

# SAS/AF Application in Sequential Clinical Trials

**Tatsuru Matsushita**

Medical and Pharmaceutical Statistics Research Unit  
The University of Reading  
UK

t.matsushita@reading.ac.uk

## (1) Introduction

Competition in pharmaceutical drug development is intense [Senn:1997]. Planning and conducting clinical trials to confirm drug efficacy is a complicated procedure, and software aids are very desirable for those managing this activity. This paper gives a brief account of sequential methodology that is a powerful but demanding approach to the conduct of clinical trials. Then a SAS application program (PEST 4) which supports sequential clinical trials will be described.

## (2) Clinical trials

The main aim of clinical trials is to obtain a firm conclusion (either positive or negative) about the efficacy of a drug through a number of experimental applications of the drug to patients. Since clinical trials use human beings as experimental subjects, it is ethically desirable to reach a conclusion involving as small number of patients as possible. However, in order to obtain results with reasonable accuracy, the number of patients cannot be reduced below a certain limit.

Another particular aspect of clinical trials is the manner of data accumulation. Unlike experiments in agriculture, from which many early statistical methodologies stemmed, drug efficacy data do not usually become available all at once. Rather, data accumulation depends on opportunities to recruit suitable patients for the drug testing.

Considering these two features of clinical trials, it seems reasonable to stop a trial as soon as it has become obvious that the drug works (or that the drug definitely does not work sufficiently), by inspecting the accumulating data from time to time. This leads to the idea of *sequential* clinical trials [Whitehead:1997].

## (3) Sequential Clinical Trials

In sequential clinical trials, data from all patients are examined every time some relatively small number of patients are recruited. Patients are randomly assigned to treatment, one group being given the drug and the other given standard treatment or a placebo. The observed difference in treatment effects between the drug and the control is summarised as a single efficacy measure (denoted by  $Z$  in PEST 4) and the amount of information (denoted  $V$ ) are calculated.

The computed  $Z$  and  $V$  will be compared with predefined upper and lower boundaries of the trial. If the point  $(V, Z)$  crosses the upper boundary, we claim that the drug does work. If the lower boundary is crossed, we conclude that either the drug has no advantage to the placebo or it may worse than placebo. In both cases, the trial terminates. Otherwise, if the point has not reached either of the boundaries, the whole procedure will be repeated when

data from another group of patients have become available. Each inspection of the data of this type is called an 'interim analysis'.

Because of the intermediate inspections, clinicians have ample chance to terminate the trial when the drug efficacy is obvious or drug efficacy is definitely not promising. However, because of intermediate inspections, the conclusion drawn from the data will be biased unless a special method of analysis is used.

#### (4) Required steps

In order to conduct a sequential clinical trial, we need to go through a sequence of steps:

- Designing trial

Confirming superiority of a drug is only one of various possible aims of a clinical trial. Clinicians might want to compare two similar drugs in their efficacy, or they might want to show their equivalence. They might aim at confirming that a drug, which is known to be effective, has relatively weak side effects. These different aims require different shapes of stopping boundary.

There are two important parameters that determine the design. The first is the probability of claiming the drug to be superior when, in fact, it is not (type I error rate). The second is the probability of claiming non-superiority when it does have intended efficacy (type II error rate). These two error rates determine the position of the boundaries.

Another principal factor in designing clinical trials is the type of patient response. The patients' responses might be qualitative (Yes/No answers to the question whether the drug relieved symptoms). It might be the length of time since the application of the drug before certain events (such as death) occur. It might be a quantitative measure such as reduction in blood pressure after administration of a drug. Response types determine the way data are converted to  $(V, Z)$  points.

Given these parameters, you can explore various designs in terms of the likely sample size, and select a design that is most suitable for each individual study.

- Accumulating and inspecting data

During the trial, each recruited patient will be allocated to one of the trial arms (experimental or control). After recording all results from the patients available for each interim analysis, the data collected will be summarised as a point in  $(V, Z)$  plane, which can be compared with the stopping boundaries.

- Analysing the final result

Once the trial path (the sequence of the  $(V, Z)$  points corresponding to the interim analysis) has crossed a boundary, a p-value, estimated improvement (quantifying the efficacy or lack of it, of the drug), and its confidence interval (range of magnitudes of improvement consistent with the data) can be computed.

#### (5) PEST 4

The computation required for the steps above is quite burdensome. In determining the equation of a stopping boundary, we need to solve a pair of simultaneous equations each involving numerical integration. In order to convert raw interim data into a  $(V, Z)$  point, we may need to do matrix inversion. To analyse the final result in a rigorous way, numerical

integration again is necessary. The software PEST 4 performs each of these functions [MPS Research Unit : 2000].

PEST 4 is a SAS application with graphic user interface working on the WINDOWS platform. While designing, the user can type in trial design parameters through a series of frames, and then PEST 4 calculates appropriate stopping boundaries. When actual data become available, the user can type in summary data or read in individual patient data from a file. Then the data will be automatically converted into V and Z and plotted against the stopping boundaries. When a boundary is crossed, the user has choice between a quick analysis by approximation or a more exact analysis using numerical integration. PEST 4 computes various statistics such as a p-value, an estimated improvement, and its confidence interval.

In addition to the three functions above, PEST 4 has two further facilities. First, you can view a completed or ongoing trial. You can examine the design of trials already entered, and if data collection has not yet started, modify the design parameters. Also you can reproduce any intermediate status of a trial. Of course you can view the result of final analysis. All these facilities would be useful when you need to report a trial.

The other additional function of PEST 4 is simulation. There are two purposes for this facility. The first is to provide the user opportunities to observe how the trial might go. PEST 4 generates data in accord with a user-specified efficacy of the drug. Simulated data are then accumulated in the same way as in monitoring, and once the simulated path crosses a boundary, the user can analyse the result. The second purpose is to conduct, automatically, thousands of simulation runs and use the statistics to verify the design assumptions such as the error rates and average sample size.

#### (6) User Interface (demonstration)

While designing a clinical trial, you can freely try out various combinations of design parameters. Once entered, the parameters are stored in a set of SAS datasets in a folder (the trial folder). All clinical data files should reside in this folder and PEST 4 allows you to specify the data file to be read next. During monitoring, all results from interim inspections will be accumulated into another SAS data set which will be passed to analysis module to compute the final statistics. The results of final analysis are again stored in a SAS datasets in the trial folder. When you are 'VIEW'ing a trial you are looking at these datasets. The dataset created through simulation facility are also stored in the folder.

#### (7) Software structure

PEST 4 consists of around 150 SAS/AF entries, (FRAME entries and SCL entries), and about 90 C program files (source files and header files). All 150 SAS/AF entries are sub-divided into five SAS CATALOGS. By using CATALOGS we can organise entries into groups for maintain software. Four of the CATALOGS correspond to the main function of PEST 4. The fifth one is a set of entries handling output files and graphics

- FRAME entries

A FRAME entry defines a window containing a collection of graphic objects, which are very powerful in constructing an unambiguous and clear user interface. You can define text input fields into which the user can type in any text strings, push buttons invoking other FRAMEs, select radio buttons allowing multiple choices. Graphic display objects can be used to facilitate a quick look at various graphic images before producing final graphic output. Hotspot objects allow you to read numeric values of specific points on a graph.

While the user is interacting with a window, you can control properties of objects, such as colour or visibility, and hence guide the user to sensible usage of the window. Also, It is possible to specify positions of windows on the screen so that the program control flow becomes visible to the user.

Most of the frequently used design parameter values are made available from pull down menus which reduces erroneous input. When the user needs to type in specific values, immediate consistency checking is done and warning messages come up if necessary.

- SCL entries

An SCL entry is a collection of program codes written in Screen Control Language, which can be associated with a FRAME entry and thus be invoked with the corresponding window. This association helps programmers to produce modularised code corresponding to the program control flow.

The SCL language provides a neat way to handle SAS datasets. By overloading SCL variables and dataset variables, the programmer can spare many lines to move variables from/to SAS datasets, resulting in a concise code.

Another important feature of SCL language is that we can write ‘submit blocks’ that can contain any SAS commands. Since ‘submit blocks’ can be interwoven with SCL code segments, you can, for example, modify graphic options (which are global parameters) depending on input data being modified interactively. Even while running PEST 4, it is possible to access intermediate SAS datasets by executing the SAS command from the program editor.

Because of the existence of ‘submit blocks’, we can enjoy benefits from a large set of SAS procedures which are already verified and widely used. We can save a huge amount of time and energy, which would have been needed in constructing and validating these procedures from scratch. This also helps us to compare results with other SAS applications. The following 15 SAS procedures are used in PEST 4:

| MODULE | PROCEDURE  |
|--------|------------|
| BASE   | PRINT      |
|        | DATASETS   |
|        | SORT       |
|        | MEANS      |
|        | APPEND     |
|        | UNIVARIATE |
| STAT   | GLM        |
|        | LOGISTIC   |
|        | PHREG      |
|        | LIFETEST   |
|        | TRANSPOSE  |
| IML    | IML        |
| GRAPH  | GPLOT      |

- C programs

Computation workload in the four main functions of PEST 4 are distributed between SCL codes and C programs. Whenever heavy computations, such as numerical integration, solving equations, or iterative processes, are required, we resort to C programs. The SAS/TOOLKIT module facilitates the writing SAS procedures in C, and hence provides the C interface.

In order to build a SAS procedure in C, you need to define a tiny ‘grammar’ describing the syntax of invocation of the SAS procedure. The grammar is transformed into a C program

that recognises the procedure name and passes its arguments to the C program. SAS *functions* have to be defined in a slightly different way. In the program file containing the entry function, several lines are needed to establish links between the SCL code invoking the procedure and the program.

SAS datasets are collections of records (observations in SAS terminology), which are again collections of items of various types and lengths. SAS/TOOLKIT provides a range of library functions for input/output. When reading in variables in SAS datasets, you can randomly access any item in a record. In order to write to a SAS dataset, you can define the format of the record in advance, and the actual construction of the record is automated. If some items are not defined at the time of writing, a predefined 'MISSING' value will be used instead. It is possible to pick up these 'MISSING' values later on through other SAS/TOOLKIT functions.

There is another set of library functions dealing with memory management. Allocating and freeing memory is one of most error-prone parts of programming that might cause a slowing down of the whole program. The library functions help programmers to avoid muddling in this. At the beginning of the program you can define a global buffer (memory pool in SAS terminology) with several properties. Later on, when you need a certain amount of memory, you can fetch it from this buffer. You can put it back for re-use when it becomes unnecessary. All operations for address management are hidden from the programmer.

## (8) Conclusion

Sequential methodology is a powerful tool for obtaining firm conclusions about drug efficacy with minimal number of patients. However, it is a complicated procedure to organise and conduct. There are several programs commercially available, which support practitioners in all phases of this complex procedure. PEST 4 is one such software package, which covers all necessary computational functions, and also has a simulation facility. Because of the availability of software, the advantages of sequential clinical trials have become much more accessible to clinicians.

## References

- Medical and Pharmaceutical Statistics Research Unit (2000). *PEST 4: Operating Manual* : University of Reading
- Senn, S (1997). *Statistical Issues in Drug Development* : Chichester, Wiley
- Whitehead, J (1997). *The Design and Analysis of Sequential Clinical Trials (Revised 2<sup>nd</sup> edition)* : Chichester, Wiley

## Acknowledgements

Development of PEST 4 is supported through sponsorship from AstraZeneca, Aventis, GlaxoSmithKline, Pfizer and Roche.