, and important NONMEM messages are documented in NONMEM Users Guide, Part I, Chapter G. Unlike certain other regression programs, NONMEM-PREDPP will not try to mask what is perceived as a real problem and to which attention must be given by the user before the computation can proceed; an error message results and often, the program terminates.

The Estimation and Covariance Steps do not always terminate successfully. This is a normal part of the process of model building.

2. Abnormal Termination of the Estimation Step

Normal termination of the Estimation Step is indicated by the message:

```
MINIMIZATION SUCCESSFUL
```

Even when this message is seen, it is possible that the Estimation Step has not run correctly. Final estimates should be different from initial estimates. If the initial and final estimates are the same and the gradients for a parameter are zero at every iteration†, this is a sign of a modelling error. The parameter does not affect any predictions, as discussed in Chapter 7, Section 4.2. If there were bounds, estimates should be well away from the bounds. A final estimate which is close to a bound is discussed in Chapter 11, Section 4.3.

Abnormal termination of the Estimation Step is indicated by a message whose first line is:

```
MINIMIZATION TERMINATED
```

One of several messages will follow, indicating the type of failure. The messages are listed in Users Guide I.

Two of the most common are discussed here separately.

2.1. "DUE TO MAX. NO. OF FUNCTION EVALUATIONS EXCEEDED"

If after any iteration the total number of evaluations of the objective function (Chapter 10, figure 10.5, line 33) is equal to or greater than the maximum allowed (Chapter 10, figure 10.2, line 51), the minimization search is halted with this message. If the $ESTIMATION record requested that a Model Specification File be written, it is possible to continue the search from this point in a subsequent NONMEM run. In Chapter 12, Section 4.3, a strategy is discussed by which the number of function evaluations is deliberately set to a low value in order to structure a lengthy run as a series of shorter runs.

Whenever this message is obtained, it is important to examine the intermediate output and evaluate the progress made so far. A poorly-specified model, for example, may cause very slow convergence of the minimization search. Raising the maximum number of function evaluations (using the MAXEVAL option of the $ESTIMATION record) may not be advisable.

---

† A gradient may reach zero at or near the final iteration; this is not an error. Note also that no gradient is computed or printed for a parameter whose value is fixed, so if a gradient is always zero, it is not for this reason.

2.2. "DUE TO ROUNDING ERRORS (ERROR=134)"

This message will be accompanied in the intermediate output by a message beginning: NUMSIGDIG: .... which gives the approximate number of significant digits obtained in each of the parameters being estimated. At least one such number will be less than the number requested.

The number of significant digits obtained should be examined. If it is at least 2, and the gradient vector appears stable throughout the last few iterations, a satisfactory minimum may well have been obtained. (It may be desirable to re-run the problem with the print interval for iteration summarization set to 1 (PRINT=1 in the $ESTIMATION record) so that the progress made at every iteration can be examined.) The final parameter estimates should be examined, and if they appear reasonable, they might be accepted. Although the user may have requested more than 2 significant digits, the data may only support about 2 digits, given the precision of the arithmetic being used. By examining the gradients carefully, it is often possible to obtain further information about which parameter estimates are less-well determined.

Even though the final parameter estimates may be adequate, it is unlikely that the minimum is sufficiently well-determined to allow the Covariance Step to run successfully, at least with the number of significant figures requested in the Estimation Step. The Estimation Step may need to be rerun, requesting only 2 significant figures, followed by the Covariance Step.

If the number of significant digits is less than 2 (or even negative), then the final estimates should not be trusted. The problem may be model misspecification or insufficient data.

Model misspecification is a very general problem involving some mismatch between the model and the data. This can result in particularly large values of the objective function or slow convergence of the minimization search. Sometimes the model is overparameterized. This means that the model has more parameters than can be well-enough estimated from the data (e.g., a biexponential model is fit to monoexponential data). When model misspecification occurs due to over-parameterization, then the Estimation Step will usually proceed smoothly, but terminate with fewer than 2 significant digits. It is best to start with simple models (see Chapter 11).

A related problem arises when a covariance element, e.g., Ω_{12} (or Σ_{12}), is being estimated. If the ID (or L2 data item) is not used correctly, it may appear as though the covariance does not affect objective function values, and then this parameter will not be well estimated. In other words, it may appear as though the model is overparameterized due to the inclusion of this parameter. See Chapter 12, Section 4.2.

3. Abnormal Termination of the Covariance Step

It is possible for the Estimation Step to terminate successfully, and yet the Covariance Step generates an error message. Error messages from the Covariance Step are printed immediately after line 46 of Figure 10.5. The messages are listed in Users Guide I.

When a message arises, often it is:

```
R MATRIX ALGORITHMICALLY NON-POSITIVE-SEMIDEFINITE
BUT NONSINGULAR
COVARIANCE STEP ABORTED
```

In order for the minimization routine to terminate successfully, it needs to determine that the final gradient vector is sufficiently small, which is a necessary condition for a minimum to have been achieved. This condition, however, is not sufficient. A sufficient

condition, that the R matrix be positive definite (and therefore, that the apparent minimum not be a saddle point) is only checked in the Covariance Step. The message means that the sufficient condition appears not to be satisfied. The final estimate is, therefore, in doubt.

Sometimes the message is:

```
R MATRIX ALGORITHMICALLY SINGULAR
COVARIANCE STEP UNOBTAINABLE
S MATRIX ALGORITHMICALLY SINGULAR
```

This arises when there exists a parameter whose values do not actually affect the predictions and whose gradient in the intermediate output is always 0.

In general, successful completion of the Covariance Step requires a better defined minimum than does the successful completion of the Estimation Step.

4. Miscellaneous Problems

This section discusses a few simple errors which prevent NONMEM-PREDPP from running successfully.

4.1. Proportional Error Model

A common error is to use the proportional error model while some predicted values for actual observations are zero or close to zero. (For example, if the first dose is an infusion and there is a "baseline" observation at the start of the infusion, the predicted level will be zero.)

With individual data this will lead to an error message similar to the following (the individual number may be different from 1):

```
PROGRAM TERMINATED BY OBJ, ERROR IN ELS
 WITH INDIVIDUAL    1 (IN INDIVIDUAL RECORD ORDERING)
 VAR-COV OF DATA FROM INDIVIDUAL RECORD ESTIMATED TO BE  SINGULAR
```

With population data this will lead to an error message similar to the following (the individual and record numbers may be different than 1):

```
 PROGRAM TERMINATED BY OBJ, ERROR IN CELS
 WITH INDIVIDUAL    1 (IN INDIVIDUAL RECORD ORDERING)
 INTRAINDIVIDUAL VARIANCE OF DATA FROM OBS RECORD  1 ESTIMATED TO BE 0
```

4.2. Errors in the Pharmacokinetic Model

When using a new model, a run should done in which the Estimation Step is not run, and a scatterplot of PRED vs DV with unit slope line is produced, to verify that the model and the initial parameter estimates are reasonable. It is much harder to diagnose errors in the model or the initial estimates after the Estimation Step fails. Make sure that the initial value of the objective function is not excessively large, and that the unit slope line shows on the plot: scaling errors can easily go undetected! E.g., if the units are incorrect at some point in the model (L vs ml), the predictions may be wrong by a factor of 1000. Similarly, if no scale was specified for the compartment being observed, the predictions for the observations are compartment amounts rather than concentrations. In both cases, the shape of the PRED vs DV plot may appear linear, but the axes may be labeled quite differently. When observations from two different compartments are present in the data (e.g., C_p and C_u), some of the observations may be scaled incorrectly. This is discussed

in Chapter 6, Section 9, and Chapter 7, Section 4.3.3.

5. Errors with PREDPP

5.1. Error Messages from a TRANS Routine

TRANS routines can produce error messages. Here is one from TRANS2:

ERROR IN TRANS2 ROUTINE: V IS ZERO

Appendix 2 shows that TRANS2 normally computes `K=CL/V`. The routine checks that V is not zero, and upon finding that it is, it prints the informative message, and terminates the run (thus avoiding a machine "division by zero" interrupt by the operating system). This error usually occurs with the initial parameter estimates. E.g., suppose the relevant $PK statement is:

```
V=THETA(1)+WT*THETA(2)
```

For some values of θ_1, θ_2, and WT, a value of zero is being computed for V. The initial estimates of θ_1 and θ_2 should be checked. The intercept θ_1 might have been fixed to zero, in which case then, the values of WT should also be checked. If WT is recorded only on the first event record of each individual's data, careful coding is required to insure that a value of zero is not used when the $PK record is evaluated with subsequent event records.

5.2. Error Messages from ADVAN Routines

A similar error message can be generated in PREDPP, e.g.

PK PARAMETER FOR OBSERVATION COMPARTMENT'S SCALE IS ZERO

Some scale parameter is modeled in such a way as to produce a zero. Again, the code for that scale parameter, and the initial estimates for the θ's used therein, should be checked. Perhaps the scale parameter is being set equal to a volume parameter, and as described above, the volume parameter is being set to zero. When TRANS1 is used, the volume parameter is neither recognized nor checked.

5.3. Numeric difficulties in PREDPP

Numeric difficulties can occur with linear pharmacokinetic models (e.g., ADVAN1-4) in the process of computing certain exponentials. They can occur from an error in the units of either a rate constant and/or the TIME data items. They can also occur from inordinately large values for a rate constant which arise during the minimization search. This might be avoided by placing appropriate constraints on θ's.

They can also occur when the system is advanced over an excessively long period of time. This can happen within an individual record, when the individual had a course of drug treatment, followed by a wash-out period, followed by another course of drug treatment. The first dose record of treatment courses other than the first should have EVID data item equal to 4 (reset-dose) rather than 1 (dose), to avoid computing excessively small compartment amounts (see Chapter 6, Section 7.3), and to reduce computational cost.

Difficulties can occur in the process of computing predictions with ADVAN2 and ADVAN4 when values of KA and K arise during the minimization search that are very close to one another. The models encoded into the ADVAN routines assume that KA and K have fairly distinct values, and the formulas for the predictions have the term KA-K in the denominator. If, for example, the typical values of K and KA are associated with θ_1 and θ_2, respectively, then one might try reparameterizing. The typical values of K and KA-K can be associated with θ_1 and θ_2, so that $\tilde{K} = \theta_1$ and $\tilde{KA} = \tilde{K} + \theta_2$. A lower bound

of zero should be placed on θ_2†.

A similar situation occurs with TRANS3, where VSS-V occurs in the denominator of the expression for K21. As above, reparameterization and a constraint on an element of θ may help.

† This technique also prevents a "flip-flop" from occurring. (In the original parameterization, the final estimates of θ_1 and θ_2 can be the final estimates of the typical values of KA and K, respectively).

Appendix 1 - Standard Pharmacokinetic Models and Parameters

ADVAN	Compartments	Basic and additional PK parameters	
ADVAN1	1 = Central	K	Rate constant of elimination
	2 = Output	S1	Scale for central compartment
		S2	Scale for output compartment
		F1	Bioavailability for central compartment
		F0	Output Fraction
ADVAN2	1 = Depot	KA	Absorption rate constant
	2 = Central	K	Rate constant of elimination
	3 = Output	S1	Scale for depot compartment
		S2	Scale for central compartment
		S3	Scale for output compartment
		F1	Bioavailability for depot compartment
		F2	Bioavailability for central compartment
		F0	Output Fraction
ADVAN3	1 = Central	K	Rate constant of elimination
	2 = Peripheral	K12	Rate constant from central to peripheral
	3 = Output	K21	Rate constant from peripheral to central
		S1	Scale for central compartment
		S2	Scale for peripheral compartment
		S3	Scale for output compartment
		F1	Bioavailability for central compartment
		F2	Bioavailability for peripheral compartment
		F0	Output Fraction
ADVAN4	1 = Depot	KA	Absorption rate constant
	2 = Central	K	Rate constant of elimination
	3 = Peripheral	K23	Rate constant from central to peripheral
	4 = Output	K32	Rate constant from peripheral to central
		S1	Scale for depot compartment
		S2	Scale for central compartment
		S3	Scale for peripheral compartment
		S4	Scale for output compartment
		F1	Bioavailability for depot compartment
		F2	Bioavailability for central compartment
		F3	Bioavailability for peripheral compartment
		F0	Output Fraction
ADVAN10	1 = Central	VM	Maximum Rate
	2 = Output	KM	Michaelis Constant
		S1	Scale for central compartment
		S2	Scale for output compartment
		F1	Bioavailability for central compartment
		F0	Output Fraction

ADVAN	Compartments	Basic and additional PK parameters	
ADVAN11	1 = Central 2 = Peripheral 1 3 = Peripheral 2 4 = Output	K	Rate constant of elimination
		K12	Rate constant from central to periph. 1
		K21	Rate constant from periph. 1 to central
		K13	Rate constant from central to periph. 2
		K31	Rate constant from periph. 2 to central
		S1	Scale for central compartment
		S2	Scale for peripheral 1 compartment
		S3	Scale for peripheral 2 compartment
		S4	Scale for output compartment
		F1	Bioavailability for central compartment
		F2	Bioavailability for periph. 1 compartment
		F3	Bioavailability for periph. 2 compartment
		F0	Output Fraction
ADVAN12	1 = Depot 2 = Central 3 = Peripheral 1 4 = Peripheral 2 5 = Output	KA	Absorption rate constant
		K	Rate constant of elimination
		K23	Rate constant from central to periph. 1
		K32	Rate constant from periph. 1 to central
		K24	Rate constant from central to periph. 2
		K42	Rate constant from periph. 2 to central
		S1	Scale for depot compartment
		S2	Scale for central compartment
		S3	Scale for peripheral 1 compartment
		S4	Scale for peripheral 2 compartment
		S5	Scale for output compartment
		F1	Bioavailability for depot compartment
		F2	Bioavailability for central compartment
		F3	Bioavailability for periph. 1 compartment
		F4	Bioavailability for periph. 2 compartment
		F0	Output Fraction

Appendix 2 - Alternative Parameterizations

Alternative Parameters		Reparameterization Lines
ADVAN1	**TRANS2**	
CL	Clearance	K=CL/V
V	Volume of distribution	
ADVAN2	**TRANS2**	
CL	Clearance	K=CL/V
V	Volume of distribution	KA=KA
KA	Absorption rate	
ADVAN3	**TRANS3**	
CL	Clearance	K=CL/V
V	Central Volume	K12=Q/V
Q	Intercompartmental clearance	K21=Q/(VSS-V)
VSS	Volume of distribution at steady state	
ADVAN3	**TRANS4**	
CL	Clearance	K=CL/V1
V1	Central volume	K12=Q/V1
Q	Intercompartmental clearance	K21=Q/V2
V2	Peripheral volume	
ADVAN3	**TRANS5**	
AOB	A/B	K21=(AOB*BETA+ALPHA)/(AOB+1)
ALPHA	alpha	K=ALPHA*BETA/K21
BETA	beta	K12=ALPHA+BETA-K21-K
ADVAN3	**TRANS6**	
ALPHA	alpha	K=ALPHA*BETA/K21
BETA	beta	K12=ALPHA+BETA-K21-K
K21	Rate constant from periph. to central	K21=K21
ADVAN4	**TRANS3**	
CL	Clearance	K=CL/V
V	Central Volume	K23=Q/V
Q	Intercompartmental clearance	K32=Q/(VSS-V)
VSS	Volume of distribution at steady state	KA=KA
KA	Absorption rate	
ADVAN4	**TRANS4**	
CL	Clearance	K=CL/V2
V2	Central volume	K23=Q/V2
Q	Intercompartmental clearance	K32=Q/V3
V3	Peripheral volume	KA=KA
KA	Absorption rate	

ADVAN4	**TRANS5**	
AOB	A/B	K32=(AOB*BETA+ALPHA)/(AOB+1)
ALPHA	alpha	K=ALPHA*BETA/K32
BETA	beta	K23=ALPHA+BETA-K32-K
KA	Absorption rate	KA=KA
ADVAN4	**TRANS6**	
ALPHA	alpha	K=ALPHA*BETA/K32
BETA	beta	K23=ALPHA+BETA-K32-K
K32	Rate constant from periph. to central	K32=K32
KA	Absorption rate	KA=KA
ADVAN11	**TRANS4**	
CL	Clearance	K=CL/V1
V1	Central volume	K12=Q2/V1
Q2	Intercompartmental clearance 1	K21=Q2/V2
V2	Peripheral volume 1	K13=Q3/V1
Q3	Intercompartmental clearance 2	K31=Q3/V3
V3	Peripheral volume 2	V3=V3
ADVAN11	**TRANS6**	
ALPHA	alpha	K=ALPHA*BETA*GAMMA/(K21*K31)
BETA	beta	V1=ALPHA+BETA+GAMMA
GAMMA	gamma	V2=ALPHA*BETA+ALPHA*GAMMA
K21	Rate constant from periph. 1 to central	+BETA*GAMMA
K31	Rate constant from periph. 2 to central	K13=(V2+K31*K31-K31*V1-K*K21)/(K21-K31)
		K12=V1-K-K13-K21-K31
ADVAN12	**TRANS4**	
CL	Clearance	K=CL/V2
V2	Central volume	K23=Q3/V2
Q3	Intercompartmental clearance 1	K32=Q3/V3
V3	Peripheral volume 1	K24=Q4/V2
Q4	Intercompartmental clearance 2	K42=Q4/V4
V4	Peripheral volume 2	V4=V4
KA	Absorption rate	KA=KA
ADVAN12	**TRANS6**	
ALPHA	alpha	K=ALPHA*BETA*GAMMA/(K32*K42)
BETA	beta	V2=ALPHA+BETA+GAMMA
GAMMA	gamma	V3=ALPHA*BETA+ALPHA*GAMMA
K32	Rate constant from periph. 1 to central	+BETA*GAMMA
K42	Rate constant from periph. 2 to central	K24=(V3+K42*K42-K42*V2-K*K32)/(K32-K42)
		K23=V2-K-K24-K32-K42
KA	Absorption rate	KA=KA

Appendix 3 - NM-TRAN Control Records

The following is an alphabetic list of NM-TRAN control records.

See Guide VIII, On-line Help, or On-line HTML for the options and for more information. See Appendix 4 for the corresponding NONMEM control records (FCON).

$ABBREVIATED
$AES
$AESINIT
$ANNEAL
$BIND
$CHAIN
$CONTR
$COVARIANCE
$DATA
$DEFAULTS
$DES
$ERROR
$ESTIMATION
$ETAS
$INDEX
$INFN
$INPUT
$LEVEL
$MIX
$MODEL
$MSFI
$NONPARAMETRIC
$OMEGA
$OMEGAP
$OMEGAPD
$OMIT
$PHIS
$PK
$PRED
$PRIOR
$PROBLEM
$SCATTER
$SIGMA
$SIGMAP
$SIGMAPD
$SIMULATION
$SIZES
$SUBROUTINES
$SUPER
$TABLE
$THETA
$THETAI
$THETAP

$THETAPV
$THETAR
$TOL
$WARNING

Appendix 4 - NONMEM Control Records (FCON)

The following is a list of NONMEM control records and options. These are generated by NM-TRAN in a file called FCON. They are listed in the order that they appear in FCON.

Records marked with * may be continued. The record name, e.g., "INDX", is not repeated on continuation(s). Any time a field may contain 0, it may contain blanks instead, which are read as 0. Constants SD, PD, LVR are from SIZES. Constants INF, INTBIG, INTSMALL indicate the largest floating point number, the largest integer value, and smallest integer value, respectively, that can be represented in the computer's architecture. (In some cases, e.g., BOOTSTRAP option of SIMULATION, it means an integer of up to 11 digits).

FILE record **(FILE) (A4,4X,A72)**

Field No.	Value	Function
1	NULL	no file stream
	72 chars	name of file stream

SUPER record **(SUPR) (A4,4X,I4,I8,I4)**

Field No.	Value	Function
1	1-9999	Number of problems in the superproblem
2	2-9999	Number of iterations of the superproblem.
3	0	Input information will be printed for first problem only
	1	Input information will be printed for all problems

PROBLEM record **(PROB) (A4,4X,A72)**

Field No.	Value	Function
1	72 chars	problem heading

DATA record **(DATA) (A4,4X,18I4)**

Field No.	Value	Function
1	0 or blank	data set is embedded in the control stream
	1	data set is in a separate file
	-1	re-use the data set from the previous problem.
2	0 or blank	FORTRAN unit not to be rewound
	1	FORTRAN unit to be rewound
3	0	data set to be read to FINISH record or end of file
	1-9999	no. of data records (low-order digits)
4	1-PD	no. of data items per data record
5	0	not data checkout
	1	data checkout only
6	0-9999	no. of data records (high-order digits)
		The no. of data records is Field 6 * 10000 + Field 3.
		When Field 6 is 0 or blank, this is simply Field 3
7	0	Simulation NOREWIND from $SIMULATION record
	1	Simulation REWIND from $SIMULATION record
8	0	NOSUPRESET
	1	SUPRESET (default)

ITEM record **(ITEM) (A4,4X,18I4)**

Field No.	Value	Function
1	0-PD	index of ID data item
2	1-PD	index of DV data item
3	0-PD	index of MDV data item
4	0-PD	no. of data item indices in INDXS
5	0	no user-supplied labels.
	1	user-supplied labels.
6	0	standard labels PRED,RES and WRES used.
	1	nonstandard labels used.
7	0-PD	index of L2 data item
8	0-PD	index of first data item specified in CONTR record
9	0-PD	index of second data item specified in CONTR record
10	0-PD	index of third data item specified in CONTR record
11	0-50	no. of user-supplied labels for tables, scatters
12	0-PD	index of MRG_ data item
13	0-PD	index of RAW_ data item
14	0-PD	no. of items on OMIT record
15	0-PD	index of RPT_ data item

INDEX record **(INDX)* (A4,4X,18I4)**

Field No.	Value	Function
1	1-PD	1st element of INDXS
2	1-PD	2nd element of INDXS
	etc.	

XVID record **(XVID) (A4,4X,18I4)**

Field No.	Value	Function
1	0-PD	XVID1
2	0-PD	XVID2
3	0-PD	XVID3
4	0-PD	XVID4
5	0-PD	XVID5

MSDT record **(MSDT) (A4,4X,20(1PE22.14E3,1X)**

Field No.	Value	Function
1	-INF-INF	MISDAT(1)
2	-INF-INF	MISDAT(2)
20	-INF-INF	MISDAT(20)

LABEL record **(LABL)* (A4,X2,A74)**

The LABL record contains a comma-delimited list of labels, beginning at position 6, with an unlimited number of continuation records. Each label is right-adjusted in a field of SD characters. By default (i.e., with

SD=20 in SIZES), there are 3 labels per line. The order is as follows:

Field No.	Function
1:	label of 1st data item
2:	label of 2nd data item
	etc.
m:	label of last data item
m+1:	label for PRED (if ITEM(6)=1)
m+2:	label for RES (if ITEM(6)=1)
m+3:	label for WRES (if ITEM(6)=1)
m+p+1:	label for 1st variable in NMPRD4†
m+p+2:	label for 2nd variable in NMPRD4† , etc.
m+p+q:	label for last displayed variable in NMPRD4

Note
m=no. of data items per data rec.=DATA(4)
p=3 if non-standard labels for PRED, RES, WRES (ITEM(6)=1)
p=0 otherwise
q=no. of user supplied labels for tables, scatters=ITEM(11)
† Blank if this variable is not displayed

Additional LABEL record (LBW1)* (A4,X2,A74)

The LBW1 record contains a comma-delimited list of labels for the additional diagnostic items, starting at position 6 in each line. The format is similar to that of the LABL record, but leading spaces are omitted. The default labels are as follows:

```
IWRS,IPRD,IRS
NPRED,NRES,NWRES
NIWRES,NIPRED,NIRES
CPRED,CRES,CWRES
CIWRES,CIPRED,CIRES
PREDI,RESI,WRESI
IWRESI,IPREDI,IRESI
CPREDI,CRESI,CWRESI
CIWRESI,CIPREDI,CIRESI
EPRED,ERES,EWRES
EIWRES,EIPRED,EIRES
NPDE,ECWRES,NPD
OBJI
```

LABEL record for THETA (LTHT)* (A4,X4,A72)
LABEL record for ETA (LETA)* (A4,X4,A72)
LABEL record for EPS (LEPS)* (A4,X4,A72)
LABEL record for RESIDUAL LABEL (LRES) (A4,X1,A5)

Symbolic names for elements of THETA, ETA, and EPS (respectively), for NONMEM to use in the report

file (*label substitution*). If label substitution is not requested, the LTHT, LETA, and LEPS records are optional and (if present) should have blanks starting in position 9.

Example: Suppose the NM-TRAN control file contains

```
$ABBR REPLACE THETA(KA,K,CL)=THETA(1 TO 3)
$ABBR REPLACE ETA(CL)=ETA(3),ETA(V)=ETA(5)
```

Then the generated LABEL records are:

```
LTHT     1=THETA(KA),2=THETA(K),3=THETA(CL)
LETA     3=ETA(CL),5=ETA(V)
LEPS
```

OMIT record **(OMIT)* (A4,4X,18I4)**

Field No.	Value	Function
1	4 chars	no. of 1st data item omitted from template matching
2	4 chars	no. of 2nd data item omitted from template matching
	etc.	

FORMAT record **(FORM) (A4,4X,A72/A80)**

Field No.	Value	Function
1	80 chars	format specification
		(field begins on first continuation record)

FIND record **(FIND) (A4,4X,18I4)**

Field No.	Value	Function
1	0	
2	0	
3	0	No Model specification file (MSFI)
	1	A Model specification file (MSFI) is to be read.
4	0	estimate on file not to be rescaled.
	1	estimate on file to be rescaled.
5	0	No ONLYREAD option
	1	ONLYREAD option
6	0	MSFTEST option (default)
	1	NOMSFTEST option
7	0	MSFI not new (default)
	1	MSFI new

MSF Version Record **(MSFV) (A4,X4,A72)**

Right after the FIND record for MSFI, if it exists, the MSFV (NM74) contains the MSF version, starting at position 9. This will be blank if not specified explicitly. Example:

```
MSFV   7.2.0
```

INITIAL VALUES record for ETA **(ETA)* (A4,I4,(comma-delimited list))**

INITIAL VALUES record for PHI (PHI)* (A4,I4,(comma-delimited list))

With FILE specified:

Line	Field No.	Value	Function
1	1	0	indicates file name is given.
1	2		File name starting in position 9 (through 88 max)

With no FILE specified:

Line	Field No.	Value	Function
1	1	1-LVR	number of initial values for etas or phis. Listed starting at position 9 of each line.

Initial STRUCTURE record (STRC) (A4,4X,18I4)

Field No.	Value	Function
1	0-999	length of THETA
2	0-999	dimension of OMEGA
3	0-999	dimension of SIGMA
4	blank	
5	blank	
6	0 or blank	OMEGA constrained with a block set partition
	1	OMEGA constrained to be diagonal
7	0 or blank	only if field 6 has value 1
	1-999	number of block sets for OMEGA

If the dimension of SIGMA is 0, the following fields may be ignored.

8	0 or blank	SIGMA constrained with a block set partition
	1	SIGMA constrained to be diagonal
9	0 or blank	SIGMA only if field 8 has value 1
	1-999	number of block sets for SIGMA
10	blank	
11	blank	
12	0 or blank	default THETA boundary test
	1	No default THETA boundary test
13	0 or blank	default OMEGA boundary test
	1	No default OMEGA boundary test
14	0 or blank	default SIGMA boundary test
	1	No default SIGMA boundary test

STRUCTURE record for OMEGA (STRC)* (A4,4X,18I4)
STRUCTURE record for SIGMA (STRC)* (A4,4X,18I4)

Field No.	Value	Function
1	1-999	size of 1st. block set

2	1-999	dimension of blocks in 1st. block set
3	1-999	size of 2nd. block set
4	1-999	dimension of blocks in 2nd. block set
		etc.

THETA CONSTRAINT record (THCN) (A4,4X,18I4)

Field No.	Value	Function
1	0 or blank	THETA unconstrained
	1	THETA constrained
2	0 or blank	use default size of initial. est. search
	1-9999	no. of points to be examined during initial est. search.
3	0 or blank	ABORT if PRED sets error return code to 1 during search
	1	NOABORT - Ignore PRED error return code during search
	2	NOABORTFIRST - Same, even with first values.

THETA record (THTA)* (A4,4X,(comma-delimited list))

Field No.	Value	Function
1		initial est. of θ_1
		(blank if NONMEM is to obtain the inital est.)
2		initial est. of θ_2
		(blank if NONMEM is to obtain the inital est.)
	etc.	

LOWER BOUND record (LOWR)* (A4,4X,(comma-delimited list))

Field No.	Value	Function
1		lower bound for θ_1
2		lower bound for θ_2
	etc.	

UPPER BOUND record (UPPR)* (A4,4X,(comma-delimited list))

Field No.	Value	Function
1		upper bound for θ_1
2		upper bound for θ_2
	etc.	

DIAGONAL record (DIAG)* (A4,1X,A1,1X,A1,(comma-delimited list))

for OMEGA or SIGMA

Field No.	Value	Function
Pos. 1	0	Diagonals Variance
	1	Diagonals standard deviation (STANDARD)
Pos. 2	blank	Not fixed.
	1	Fixed.

	2	NONMEM is to obtain the inital estimate(s).
1		initial est. of (1,1) element of matrix
2		initial est. of (2,2) element of matrix
	etc.	

BLOCK SET record (BLST)* (A4,1X,A1,1X,A1,(comma-delimited list))

for OMEGA or SIGMA

Field No.	Value	Function
Pos. 1	blank	Diagonals Variance, Off-diagonals covariance
	1	Diagonals standard deviation, Off-diagonals covariance (STANDARD)
	2	Diagonals Variance, Off-diagonals correlation (CORRELATION)
	3	Diagonals standard deviation, Off-diagonals correlation (STANDARD CORRELATION)
	4	Cholesky format (CHOLESKY)
Pos. 2	blank	Not fixed.
	1	Fixed.
	2	NONMEM is to obtain the inital estimate(s).
1		initial est. of (1,1) element of matrix
2		initial est. of (1,2) element of matrix
	etc.	
		use symmetric enumeration

SIMLUATION record (SIML) (A4,4X,I2,I3,I2,I13,5I2,I13,I6,I6,1X,A16)

Field No.	Value	Function
1	0 or blank	Simulation Step implemented
	1	Simulation Step not implemented

If the value is 1, the subsequent fields may be ignored.

2	1-10	no. of random sources (SORC records)
3	0	eta (eps) changes with each record
	1	eta (eps) changes with new ind.rec. (L2 rec) (NEW)
4	0-9999	no. of subproblems
5	0	compute objective function and other steps
	1	only the simulation step
6	0 or blank	no partial derivatives from PRED needed
	1	PRED should compute 1st. derivatives (REQUESTFIRST)
	2	PRED should compute 2nd. derivatives (REQUESTSECOND)
7	0 or blank	simulated observation is Y or F (PREDICTION)
	1	simulated observation is DV (NOPREDICTION)
8	0 or blank	Use inital ests. (TRUE=INITIAL)
	1	with MSFI, use final ests. (TRUE=FINAL)
	2	use values in THET_P, OMEG_P, SIGM_P set by the PRIOR routine (TRUE=PRIOR)
9	0	REPLACE
	1	NOREPLACE
10	-1	BOOTSTRAP using as many subjects as are in the data set
	0	No BOOTSTRAP (the default)
	1-INTBIG	BOOTSTRAP using the given number of subjects
11	0-PD	STRAT data column number
12	0-PD	STRATF data column number

13	ALPHA	RANMETHOD

ADDITIONAL RECORDS FOR SIML (A4,4X,I12,A)

Record	Function
SFIL	PARAFPRINT (0-INTBIG),PARAFILE (line may accomodate up to 80 character file for total of 88 characters)

SOURCE record (SORC) (A4,4X,2A12,I4)

Field No.	Value	Function
1	-1-21474836447	first seed
2	0-21474836447	second seed
3	0 or blank	random numbers are pseudo-normal (NORMAL)
	1	random numbers are pseudo-uniform (UNIFORM)
	2	random numbers are from a nonpar. distrib (NONPARAMETRIC)

DEFAULT record (DFLT) (A4,4X,I4)

Field No.	Value	Function
1	-1-1	NOSUB option of DEFAULT record

CHAIN record (CHN)

Line	Format	Field No.	Value	Function
1	(A4,4I12,F12.5,I12)	1	0-4	CTYPE
		2	INTSMALL-INTBIG	SEED
		3	INTSMALL-INTBIG	ISAMPLE
		4	0-INTBIG	NSAMPLE
		5	0.0001-1.0	IACCEPT
		6	0-INTBIG	DF
2	(4X,4I12,A12)	1	INTSMALL-INTBIG	ISAMPEND
		2	0-3	SELECT
		3	0-3	NOTITLE(1,3),NOLABEL(2,3)
		4	-1-INTBIG	DFS
		5	ALPHA	RANMETHOD

ADDITIONAL RECORDS FOR CHAIN (A4,4X,A)

Record	Function
CFIL	FILE (line may accomodate up to 80 character file for total of 88 characters)
CDLM	FORMAT
ORDR	ORDER
CHFL	PARAFPRINT (I12:0-INTBIG),PARAFILE (line may accomodate up to 80 character file for total of 88 characters)

LEVEL record (OLEV)* (A4,X4,A20,A52)

Field No. Value Function
1 ALPHA Data item type
2 ALPHA level description

LEVEL rec. contination rec. (OLEV)* (A4,X4,X20,A52)

Field No. Value Function
1 blank
2 ALPHA level description (continues level description from previous record)

ANNEAL record (ANNL) (A4,X4,A6,A6)

Field No.	Value	Function
1	1-LVR	Eta Number
2	0-INF	Starting Omega value

ESTIMATION record (ESTM) (A4,4X,18I4)

Field No.	Value	Function
1	0 or blank	Estimation Step implemented
	1	Estimation Step not implemented

If the value is 1, the subsequent fields may be ignored.

Field No.	Value	Function
2	0-9999	maximum no. of function. evaluations (low-order digits)
	-1	Reuse the value from the previous run (with MSFI)
3	1-8	number of significant figs. required in final est.
4	0 or blank	no summarization of iterations
	n>0	every nth iteration summarized
5	0 or blank	no second search (REPEAT)
	1	second search (REPEAT) implemented
6	0 or blank	MSF not output
	1	MSF output
7	0 or blank	First order (FO) method
	1	Conditional method (METHOD=COND)
8	0 or blank	No POSTHOC etas are to be estimated.
	1	POSTHOC etas are to be estimated.
9	0 or blank	Etas are 0 for comp. of intraind. error (NOINTERACTION)

	1	Nonzero etas for comp. of intraind. error INTERACTION
10	0 or blank	Do not use Laplacian method.
	1	Laplacian method is to be used.
11	0 or blank	ABORT if PRED sets error return code to 1
	1	NOABORT - Attempt theta-recovery when PRED error code 1.
	2	NOHABORT - Attempt recovery even at first iteration
12	0 or blank	Faster method of compuation (NOSLOW)
	1	Slower method of computation (SLOW)
	2	Slower method of computation (SLOW=2); for Stieltjes
	3	Fast analytical derivative method of compuation (FOCE only)
13	0 or blank	avg. cond. est. of etas unconstrained (NOCENTER)
	1	avg. cond. est. of etas constrained close to 0. (CENTER)
14	0 or blank	First-order model not used (NOFO)
	1	First-order model used with METHOD=1 CENTERING (FO)
15	0 or blank	Second eta-derivs. computed by PRED (NONUMERICAL)
	1	Second eta-derivs. for Laplacian to be obtained numerically.
16	0 or blank	Y or F (with user-supplied code) is a prediction.
	1	Y or F is a LIKELIHOOD.
	2	Y or F is a -2LOGLIKELIHOOD
17	0 or blank	Not the Hybrid method
	1-99	no. of etas fixed to zero by ZERO recs. (Hybrid method)
18	0 or blank	Not the Stieltjes method.
	1	Stieltjes method; no GRID option.
	2	Stieltjes method; GRID was specified.

ESTIMATION rec. continuation rec. ( )(A4,4X,18I4)

Field No.	Value	Function
1	0 or blank	Required if estimation step is omitted, otherwise:
	0 or blank	The REPEAT2 option is not coded; same as NOREPEAT2
	1	REPEAT2 (with Stieltjes)
2	0 or blank	No ETABARCHECK.
	1	ETABARCHECK option is coded.
3	0 or blank.	Sum contrib. to obj. func. in data set order.
	1	Sort contrib. to obj. func. prior to sum (SORT)
4	0-9999	maximum no. of function evaluations (high-order digits)
		The no. of func. evals. is Field 4 * 10000 + low-order
		When Field 4 is 0 or blank, this is simply low-order
5	-1,0,100	SIGL default
	1-15	SIGL value
6	-1,0,100	SIGLO default
	1-15	SIGLO value

BAYES ESTIMATION record (BEST)

Line	Format	Field No.	Value	Function
1	(A4,4I12,F12.5,I12)	1	-1-16	BAYES METHOD
			<=0	FO/FOCE/Laplace

			10	DIRECT
			11	BAYES
			12	ITS
			13	SAEM
			14	IMP
			15	IMPMAP
			16	CHAIN
		2	-1-INTBIG	PSAMPLE_M1
		3	-1-INTBIG	PSAMPLE_M2
		4	0-INTBIG	PSAMPLE_M3
		5	0.0001-1.0	PACCEPT
		6	-1-INTBIG	OSAMPLE_M1
2	(X4,2I12,F12.5,3I12)	1	-1-INTBIG	OSAMPLE_M2
		2	0-INTBIG	OSAMPLE_M3
		3	0.0001-1.0	OACCEPT
		4	1-INTBIG	ISAMPLE/ICHAINS (non-CHAIN)
			INTSMALL-INTBIG	ISAMPLE (CHAIN)
		5	0-INTBIG	ISAMPLE_M1
		6	0-INTBIG	ISAMPLE_M2
3	(X4,I12,F12.5,4I12)	1	0-INTBIG	ISAMPLE_M3
		2	0.0-1.0	IACCEPT
		3	0-INTBIG	NSAMPLE/NITER
		4	0-INTBIG	NBURN
		5	0-INTBIG	DF
		6	0-3	EONLY
4	(X4,5I12,F12.5)	1	INTSMALL-INTBIG	SEED
		2	0-1	NOPRIOR
		3	0-3	NOTITLE(1,3),NOLABEL(2,3)
		4	0-4	CTYPE
		5	1-INTBIG	CITER/CNSAMP
		6	0.0000001,1	CALPHA
5	(X4,4I12,2E12.5)	1	0-INTBIG	CINTERVAL
		2	0-INTBIG	MAPITER
		3	-1-INTBIG	MAPINTER
		4	0-INTBIG	ISAMPLE_M1A
		5	0-INF	ISCALE_MIN
		6	0-INF	ISCALE_MAX
6	(X4,5I12,A12)	1	0-INTBIG	CONSTRAIN
		2	0-15	ATOL
		3	0-2	FNLETA
		4	0-2	OPTMAP
		5	0-3	ETADER
		6	ALPHA	RANMETHOD
7	(X4,5I12,E12.5)	1	0-INTBIG	MCETA
		2	0-2	NONINFETA
		3	INTSMALL-INTBIG	ISAMPEND (CHAIN)
		4	0-1	ETASTYPE
		5	0-1	AUTO
		6	0.0-1000	STDOBJ

8	(X4,I12,2E12.5,I12,I12,I6)	1	0-3	NUMDER
		2	0-INF	PSCALE_MIN
		3	0-INF	PSCALE_MAX
		4	-1-INTBIG	DFS
		5	0-3	SELECT
		6	0-1	NOCOV
9	(4X,I6,I6,I6,E12.5,E15.8,I6)	1	0-1	DERCONT
		2	-1-1	NOSUB
		3	0-2	MAPCOV
		4	0.0-1.0	IACCEPTL
		5	-10.0-INF	GRDQ
		6	0-INTBIG	ISAMPLE_M1B
10	(4X,4(I12),2(E15.8))	1	-1-1	MASSRESET
		2	0-1	BAYES_METHOD (0:BAYES, 1:NUTS)
		3	-1-INTBIG	MADAPT
		4	0-INTBIG	IMADAPT (NOT USED)
		5	0.0001-INF	KAPPA
		6	0.0001-INF	IKAPPA
11	(4X,6(E15.8))	1	0.0-INF	NUTS_GAMMA
		2	0.0-INF	IGAMMA (NOT USED)
		3	0.O-1.0	NUTS_DELTA
		4	0.0-1.0	IDELTA (NOT USED)
		5	0.0-INF	OLKJDF
		6	0.0-INF	SLKJDF
12	(4X,3(E15.8))	1	0.0-INF	TTDF
		2	-INF-INF	OVARF
		3	-INF-INF	SVARF
13	(4X,A12,A12,I4,I4,I4,I4,I4,I4)	1	ALPHA	NUTS_TYPE (NOT USED)
		2	ALPHA	NUTS_MASS
		3	0-1	NUTS_TRANSFORM
		4	0-1	INUTS_TRANSFORM (NOT USED)
		5	0-2	NUTS_EPARAM
		6	0-10	WISHTYPE (NOT USED)
		7	0-1	NUTS_OPARAM
		8	0-1	NUTS_SPARAM
14	(4X,I12,I12,I12,3(E15.8))	1	0-INTBIG	NUTS_STEPITER
		2	0-INTBIG	NUTS_STEPINTER
		3	0-1	NUTS_TEST
		4	0-INF	NUTS_INIT
		5	-99-INF	NUTS_BASE
		6	0-INF	NUTS_TERM
15	(4X,I12,I12,I12,E15.8,E15.8,I4)	1	0-1	KNUTHSUMOFF
		2	0-1	LEVWT
		3	-1-INTBIG	NUTS_MAXDEPTH
		4	0.0-INF	NUTS_CHOLBND (NOT USED)
		5	0.0-INF	NUTS_REG
		6	0-INTBIG	SADDLE_RESET
16	(4X,I12,I12,I12,I12,I12)	1	0-1	SADDLE_HESS
		2	0-INTBIG	THIN

3	0-1	ETASAMPLES
4	0-1	PHITYPE
5	0-4	LEVSWITCH (NOT USED)

ADDITIONAL RECORDS FOR ESTIMATION (A4,4X,A)

Record	Function
BFIL	FILE (name be up to 256 characters)
BDLM	FORMAT/DELIM
BMUM	MUM (may go beyond 80 characters)
BGRD	GRD (may go beyond 80 characters)
ORDR	ORDER
PFIL	PARAFPRINT (I12:0-INTBIG),PARAFILE (name be up to 256 characters)
FFIL	FPARAFPRINT (I12:0-INTBIG),FPARAFILE (name be up to 256 characters)

ZERO record (ZERO)* (A4,4X,18I4)

Field No.	Value	Function
1	0	conditional estimate for eta(1)
	1	eta(1) is fixed to 0 (HYBRID method)
2	0	conditional estimate for eta(2)
	1	eta(2) is fixed to 0 (HYBRID method)
	etc.	

GRID record (GRID) (A4,4X,9A8)

Field No.	Value	Function
1	nr	as specified in GRID=(nr,ns,r0,r1)
2	ns	as specified in GRID=(nr,ns,r0,r1)
3	r0	as specified in GRID=(nr,ns,r0,r1)
4	r1	as specified in GRID=(nr,ns,r0,r1)

NONPARAMETRIC record (NONP) (A4,4X,6I4,I12,I6,I6)

Field No.	Value	Function
1	0 or blank	Nonparametric step implemented conditionally
	1	Nonparametric step implemented unconditionally
2	0 or blank	use nonparametric estimate from input MSF
	1	recompute nonparametric estimate
3	0 or blank	obtain marginal cumulatives
	1	compute conditional nonpar. etas (CNPE ETAS)
4	0 or blank	no model specification file is output
	1	a model specification file is output
5	0,1	1=BOOTSTRAP
6	0-3	EXPAND(1,3),NPSUPPE(2,4)

7	0-INTBIG	NPSUPP(E) value
8	1-INTBIG	STRAT data column number
9	1-INTBIG	STRATF data column number

ADDITIONAL RECORD FOR NONPARAMETRIC (A4,4X,I12,A)

Record	Function
NFIL	PARAFPRINT (0-INTBIG), PARAFILE (name be up to 256 characters)

COVARIANCE record (COVR) (A4,4X,18I4)

Field No.	Value	Function
1	0 or blank	Covariance Step conditionally implemented
	1	Covariance Step unconditionally implemented
	2	Covariance Step not implemented
2	0 or blank	covariance matrix set to (R inverse) S (R inverse)
	1	covariance matrix set to R inverse
	2	covariance matrix set to S inverse
3	0 or blank	neither R nor S printed.
	1	R matrix printed
	2	S matrix printed
	3	both R and S printed
4	0 or blank	eigenvalues not printed
	1	eigenvalues printed.
5	0 or blank	default computation.
	1	Special computation with a recursive PRED subroutine.
6	0 or blank	Print Covariance Step arrays in normal format.
	1	Print Covariance Step arrays in compressed format.
7	1	
8	0 or blank	
9	0 or blank	Normal method of computation
	1	Slower method of computation (SLOW)
	3	Fast analytical derivative method of compuation (FOCE only)

Additional COVARIANCE record (COVT) (A4,4X,6I4,I12,I6,I6,I6,I6,I6,I6,I2,I2,A10)

Field No.	Value	Function
1	-1,0,100	SIGL default
	1-15	SIGL value
2	1-15	TOL
3	-1,0,100	SIGLO default
	1-15	SIGLO value
4	1-15	ATOL
5	0-1	1=NOFCOV
6	0-1	1=COVRESUME
7	0-INTBIG	SIRSAMPLE value
8	0-INTBIG	SIRNITER value

9	0-1	SIRCENTER value
10	0-INTBIG	PRECOND value
11	0-INTBIG	PFCOND value
12	0-2	PRETYPE value
13	0-1	FPOSDEF value
14	0-1	1:THBND=0, 0:THBND=1
15	0-1	1:SIRTHBND=0, 0:SIRTHBND=1
16	ALPHA	PRECONDS value

Second line of COVT Record (COVT) (8X,I8,I4,I4,E12.5,E12.5,E12.5,A16)

Field No.	Value	Function
1	0-INTBIG	SIRPRINT value
2	0-1	CHOLROFF value
3	0-1	KNUTHSUMOFF value
4	0.0-INF	SIRDF
5	0.0-1.0	IACCEPT
6	0.0-1.0	IACCEPTL
7	ALPHA	RANMETHOD

ADDITIONAL RECORDS FOR COVARIANCE (8X,I12,A)

Record	Function
CPAR	PARAFPRINT (0-INTBIG), PARAFILE (name be up to 256 characters)

Initial TABLE record (TABL) (A4,4X,18I4)

Field No.	Value	Function
1	0 or blank	Table Step conditionally implemented
	1	Table Step unconditionally implemented
	2	Table Step not implemented

If the value is 2, the next field may be ignored, and there should not appear
any individual TABLE records.

2	1-10	number of tables

ADDITIONAL RECORDS FOR TABLE (A4,4X,I12,A)

Record	Function
PPAR	PARAFPRINT (0-INTBIG),PARAFILE (name be up to 256 characters)

Individual TABLE record (TABL) (A4,1X,3A1,I4,9(I6,I2)

Field No.	Value	Function
Pos. 1	blank	no option record.
	1	an option record follows.
		(only if at least one item on the option rec. is non-blank)
1	0-PDT	number of selected data item types
2	1-9999	index of 1st selected data item type
3	0-8	sort code for data items of 1st selected type
	-1	Exclude-by item marked by -1
4	1-9999	index of 2nd selected data item type
5	0-8	sort code for data items of 2nd selected type
	etc.	

`Individual TABLE rec. contin. rec. (    )*(A4,1X,3A1,I4,9(I6,I2)`
(as needed)

Field No.	Value	Function
1	1-999	index of 9th. selected data item type
2	0	
3	1-999	index of 10th. selected data item type
4	0	
	etc.	

`Individual TABLE record option rec.(    )(A4,4X,5I4,I12,I12,A12,I2,I2,I2,I2,1X,A`

Field No.	Value	Function
1	blank	Every data record appears in the table.
	1	Only the first data rec. from each ind. rec. (FIRSTONLY)
	2	Only the last data rec. from each ind. rec. (LASTONLY)
	3	Only the first and last data rec. from each ind. rec. (FIRSTLASTONLY)
2	1	With TABLE file, no printed table (NOPRINT)
	2	With TABLE file, printed table appears in the NONMEM output.
3	0	default
	1	ONEHEADER
	4	NOTITLE
	8	NOLABEL
	5	ONEHEADER NOTITLE
	9	ONEHEADER NOLABEL
	14	NOHEADER (same as NOTITLE NOLABEL)
4	blank	The TABLE file is opened and is positioned at the start.
	1	The TABLE file is positioned at the end (FORWARD)
5	blank	DV, PRED, RES, WRES appear automatically
	1	DV, PRED, RES, WRES do not appear unless listed (NOAPPEND)
6	INTSMALL-INTBIG	SEED
7	3-INTBIG	ESAMPLE
8	ALPHA	RANMETHOD
9	0-1	WRESCHOL
10	-1-1	NOSUB
11	0	No SE to PRED item
	1	SE to PRED item

12	0-1	NPDTYPE
13	ALPHA	FORMAT (May be up to 20 characters, so total line length may be up to 89 characters)

Additional TABLE records (A4,4X,A)

Record	Function
FRML	LFORMAT (record may be longer than 80 characters)
FRMR	RFORMAT (record may be longer than 80 characters)
FETA	FIXEDETAS number list (record may be longer than 80 characters)

Initial SCATTERPLOT record (SCAT) (A4,4X,18I4)

Field No.	Value	Function
1	0 or blank	Scatterplot Step conditionally implemented
	1	Scatterplot Step unconditionally implemented
	2	Scatterplot Step not implemented

If the value is 2, the next field may be ignored, and there should not appear any individual SCATTERPLOT records.

2	1-20	number of families

Individual SCATTERPLOT record (SCAT) (A4,4X,9I8/8X,9I8)

Field No.	Value	Function
1	1-23	index of data items plotted on abcsissa axis
2	1-23	index of data items plotted on ordinate axis
3	0 or blank	a single scatterplot
	1	a one-way partitioned scatterplot
	2	a two-way partitioned scatterplot

If the value of field 3 is 0 or blank, the next two fields should be ignored.

4	1-23	index of 1st separator

If the value of field 3 is 1, the next field should be ignored.

5	1-23	index of 2nd separator
6	0 or blank	no unit slope line appears
	1	unit slope line appears
7	0-99999999	no. of the first data rec. for the scatter (FROM)
8	0-99999999	no. of the last data rec. for the scatter (TO)
9	0 or blank	a line through zero on the ordinate axis if appropriate.
	1	a line through zero on the ordinate axis. (ORD0)
	-1	no line through zero on the ordinate axis.
10	0 or blank	a line through zero on the abscissa axis if appropriate.

	1	a line through zero on the abscissa axis. (ABS0)
	-1	no line through zero on the abscissa axis.
11	0 or blank	Every data record appears in the scatter.
	1	Only the first data rec. from each ind. rec. (FIRSTONLY)
12	0 or blank	Every data record appears in the scatter
	1	Only data records with MDV=0 (OBSONLY).
12	-1-1	NOSUB

INDEX

100,1000 CMT data item 155
100,1000 PCMT data item 155
100,101 MDV data item 153
1.0E+06 140, 153
154
2000 54
-2LL option of $ESTIMATION record 152, 159

- A -

A_0,A_0FLG reserved variable 155
α-level test 48
Abbreviated Code 156
abbreviated code 5, 10
$ABBREVIATED record 148
ABORT 148
ABS0 option of $TABLE record 146
ABS 73
absorption, first-order 8
Absorption lag parameter ALAG 135
absorption rate KA 10
ACOS 73
additional dose 74
additional PK parameter 55, 72
additive error model 27, 37, 41, 85, 90
ADDL data item 135
ADVAN1 13
ADVAN2 9
ADVAN57 133
ADVAN68,13-14 133
ADVAN9_15 133
ADVAN 4, 71
advance 35, 71, 74
$AESINIT record 134
$AES record 134
AES subroutine 132
A(i) reserved variable 155
ALAGn 135
All Options for $COVARIANCE 166
All Options for $ESTIMATION 159
AMT data item 56, 58
ANALYSIS TYPE 140
$ANNEAL record 153
ANSI FORTRAN 4
APPEND option of $TABLE record 146
ASIN 73
assay 29
assay, bias of 25
ATAN 73
ATOL option of $COVARIANCE record 134, 166
ATOL option of $ESTIMATION record 134, 159
auto-correlation 142
AUTO option of $ESTIMATION record 159

- B -

basic PK parameter 55, 72
bias of assay 25
bias of estimate 43
$BIND record 136
bioavailability 59-62, 72, 79
BLANKOK option of $DATA record 156
BLOCK option of $OMEGA record 139
BLOCK option of $SIGMA record 139
BLOCK SAME(m) option of $OMEGA record 141
BLOCK SAME(m) option of $SIGMA record 141
bolus dose, instantaneous 59
bolus dose, multiple 60
bolus dose, zero-order 134-135
bootstrap 146, 153
BOOTSTRAP option of $NONPARAMETRIC record 153
BOOTSTRAP option of $SIMULATION record 153
BY option of $TABLE record 146

- C -

calendar date 66
CALL data item 58, 74, 84, 137
CALLFL reserved variable 134, 136
calling protocol phrase 136-137
call to ERROR subroutine 58, 136
call to PK subroutine 58, 136
CALPHA option of $ESTIMATION record 159
case insensitivity 155
CCONTR subroutine 149
CCV error model 27, 37, 41, 85, 90
CENTERING option of $ESTIMATION record 147, 159
central compartment 9, 71
CHAIN option of $CHAIN record 153
CHAIN option of $ESTIMATION record 153
Changes to NONMEM Outputs 154
CHECKOUT option of $DATA record 147
checkpoint-restart 142
chi-square 48
CHOLESKY option of $OMEGA record 140
CHOLESKY option of $SIGMA record 140
CHOLROFF option of $COVARIANCE record 166
CI 44, 129
CINTERVAL option of $ESTIMATION record 159
CIPRED,CIRES,CIWRES reserved label of $TABLE,$SCATTER record 154
CIPREDI,CIRESI,CIWRESI reserved label of $TABLE,$SCATTER record 154
CITER/CNSAMP option of $ESTIMATION record 159
CL, clearance 13, 24
clearance CL 13, 24
clock time 56, 65, 69
CMT data item 57, 71
CNTID reserved variable 158
Code, Abbreviated 156
coefficient of variation 27
colon ":" in II data item 69
colon ":" in TIME data item 66
COM,COMACT reserved variable 156
command, operating system 5
COMPACT 155
compartment, central 9, 71
compartment 9
compartment, default dose 9, 71-72
compartment, depot 9, 60, 72

compartment, dose 57, 59
compartment, equilibrium 133
compartment number 57, 71, 93
compartment, output 25, 57, 60, 64, 72
compartment, prediction 57
compartment, zero-out a 72
COMPRESS option of $COVARIANCE record 144, 166
COMRES,COMSAV option of $ABBREVIATED record 156
concentration 23
concentration, plasma 8, 25, 30, 35
concentration, urine 25, 30, 35, 38, 64
concomitant data 55
conditional estimate 147
conditional estimation method 147
conditional estimation method, first-order 147
CONDITIONAL option of $COVARIANCE record 143, 166
CONDITIONAL option of $ESTIMATION record 159
CONDITIONAL option of $SCATTERPLOT record 92
CONDITIONAL option of $TABLE record 92
conditional statement 77
confidence interval 44-46, 129
console control characters 153
constant coefficient of variation (CCV) error model 27, 37, 41, 85, 90
constant infusion 62
CONSTRAIN option of $ESTIMATION record 153, 159
constraint, parameter 10, 87, 114
CONSTRAINT subroutine 153
& continuation character 155
continuation line 155
control language, NONMEM 5
$CONTR record 149
CONTR subroutine 149
correlation 36
correlation matrix of estimate 92, 102
Correlation matrix OMEGA,SIGMA output 154
correlation of parameter estimate 45
correlation of residual vs y 117
CORRELATION option of $OMEGA record 140
CORRELATION option of $SIGMA record 140
CORRL2 reserved variable 142
COS 73
covariance 36, 89, 139
covariance matrix 36, 39, 103
covariance matrix, full 139
covariance matrix of estimate 92
Covariance matrix OMEGA,SIGMA output 154
COVARIANCE option of $OMEGA record 140
COVARIANCE option of $SIGMA record 140
$COVARIANCE record 10, 87, 91-92, 97, 102
covariance step 91-92, 102, 168
CPRED,CRES,CWRES reserved label of $TABLE,$SCATTER record 154
CPREDI,CRESI,CWRESI reserved label of $TABLE,$SCATTER record 154
CRIT subroutine 149
CTLO CTUP reserved variable 152
CTYPE option of $ESTIMATION record 159

- D -

DAT1,DAT2,DAT3 data item 67
data checkout 96, 147
data, effect 132, 150
data file name 52
data, individual 8, 23, 69
data item 8, 29, 50
data item, dropping 55, 67, 70
data item label 54
data item, null 50
data items, maximum number of 51, 55
data item, steady-state 58
data, population 55, 89
Data Preprocessor 156
Data Preprocessor 65
data record 50
$DATA record 52, 65, 97, 144
data set 50
data set, deleting records from 50-51
data set, sequence of 50, 52
data set, size of 50
data, single-response population 23
date, calendar 66
DATE data item 66
days 54, 67
day-time translation 65, 69
ddexpand utility 159
DECLARE option of $ABBREVIATED record 157
default dose compartment 9, 71-72
degrees of freedom 48, 141
deleting records from data set 50-51
DELIM option of $TABLE record 146
dependent variable 8-9, 23
depot compartment 9, 60, 72
DERCONT option of $ESTIMATION record 159
DERIV1 option of $ABBREVIATED record 156
DERIV2 option of $ABBREVIATED record 156
$DES record 134
DES subroutine 132
DF option of $ESTIMATION record 159
DFS option of $ESTIMATION record 159
DIAG 157
diagonal elements of OMEGA 37
DIAGONAL option of $OMEGA record 139
DIAGONAL option of $SIGMA record 139
diagonal variance-covariance matrix 89, 139
dispersion factor 141
distribution of parameter estimate 44
do directive 158
doexpand utility 158
dose, additional 74
dose amount 58
dose compartment, default 9, 71-72
dose compartment 57, 59
dose event record 56, 58
dose, implied 60
dose, infusion 59
dose, instantaneous bolus 59
dose, lagged 74
dose, multiple bolus 60
dose, multiple 12
dose-related data item 58
doses, multiple steady-state 63
dose, steady-state 60
dose, zero-order bolus 134-135
DOSTIM reserved variable 135
DOWHILE(DATA) 138
DROP option of $INPUT record 55, 67, 70, 97
dropping data item 55, 67, 70
drug level 61
duration, modeled 135
duration of infusion 59
DV data item 8, 55-56
dynamic memory allocation 154

- E -

EBV 154
EBVshrink output 154
ECWRES reserved label of $TABLE,$SCATTER record 155
effect data 132, 150
eigenvalue 133, 143
EIPRED,EIRES,EIWRES reserved label of $TABLE,$SCATTER record 155
elimination rate K 10, 24
ELS, extended least squares 42
Emax model 132
EM 7
EM method 152
enddo directive 158
end of infusion 59
EONLY option of $ESTIMATION record 159
Epectation-Maximation 7
EPRED,ERES,EWRES reserved label of $TABLE,$SCATTER record 154
ε 32
EPSshrink output 154
EPS variable 84, 89
equilibrium compartment 133
error, estimation 43
error, interindividual 36
error, intraindividual 25, 36
error message, operating system 53
error message, PREDPP 171
error message, TRANSLATOR 171
error model, additive 27, 37, 41, 85, 90
error model, CCV 27, 37, 41, 85, 90
error model, constant coefficient of variation (CCV) 27, 37, 41, 85, 90
error model 25, 84, 90
error model, exponential 28, 37, 85
error model, log-normal 28, 37, 85
error model, power function 29, 42, 85
error model, proportional 27, 37
error model, statistical 23
error, MSE mean squared 43
$ERROR record 5, 14, 55, 150
$ERROR record 84, 86
error recovery option 148
error return code 148
ERROR subroutine, call to 58, 136
ERROR subroutine 4-5, 55, 72, 132
error variance 15, 26
ERR variable 25, 84, 89
ESAMPLE option of $TABLE record 147
estimate, bias of 43
estimate, conditional 147
estimate, initial 10, 133
estimate of ETA, individual 147
estimate of theta, initial 87
estimate of variance, initial 90, 142, 150
estimate, perturbed initial 88
estimation error 43
estimation method, first-order conditional with interaction 147
estimation method, first-order 147
estimation method, laplacian 147
$ESTIMATION record 10, 87, 91, 97, 153
estimation step 91, 101, 143, 153, 168
ETABARCHECK option of $ESTIMATION record 154, 159
ETABAR output 154
ETADER option of $ESTIMATION record 159
η 25
ETA, individual estimate of 147
ETA(k:n) reserved label of $TABLE record 154
ETAn reserved label of $TABLE record 154
ETASAMPLES option of $ESTIMATION record 159
ETAS(...) 154
ETAshrink output 154
ETAS option of $NONPARAMETRIC record 152
$ETAS record 153
ETASTYPE option of $ESTIMATION record 159
ETASTYP option of $ESTIMATION record 154
ETASXI reserved variable 154
ETA variable 14, 25, 84, 89
event record, dose 56, 58
event record 56
event record, observation 56, 84
event record, other 56-57, 74
event record, reset-dose 57
event record, reset 57
EVID data item 56, 65
EVID data item, generated 69
exception, floating-point 148
EXCLUDE_BY 155
EXIT statement 148
EXPAND option of $NONPARAMETRIC record 152
expectation feature 152
experiment, replication of 43
EXP 73
exponential error model 28, 37, 85
extended least squares ELS 42
external table file 146

- F -

FAST option of $COVARIANCE record 166
FAST option of $ESTIMATION record 159
Features, NONMEM 152
F distribution 48
F_FLAG reserved variable 152
filename option of $DATA record 52
FILE option of $COVARIANCE record 166
FILE option of $ESTIMATION record 159
FILE option of $TABLE record 146
final parameter estimate 91
finedata utility 158
FINISH record 52
FIRSTLASTONLY option of $TABLE record 146
FIRSTONLY option of $TABLE record 146
first-order absorption 8
first-order conditional estimation method 147
first-order conditional with interaction estimation method 147
first-order estimation method 147
fixed effects 23, 55
fixed effects parameter 23, 33
FIXEDETAS 155
FIXED option of $OMEGA record 89, 100, 140
FIXED option of $SIGMA record 89, 100, 140
FIXED option of $THETA record 88, 100
flip-flop 172
floating-point exception 148
FNLETA option of $ESTIMATION record 159
FOCE method 91
FO method 91
FORMAT/DELIM option of $ESTIMATION record 159
FORMAT option of $COVARIANCE record 166
FORMAT option of $TABLE record 146
format specification 50, 52, 65, 69-70

FORTRAN 6, 50
FORTRAN OPEN statement 52
FORTRAN READ statement 53
FORWARD option of $TABLE record 146
FPARAFILE option of $ESTIMATION record 154, 159
FPOSDEF option of $COVARIANCE record 166
fraction, output 72, 79
FROM option of $SCATTERPLOT record 104
FSUBS 6
full covariance matrix 139
FULL 155, 157
full model 47, 118
FUNCA reserved variable 157
function of parameters 43

- G -

GAMLN 73
general mixed effects model 39
generated EVID data item 69
generated ID data item 69
generated MDV data item 69
generated subroutine 6
GG array 98
Gn1 reserved label of $TABLE record 154
goodness of fit 10, 47, 118
gradient 100, 168
GRD option of $ESTIMATION record 159
GRDQ option of $ESTIMATION record 159
GRID option of $ESTIMATION record 152, 159

- H -

half-life 43, 46
hierarchical file 51
Hn1 reserved label of $TABLE record 154
hours 54, 67
HYBRID option of $ESTIMATION record 152, 159
hyperbolic model 34
hypotheses, joint 47
hypothesis, null 46
hypothesis test 19, 46, 105

- I -

IACCEPTL option of $COVARIANCE record 166
IACCEPTL option of $ESTIMATION record 159
IACCEPT option of $COVARIANCE record 166
IACCEPT option of $ESTIMATION record 159
ICALL reserved variable 138, 157
ID data item, generated 69
ID data item 51, 55, 141, 169
identification number, patient 51
IFIRSTEM reserved variable 158
IGNORE option of $DATA record 53
II/24 54
II data item, colon ":" in 69
II data item 56, 59, 69, 135
IIDX reserved variable 158
II option of $DATA record 156
ill-formed data file 156
implied dose 60
INCLUDE record 94
index plot 107, 116
indicator variable 29, 35, 77, 81, 85-86
individual data 8, 23, 69
individual estimate of ETA 147
individual parameter estimate 147
individual record 55
INDR1,INDR2 reserved variable 158
infinite infusion 62
INFINITY 87, 97
$INFN record 138
INFN subroutine 132, 138
informative form,informative record name 141
infusion, constant 62
infusion dose 59
infusion, duration of 59
infusion, end of 59
infusion, infinite 62
infusion, multiple 61
infusion, rate of 58
Initial condition 134
initial estimate 10, 133
initial estimate of theta 87
initial estimate of variance 90, 142, 150
initial estimate, perturbed 88
initial estimate step 88, 91, 142
initialization/finalization 138
initial parameter estimate 142
$INPUT record 54, 59, 96
installation of NONMEM 2, 4
instantaneous bolus dose 59
INTERACTION method 91
INTERACTION option of $ESTIMATION record 147, 159
interactive control 153
interdose interval 59, 69, 135
INTER file 142
interindividual error 36
interindividual variability 36
interrupt 148
INT 73
intraindividual error 25, 36
intraindividual variability 25, 36
inverse covariance matrix of estimate 92
IPRD,IRS,CIWRS reserved label of $TABLE,$SCATTER record 154
IPROB reserved variable 144
IREP reserved variable 145
ISAMPEND option of $ESTIMATION record 159
ISAMPLE_M1A option of $ESTIMATION record 159
ISAMPLE_M1 option of $ESTIMATION record 159
ISAMPLE_M2 option of $ESTIMATION record 159
ISAMPLE_M3 option of $ESTIMATION record 159
ISAMPLE option of $ESTIMATION record 159
ISCALE_MAX option of $ESTIMATION record 159
ISCALE_MIN option of $ESTIMATION record 159
ISFINL reserved variable 155
I_SS 133-134, 155
ISSMOD reserved variable 133, 155
iteration 100, 168
ITS method 152

- J -

joint hypotheses 47

- K -

KA, absorption rate 10
KAPPA option of $ESTIMATION record 159
K, elimination rate 10, 24
KNUTHSUMOFF option of $COVARIANCE record 166
KNUTHSUMOFF option of $ESTIMATION record 159

- L -

L1 data item 141
L2 data item 141, 150, 169
labels, reserved 54
label substitution 156
lagged dose 74
LAPLACE option of $ESTIMATION record 159
laplacian estimation method 147
LAPLACIAN option of $ESTIMATION record 147
LAST20 option of $DATA record 54
LASTONLY option of $TABLE record 146
least squares criterion 41
least squares ELS, extended 42
least squares OLS, ordinary 41
least squares WLS, weighted 41
$LEVEL record 153
LEVWT option of $ESTIMATION record 159
LFORMAT option of $TABLE record 146
likelihood ratio test 48, 131
LIKE option of $ESTIMATION record 152, 159
linear model 33
linear system 63
link editing 5
LIREC reserved variable 158
load module 5
log likelihood 48, 119
LOG,LOG10 73
log-normal error model 28, 37, 85
Loops 157
lower case 155
lower triangular elements of OMEGA 37
LRECL option of $DATA record 53

- M -

MADAPT option of $ESTIMATION record 159
MAPCOV option of $ESTIMATION record 159
MAPINTER option of $ESTIMATION record 159
MAPITER option of $ESTIMATION record 159
MARGINALS option of $NONPARAMETRIC record 152
mass balance 72
MASSRESET option of $ESTIMATION record 159
MATRIX option of $COVARIANCE record 143, 166
MAXEVAL option of $ESTIMATION record 92, 159, 168
maximum number of data items 51, 55
maximum number of observation records 56
MAX 73
MCETA option of $ESTIMATION record 159
MDV data item 100,101 153
MDV data item, generated 69
MDV data item 55, 139
MDVI1,2,3 reserved variable 153
MDVRES reserved variable 155
mean squared error, MSE 43
METHOD option of $ESTIMATION record 91, 147, 159
Michaelis-Menten model 34, 71
microconstant 72
minimal model 112
minimum value of objective function 10, 48, 91, 101
MIN 73
MISDAT option of $DATA record 156
mixed effects model, general 39
mixed effects model 26, 31
MIXEST reserved variable 148
MIXNUM reserved variable 148
MIXP reserved variable 148
MIXPT reserved variable 148
$MIX record 148
MIX subroutine 132, 148-149
mixture model 148-149
MNEXT reserved variable 136
MNOW reserved variable 136
model building 105
modeled duration 135
modeled rate 134
model, Emax 132
model, full 47, 118
model, general mixed effects 39
model, linear 33
model, Michaelis-Menten 34, 71
model, minimal 112
model misspecification 102
model, mixed effects 26, 31
model, mixture 148-149
model, multiplicative 34, 80
model, one-compartment 8, 13, 23
model, parameter 32
model, pharmacodynamic 132, 139, 150
model, pharmacokinetic 2, 71, 133
model, population 32
model, random effects 126
$MODEL record 133
model, reduced 47, 118
model specification file 92, 142-143
model, statistical 125-126
model, structural 23, 36, 97, 112
MODEL subroutine 132
model, user-defined 133
MOD 73
monitoring of search 91
monte-carlo method 152, 155
monte-carlo 147
MPAST reserved variable 136
MRG_ data item 152
MSEC,MFIRST reserved variable 158
MSE mean squared error 43
$MSFI record 92, 142
MSFO option of $ESTIMATION record 142, 159, 168
MSFO option of $NONPARAMETRIC record 152
MTDIFF reserved variable 136
MTIME 135
multiple bolus dose 60
multiple dose 12
multiple $ESTIMATION records 153
multiple infusion 61
multiple steady-state doses 63
multiplicative model 34, 80
multivariate observation 141
MU modeling 152-153
MUM option of $ESTIMATION record 159
MU_reserved varaibles 152-153
MXSTEP reserved variable 134

- N -

NBURN option of $ESTIMATION record 159
neffi utility 159
neff utility 159
negative objective function 42
nested if 74
nested parentheses 74
NEWIND reserved variable 157
NEW option of $MSFI record 142
NEW option of $SIMULATION record 145
NINDR reserved variable 158
NIPRED,NIRES,NIWRES reserved label of $TABLE,$SCATTER record 154
NIREC,NDREC reserved variable 157
NITER/NSAMPLE option of $ESTIMATION record 159
nmfe74 5
nmfe74 utility 158
nmfe 5
nmtemplate utility 158
NM-TRAN defined 2
NOABORTFIRST 148
NOABORTFIRST option of $THETA record 149
NOABORT 148
NOABORT option of $ESTIMATION record 159
NOAPPEND option of $TABLE record 146
NOCHECKMU option of $ABBREVIATED record 153
NOCOV option of $ESTIMATION record 144, 159
NOFASTDER option of $ABBREVIATED record 156
NOFCOV option of $COVARIANCE record 144, 166
NOFORWARD option of $TABLE record 146
NOHABORT 148
NOHABORT option of $ESTIMATION record 159
NOHABORT option of $THETA record 149
NOHEADER option of $TABLE record 146
NOLABEL option of $ESTIMATION record 159
NOLABEL option of $TABLE record 146
NOMSFTEST option of $MSFI record 142
Non continuous 152
NONINFETA option of $ESTIMATION record 159
NONMEM control language 5
NONMEM Features 152
nonmem_general_reserved 7
NONMEM 1
NONMEM Outputs, Changes to 154
nonmem_reserved_general 153, 158
nonmem_reserved 157
$NONPARAMETRIC record 152
NOOMEGABOUNDTEST option of $ESTIMATION record 159
NOPREDICTION option of $SIMULATION record 145
NOPRINT option of $TABLE record 146
NOPRIOR option of $ESTIMATION record 149, 159
NOREPLACE option of $MSFI record 142
NOREPLACE option of $SIMULATION record 146, 153
NOREWIND 52
NORMAL option of $SIMULATION record 145
NOSIGMABOUNDTEST option of $ESTIMATION record 159
NOSLOW option of $COVARIANCE record 166
NOSUB option_of $ESTIMATION record 156
NOSUB option of $ESTIMATION record 159
NOSUB option_of $SCATTER record 156
NOSUB option_of $TABLE record 156
NOSUPRESET option of $SIMULATION record 146
NOTHETABOUNDTEST option of $ESTIMATION record 159
NOTITLE option of $ESTIMATION record 159
NOTITLE option of $TABLE record 146
N output 154
NPDE,NPD reserved label of $TABLE,$SCATTER record 155
NPOPETAS option of $MSFI record 142
NPROB reserved variable 144
NPSUPP,NPSUPPE option of $NONPARAMETRIC record 152
NREP reserved variable 145
NSIG option of $ESTIMATION record 159
NSPOP reserved variable 148
null data item 50
null hypothesis 46
NULL option of $DATA record 53
null value 46
null value of parameter 128-129
number of significant digits 97
NUMBERPOINTS option of $THETA record 142
NUMDER option of $ESTIMATION record 154, 159
NUMERICAL option of $ESTIMATION record 159
NUTS_BASE option of $ESTIMATION record 159
NUTS_DELTA option of $ESTIMATION record 159
NUTS_EPARAM option of $ESTIMATION record 159
NUTS_GAMMA option of $ESTIMATION record 159
NUTS_INIT option of $ESTIMATION record 159
NUTS_MASS option of $ESTIMATION record 159
NUTS_MAXDEPTH option of $ESTIMATION record 159
NUTS_OPARAM option of $ESTIMATION record 159
NUTS_REG option of $ESTIMATION record 159
NUTS_SPARAM option of $ESTIMATION record 159
NUTS_STEPINTER option of $ESTIMATION record 159
NUTS_STEPITER option of $ESTIMATION record 159
NUTS_TERM option of $ESTIMATION record 159
NUTS_TEST option of $ESTIMATION record 159
NUTS_TRANSFORM option of $ESTIMATION record 159
NWPRI subroutine 141, 149

- O -

OACCEPT option of $ESTIMATION record 159
objective function, minimum value of 10, 48, 91, 101
objective function 42, 91, 119
OBJI reserved label of $TABLE record 154
observation event record 56, 84
observation records, maximum number of 56
observed value 4, 55-56, 90
OBSONLY option of $TABLE record 146
Odd type data 152
OLKJDF option of $ESTIMATION record 159
OLNTWOPI option of $ESTIMATION record 159
OLS, ordinary least squares 41
OMEGABOUNDTEST option of $ESTIMATION record 159
OMEGA, diagonal elements of 37
$\hat{\omega}$ 41
OMEGA, lower triangular elements of 37
OMEGA 10, 26, 36
Ω 36, 88
$OMEGAPD record 141
$OMEGAP record 141
$OMEGA record 10, 14, 88-89, 97, 139
OMEGA reserved variable 158
$OMIT record 152
OMITTED option of $COVARIANCE record 166
OMITTED option of $ESTIMATION record 159
one-compartment model 8, 13, 23
ONEHEADERALL option of $TABLE record 146
ONEHEADER option of $TABLE record 146

ONEHEADERPERFILE option of $TABLE record 146
ONLYREAD option of $MSFI record 142
ONLYSIMULATION option of $SIMULATION record 145
on/off status 57, 65, 78
operating system command 5
operating system error message 53
OPTMAP option of $ESTIMATION record 159
ORD0 option of $SCATTERPLOT record 93, 146
ORDER option of $ESTIMATION record 154, 159
ordinary least squares OLS 41
OSAMPLE_M1 option of $ESTIMATION record 159
OSAMPLE_M2 option of $ESTIMATION record 159
other event record 56-57, 74
OTHER option of $SUBROUTINE record 149
outlier 108
output compartment 25, 57, 60, 64, 72
output fraction 72, 79
Outputs, Changes to NONMEM 154
output-type 57, 65, 72, 137
OVARF option of $ESTIMATION record 159

- P -

PACCEPT option of $ESTIMATION record 159
PARAFILE option of $COVARIANCE record 154, 166
PARAFILE option of $ESTIMATION record 154, 159
PARAFILE option of $SIMULATION record 154
PARAFILE option of $TABLE record 154
PARAFPRINT option of $COVARIANCE record 166
PARAFPRINT option of $ESTIMATION record 159
parallel computing 154
parameter, additional PK 55, 72
parameter ALAG, Absorption lag 135
parameter, basic PK 55, 72
parameter constraint 10, 87, 114
parameter estimate, correlation of 45
parameter estimate, distribution of 44
parameter estimate, final 91
parameter estimate, individual 147
parameter estimate, initial 142
parameter estimate 41
parameter estimate, precision of 43, 92, 128, 131
parameter, fixed effects 23, 33
parameterization 24, 43, 46, 72, 76
parameter model 32
parameter, null value of 128-129
parameter, PK 10, 71
parameter, random effects 26, 126
parameter, scale 10, 24, 71, 77
parameters, function of 43
parameter, time varying 35
PARAMETRIC option of $SIMULATION record 145
partial derivative 6
partitioned scatterplot 93, 104
PASSRC reserved variable 138
PASS subroutine 138
patient identification number 51
PCMT data item 57, 71
PD in $SIZES 154
PD in $SIZES 51
PD in SIZES 51
PDT in $SIZES 146, 154
perturbed initial estimate 88
PFCOND option of $COVARIANCE record 166
pharmacodynamic model 132, 139, 150
pharmacokinetic model 2, 71, 133
phenobarbital example 12, 24, 89-90, 105, 132, 151
ϕ 23
PHI 73
$PHIS record 153
PHITYPE option of $ESTIMATION record 159
PK parameter, additional 55, 72
PK parameter, basic 55, 72
PK parameter 10, 71
$PK record 5, 10, 55, 71
PK subroutine, call to 58, 136
PK subroutine 4-5, 55, 72, 132
plasma concentration 8, 25, 30, 35
plot, index 107, 116
population data 55, 89
population data, single-response 23
population model 32
positive definite 140
POSTHOC option of $ESTIMATION record 147, 159
power function error model 29, 42, 85
PR_CT reserved variable 152
PRDERR 149
PRDFL reserved variable 152
precision of parameter estimate 43, 92, 128, 131
PRECOND option of $COVARIANCE record 166
PRECONDS option of $COVARIANCE record 166
PRED error recovery option 148
PRED error return code 148
predicted value 4, 56
predicted value PRED 11, 92
prediction compartment 57
PREDICTION option of $ESTIMATION record 159
PREDICTION option of $SIMULATION record 145
PREDI,RESI,WRESI reserved label of $TABLE,$SCATTER record 154
PREDPP error message 171
PREDPP library 4
PREDPP 1, 71, 155
PRED, predicted value 11, 92
$PRED record 6, 97
PRED_,RES_,WRES_ reserved variable 158
PRED subroutine 1, 71, 97, 132, 138
PRED subroutine, recursive 35, 143
Preprocessor, Data 156
P reserved variable 148
PRETYPE option of $COVARIANCE record 166
PRINT option of $COVARIANCE record 143, 166
PRINT option of $ESTIMATION record 92, 159, 169
PRINT option of $TABLE record 146
prior 141
$PRIOR record 149
PRIOR subroutine 149
$PROBLEM record 8, 96, 144
proportional error model 27, 37
PR_Y reserved variable 152
PSAMPLE_M1 option of $ESTIMATION record 159
PSAMPLE_M2 option of $ESTIMATION record 159
PSAMPLE_M3 option of $ESTIMATION record 159
PSCALE_MAX option of $ESTIMATION record 159
PSCALE_MIN option of $ESTIMATION record 159
P VAL output 154
p-value 131

- R -

random effects model 126
random effects parameter 26, 126
random effects 26, 32, 80
RANDOM subroutine 145, 152
random variable 6, 73, 81, 84

random variable 81
RANMETHOD option of $COVARIANCE record 166
RANMETHOD option of $ESTIMATION record 159
RANMETHOD option of $TABLE record 147
RATE data item 56, 58
rate, modeled 134
rate of infusion 58
raw data average feature 152
RAW_ data item 152
raw output file 154
RECOMPUTE option of $NONPARAMETRIC record 152
record length 53
RECORDS option of $DATA record 52, 156
recovery option, error 148
recursive PRED subroutine 35, 143
reduced model 47, 118
relative time 56, 66
REPEAT1 option of $ESTIMATION record 152, 159
REPEAT2 option of $ESTIMATION record 152, 159
repeated values xn 141
REPEAT option of $ESTIMATION record 143, 159
Repetition feature 152
REPLACE option of $ABBREVIATED record 156
REPLACE option of $SIMULATION record 146, 153
replication of experiment 43
REQUESTFIRST option of $SIMULATION record 146
REQUESTSECOND option of $SIMULATION record 146
RESCALE option of $MSFI record 143
reserved labels 54
reserved variable 7
reset-dose event record 57
reset event record 57
residual error 25, 32
residual RES 16, 92, 115
residual vs y, correlation of 117
RES, residual 16, 92, 115
RESUME option of $COVARIANCE record 144, 166
return code, error 148
REWIND option of $SIMULATION record 145
REWIND 52
RFORMAT option of $TABLE record 146
root.agh output file 154
root.fgh output file 154
root.xxx output file 154
RPT_ data item 152
RPTI,RPTO,RPTON reserved variable 152
R^{-1} covariance matrix 143

- S -

S1IT,S2IT reserved variable 144
S1NIT,S2NIT reserved variable 144
S1NUM,S2NUM reserved variable 144
SADDLE_HESS option of $ESTIMATION record 159
SADDLE_RESET option of $ESTIMATION record 159
SAME option of $SIGMA record 140
saturation model 34, 80
scale parameter 10, 24, 71, 77
scatterplot, partitioned 93, 104
$SCATTERPLOT record 87, 92-93, 97
scatterplot 11, 51, 57, 59, 92-93
scatterplot step 104
SD in SIZES 54
SD option of $OMEGA record 140
SD option of $SIGMA record 140
SEED option of $ESTIMATION record 159
SEED option of $TABLE record 147
SEOMEG reserved variable 158
SE output 154
sequence of data set 50, 52
SESIGM reserved variable 158
SE (standard error) 44
SETHET reserved variable 158
SETHETR reserved variable 158
SIGDIGITS option of $ESTIMATION record 91
SIGLO option of $COVARIANCE record 144, 166
SIGLO option of $ESTIMATION record 159
SIGL option of $COVARIANCE record 144, 166
SIGL option of $ESTIMATION record 159
SIGMABOUNDTEST option of $ESTIMATION record 159
$\hat{\sigma}$ 41
$SIGMAPD record 141
$SIGMAP record 141
$SIGMA record 14, 87-89, 97, 139
SIGMA reserved variable 158
SIGMA 15, 32
Σ 32, 88
significant digits, number of 97
SIMEPS subroutine 145
SIMETA subroutine 145
$SIMULATION record 144
simulation step 99, 144
single-response population data 23
SIN 73
SIRCENTER option of $COVARIANCE record 166
SIRDF option of $COVARIANCE record 166
SIRNITER option of $COVARIANCE record 166
SIRPRINT option of $COVARIANCE record 166
SIRSAMPLE option of $COVARIANCE record 166
SIRTHBND option of $COVARIANCE record 166
size of data set 50
$SIZES record 154
SKIP_ reserved variable 144
SLKJDF option of $ESTIMATION record 159
SLOW option of $COVARIANCE record 144, 166
SLOW option of $ESTIMATION record 159
SORT option of $ESTIMATION record 152, 159
SPECIAL option of $COVARIANCE record 143, 166
SQRT 73
SS data item 56, 58
S^{-1} covariance matrix 143
standard deviation 27, 90, 116
standard error of estimate 92
standard error 8, 10, 44-45, 102, 129, 143
standard error vs. standard deviation 44
STANDARD option of $OMEGA record 140
STANDARD option of $SIGMA record 140
statistical error model 23
statistical model 125-126
STDOBJ option of $ESTIMATION record 159
steady-state data item 58
steady-state doses, multiple 63
steady-state dose 60
steady-state level 61-62, 64
steady-state 135
STIELTJES option of $ESTIMATION record 152, 159
STRAT,STRATF option of $SIMULATION record 146, 153
structural model 23, 36, 97, 112
SUBPROBLEM option of $SIMULATION record 144
subroutine, generated 6
$SUBROUTINE record 134
$SUBROUTINE record 71, 73, 97
$SUBOUTINES record 149
subroutine, user-supplied 6

subroutine, user-written 149
sum of squares 10, 41
superposition 63
superproblem 144
$SUPER record 144
SUPRESET option of $SIMULATION record 146
SVARF option of $ESTIMATION record 159
symbolic label substitution 156
synonym 54
system, linear 63

- T -

table_compare utility 158
table file, external 146
table_quant utility 158
$TABLE record 87, 92-93, 97
table_resample utility 158
table step 104
table 57, 59, 92
table_to_xml utility 158
tabs in data file 156
tag label output 154
tag label # output 154
TAN 73
TEMPLT reserved variable 148, 152
THBND option of $COVARIANCE record 166
theophylline example 8
THETABOUNDTEST option of $ESTIMATION record 159
THETAFR reserved variable 158
$\hat{\theta}$ 41
theta, initial estimate of 87
$THETAI record 153
$THETAP record 153
$THETA record 10, 87, 97
$THETAR record 153
THETA 10, 73
THIN option of $ESTIMATION record 159
TIME/24 54, 67
TIME data item, colon ":" in 66
TIME data item 56
TIME option of $DATA record 156
time, relative 56, 66
time varying parameter 35
TNPRI subroutine 149
TOL option of $COVARIANCE record 134, 166
TOL option of $ESTIMATION record 134
TOL option of $SUBROUTINE record 134
$TOL record 134
TOL subroutine 132
to NONMEM Outputs, Changes 154
TO option of $SCATTERPLOT record 104
TRANS1 73
transgeneration 138, 157
TRANSLATE option of $DATA record 54, 67, 156
translation 24, 72, 98
TRANSLATOR error message 171
TRANSLATOR warning message 5
TRANS 4, 24, 72
TRUE option of $SIMULATION record 145
true-value variable 81
TSTATE reserved variable 155
TTDF option of $ESTIMATION record 159
typical value 36, 80, 90

- U -

UNCONDITIONAL option of $COVARIANCE record 143, 166
UNCONDITIONAL option of $SCATTERPLOT record 92
UNCONDITIONAL option of $TABLE record 92
UNCONSTRAINED ETAS 140, 153
unexplained variability 119, 126
UNIFORM option of $SIMULATION record 145
UNIT option of $SCATTERPLOT record 93
unit slope line 11, 93
units 24, 77
upper case 155
urine collection 30, 64
urine concentration 25, 30, 35, 38, 64
urine volume 25, 30, 64
user-defined model 133
user-supplied subroutine 6
user-written subroutine 149
USMETA subroutine 149

- V -

value, null 46
value, observed 4, 55-56, 90
value, predicted 4, 56
VALUES option of $OMEGA record 141
VALUES option of $SIGMA record 141
value, typical 36, 80, 90
VARCALC 155
variability, interindividual 36
variability, intraindividual 25, 36
variability, unexplained 119, 126
variable, dependent 8-9, 23
variable, indicator 29, 35, 77, 81, 85-86
variable, random 6, 73, 81, 84
variance-covariance matrix, diagonal 89, 139
variance-covariance matrix 42
VARIANCE option of $OMEGA record 140
VARIANCE option of $SIGMA record 140
variance 10, 36
VECTRA reserved variable 157
VERSION option of $MSFI record 142
volume of distribution V 8, 24-25
V, volume of distribution 8, 24-25

- W -

warning message, TRANSLATOR 5
$WARNING record 156
weighted least squares WLS 41
weighted residual 92
weighted residual WRES 92, 115, 117
WIDE option of $DATA record 53
WLS, weighted least squares 41
WRESCHOL option of $TABLE record 147
WRES, weighted residual 92, 115, 117

- X -

xml_compare utility 158
XVID1-5 data item 155
LNTWOPI 159

- Y -

year 2000 54
YLO YUP reserved variable 152
ỹ 41
y 23

- Z -

ZERO option of $ESTIMATION record 152, 159
zero-order bolus dose 134-135
zero-out a compartment 72