

Estimating Time to Steady State Analysis in SAS

Richard Moreton and Lata Maganti, Merck & Co., Inc., Rahway, NJ, USA

ABSTRACT

For chronic use drugs, the time it takes to reach steady state drug concentrations needs to be estimated. Steady state is said to be reached when after multiple doses of a drug there is no meaningful difference between pharmacokinetic (PK) profiles of successive doses. An overview of three statistical methodologies is presented along with SAS® code. These methodologies include NOSTASOT (No-Statistical-Significance-of-Trend), effective half-life for accumulation, and nonlinear mixed effects modeling. NOSTASOT requires the least number of assumptions while effective half-life and nonlinear modeling have several assumptions. To perform the analyses, we will touch upon advanced programming with PROC IML, PROC MIXED, and PROC NL MIXED.

INTRODUCTION

When drugs are taken at regular intervals, there will come a time when the amount of drug in a subject's system will be the same as for the previous dose. When this occurs the concentration of the drug in the subject is said to be at steady state. The time until this occurs is of interest for regulatory agencies and may also be needed for designing drug-drug interaction studies.

REACHING STEADY STATE

To understand how steady state occurs, let's first walk through what happens when a drug is taken repeatedly. For this example, we will assume a once daily oral dosing with a one compartment PK model. The time between doses is denoted as τ . For once daily dosing $\tau=24$ hours.

After a single dose, the drug is absorbed and then shortly after reaching a maximum concentration it is eliminated in an exponential manner. The concentration right before the second dose (at time $\tau=24$ hours) is the trough concentration (C_{trough}) of the first dose. Figure 1 illustrates the cumulative effect of multiple doses and how C_{trough} approaches steady state for a simulated subject.

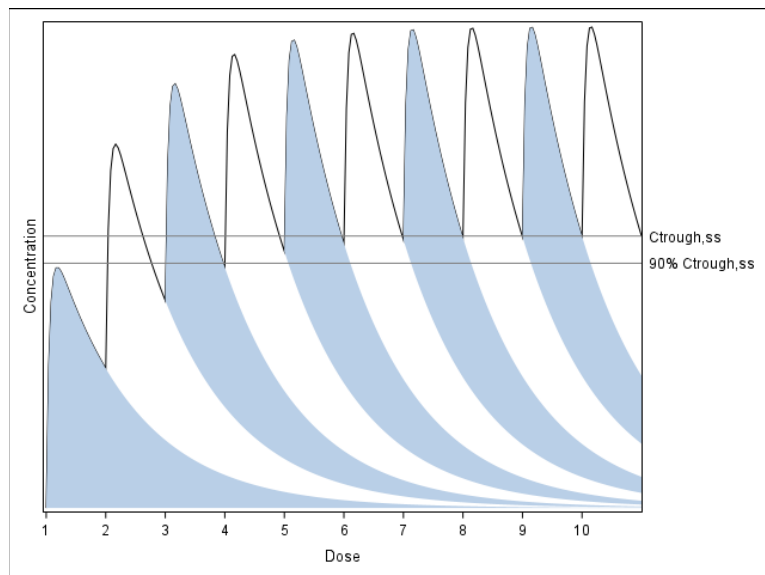


Figure 1 Simulated Concentration After Multiple Doses

After the second dose, the concentration is equal to the amount of drug absorbed from the second dose (unshaded region), as well as the amount left over from the first dose (the first shaded region). The concentration after the third dose is equal to the amount of drug absorbed from the third dose, plus the

amount left over from the second dose, plus the even smaller amount left over from the first dose. This goes on until the amount of drug from the first dose is practically zero (approximately 5 half-lives). At this point, the subject has reached steady state.

ESTIMATING TIME TO STEADY STATE

Using multiple dose Phase 1 studies, the time to steady state can be derived in several ways. This can be seen by comparing either C_{trough} or the area under the curve (AUC) of successive doses. In practice, it is sufficient to define the time to steady state as time until 90% of the theoretical steady state has been achieved. Three methods of estimating time to steady state will be discussed in this paper; No-Statistical-Significance-of-Trend (NOSTASOT), effective half-life for accumulation, and nonlinear mixed effects modeling (Maganti et al., 2008).

NOSTASOT

NOSTASOT is a stepwise test of linear trend. At steady state, C_{trough} values generally appear to fall in a straight line when plotted against time. The main assumption for using this method is that the C_{trough} values are monotonically non-decreasing.

The time to achieve steady state plasma concentrations of a compound is assessed using repeated measures analysis of variance using PROC MIXED. Natural log transformed individual C_{trough} is the response variable with dosing-interval units as the repeated (categorical) measure. All available timepoints at which C_{trough} were collected are included in the model. Attainment of steady state is determined by step-wise testing for linear trend over successive ranges of timepoints using linear contrasts. The linear contrasts are constructed such that each represents the slope of a simple linear regression line through the timepoints included in the contrast (that is, change in natural log-transformed plasma concentration per unit dosing interval).

Performing the NOSTASOT test is carried out in SAS in two procedures. First, a linear contrast is constructed with the ORPOL function in PROC IML. Next, that contrast is tested in PROC MIXED.

ORPOL FUNCTION

In the first step of this procedure, a linear contrast including all timepoints is constructed. For a Phase 1 multiple dose study, intensive PK samples are not collected throughout the entire study. However, C_{trough} is collected at most, if not every dosing interval. Tables of contrasts for equally spaced timepoints exist, but if C_{trough} is not collected at every dosing interval we need to construct the linear contrast ourselves. To construct the linear contrast, the ORPOL function in PROC IML is used.

Syntax: ORPOL (x <, maxdegree> <, weights>)

X is a $n \times 1$ vector of values, in our case the unique days/dose numbers where C_{trough} was collected.

Maxdegree is the maximum degree of polynomial to be calculated. Degree of 0 would be a constant, degree of 1 is a linear polynomial, degree of 2 is a quadratic polynomial, etc. The default is the $\min(n, 19)$. For our analyses we only want a linear polynomial. Specifying maxdegree of 1 will return both the constant and linear polynomials.

Weights is also an optional argument to specify the weight to give to each value in X. The default is 1 for all values in X. For our analyses we will only use the default value.

The result of the ORPOL function is a matrix with n rows and maxdegree+1 columns.

The following is an annotated program to calculate the values for the contrast:

```
data timepoints;
  do dose_n=1,3,4,5,6;
    output;
  end;
run;
proc iml;
  use timepoints;*include the dataset with all unique timepoints;
```

```

read all var{dose_n} into intervals;
  *reads timedata dataset into vector named intervals;
m=NROW(intervals);*calculate the number of timepoints;
poly=ORPOL(intervals,1);
  *matrix of all orthogonal polynomials up to degree 1 (linear);
k=T(poly) [2,];
  *transpose matrix and keep only second row(linear contrast);
c=SQRT(SSQ(intervals-SUM(intervals)/m));
  *standard deviation of timepoint values;
estimate=k/c;*orthonormal contrast to be used in estimate statement;
print intervals, m, poly, k, c, estimate;
  *print values to output destination;
create est from estimate;*creates new SAS dataset named est;
append from estimate;
  *adds observations from estimate matrix to dataset est;
quit;

```

In Table 1 we can see the results from the print statement.

Intervals	M	Poly		K	C	Estimate
1	5	0.4472136	-0.727825	-0.727825	3.8470768	-0.189189
3		0.4472136	-0.20795	-0.20795		-0.054054
4		0.4472136	0.0519875	0.0519875		0.0135135
5		0.4472136	0.3119251	0.3119251		0.0810811
6		0.4472136	0.5718628	0.5718628		0.1486486

Table 1 Results From PROC IML

The matrix Poly contains the constant and linear polynomials from the ORPOL function for the given values in the vector Intervals. The values of K and Estimate have been transposed to fit the table, in the output these have 1 row and 5 columns. The dataset est has one row with default columns COL1-COL5.

PROC MIXED

From this point, programmers can use the more familiar data step functions or dive deeper into more PROC IML functions to build a macro variable with the required estimate statement for PROC MIXED. Below is the pre-specified PROC MIXED code with the estimate statement using the contrast from PROC IML.

```

proc mixed data=indata;
  class subjid dose_n;
  model ln_ctrough = dose_n;
  repeated dose_n/subject=subjid type=cs;
  estimate 'Estimate Dose 1-6' dose_n -0.189189 -0.054054 0.0135135
    0.0810811 0.1486486 / cl alpha = 0.05;
  lsmeans dose_n;
quit;

```

The CLASS statement specifies that the variables subjid and dose_n are treated as categorical variables. The MODEL statement specifies that ln_ctrough is the dependent variable and dose_n (dose number) is the independent variable.

The REPEATED statement specifies the R matrix in the mixed model. The default assumption in PROC MIXED for the R matrix is that the variance of all residuals is the same and that the covariances are all zero. The REPEATED statement controls the covariance structure imposed upon the residuals or errors. In PROC MIXED there are various structures to specify relationships among the errors. In repeated measures models the SUBJECT statement parameter defines which observations belong to the same subject, and which belong to different subjects, where different subjects are independent. The TYPE optional statement parameter specifies the model for the covariance structure of the errors. The covariance structure may be changed if there are problems of convergence of the model. In the above model, compound symmetry (CS) is specified. Other commonly used covariance structures are unstructured (UN) and heterogeneous CS (CSH).

The ESTIMATE statement consists of a label, the independent variable (dose_n), and the contrast we created with PROC IML. Optional confidence limits can be computed with CL option and specifying ALPHA. The LSMEANS statement will give the least square means at each level of dose_n, which while not required for the analysis, it is nice to report in a summary table. It is also possible to have multiple estimate statements in a single PROC MIXED call.

If the ESTIMATE statement using all the dosing timepoints is significant (<.05), that means that there is a significant slope and steady state is not reached after the first dose. Another contrast must be created with all but the first dose and the estimate in PROC MIXED must be checked again.

```
proc iml;
  use timepoints;
  read all var{dose_n} into intervals;
  m=NROW(intervals);
  subset=intervals[2:m]; *exclude the first timepoint;
  poly=ORPOL(subset,1);
  k=T(poly)[2,];
  c=SQRT(SSQ(subset -SUM(subset)/(m-1)));
  estimate=k/c;
  print intervals, m, poly, k, c, estimate;
quit;
```

In Table 2 we can see the results with the first timepoint removed.

Subset	M	Poly		K	C	Estimate
3	5	0.5	-0.67082	-0.67082	2.236068	-0.3
4		0.5	-0.223607	-0.223607		-0.1
5		0.5	0.2236068	0.2236068		0.1
6		0.5	0.6708204	0.6708204		0.3

Table 2 Results From Second PROC IML Call

Because we removed the first timepoint, when we create the ESTIMATE statement for PROC MIXED we need to add an additional 0 to the beginning of the contrast.

```
proc mixed data=indata;
  class subjid dose_n;
  model ln_ctrough = dose_n;
  repeated dose_n/subject=subjid type=cs;
  estimate 'Estimate Dose 3-6' dose_n 0 -0.3 -0.1 0.1 0.3 /
    cl alpha = 0.05;
  lsmeans dose_n;
quit;
```

This process repeats until the first non-significant estimate statement or until only 3 timepoints are included. If only the last 3 timepoints are included and the estimate is still significant, then we cannot make an inference to what the time until steady state is.

Also note that since the PROC MIXED code doesn't change except for the estimate statements, we can simplify programming by creating all possible contrasts in one PROC IML call with some additional do-loop processing. Then a single PROC MIXED call can be used to evaluate all contrasts at once.

EFFECTIVE HALF-LIFE FOR ACCUMULATION

In a typical Phase 1 multiple dose study, the AUC from time 0 to the end of the dosing interval (τ) is calculated for the first and last dose. Under the assumption of linear PK, we can write a system of equations and iteratively solve them for time to steady state, based on one compartment model. Figure 2 is a visualization of $AUC_{(0-\tau)}$ for the first and tenth dose for a simulated subject.

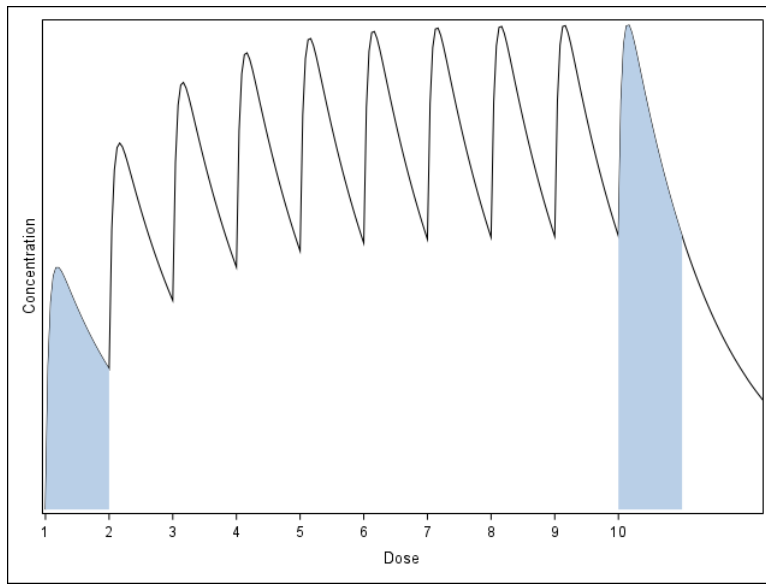


Figure 2 AUC(0- τ) for First and Last Dose

If we can assume linear PK, then the following inequality using the $AUC_{(0-\tau)}$ of the first dose (time a) and the $AUC_{(0-\tau)}$ of the last dose (time b) should hold true.

$$1 < \frac{AUC_{b(0-\tau)}}{AUC_{a(0-\tau)}} < \frac{b}{a}$$

We can then write the individual subject accumulation ratio as

$$\frac{AUC_{b_i(0-\tau)}}{AUC_{a_i(0-\tau)}} = \frac{1 - e^{-\eta_i \cdot b_i \cdot \tau}}{1 - e^{-\eta_i \cdot a_i \cdot \tau}}$$

where η_i is equal to $\ln(2)/\text{effective half-life}$.

We can solve this system of equations for each subject by using the observed accumulation ratio for the left-hand side. We can use the observed half-life as the initial estimate of η_i and then iteratively change η_i until the above formula on the right-hand side is equal to the left-hand side. This can be done using a DO UNTIL statement in the data step.

```
data dsn;
  call streaminit(123);
```

```

a=1;b=10;tau=24;
do subjid=1 to 12; ;*simulated data, similar to above figures;
  aucratio=exp(rand('normal',log(2.5),0.1));
  thalf=rand('normal',23,1);
  output;
end;
run;
data eta;
set dsn;
length note $50.;
min=0;*initial minimum bound for eta;
max=1;*initial maximum bound for eta;
eta = log(2)/thalf;*starting estimate for eta;
acc = round(aucratio,0.0001);
  *rounding enforces tolerance limit for convergence;
count = 0;*count value is added as a precaution against infinite loop;
if 1< acc <(b/a) and 0 < eta < 1 then do;
  *inequality must hold true for analysis;
do until (ratio=acc or count=1000);*loop continues until condition met;
  calcratio = ( 1 - exp(-1*b*eta*tau) ) / ( 1 - exp(-1*a*eta*tau) );
  ratio = round(calcratio,0.0001);
  if ratio > acc then do;
    *if calculated ratio greater than observed, increase eta;
    min = eta;
    eta = eta + (max-eta)/2;
  end;
  else if ratio < acc then do;
    *if calculated ratio less than observed, decrease eta;
    max = eta;
    eta = min + (eta - min)/2;
  end;
  count = count + 1;
end;
end;
else do;
  eta=.;
  note = "The condition 1 < aucratio < (b/a) is not met";
end;
if count = 1000 then do;
  note = "eta cannot be found in 1000 iterations";
  eta = .;
end;
run;

```

Once we have an estimate of η_i , we can calculate the fraction of steady state (f_{ss}) achieved at each dose number (N) with the following formula.

$$f_{ss_i} = (1 - e^{-N \cdot \eta_i \cdot \tau})$$

The number of dosing intervals needed to reach steady state, $t_{0.9i}$, can be solved by setting the above equation to 0.9 and solving for N.

$$t_{0.9_i} = \frac{-\ln 0.1}{\eta_i \cdot \tau} \approx \frac{2.3}{\eta_i \cdot \tau}$$

From here we can calculate summary tables and estimate the average time to steady state. Table 3 below is a summary of the simulated data.

Mean	Median	Minimum	Maximum	Lower 90% CI	Upper 90% CI
4.45	4.51	3.74	5.13	4.27	4.64

Table 3 Simulated Time to Steady State Results

NON-LINEAR MIXED EFFECTS MODELING

Non-linear mixed effects modeling allows us to calculate both population and subject level estimates of the time to steady state. If we can assume a one compartment PK model with continuous infusion or oral dosing with rapid absorption, then we can model the C_{trough} values with a non-linear model.

We can write a function of timepoints and unknown parameters (θ_i) as

$$f(t_{ij}, \theta_i) = C_{ss(i)}(1 - e^{-2.3t_{ij}/t_{0.9i}})$$

Where C_{ss_i} is the concentration at steady state for the i th subject, t_{ij} is the dose number of the j th dosing interval for the i th subject, and $t_{0.9i}$ is the time to steady state for the i th subject.

C_{trough} can then be modeled as

$$C_{ij} = f(t_{ij}, \theta_i) e^{\varepsilon_{ij}}$$

Where $\varepsilon_{ij} \sim N(0, \sigma^2)$. Inter-subject variation is modeled as

$$\theta_i = \theta e^{\gamma_i}$$

Where θ is the vector of population parameters of C_{ss} and t_{90} , $\gamma_i \sim N([0,0], \omega)$ is a vector of inter-subject random effects, and ω is a covariance matrix. All error terms are assumed to be log-normally distributed.

From these equations we can build a model in PROC NL MIXED.

PROC NL MIXED

While PROC MIXED is used for linear models, PROC NL MIXED is used for non-linear models. In PROC NL MIXED users need to define their parameters and error structures. We can write statements, similar to a data step, and any underlying relationships between parameters can also be defined. This makes it extremely powerful and flexible. At the same time, this makes it very intimidating. Here we will build our PROC NL MIXED code step by step.

```
proc nlmixed data=dsn2;
  t90=exp(beta1+b1);
  Css=exp(beta2+b2);
```

First, we have defined the population variables t_{90} and C_{ss} . These are both log-normally distributed, so we write them as $\exp(\text{beta}+b)$. In this expression beta is the mean value and b is the inter-subject error term. We can define the error structure of these error terms below.

```
random b1 b2~normal([0,0],[s2b1,cb12,s2b2]) subject=subject;
```

The error terms b_1 and b_2 are random effects that are normally distributed with mean $[0,0]$ and unstructured variance. Only the upper triangle of the variance matrix needs to be specified (i.e., variance of b_1 , covariance of b_1 and b_2 , and variance of b_2). This is the error term that was specified by γ_i above.

```
pred = log(Css * (1-exp(-2.3*n/t90)));
```

To simplify the formula for C_{trough} and its error term above we can take the log of both sides. Here pred is the predicted, or average, of the function.

```
model lnctrough~normal(pred,s2);
```

Here we have finally defined our model. The natural log of C_{trough} is defined as normally distributed with a mean of pred and variance of s_2 .

```
parms beta1=1.8 beta2=3.4;*initial estimate of t90=6 and Css=30;
```

PROC NL MIXED solves the above equations through an iterative method, similar to our do-while loop above. The method can be specified in the PROC NL MIXED statement with the METHOD option. When we started our do-while loop, we used the observed half-life as an initial estimate. By default, PROC

NLMIXED uses 1 as an initial estimate for all parameters. We can use the PARMS statement to specify a value or a list of values for the initial estimate. Different initial estimates can lead to final estimates, so it is important to try different values and make sure they all converge to roughly the same value.

```
estimate "t90" exp(beta1);  
estimate "Css" exp(beta2);
```

In PROC NLMIXED we can specify a variable or any valid SAS expression in the estimate statement. We can obtain an estimate of t90 and Css by requesting the exponentiation of beta1 and beta2.

```
predict Css out=predcss;  
predict pred out=predc;  
predict t90 out=predt90;  
quit;
```

Lastly, subject level estimates of our parameters can be obtained with some PREDICT statements.

Sometimes the model may not fully converge. There are dozens of PROC NLMIXED options that can affect the outcome, in addition to the above mentioned parms statement. The SAS log will give a fairly detailed warning or error message. Searching the exact wording of the message online can give some settings to play around within your code. Again, it is best to run the model multiple ways and make sure you are getting a consistent result.

CONCLUSION

The three statistical methods generally give consistent results for the time to steady state, but care must be given to the assumptions for using them. NOSTASOT is the least restrictive among the three methods. If the proper studies to determine the underlying PK model have not been completed yet, NOSTASOT should be the method used for analysis. If the PK data follows a one compartment model and we have collected trough concentrations, we can use a non-linear mixed effects model. If dose linearity and one compartment model has been shown, effective half-life for accumulation can be used if AUC is collected over two dosing intervals.

REFERENCES

Maganti L, Panebianco DL, Maes AL. Evaluation of methods for estimating time to steady state with examples from phase 1 studies. AAPS JOURNAL. 2008;10(1):141-147. doi:10.1208/s12248-008-9014-y

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Richard Moreton
Richard.moreton@merck.com

Any brand and product names are trademarks of their respective companies.