University of Washington
Seattle, WA. 98195-2255

Resource Facility for Population Kinetics
Department of Bioengineering 352255
Tel: (206)685-2009
(800)421-SAAM
Fax: (206)543-3081
e-mail: foster@saam.washington.edu

January 29, 1999

To:

RFPK Advisors	RFPK Collaborators/Consultant	RFPK Personnel
Atam Dhawan	David Bourne	Gail Badner
Alan Forrest	George Drusano	Hugh Barrett
David Giltinan	Crispin Pierce	Brad Bell
Don Heald	Danny Shen	Jim Burke
Tom Ludden	John Slattery	Claudio Cobelli
Loren Roskos		Michael Macaulay
Lew Sheiner		Roger Neate
David Young		Alan Schumitzky
JonWakefield		Mary Smith
		Paolo Vicini
		Mitch Watrous

From: David Foster

Subject: **RFPK Advisory Committee Meeting February 11, 1999, 202 Electrical Engineering/Computer Science Building (EECS)**

Thank you all for being willing to participate in our first Advisory Committee meeting for RFPK. The Agenda for the meeting follows. This first meeting should be very interesting since the overall goals are to review the work we have done in the first 11 months of RFPK, and to assess what we plan to do both in the short (next year) and longer terms. In keeping with the wishes of NCRR, the funding unit of the NIH for RFPK, there are very few faces who were involved with the various Advisory Committee meetings associated with the SAAM II project. However, we have learned much from the SAAM II project, and this is reflected in our overall strategy for software development in RFPK.

The overall goals for RFPK are the development and application of new software tools for population kinetics. While most of the money and effort in RFPK itself is on the algorithm and software design/development, we must also focus on the best way to promote the application of the new software tools. I would like to keep this in mind as the meeting progresses.

For this first meeting, the overall goals are:

- To discuss/assess the computational functionality we would like to see in the software;
- To discuss/assess the standard operating procedures and quality assurance plans;
- To discuss plans for service, training and dissemination.

As with the SAAM II Advisory Committee meetings, I plan to serve primarily as a moderator of discussions. **Please keep whatever notes you wish, but leave them (or a copy) when you leave since this will help us write the minutes of the meeting.** A draft of these minutes will be sent to you to annotate, and the final version created. These minutes are extremely important for us since they serve as a top level guide in decision making.

cc: Dr. Richard Dubois

RFPK ADVISORY COMMITTEE MEETING

AGENDA

Wednesday, February 10

6:00PM Wine reception and light dinner Casa Foster (Pick up will be arranged)

Thursday, February 11 (Electrical Engineering/Computer Science Building – Room 202)

8:00 Take shuttle from Silver Cloud Inn (coffee and breakfast rolls will be provided at the meeting site)

8:30 Welcome and introduction to RFPK – David
{Discuss the agenda and goals of the meeting. Discuss the organizational chart for RFPK, and personnel changes that have been made, or will be made.}

9:00 Project 1. Modeling Theory (Bell, Burke, Schumitzky) and Review of Existing Software: NONMEM (Watrous)

What has been done
How this will impact software design/development
What needs to be done
Comments on the critique of the original proposal *{Point here is to assess the critique of the study section (see enclosed), and how we are dealing with the issues raised.*

10:45 Coffee break

11:00 Project 2. Design and Development of Software Systems (Neate, Badner, Bell, Watrous, Vicini)

Life Cycle Overview
SPK Concept Description *{SPK is the first software deliverable; we will present an overview of this document, and after lunch, spend time on the details.}*

12:00 Lunch *{1 ½ hours is planned so Advisory, Collaborators and RFPK personnel will have plenty of time to interact on an individual basis}.*

1:30 SPK Concept Description (Continued) and Project Strategy (Neate, Badner, Bell, Watrous, Vicini)

2:30 Collaboration status report – Barrett,Vicini

3:00 Coffee/tea break

3:30 Service and Training: planned teaching meetings – Foster

4:00 Dissemination: use of the web – Foster, Bourne

4:30 Forgotten items

5:00 Return to Silver Cloud, or remain at RFPK for additional discussion

7:00 Dinner together (Silver Cloud Inn visitors will be picked up at 6:45)

RFPK ADVISORY COMMITTEE MEETING

February 11, 1998, 202 Electrical Engineering/Computer Science Building

INFORMATION ON TOPICS TO BE CONSIDERED

Introduction

RFPK, being in its first year and being the natural successor to the Resource Facility for Kinetic Analysis, the resource which developed the SAAM II software system, provides an excellent framework for us all to work together to produce new and innovative software tools for population kinetic analysis. In addition, we can work to develop new ideas to promote the application of these tools so that they can be more widely applied in biomedical and pharmaceutical research.

The following information is in support of the topics to be considered for discussion at the Advisory Committee Meeting. First let me summarize the structure of RFPK in terms of the projects. Remember each NCRR/BTP Resource Facility must have the following five components:

- Core Research and Development
- Collaborative Projects
- Service
- Training
- Dissemination

We are enclosing a copy of the proposal for Projects 1 and 2 in the RFPK grant application. This provides additional information for those wishing it beyond what is summarized below.

Structure of RFPK

The two core research and development are summarized below.

Project 1. Modeling Theory

In this project, we will develop new or modify existing theories for the algorithms necessary for parametric, nonparametric and two-stage methods of population kinetics. For each algorithm to be used in our software, a precise statement of the required functionality will be written with the assistance of our collaborators to ensure their needs are met. This functionality document will be used as the basis for the software specification, design and development in Project 2.

Project 2. Software Design and Development

This project will implement the theories and algorithms developed in Project 1 by designing and coding software tools which will make them easily available to the biomedical research community. It will include specifying and designing how the user will access the theory and algorithms. The project will include the specification, design, implementation, testing, validation, release and maintenance of the software. RFPK will write a set of standard operating procedures and establish a quality assurance plan that will meet IEEE standards. The software tools will be developed incrementally to make testing and validation easier.

The six collaborative projects are summarized below. We have chosen these projects to include both traditional areas of application, i.e. pharmacokinetic and pharmacokinetic/pharmacodynamic studies, and new areas such as traditional metabolic tracer kinetic studies. We would anticipate, as was the case in RFKA, that some service projects will lead to new collaborations, so this list should change over the course of the grant. These projects are in various states of activity; they will be reviewed at the meeting.

Project 3. Application of Population Kinetics in Clinical Pharmacology Studies (Schumitzky, Foster and Jelliffe.

This project includes, among other things, the possibility of porting SPK to the supercomputer; we need to assess this strategy.

Project 4. Application of Population Kinetics in PK/PD Studies: Development of Integrated Population Pharmacokinetic/Pharmacodynamic Relationships for Antiviral Chemotherapy (Schumitzky and Drusano)

This project will start next year.

Project 5. Application of Population Kinetics in PK/PD Studies: Population Concentration-Effect Relationships of Cyclophosphamide Metabolites in Hematopoietic Stem Cell Transplantation. (Schumitzky, Foster and Slattery)

This project will start next year.

Project 6. Application of Population Kinetics to Lipid and Lipoprotein Metabolic Studies (Barrett, Foster and Huff).

Dr. Barrett will review progress and plans on this project.

Project 7. Application of Population Kinetic Studies in Intermediary Metabolism (Cobelli, Foster, Toffolo, Vicini, Bergman, Avogaro and Vettor).

Drs. Cobelli and Vicini will review progress and plans on this project.

Project 8. Application of Population Kinetics to Environmental Toxicokinetic Studies (Schumitzky, Foster, Shen, Pierce, Vicini).

Dr. Vicini will review progress and plans on this project.

Plans for Service and Training will be discussed at the Meeting. A review of where we are with Dissemination on the Web is given below.

<u>Software Deliverables</u>

While the end goal of RFPK will be comprehensive compartmental population kinetic deliverables with an MS-Windows interface, we will achieve this goal by sequentially developing a series of deliverables which can be used by select collaborators or other software developers seeking comparable computational capabilities. *To evaluate where we are in the process, it is necessary to summarize our overall plan in the development process.*

There are five methodologies which were described in the proposal as potential for implementation: the global two-stage, the "NONMEM" method, Lindstrom/Bates, Maximum Likelihood, and nonparametric. Each will be evaluated for potential implementation. Others will also be developed and/or evaluated as required. Fully Bayesian will not be implemented in the first five-year cycle of RFPK; *we feel this can serve as the basis for the first competing renewal for RFPK.*

The deliverables for each of the five methods are:

SPK	The basic population kinetic subroutine
SDPK	The subroutine that simulates population kinetic data
SCPK	The compartmental population kinetic subroutine
PCPK	A batch-mode compartmental population kinetic program
GCPK	A compartmental population kinetic program with an MS-Windows interface

Computationally, specialized integrators and optimizers will be required. Internally, these will be created as modules. When deemed appropriate, these modules will be fully documented and made available to other investigators wishing to incorporate these specialized technologies in their software products.

Each deliverable is explained in detail below.

SPK: Subroutine for Population Kinetics

This will be a subroutine that performs population kinetic analysis using the methods developed in Project 1. This deliverable, since it is the computational workhorse for each method, will be scheduled for the earliest possible release for use by our collaborators. It will thus be tightly linked to Project 1 since all theoretical work for each method must be completed before the design can start. It will be designed to run on distributed systems and perhaps a supercomputer.

The user will be required to program a specific model in order to use this routine. Thus this product will not be restricted to any specific type of modeling. Its input values will be checked and if an error is detected, an error code will be returned. This error code will specify the exact nature of the error from the user's point of view.

An example of SPK user defined input is:

- a model that describes the mean and variance for each data point as a function of all parameters;
- define the fixed and adjustable parameters
- define the covariates on an experimental time line (that also includes any perturbations, i.e. "changing the conditions" of the experiment;
- prescribe the measurement values (data); and
- provide initial parameter estimates for the adjustable parameters.

The output from SPK will be the population kinetic parameters.

One major goal of this meeting is to review the Concept document for SPK. This document is at a point where it needs to be thoroughly evaluated since we are about to hand it to the software engineers.

SDPK: Simulation of Data for Population Kinetics

This deliverable will be a subroutine that performs Monte Carlo simulations of population kinetic data. The input to this routine will include the values that SPK determines (i.e. a set of parameter values in a format similar to that used by SPK). The output of this routine will be a simulated version of the data that is required by SPK as input.

SCPK: Subroutine for Compartmental Population Kinetics

This deliverable will be a subroutine that performs compartmental population kinetic analysis. It will use SPK to perform the population kinetics part of its work. It will include multiple methods for solving the differential equations for compartmental models (e.g. Runge-Kutta and stiff integrators). The subroutine will have a standard interface. The user will define the compartmental model using a data structure instead of having to program it directly. Thus this product will require less work, on the part of the user, than SPK. On the other hand, it will be restricted to the compartmental modeling case. Its input values will be checked

and, if an error is detected, an error code will be returned. This error code will specify the exact nature of the error from the user's point of view.

PCPK: Program for Compartmental Population Kinetics

This deliverable will be a program that both simulates and analyzes population kinetic data sets. It will use SDPK and SCPK to perform the majority of its work. It will be restricted to population kinetics of compartmental models. The user will define the compartmental model and data values using a file structure. This product will require less work on the part of the user than the routine described above. This program will be validated in accordance with IEEE guidelines. Its input values will be checked and if, an error is detected, an error message will be printed. This message will specify the exact nature of the error including exactly where in the input file the error was detected.

GCPK: A GUI (graphical user interface) for Compartmental Population Kinetics

This deliverable will contain a graphical user interface to invoke SDPK to simulate data sets and SCPK to analyze data sets. The user will define a specific model using graphical drawing tools. This deliverable will have a context sensitive help system so that specific help can be obtained for each dialog, window, or error message. It will also read and write files that have the format used by the program PCPK. Its input values will be checked and, if an error is detected, an error message will be displayed. This message will specify the source of the error from the user's point of view.

It is our intent that GCPK will make the initial model development and testing process for population kinetic analysis easy, but that PCPK will be the deliverable which, because it operates in batch mode, will be used to generate final numerical values to FDA standards.

Standard Operating Procedures and Quality Assurance Plan

We have spent considerable time working on the Standard Operating Procedures that will govern all aspects of the software development conducted by RFPK, and on developing a Quality Assurance Plan. These will be discussed as part of the agenda. They are included as an appendix to this document since they are written in html and hence, once we decide to do it, can be put on the RFPK homepage.

Concept Description for SPK

The Concept Description will be reviewed at the meeting.

Plans for the RFPK Web site

RFPK has started to work on its website. You can find it at

http://muir.saam.washington.edu

We would request that you check the site out. There are a number of decisions to be made as to what information should be made available on this site.

Points raised by Site Visitors

The Study Section, in the overall critique, offered a number of suggestions to be considered. These need to be reviewed/addressed at this meeting.

1. Enhance expertise in GUI development.
2. Work out a detailed time line which includes additional user feedback during the design process with resulting interaction. Specify intermediate milestones for the next product release.
3. In addition to input from "expert" users, "naïve" users should also be engaged early in the process. Perhaps educational programs offered would be a medium for this input.

4. Specify standard operating procedures.
5. Consider the links between Projects 1 and 2, and attempt to strengthen them.
6. More graduate students as part of RFPK would be desirable.

For Project 1 (Theory), there were essentially no concerns except when designing the Monte Carlo studies, there was a question concerning the decision to base all parametric algorithms on the ELS procedure; alternative methods more robust to model (of the mean and of the variance) misspecification are recommended. The reviewers pointed out that various quasi-ELS methods have been proposed in which the mean and covariance are estimated interactively in separate steps. The question is whether or not to deal with the quasi-ELS methods.

There were a number of points raised in terms of Project 2 (Software Development)

- The design process needs feedback from (potential) user groups.
- Appropriate time-lines for intermediate validation and evaluation including GUI design need to be established.
- SOP's should be specified. The software is to be designed according to IEEE standards.
- A software engineer with expertise in GUI design needs to be brought on board.
- The linkage between Projects 1 and 2 is fuzzy, and can be worked out by carefully designing intermediate goals and feedback strategies.
- Adding only one high-end PC per year might be too little.

Service/Training. The Study Section recommended:

- As usage builds, it will be necessary to name a full-time service and training coordinator thus freeing the time of Hugh Barrett. There are concerns about the over-commitment of Hugh Barrett.
- It will also be important to track usage. Tracking would indicate times when more user support or teaching meetings are needed.
- Use of the Internet (e.g. with FAQ) will be important
- Adding graduate students to the Resource would be a benefit. They are an area of training not considered in the grant. Also, a course should be planned in the Department of Bioengineering.
- Department support of release time and a teaching assistant for internal courses developed would help promote the Resource and increase its visibility at the University.
- Engage early naïve users. Early feedback would be important to determine overall strategies. It might be important to start training on a slower basis, as the new software comes on board. Also, the National Advisory Committee should be engaged in reviewing NIH suggestions.
- Space available and workstations are thought adequate for one to two persons at the time.

Dissemination: The Study Section commented:

- A local full-time Webmaster should be named.
- Every Resource document should be considered for posting in HTML format; this includes the SOP and other design documents which could invite user feedback.

Participant List

Advisors

1. Atam Dhawan
   Medical College of Ohio
   1014 Nitschke Hall
   University of Toledo
   2801 W. Bancroft Street
   Toledo, OH 43606-3390
   Phone: 419-530-8267 or 419-530-7391
   Fax: 419-530-7392
   Email: adhawan@eng.utoledo.edu

2. Alan Forrest
   Department of Pharmaceutics
   517 Hochstetter Hall
   SUNY/Buffalo
   Buffalo, New York 14260-1200
   Phone: (716) 645-2842
   Fax: (716) 645-3693
   Preferred Mailing Address: Millard Fillmore Hospital/Pharmacokinetics
   3 Gate Circle
   Buffalo, NY 14209

3. Donald Heald
   Rhone-Poulenc Rorer Pharmaceuticals Inc
   500 Arcola Road, 4C58
   P. O. Box 5094
   Collegeville, PA 19426-0998
   Phone: (610)454-5135
   Fax: (610)454-8107
   Email: donald.heald@rp-rorer.com

4. David Giltinan
   Medical Affairs
   Genentech, Inc.
   1 DNA Way
   South San Francisco, CA 94080
   Phone: (650) 225-1000
   Email: giltinan@gene.com

5. Thomas Ludden
   Pharmaceutical Sciences
   University of Nebraska Medical Center
   College of Pharmacy
   COP 3026B
   Omaha, Nebraska 68198-6025
   Phone: 402-559-9261
   Fax: 402-559-9543
   Email: tludden@unmc.edu

6. Lewis Sheiner
   Box 0626
   University of California at San Francisco
   San Francisco, CA 94143
   Phone: (415) 476-1965
   Email: lbs@c255.ucsf.edu

7. David Young
   GloboMax LLC
   7250 Parkway Drive, Suite 430
   Hanover, MD 21076
   Phone: 410-712-9500
   Fax: 410-712-0737
   Email: youngd@globomax.com

8. Loren Roskos
   Amgen Inc.
   1840 DeHavilland Drive
   Thousand Oaks, CA 91320-1789
   Phone: (805) 447-6302
   Email: lroskos@amgen.com

9. Jon C. Wakefield
   Department of Epidemiology and Public Health
   Imperial College School of Medicine
   Norfolk Place, LONDON W2 1PG
   Phone: +44 (0) 171 594 3328
   Fax: +44 (0) 171262 1034
   E-mail: j.c.wakefield@ic.ac.uk

Collaborators

10. David Bourne
    University of Oklahoma HSC College of Pharmacy
    Oklahoma City, OK 73117
    Phone: (405) 271-6481 x 47213
    Fax: (405) 271-7505
    Email: david@boomer.org or david-bourne@ouhsc.edu

11. George Drusano
    Division of Clinical Pharmacology
    Departments of Medicine and Pharmacology A-142
    Albany Medical College
    47 New Scotland Avenue
    Albany, NY 12208
    Phone: (518) 262-6330
    Fax: (518) 263-6333
    Email: gldrusano@aol.com

12. Roger Jelliffe
    USC Lab of Applied Pharmacokinetics
    2250 Alcazar St
    Los Angeles CA 90033
    Phone: (323) 442-1300,
    Fax: (323) 442-1302,
    Email: jelliffe@hsc.usc.edu

13. Crispin Pierce
    Environmental Health
    Box 357234
    University of Washington
    Seattle, WA 98195-7234
    Phone: 206-685-9537
    Email: crispo@u.washington.edu

Collaborators (Continued)

14. Danny Shen
    Department of Pharmaceutics
    H-272F Health Sciences, Box 357610
    University of Washington
    Seattle, WA 98195-7610
    Phone: 206-685-2920, 206-667-4583 or Vmail: 206-667-4583
    Fax: 206-543-3204
    Email: ds@u.washington.edu

15. John Slattery
    Department of Pharmaceutics
    H-272J Health Sciences, Box 357610
    University of Washington
    Seattle, WA 98195-2255
    Phone: 206-543-7736, 206-543-5900 or Vmail: 206-667-3372
    Fax: 206-543-3204
    Email: jts@u.washington.edu

Personnel

16. James Burke
    Department of Mathematics
    C-421 Padelford, Box 354350
    University of Washington
    Seattle, WA 98195-4350
    Phone: 206-543-6183, 206-543-1150
    Fax: 206-543-0397
    Email: burke@math.washington.edu

17. Claudio Cobelli
    Department of Electronics and Informatics
    University of Padova
    35131 Padova, Italy
    Tel. +39 (049) 827-7616, Fax: +39 (049) 827-7699
    Email: cobelli@dei.unipd.it

18. Alan Schumitzky
    Department of Mathematics
    Denney Research Building, DRB 155
    University of Southern California
    Los Angeles, CA 90089-1113
    Tel. (213) 740-2392, Fax: (213) 740-2424
    Email: schum@mtha.usc.edu

Note-- All personnel below are located in the RFPK Office, 241 Aerospace Research Building, Box 352255, University of Washington, Seattle, WA 98195-2255, Fax: 206-543-3081:

19. Gail Badner
    Software Engineer
    Phone: 206-543-3016
    Email: badner@saam.washington.edu

20. Hugh Barrett
    Associate Professor
    Phone: 206-685-2009
    Email: barrett@saam.washington.edu

21. Bradley Bell
    Senior Mathematician
    Phone: 206-616-9424
    Email: brad@apl.washington.edu

22. David Foster
    Principal Investigator
    Phone: 206-685-2009
    Email: foster@saam.washington.edu

23. Michael Macaulay
    Systems Administrator
    Phone: 206-685-2009
    Email: macaulay@apl.washington.edu

24. Roger Neate
    Project Manager
    Phone: 206-543-4120
    Email: neate@saam.washington.edu

25. Mary Smith
    Operations Manager
    Phone: 206-685-3021
    Email: mchsmith@u.washington.edu

26. Paolo Vicini
    Assistant Professor
    Phone: 206-685-2009
    Email: vicini@saam.washington.edu

27. Mitchell Watrous
    Software Engineer
    Phone: 206-543-3016
    Email: watrous@saam.washington.edu

SECTION 2. RESEARCH PLAN

A. Specific Aims

The objective of this proposal is to establish a new Resource Facility for Population Kinetics (RFPK) to promote the application of computer modeling in biomedical research focusing on population kinetic analyses. We will develop and maintain new software systems designed to address issues in population kinetics, and support the application of these systems in biomedical research through collaborative studies, service, training and dissemination.

The **overall goal** of our Resource Facility will be to provide computer modeling assistance to the biomedical research community. To achieve this goal, we will:

1. develop new modeling technologies and apply them to study biological systems;
2. specify, design, develop, test, validate and maintain new software systems which incorporate the parametric, nonparametric and two-stage methods of estimating population parameters in kinetic studies;
3. provide service to the biomedical community via consultation in mathematical modeling, and in the experimental design and analysis of metabolic kinetic and pharmacokinetic/pharmacodynamic data;
4. educate and train individuals in the use of modeling technology in biomedical research; and
5. disseminate our technology, expertise and accomplishments.

The **development of new modeling technologies and implementing them in our software systems** (goals 1 and 2) are described in Projects 1 and 2:

Project 1. Modeling Theory

In this project, we will develop new or modify existing theories for the algorithms necessary for parametric, nonparametric and two-stage methods of population kinetics. For each algorithm to be used in our software, a precise statement of the required functionality will be written with the assistance of our collaborators to ensure their needs are met. This functionality document will be used as the basis for the software specification, design and development in Project 2.

Project 2. Software Design and Development

This project will implement the theories and algorithms developed in Project 1 by designing and coding software tools which will make them easily available to the biomedical research community. It will include specifying and designing how the user will access the theory and algorithms. The project will include the specification, design, implementation, testing, validation, release and maintenance of the software. These activities will follow the guidelines established by the Food and Drug Administration, and will meet IEEE standards. The software tools will be developed incrementally to make testing and validation easier.

The **application of the new technologies to study biological systems** (goal 1) is described in our collaborative projects 3 - 7. All take advantage of the collective strengths of resource personnel and our collaborators. The collaborative projects will not only bring together investigators from around the world to help in their research, but will aid in designing the software, and opening new areas of application for the software. With the integrated packages we propose to develop, we expect significant growth in both the pharmacokinetic and metabolic kinetic areas of research.

D. Research Design and Methods

In accordance with BTP guidelines, we have divided the Research Design and Methods into the following sections:

- Core Research and Development

  Project 1: Theory and algorithms for population kinetics
  Project 2: Design and development of software systems for population kinetics

- Collaborative Research

  Project 3: Application of population kinetics in clinical pharmacology studies
  Project 4: Application of population kinetics in pharmacokinetic/pharmacodynamic studies
  Project 5: Application of population kinetics in lipid and lipoprotein kinetic studies
  Project 6: Application of population kinetics in intermediary metabolism
  Project 7: Application of population kinetics in environmental toxicokinetic studies

CORE RESEARCH AND DEVELOPMENT

Introduction

The goal of Projects 1 and 2 is the production of the software system deliverables described below. In Project 1, we describe how we will develop the necessary theory and algorithms for our population kinetic deliverables. In Project 2, we describe how these theories and algorithms will be designed to produce easy-to-use software tools.

Project 1: Modeling Theory and Development of Algorithms

Project Leaders: Alan Schumitzky and Brad Bell
Co-Investigators: Jim Burke, Claudio Cobelli, David Foster, Paolo Vicini
Collaborators: David Bourne, University of Oklahoma
Roger Jelliffe, University of Southern California

Specific Aims

1. Develop convergent numerical algorithms for the parametric and nonparametric estimators in population kinetic analysis to be implemented in Project 2.

2. Analyze the statistical properties of the estimators in Specific Aim 1: consistency, asymptotic normality, asymptotic confidence regions and hypothesis testing.

3. Investigate efficiency and robustness of the estimators in Specific Aim 1 via simulation studies.

For the parametric case, we will develop four algorithms: a "true" maximum likelihood (ML) algorithm, a Global Two Stage (GTS) algorithm, a NONMEM type algorithm, and a Lindstrom-Bates type algorithm. The ML algorithm is based on Monte Carlo integration for evaluating the objective function. The GTS, NONMEM, and Lindstrom-Bates type algorithms are all based on the extended least squares (ELS) method. For the nonparametric case, we will develop a Mallet type algorithm for mixed effects models.

For the parametric case, consistency is a difficult issue. Consistency means that the estimated values converge to the true values as the number of subjects gets arbitrarily large. It is important to note that the original estimation procedures of NONMEM and Lindstrom-Bates are not consistent for general nonlinear models. The only algorithm that is consistent relative to the true parameter values is the true maximum likelihood algorithm. For the class of ELS algorithms we develop, there is a generalized notion of consistency, which means that the estimated values converge to the values that best approximate the model. We will investigate the required theory for the generalized consistency and asymptotic normality of these algorithms. The formulas for the asymptotic confidence intervals and hypothesis testing will follow from the same theory.

For the nonparametric case, the consistency of the method, relative to the true values of the model, has already been established. What remains then is the determination of the asymptotic confidence intervals for estimated parameters such as means, medians, trimmed means, etc. At present these results have not been derived for the nonparametric case. We will use the theory of maximum likelihood estimation in infinite dimensional spaces for this purpose.

Efficiency of an (unbiased) estimator is measured by the generalized variance of estimated values, with the Cramer-Rao lower bound being optimal. Relative efficiency of two estimators compares the corresponding generalized variances. Robustness measures how an algorithm performs when there are violations in the model and/or probability distribution assumptions. By utilizing Monte Carlo simulation studies, these properties can be investigated without requiring asymptotic conditions.

Background

Of central interest in kinetic analysis is the relationship between a given experimental protocol and the resulting effect. This cause/effect relationship usually differs markedly between individual subjects. By definition, population kinetic analysis is the methodology used to quantify this intersubject variability.

Population kinetic analysis is widely used in pharmacokinetic/pharmacodynamic (PK/PD) studies since it is the key to understanding how drugs behave in humans and animals. More specifically, it provides the foundation for the intelligent design of dosage regimens to treat disease processes. In metabolic studies, it is used to identify which parameters in a model change when a population of normal subjects is compared to a population of subjects with a known pathological condition. Here it is used to identify potential aberrant pathways which is a first step in planning an intervention. It is also used to analyze the action of a specific intervention in a baseline and treated state. Finally, population kinetic analysis is necessary in situations where there is not sufficient data on each subject to estimate the individual subject parameters.

There are three significant obstacles in such modeling efforts: (i) there is no one software package that includes both parametric and nonparametric methods, (ii) certain methods currently in use have not had the rigorous mathematical analysis and software design that one could desire, and (iii) current software is either limited in modeling capabilities or is not user friendly. That is, available software packages that have general modeling capabilities, do not have a user friendly graphical interface for model definition and data handling; and available

software packages that have a user friendly graphical interface, allow only a limited pre-determined set of possible models.

In this research project, we propose to develop a population modeling package which corrects this situation.

This proposal brings together a unique group of researchers to overcome these problems by developing a population kinetic analysis program that includes both parametric and nonparametric methods. Existing theory will be rigorously reviewed, and new theories developed. Access to the computational machinery and data handling will be implemented using modern software engineering philosophy. The result of the proposed work will be a software package that is powerful, flexible and easy to use.

Preliminary Results

Our results here apply to both parametric and nonparametric population models. They are described in detail in Schumitzky (1991ab, 1992, 1993, 1995) and Spieler and Schumitzky (1993). (Because of the large number of references in Project 1, references are cited using parenthetical documentation. The Reference List is at the end of the Project 1 description.)

Theoretical and algorithmic development

Schumitzky (1991a, 1992, 1993) considered the nonparametric approach. Maximum likelihood estimation was analyzed. Various implementations of this method were studied. These included the Nonparametric Expectation-Maximization (NPEM) method (Schumitzky, 1991a), and the variation (Schumitzky, 1993) of the Nonparametric Maximum Likelihood (NPML) method of Mallet (1986). In these papers, consistency and the ability of these nonparametric methods to handle routine clinical data were confirmed by careful simulation studies.

Our work on extended least squares (ELS) estimators was initiated in Spieler and Schumitzky (1993). There we analyzed the NONMEM algorithm based on first order linearization (Beal and Sheiner, 1982, 1992). Our main results concerned the consistency of such estimators. When the linearization is centered about the assumed true population parameters, we showed that the resulting estimator is not consistent. When the linearization is centered about the current Bayesian estimates of the individual parameters (Lindstrom and Bates, 1990), we showed that the resulting estimator has better consistency properties. Although classical consistency was not expected for this latter case either.

More general work on ELS estimators was done in Bell and Schumitzky (1997ab). There we developed a numerically convergent algorithm for calculating these estimators, and derived the generalized notion of consistency and asymptotic normality for this situation. Formulas for the asymptotic confidence intervals for such estimators follow directly from these results.

Software development.

Schumitzky and his colleagues at USC have developed a population kinetic analysis program package: NPEM (nonparametric EM algorithm). The current NPEM program accommodates the linear three compartment model used by the current USC*PACK clinical programs and by the Multiple Model dosage regimen program, and has been part of that package for over three years, see (Schumitzky *et al.* 1994) and (Bayard *et al.* 1994). This model permits up to 7 PK parameters.

The NPEM program estimates the population parameter distributions in PK models from routine clinical data. It runs on a PC. It is based on the nonparametric EM algorithm (Schumitzky, 1991a). A key feature is that no parametric assumptions about the form of the population distribution are required. Thus this approach will discover populations with unanticipated non-normal and multi-modal distributions. The parametric methods, in contrast, cannot do this.

Bell has designed the numerical kernel of SAAM II, which includes a number of new numerical routines. One such routine allows for estimation of parameters and weights in multiple data sets (Bell, Burke and Schumitzky, 1996). Bell (1995) has also developed a new numerical algorithm for implementing the Iterated Kalman filter which will be useful in the population kinetic analysis program in the parametric case. Bell has worked on design of large systems programs and graphical user interfaces. Bell is also the co-developer of the Matrix Language O-Matrix (Bell, Paisley and Trippel, 1994).

POP3CM

Substantial progress has been made in the development of a mixed effects parametric program. The program accommodates the open two compartment model with multiple inputs with a graphical user interface. More general models are readily available. For example, the nonlinear glucose/insulin minimal model has been developed. For the two compartment model, a general analytic solution for the multiple dosing case is employed. The partial derivatives of the solution with respect to the microparameters are also calculated analytically. These analytic derivatives are used to provide analytic gradients for the resulting optimization algorithms described in Specific Aim #1. A prototype version of the parametric program has been written by Bell in the language O-Matrix.

Methods

SPECIFIC AIM #1. Develop convergent numerical algorithms for the parametric and nonparametric estimators in population kinetic analysis to be implemented in Project 2.

We begin by stating the problem precisely and define the relevant notation. Also for convenient equation referencing, we will label sections and equations numerically.

1. Problem Statement / Mixed Effects Models

Consider a sequence of experiments. Each experiment can have its own experimental protocol. Mathematically, each experiment is represented by a collection of finite dimensional random vectors (y_i, x_i, v), $i=1,\ldots, N$. The random vector y_i is observed. The random vector x_i takes values in a finite dimensional set X and is not observed. (The $\{x_i\}$ are the random effects.) The vector v takes values in a finite dimensional set V and is not observed. (The components of v are the fixed effects.) The components of y_i contain all the measurements for the i^{th} experiment, e.g., serum concentrations, urine amounts, etc. The components of x_i represent the unknown model and noise parameters for the i^{th} experiment that change from subject to subject, e.g., rate constants, volumes, Michaelis-Menten constants, etc. The components of v represent the unknown model and noise parameters that do not change from subject to subject, e.g., scale parameters in the variance of the measurement noise.

It is assumed that the conditional density $p_i(y_i \mid v, x_i)$ of y_i given v and x_i is known. This conditional density $p_i(y_i \mid v, x_i)$ completely describes the probabilistic model for the i^{th} subject. In the biostatistical literature, such models are said to have "mixed effects", i.e., both fixed effects v and random effects x_i.

We consider the general nonlinear regression model of the form:

(1.1) $$y_i = H_i(v^*, x_i) + G_i(v^*, x_i)\, e_i$$

where H_i is a known continuous vector-valued function and G_i is a known continuous matrix-valued function which is assumed to be positive definite. H_i and G_i depend on all the experimental conditions of the i^{th} experiment, e.g., dose amounts and dose and sampling times, forcing functions, and other covariates. The vector v^* is the true but unknown vector of fixed effects. The $\{e_i\}$ are independent multivariate normal random (noise) vectors, each with zero mean and unit covariance. The $\{e_i\}$ and $\{x_i\}$ are mutually independent. One of the strengths of our estimation procedure is the ability to include unknown covariance and model parameters in the matrix G_i (Bell and Schumitzky, 1997ab).

Given the model (1.1), it is easy to define $p_i(y_i \mid v, x_i)$, namely

(1.2a) $$p_i(y_i \mid v, x_i) = k(y_i - H_i(v, x_i),\ R_i(v, x_i)), \qquad R_{i\,i}(v, x_i) = G_i(v, x_i)^T G_i(v, x_i)$$

where $k(y, D)$ is the multivariate normal density function

(1.2b) $$k(y, D) = [(2p)^q \det D]^{-1/2} \exp\{-\frac{1}{2}\ y^T D^{-1} y\},\ q = \text{dimension of } y.$$

What connects all the experiments together is the following basic assumption:

The $\{x_i\}$ are independent and identically distributed with common (but unknown) probability distribution F^* defined on X.

The population kinetic analysis problem is then to estimate v^* and F^* based on the data $\{y_i\}$.

Definition of Maximum Likelihood

Let the data $\{y_1, \ldots, y_N\}$ be fixed. Given a parameter v, and a distribution F, the density of y_i is:

(1.3) $$p_i(y_i \mid v, F) = \int p_i(y_i \mid x, v)\, dF(x).$$

By the independence assumptions on $\{e_i\}$ and $\{x_i\}$, the $\{y_i\}$ are independent, and the log likelihood of the data is

(1.4) $$L(v, F) = \sum_{i=1}^{N} \log p_i(y_i \mid v, F).$$

Let F be a set of probability distributions on X. A probability distribution $F^{ML} \in F$ and a vector $v^{ML} \in V$ will be called a maximum likelihood estimate of (v^*, F^*), if

(1.5) $$(v^{ML}, F^{ML}) = \arg\min \{L(v, F):\ v \in V, F \in F\}$$

In the parametric case, F is assumed to be a family of distributions defined by a finite number of parameters. In the most important example, this family F is the class of all multivariate normal distributions with unknown mean vectors and unknown covariance matrices. In the nonparametric case, F is assumed to be the family of all distributions defined on X.

Basic Requirements

No matter what estimation procedure is used, the following requirements should ideally be satisfied.

1. Procedure is numerically convergent

2. Procedure is consistent.

3. Procedure can handle routine clinical data. (As little as one data point per subject.)

4. Procedure provides confidence intervals for all estimated parameters.

For all of the methods we propose in this grant application, we will investigate these basic requirements. Note, for the nonparametric case, Item 4 will apply to the estimated means, medians, covariances, etc. of the estimated distribution $\hat{F}$.

2. Two Stage Methods

Two Stage Methods are the simplest algorithms in population kinetic analysis. They only apply in data rich situations, so they do not satisfy our Basic Requirement 3 (routine clinical data). However, they will be appropriate for many of the studies in the Lipid Metabolism and Intermediary Metabolism collaborations. They are important because they can be applied in cases where the models are so large that most other methods of population kinetic analysis are not be feasible.

Assume enough data are available for each subject in the population to accurately estimate that subject's unknown parameters. (This assumption violates Requirement 3.) Thus for the i^{th} subject let

$$\hat{x}_i = \text{some estimate of } x_i \text{ based on } y_i$$

and

$$V_i = \text{some estimate of } Cov[\hat{x}_i].$$

Usually $\hat{x}_i$ is determined by an extended least squares algorithm and V_i is determined by the corresponding asymptotic analysis (Bell, Burke and Schumitzky, 1996) and (Bell and Schumitzky, 1997ab).

Standard Two Stage

The Standard Two Stage (STS) method is nonparametric and can be defined as follows: The nonparametric estimator F^S of F* is given by the empirical (or sample) distribution of the $\{\hat{x}_i\}$. F^S is a discrete distribution with mass $1/N$ at each point $\hat{x}_i$. Note that F^S uses the original data $\{y_i\}$ only through the estimates $\{\hat{x}_i\}$, and completely ignores the $\{V_i\}$ information. Nevertheless this estimator is easy to calculate and can be very useful.

From the distribution F^s any desired population statistics can be estimated. For example, the estimates of the mean and covariance are given by:

$$m^s = \frac{1}{N}\sum_{i=1}^{N} \hat{x}_i \qquad D^s = \frac{1}{N}\sum_{i=1}^{N} (\hat{x}_i - m^s)(\hat{x}_i - m^s)^T$$

Finally, the STS estimates of the population mean vector and covariance matrix can be used as starting values for the more complicated methods to be described below.

Global Two Stage

The Global Two Stage (GTS) method is parametric and is considerably more complicated than the STS method. The GTS method is one of the oldest population analysis methods in the literature. It first appears in the work of Mones Berman in his original plans for SAAM. One of the first references to this method appears to be a 1967 thesis of K. Pettigrew (1964), see Lyne *et al.*. (1992).

The GTS method is motivated as follows. First it is tentatively assumed that

$$\hat{x}_i = x_i + u_i \quad u_i \sim N(0, V_i)$$

In all that follows, we use the notation: $x \sim N(m, D)$ to mean that the random vector x has a multivariate normal distribution with mean vector m and covariance matrix D. This is known to be approximately true if there are sufficiently many data for the i^{th} subject (Bell, Burke and Schumitzky, 1996). Next, it is assumed that $x_i \sim N(m^*, D^*)$, where m^* and D^* are unknown. For independent $\{x_i\}$ and $\{u_i\}$ it follows:

(2.1) $$\hat{x}_i = m^* + w_i, \qquad w_i \sim N(0, V_i + D^*).$$

Given these assumptions, the GTS estimates of m^* and D^* are the vector $\hat{m}$ and positive definite matrix $\hat{D}$ that minimize the function

(2.2a) $$J_0(m, D) = \sum_{i=1}^{N} \log\det S_i + r_i^T S_i^{-1} r_i \qquad S_i = V_i + D \text{ and } \quad r_i = \hat{x}_i - m$$

i.e.,

(2.2b) $$(\hat{m}, \hat{D}) = \arg\min\{J_0(m, D)\}$$

Note that the GTS method uses the original data $\{y_i\}$ only through the estimates $\{\hat{x}_i, V_i\}$.

GTS Algorithms

The GTS optimization problem (2.2) is of relatively high dimension. If the dimension of x is equal to q, there are $d = q+q*(q+1)/2$ free variables in the pair (m, D). (Thus for example, if q = 9, then d = 54.) Because of this, there have been many algorithms proposed for its solution.

GTS algorithms basically fall into two categories: direct and indirect. The indirect approach relates the GTS to an EM algorithm. See Steimer *et al.* (1985) and Schumitzky (1995) for a more detailed discussion. In the case of very large models, this EM approach may be the only one feasible. The direct approach minimizes (2.2) directly. This gives rise to a problem of the ELS type. This is the method we will investigate, as we will also use this ELS method in the NONMEM and Lindstrom-Bates type algorithms.

For the direct approach, our algorithm takes advantage of three important numerical features of the problem: elimination of constraints by factorization, reduction of dimensionality, and the use of analytic derivatives.

Elimination of constraints by factorization. The matrix D in $J(m, D)$ is constrained to be positive definite. This is equivalent to a complicated system of nonlinear constraints involving the components of D, namely, that all the principal minors of D are positive. On the other hand, D is positive definite if and only if D admits the Cholesky factorization: $D = C\,C^T$, where C is a lower triangular matrix with positive diagonal elements. Thus the original constraints on D are equivalent to the simple linear constraints that the diagonal elements of C are positive. An equivalent objective function is then given by $K(m, C) = J(m, C\,C^T)$.

Further reduction of dimensionality. Define

$$m(C) = [\sum_{i=1}^{N} S_i^{-1}]^{-1} \sum_{i=1}^{N} S_i^{-1} \hat{x}_i \tag{2.3}$$

Then for each fixed C, $m(C)$ minimizes $K(m, C)$. (Note that the partial of $K(m, C)$ with respect to m is zero at $m(C)$ and that K is convex in m.) Thus an equivalent objective function is given by $L(C) = K(m(C), C)$, where C is positive definite. Now the objective function $L(C)$ has q fewer variables than the function $K(m, C)$.

Analytic derivatives. Many optimization methods require the gradient of the objective function. The following result yields a formula that is faster and more accurate than approximating the derivative by function differences. The partial derivative of $K(m, C)$ with respect to the $(i,j)^{th}$ component of C, is the (i,j) component of the matrix M:

$$M = 2\,[\sum_{i=1}^{N} S_i^{-1} - S_i^{-1} r_i r_i^T S_i^{-1}]\,C\,. \tag{2.4}$$

A similar formula is given in Jamshidian and Jennrich (1993, eq. (27)).

We propose to implement the above GTS type algorithm for Project 2.

Linear Models

The GTS machinery described above can easily be extended to a slightly more general situation. This will be important in the next section. Therefore define the linear (Gaussian) model by the system:

$$y_i = A_i x_i + u_i\,, \quad i = 1, \ldots, N. \tag{2.5}$$

Here A_i is a known matrix; $x_i \sim N(m^*, D^*)$, where m^* and D^* are unknown; $u_i \sim N(0, V_i)$, where V_i is known. Further assume $\{x_i\}$ and $\{u_i\}$ are independent. The problem is to estimate (m^*, D^*) based on the data $\{y_i\}$.

The objective function for this problem is exactly the same as that given in (2.2) where now:

(2.6) $$S_i = V_i + A_i D A_i^T, \quad r_i = y_i - A_i m.$$

Formulas (2.3) and (2.4) can then be easily extended to this case as seen in Jamshidian and Jennrich (1993) eqs. (26)-(27).

3. Parametric Case - Nonlinear Mixed Effects Models

Consider again the model of eq. (1.1):

(3.1a) $$y_i = H_i(v^*, x_i) + G_i(v^*, x_i)\, e_i$$

and now assume that

(3.1b) $$x_i \sim N(m^*, D^*), \quad e_i \sim N(0, I_i)$$

where m^* is an unknown mean vector, D^* is an unknown covariance matrix, and I_i is the identity matrix of dimension depending on i. Thus each subject may have a different number of observations.

Thus for given values of m, D and v, eqs. (3.1ab) imply that the individual likelihood and log likelihood are:

(3.2a) $$p_i(y_i \mid v, m, D) = \int p_i(y_i \mid v, x)\; p(x \mid m, D)\, dx$$

(3.2b) $$L(v, m, D) = \sum_{i=1}^{N} \log p_i(y_i \mid v, m, D)$$

where

$$p_i(y_i \mid v, x) = k(y_i - H_i(v, x),\; R_i(v, x)), \quad p(x \mid m, D) = k(x-m, D).$$

Recall that $k(\cdot, \cdot)$ is the multivariate normal density function defined in eq. (1.2b).

The maximum likelihood estimate $(\hat{v}, \hat{m}, \hat{D})$ of (v, m^*, D^*) is then defined as the maximizer of $L(v, m, D)$ in eq. (3.2b), i.e.,

(3.2c) $$(\hat{v}, \hat{m}, \hat{D}) = \arg\max \{L(v, m, D)\}$$

Computational Considerations

There are two computational problems in determining the maximum likelihood estimate $(\hat{v}, \hat{m}, \hat{D})$: the integration in eq. (3.2a) and the optimization in eq. (3.2c). Of these two problems, the integration is the most serious one. For typical PK/PD problems, x can be 10 dimensional or greater, so the integration in eq. (3.2a) poses a formidable numerical problem.

There have been only a few algorithms proposing to attack this problem directly. One such is the "smooth nonparametric" method of Davidian and Gallant (1993). In their approach, the integration is performed by Gauss-Hermite quadrature. This quadrature method is very efficient for integrals of the type in eq. (3.2a). The drawback is that this quadrature formula depends on the values of *(m, D);* and must be reapplied at each evaluation of the objective function in the optimization of eq. (3.2c). We therefore propose using a quadrature formula which is independent of the values of *(m, D).* For the NPEM algorithm (Schumitzky, *et al.*, 1994), we have had particular success with the quasi Monte-Carlo integration method of Korobov, see Deak (1990). Recent work on the Monte-Carlo integration required for the optimization in eq. (3.2c) (for relatively high dimensional *x*) is discussed in Geyer (1996).

The optimization problem in (3.2c) must also be addressed. We have used the derivative-free Nelder-Mead algorithm (Press, *et al.* 1987, p. 289) for similar optimization problems in NPEM2 (Schumitzky, et al. 1994). Also gradient methods, like conjugate gradients (Press, *et al.* 1987, p. 301), can be used. In this case the derivative of *L(v, m, D)* must be calculated. Formulas for these derivatives are given in (Schumitzky 1995). These derivatives can be calculated using the same type of quadrature formula as used for *L(v, m, D).*

We propose implementing this true Maximum Likelihood (ML) algorithm in Project 2, at least in situations where the dimension of the vector *x* is small to moderate.

This ML method is consistent and asymptotically efficient under very general hypotheses. Further all of the Basic Requirements are satisfied. Thus this ML method can then be used as a standard to which the two stage methods and linearization methods (described below) can be compared.

NONMEM

The first serious population kinetic analysis method was the NONMEM (NONlinear Mixed Effects Model) algorithm of Beal and Sheiner (1982). In the original version, a first order linearization of eq. (1.1) resulted in the approximate equation:

(3.3) $$y_i \approx H_i(v,m) + \frac{\partial}{\partial x} H_i(v,m)(x_i - m) + G_i(v,m)e_i$$

It follows that $p_i(y_i \mid v, m, D) \approx N(r_i, U_i)$ where

$$r_i = y_i - H_i(v,m), \quad U_i = [\frac{\partial}{\partial x} H_i(v,m)]^T D[\frac{\partial}{\partial x} H_i(v,m)] + R_i(v,m)$$

In this case, maximizing the corresponding approximate likelihood function is equivalent to minimizing the function:

(3.4a) $$J_1(v,m,D) = \sum_{i=1}^{N} [\log\det U_i + r_i^T U_i^{-1} r_i]$$

The NONMEM estimate $(\hat{v}, \hat{m}, \hat{D})$ of *(v*, m*, D*)* is then the minimizer of J_1 with respect to *(v, m, D);* i.e.

(3.4b) $$(\hat{v}, \hat{m}, \hat{D}) = \arg\min\{ J_1(v, m, D) \}$$

The NONMEM algorithm satisfies the Basic Requirements 3-4. But it was shown by Spieler and Schumitzky (1993), and Vonesh and Chinchilla (1997, p. 353 - 358) that for nonlinear models, the NONMEM algorithm using " first order linearization" is not necessarily consistent.

This inconsistency of NONMEM is not surprising. Whenever there is an approximation to the original maximum likelihood problem, the consistency of the maximum likelihood method is lost. However, the NONMEM method defined above satisfies a generalized notion of the consistency property found in all ELS estimators. This means that the estimated values converge to the values that best approximate the model relative to the limiting objective function, see White (1994) for more details. Further, under suitable hypotheses, such ELS estimators also are asymptotically normal. Whether or not these generalized notions of consistency and asymptotic normality can be exploited will be studied. At the minimum, these generalized notions imply that the corresponding estimators are converging to something in a well defined way.

We propose to implement the above NONMEM type algorithm in Project 2. Special attention will be focused on the numerical convergence of the minimizer in (3.4b), see Bell and Schumitzky (1997a).

Lindstrom - Bates Method

In the linearization of eq. (3.3), the random vector x_i is approximated by its mean value m. To improve upon this approximation, Lindstrom and Bates (1990) proposed a refinement of the linearization whereby the random vector x_i is approximated by its Bayesian estimate $\hat{x}_i$ (defined below). The original method as derived in Lindstrom and Bates (1990), required that

$$G_i(v, x) = G_i(v)$$

be a function of v only. This then leads to a linearization of eq. (1.1) of the form:

$$y_i \approx H_i(v, \hat{x}_i) + \frac{\partial}{\partial x} H_i(v, \hat{x}_i)(x_i - \hat{x}_i) + G_i(v)e_i \tag{3.5}$$

where $\hat{x}_i$ is defined by the optimization problem

$$\hat{x}_i = \arg\min\{J_2(v, m, D, x): \ x \in X\} \tag{3.6}$$

$$J_2(v, m, D, x) = (x - m)^T D^{-1} (x - m) + (y_i - H_i(v, x))^T [R_i(v)]^{-1} (y_i - H_i(x))$$

Now, conditional on $\hat{x}_i$, it follows that

$$p(y_i \mid v, m, D, \hat{x}_i) = k(q_i, W_i)$$

where

$$q_i = y_i - H_i(v, \hat{x}_i) + \frac{\partial}{\partial x} H_i(v, \hat{x}_i)(\hat{x}_i - m)$$

$$W_i = [\frac{\partial}{\partial x} H_i(v, \hat{x}_i)^T D \frac{\partial}{\partial x} H_i(v, \hat{x}_i)] + R_i(v)$$

In this case, maximizing the corresponding approximate likelihood function is equivalent to minimizing the function:

(3.7) $$J_3(v, m, D) = \sum_{i=1}^{N} \log \det W_i + q_i^T (W_i)^{-1} q_i.$$

Since the optimization problem ineq. (3.6) is defined in terms of the unknown parameters *v, m, D,* an iterative algorithm is proposed as follows:

(3.8a) Choose initial conditions $(v, m, D)^0$. Substitute these values in eq. (3.6)

(3.8b) Calculate $\hat{x}_i$ from eq. (3.6), for $i = 1, ..., N$. Substitute these vectors into eq. (3.7).

(3.8c) Optimize J_3 with respect to (v, m, D) to get updated estimate $(v, m, D)^1$. Go to (3.8a).

Computational Considerations

A prototype version of this algorithm has been written by Bell in the language O-Matrix (Bell, Paisley and Trippel, 1994). There are two optimizations in the algorithm defined by eq. (3.8). The optimization in eq. (3.8b) is done using a nonlinear least squares algorithm. The optimization in eq. (3.8c) is done using the conjugate gradient algorithm. This version of the Lindstrom-Bates method is employed in our current version of the POP3CM program.

Generalized Lindstrom - Bates Method

We now consider the case when $G_i(v, x)$ is a function of v and x. This corresponds to model weighting of data as opposed to the less rigorous data weighting. Equations (3.5)-(3.7) become:

(3.5)' $$y_i \approx H_i(v, \hat{x}_i) + \frac{\partial}{\partial x} H_i(v, \hat{x}_i)(x_i - \hat{x}_i) + G(v, \hat{x}_i) e_i$$

where $\hat{x}_i$ is defined by the optimization problem

(3.6)' $$\hat{x}_i = \arg\min \{ J_2(v, m, D, x): \; x \in X \}$$

$$J_2(v, m, D, x) = \sum_{i=1}^{N} [(x_i - m)^T D^{-1} (x_i - m) + (y_i - H_i(v, x_i))^T [R_i(v, x_i)]^{-1} (y_i - H_i(x_i)) + \log \det R_i(v, x_i)]$$

Solving the optimization problem (3.6)' is more complicated than (3.6). However, such problems were studied in Bell and Schumitzky (1997b).

Now conditional on $\hat{x}_i$, it follows that $p(y_i \mid v, m, D, \hat{x}_i) = k(q_i, W_i)$, where

$$q_i = y_i - H_i(v, \hat{x}_i) + \frac{\partial}{\partial x} H_i(v, \hat{x}_i)(\hat{x}_i - m)$$

$$W_i = = [\frac{\partial}{\partial x} H_i(v, \hat{x}_i)^T D \frac{\partial}{\partial x} H_i(v, \hat{x}_i)] + R_i(v, \hat{x}_i)$$

In this case, maximizing the corresponding approximate likelihood function is equivalent to minimizing the function:

(3.7)' $$J_3(v, m, D) = \sum_{i=1}^{N} \log \det W_i + q_i^T (W_i)^{-1} q_i.$$

The new algorithm then is the same as defined in (3.8) but with (3.6), (3.7) replaced by (3.6)', (3.7)'. We propose to implement this Lindstrom-Bates type algorithm in Project 2.

Remarks,

a) The algorithm defined in (3.8) does not lead to an Optimization Estimator as defined in Specific Aim #2. That is because there is no one fixed objective function to be optimized. Further, there is no proof of convergence for this algorithm. One way of modifying this algorithm is as follows. For fixed (v, m, D) write $\hat{x}_i$ in eq. (3.6) as $\hat{x}_i(v, m, D)$. Then the function $J_3(v, m, D)$ defined by (3.7) is a function only of (v, m, D), albeit more complicated than before. We will investigate the possibility of modifying the Lindstrom-Bates algorithm so that there is a proof of convergence and the resulting algorithm is computationally feasible.

b) We emphasize that the three algorithms proposed of GTS type, NONMEM type and Lindstrom-Bates type, are all formulated as an ELS problem. So much of the work on these three algorithms will overlap. Further, the numerical algorithm proposed for solving ELS problems will be based on the results of Bell and Schumitzky (1997ab). This should lead to a proof of convergence for these algorithms.

4. Nonparametric Methods

In the nonparametric approach, no parametric assumptions about the form of the underlying population distribution are made. The entire distribution is estimated from the population data. It thus allows for heavy-tailed, non-normal, and multimodal distributions.

From a computational standpoint, the nonparametric approach has many desirable features which are not shared by any of the parametric methods.

- The likelihood function $L(v, F)$ is a concave function of the distribution F.
- The likelihood function $L(v, F)$ has an unique global maximum with respect to F.

  No local extrema or critical points as in parametric methods

- Algorithm for calculation maximum likelihood estimate is based on convexity theory

  Avoids linearization. Avoids numerical integration.

From a statistical standpoint, the nonparametric approach also has many desirable features.

- Estimator is consistent.
- Estimated distribution can discover "hidden"covariates, such as would occur, for example, in a population of fast and slow acetylators.
- Estimated distribution has a discrete density.

The detection of hidden covariates was used by Preston and Drusano (1996) to discover unexpected subpopulations in clinical trials withganciclovir. The discrete density was used in Bayard *et al.* (1994) to determine optimal dosage regimen design. Further, this discrete density for the estimated prior distribution is the basis for the Multiple Model dosage regimen programs described in Project 3.

Random Effects Models

We again focus our attention on the maximization problem ofeq. (1.5). We first describe our approach only for the random effects case. Thus assume that the conditional density $p_i(y_i \mid x)$ does not depend on any unknown fixed parameter v. In this case, for an assumed distribution F, the density of y_i is given by:

$$p_i(y_i \mid F) = \int p_i(y_i \mid x)\, dF(x).$$

By the independence of the $\{y_i\}$, the log likelihood of the data is

$$L(F) = \sum_{i=1}^{N} \log p_i(y_i \mid F).$$

But now assume that the family of allowable distributions $\mathcal{F}$ is the family of all distributions defined on X.

A probability distribution F^{ML} will be called a maximum likelihood estimate of F^* if :

(4.1) $$F^{ML} = \arg\min\{L(F): F \in \mathcal{F}\}.$$

The following theorem was proved by Lindsay (1983, Section 3) and by Mallet (1986, Section 3):

Let $g_i(x) = p_i(y_i \mid x)$ be continuous on X. Then F^{ML} can be found in the class of probability distributionswith discrete measures supported at N (or fewer) points in X. (N = number of subjects.)

This beautiful result still leaves a formidable optimization problem to be solved if attacked by direct methods. A discrete distribution F supported at M points depends on parameters:

$$Q = \text{support}(F) = \{x_1, .., x_M\}, \quad x_j \in X$$

$$W = \text{weighs}(F) = \{w_1, \cdots, w_M\} \quad w_j \geq 0 \text{ and } \sum_{i=1}^{M} w_i = 1.$$

In this case write: $F = F(W, Q)$. The number of free parameters in $F(W, Q)$ is then $d = M*q + M - 1$, $q=dim(X)$. For example, if q= 4, M = 188, then d = 939.

However, it is also shown in Lindsay (1983) and Mallet (1986) that (4.1) is equivalent to certainwell known problems in optimal design and convexity theory. Additionally, Schumitzky (1991a) provided two nonparametric EM algorithms for solving (4.1); one "Continuous" and one "Discrete." The Continuous EM algorithm is utilized in the USC-PC Pack programs (Schumitzky, Jelliffe and Van Guilder, 1994). Other optimization techniques are discussed inSchumitzky (1992, 1993), Bohning (1988) and Laird (1978).

Algorithms

To state these algorithms, we need some additional notation. We first discuss a special case of the relationship between a probability distribution F and its corresponding probability measure dF: namely the case where the measure is discrete with finite support.

Define the Dirac measure d_q by: $d_q(A) = 1$, if $q \in A$ and $d_q(A) = 0$, if $q \notin A$,

for any Borel set A in X. Then a probability measure dF is said to be ***discrete*** with ***finite support*** if:

$$dF = \sum_{i=1}^{n} w_j d_{q_j} \quad \text{where} \quad q_j \in X, w_j \geq 0 \quad \text{and} \quad \sum_{j=1}^{n} w_j = 1$$

The set of points $\{q_1,..., q_n\}$ is called the ***support*** of F. For any function $g(q)$ defined on X, it follows:

$$\int g(q)dF(q) = \sum_{j=1}^{n} w_j g(q_j)$$

Next, we define a function which is important in allnonparametric approaches:

$$D(x,F) = \sum_{i=1}^{N} \frac{p(y_i \mid x)}{p(y_i \mid F)} - N \qquad D(F) = \max\{D(x,F) : x \in X\}$$

where S is some subset of X. The function $D(x, F)$ is essentially the derivative of $L(F)$ in the direction of the discrete distribution supported at x. The importance of this function is seen in the following Equivalence Theorem of Kiefer and Wolfowitz(see Lindsay (1983) and Mallet (1986)):

(4.2) $F = F^{ML}$ *if and only if* $D(F) = 0$.

The basic algorithm is due toFedorov (1972).

Basic Algorithm

1. Let $dF = \sum_{i=1}^{n} w_j d_{q_j}$ be the current candidate for the measure dF^{ML}.

2. Calculate $D(F)$. If $D(F) = 0$, then $F = F^{ML}$. Stop.

3. Let $q_{max} \in \arg\max\{ D(q, F): q \in X \}$ and define $dF_s = (1-s)\, dF + s\, d_{q_{max}}$.

   Then let $s_{max} = \arg\max\{ L(F_s): s \in [0,1] \}$ and $dF_{new} = dF_{smax}$.

4. Set $dF = dF_{new}$ and go to Step 1.

It is shown in (Fedorov 1972) that the Basic Algorithm generates a sequence of likelihood increasing discrete measures which tend to F^{ML} as the number of iterations tends to infinity. It is further shown that the line search problem required in the definition of s_{max} has an analytical solution. The main drawback of this algorithm is that the number of support points of F is increased at each iteration.

Mallet Algorithm

Mallet has suggested two improvements to the Basic Algorithm. The first improvement is to add an additional optimization of F_{new} in Step 3, now with respect to all the weights $\{w_j\}$ keeping the support points $\{q_j\}$ fixed. The second improvement is to restrict the number of support points in F_{new} to be less than or equal to N without decreasing the likelihood. The resulting algorithm has the same convergence properties as the Basic Algorithm. The Mallet algorithm is significantly more efficient at the expense of being more computationally intensive. The exact details of this algorithm are found in (Mallet 1986).

Remarks.

a) The Federov/Mallet algorithm generates a sequence of likelihood increasing discrete measures which tend to F^{ML} as the number of iterations tends to infinity. Consistency of the corresponding estimate F^{ML} then follows from the theory of Kiefer-Wolfowitz (1956, Result (2.12)).

b) In the Mallet algorithm, the optimization of F_{new} in Step 3, with respect to all the weights $\{w_j\}$ is done by a convex program. This optimization can be more simply performed by an EM algorithm (Schumitzky 1991a, 1993) as follows:

If $F_{new} = F(W, Q)$ with $W = \{w_1,..,w_n\}$, then set

(4.3) $$w_j' = \sum_{i=1}^{n} p(x_j \mid y_i, F) \quad W' = \{w_1', \cdots, w_n'\}$$

where $$p(x_j \mid y_i, F) = w_j \frac{p(y_i \mid x_j)}{p(y_i \mid F)}.$$

It is shown in (Schumitzky 1991a) that $L(F(W,Q)) \le L(F(W',Q))$. The iteration in (4.3) is then performed until "convergence".

The only drawback of this simple algorithm is that the convergence is very slow. However, numerous acceleration methods are available (see, Jamshidian and Jennrich (1993), Silvey, Titterington and Torsney (1978), and Bohning, D. (1985)). Additional variations on the Federov/Mallet Algorithm have been suggested by Bohning (1985, 1988).

c) The stopping rule in Step 3 is implemented in practice as follows:

Given eps: If $D(F) \leq eps$. Stop.

It then follows from Federov (1972) that $0 \leq L(F^{ML}) - L(F) \leq eps$.

d) Starting values for Q and W are easily found. For example, the ML or MAP estimate for each individual x_i is usually very good. And it suffices to take $w_i = 1/M$.

We propose to implement the above Mallet type algorithm in Project 2.

Additionally, we will investigate improvements on this algorithm based on recent developments in convexity theory. For example, the Mallet algorithm is essentially a steepest descent method on the space of probability measures. Steepest descent methods in general can be very slow. Higher order convergence rates may be possible using interior point methods, see Nesterov and Nemirovsky (1994).

Semi-Infinite Programming

One approach that has yet to be seriously considered is the connection between the optimization problem (4.1) and certain semi-infinite programming problems (Lesperance and Kalbfleisch 1992, Section 3.3). Let W be the positive orthant in N dimensional Euclidean space and for $u = (u_1, .. , u_N)$ in U, define the functions

$$k(u) = \sum_{i=1}^{N} \log u_i \quad K(u,x) = \sum_{i=1}^{N} g_i(x) u_i - N \quad \text{where} \quad g_i(x) = p_i(y_i \mid x)$$

Consider the semi-infinite programming problem:

(4.4) Maximize $k(u)$ over U subject to the constraints: $K(u, x) \leq 0$, for all x in X.

It is shown by Lindsay (1983) that from the solution to (4.4), one can obtain F^{ML}. In particular, if u^* is the optimal solution to (4.4), then the support of F^{ML} is just the finite set on which the constraints are active, i.e. $K(u^*, x) = 0$.

Problem (4.4) requires the maximization of a simple concave function subject to an infinite number of linear constraints, of which only a finite number can be active. Such semi-infinite programming problems were studied by Bell (1990). Also relevant are new results in the theory of barrier methods, see Kaliski *et al.* (1966) and interior point methods, see Nesterov and Nemirovsky (1994). We also note that the methods of Kaliski, *et al.* (1996) have been developed for parallel computers. We propose to investigate this semi-infinite programming problem as a means of solving the nonparametric maximum likelihood problem.

Mixed Effects Models

Assume now that the conditional density $p(y_i \mid v, x_i)$ also depends on the fixed parameter vector v as in eq. (1.2). Here there is a problem with most of the nonparametric algorithms. Namely, solving the optimization problem:

(4.5) $(v^{ML}, F^{ML}) = \arg\max \{L(v, F): (v, F) \in (V, F)\}$

requires finding: $F^{ML} = \arg\max \{L(v, F): F \in F\}$, for each fixed v and then optimizing over v. This is called "Profiling." For small dimensional v this is a reasonable method. But for moderate to large dimensional v, this procedure is very inefficient. The theory of convex optimization shows how to do the maximization in v and F simultaneously, see Clark (1983). Important in this approach is the fact that Step 3 in the basic algorithm above results in a steepest descent method. We propose to develop this approach.

SPECIFIC AIM #2. Analyze the statistical properties of the estimators in Specific Aim 1: consistency, asymptotic normality, asymptotic confidence regions and hypothesis testing.

Consistency - Parametric Case

In the parametric case, the only method with proven consistency properties is the true maximum likelihood method of eqs. (3.2). In Spieler and Schumitzky (1993), a methodology based on the Prediction Error Method of Ljung, see Caines(1988), was initiated to analyze the asymptotic properties of estimators defined by algorithms such as NONMEM and Lindstrom-Bates. Using this methodology it was proved, for example, that the NONMEM method was not consistent for general nonlinear models. Also a simple example was given to illustrate this fact. A similar example is given in Vonesh and Chinchilla (1997,pp 354-357). The Lindstrom-Bates algorithm (defined above by eqs. (3.6)-(3.8)) was consistent for this example, but this was just a special case. In general, consistency of the Lindstrom-Bates estimator has not been proved or disproved. The results of Vonesh (1996) show that consistency for a special case of the Lindstrom-Bates algorithm was held when the number of subjects and the number of measurements per subject both tend to infinity.

Optimization Estimators and ModelMisspecification

Classical consistency means that the parameter estimates tend to the true parameter values as the number of observations goes to infinity. This assumes that the model specified by the estimation method is correct. It is shown in White (1994), that estimators defined by optimization problems, called optimization estimators, have, under very general conditions, certain consistency and asymptotic normality properties. However, when there is model misspecification, these properties are not necessarily related to the true model parameters, but to certain auxiliary parameters defined by the misspecified models. These auxiliary parameters are the best approximation to the true model parameters based on the limiting form of the objective function.

For the linearization type methods of NONMEM and Lindstrom-Bates, there is always one level of model misspecification. Namely, the model specification given byeqs. (3.3) or (3.5) does not correspond exactly to the model specified by eq. (1.1), which is presumed to generate the data. A similarmisspecification holds for the GTS algorithm. On the other hand, the GTS, NONMEM and the Lindstrom-Bates methods result in optimization estimators. The resulting generalized notion of consistency and asymptotic normality and corresponding asymptotic confidence intervals were derived in Bell andSchumitzky (1997b). The results will be applied here. In addition to this theoretical investigation, a practical analysis of this generalized consistency and asymptotic normality will be investigated by the Monte Carlo simulation methods described in Specific Aim 3.

Confidence Regions and Hypothesis Testing

Asymptotic confidence regions and hypothesis testing criteria are related to asymptotic normality as follows. Let $\hat{w}_n$ be any estimator based on n measurements such that $\hat{w}_n \to w^0$ a.e. for some point w^0 in a p-dimension compact set W. Assume $n^{1/2}(\hat{w}_n - w^0)$ converges in distribution to a normal random vector with mean 0 and covariance matrix S^{-1}. Let S_n be a consistent estimator of S. From this result we obtain the formul as for asymptotic confidence regions and asymptotic hypothesis tests as follows. Let $k(w)$ be a continuously differentiable function defined on W with values in R^q, $q \le p$; and let $K(w) = \frac{d}{dw} k(w)$ be the qxp-dimensional Jacobian. Consider the hypothesis H_0: $k(w^0) = k^0$ vs. H_1: $k(w^0) \le k^0$. Define the statistic:

$$T^2 = n(k(\hat{w}_n) - k^0)^T [K(\hat{w}_n) S_{nn} K(\hat{w}_n)^T]^{-1} k(\hat{w}_n) - k^0)$$

It is can be shown that under the hypothesis H_0, T^2 approaches a chi-squared distribution with q degrees of freedom, as $n \to \infty$. See White (1994, Theorem 8.10) for this and other related results. From this result follow the formula for the asymptotic confidence region for k^0, and the criteria for the test of H_0 vs. H_1.

The asymptotic normality of the estimators in this proposal are a consequence of the theory of optimization estimators, see White (1994). To keep this proposal as self contained as possible, we present briefly the results which are relevant to the concept of generalized asymptotic confidence intervals. We first consider the Optimization Estimator (OE) defined as follows. Let

$$Q_n(w) = \sum_{i=1}^{n} q_i(w, y_i)$$

where w belongs to a compact set W in p-dimensional space; $q_i(w, y)$ is a twice differentiable function of w; and the $\{y_i\}$ are independent random vectors. Note that the objective function in eqs. (2.2), (3. 2), and (3.4) are of this form with

$$q_i(w, y_i) = \frac{1}{2}[\frac{(y_i - H_i(w))^2}{V_i(w)} + \log(V_i(w))]$$

for suitable functions $H_i(w)$ and $V_i(w)$.

In general, an OE estimate is defined as a vector $\hat{w}_n$ which minimizes $Q_n(w)$ subject to $w \in W$. The following results are proved in White (1994) under appropriate hypotheses:

Assume as $n \to \infty$, the limit $E(\frac{Q_n(w)}{n}) \to Q(w)$ exists, and there is a w^0 which is the unique minimizer of $Q(w)$. Then $\hat{w}_n \to w^0$ a.e. Further define

$$A_n(w) = \frac{1}{n} E_w[\frac{d}{dw} Q_n(w)^T \frac{d}{dw} Q_n(w) \quad \text{and} \quad B_n(w) = \frac{1}{n} E_w[\frac{d^2}{dw^2} Q_n(w)]$$

and assume that $A_n(w) \to A(w)$ and $B_n(w) \to B(w)$ uniformly in W, where $A(w)$ and $B(w)$ are positive definite matrices. Write $S_n(w) = A_n(w)^{-1} B_n(w) A_n(w)^{-1}$ and $S(w) = A(w)^{-1} B(w) A(w)^{-1}$. Then $n^{1/2}(\hat{w}_n - w^0)$ converges in distribution to a normal random vector with mean 0 and covariance matrix $S(w^0)$. Further, $S(\hat{w}_n) \to S(w^0)$ a.e.

In the special case that $q_i(w^*, y_i) = -\log f(y_i \mid w^*)$, where $f(y_i \mid w^*)$ is the probability density of y_i for some $w^* \in W$, then $w^0 = w^*$, $A(w) = B(w)$ and $S(w) = A(w)^{-1}$. This gives the traditional asymptotic confidence intervals for the "true" w^*. This special case applies to the ML estimator defined by (3.2). Calculation of the asymptotic confidence intervals in this case will also require numerical calculation of certain integrals.

For the GTS, NONMEM and Lindstrom-Bates methods, the situation is more complicated. The above theory gives a computational formula for the asymptotic confidence intervals for w^0. This involves both the A and B matrices. Traditionally, what is done is to assume the approximate models in these methods are exact. This gives a simpler computational formula for the asymptotic confidence intervals, where it is assumed that $A=B$, see Davidian and Giltinan (1995, p. 141; p. 172). And further it is presumed that these confidence intervals are for the true parameter w^* rather than the parameter w^0 that is proscribed by the above theory. In our Monte Carlo simulations in Specific Aim 3, we will investigate this situation to see if the traditional asymptotic confidence intervals are valid.

Consistency - Nonparametric Case

Consistency of the maximum likelihood estimator (v^{ML}, F^{ML}) in eq. (4.5) of the nonparametric case is proved in the theory of Keifer and Wolfowitz (1956, Result 2.12).

Confidence Intervals

As opposed to the parametric case, it appears that no one has used the consistency of (v^{ML}, F^{ML}) to analyze the asymptotic confidence intervals for moments and other functional of F^{ML}, i.e., means, medians, standard deviations, trimmed means, etc. Such asymptotic estimates are important in providing confidence intervals for inferential statements. These asymptotic estimates can be gotten using the infinite dimensional maximum likelihood theory outlined in Wong and Severini (1991).

SPECIFIC AIM #3. Investigate efficiency and robustness of the estimators in Specific Aim 1 via simulation studies.

Efficiency Efficiency measures the dispersion of the estimator about the true values. The theorem of Cramer and Rao gives a lower bound for the generalized variance of an unbiased estimator. If a sequence of estimators is consistent and has the Cramer -Rao lower bound as its limiting generalized variance, then it is said to be asymptotically efficient. The relative efficiency of two estimators can be determined by comparing their asymptotic generalized variances.

Monte Carlo Simulations

The bias and efficiency of an estimator and relative efficiency between two estimators can be investigated using Monte Carlo simulation methods. Consider any estimation procedure applied to any population kinetic analysis

problem. The estimator $q = q(w)$ is a random vector which depends on the stochastic elements w of the problem. By replicating this problem stochastically M times, we generate a sequence $q_k = q(w_k)$, $k=1, ..., M$, of independent samples whose common distribution is the unknown distribution of q. For M sufficiently large (M ~ 1000), the empirical distribution of the $\{q_k\}$ will be a good approximation of the true (but unknown) distribution of q.

Many statistical questions can then be answered by employing standard statistical tests. Is the distribution of q approximately normal? This could be the case if the asymptotic conditions were valid in the original population problem. But normality is not necessary for further analysis. What is the bias (if any) in q? What is the variability of q? Given two estimation methods A and B, with corresponding estimators q_A and q_B, is q_A more efficient than q_B?

Further, the questions raised in Specific Aim 2 concerning the "correct" asymptotic confidence intervals can be answered by these Monte Carlo methods.

We only note here, that such Monte Carlo simulation methods are extremely powerful. Properties of estimation procedures can be investigated in many practical situations where asymptotic theory does not apply.

Robustness Robustness measures how an algorithm performs when there are violations in the model assumptions. In the population problem there are two models. The first is the intra-individual model which depends on certain PK/PD parameters. The second is the inter-individual model for the distribution of the PK/PD parameters over the population. In the parametric case, the inter-individual model for the distribution of the PK/PD parameters is multivariate normal. In cases where the true distribution is multimodal or heavy tailed, this is a bad assumption. The nonparametric methods make no such assumptions about this distribution and should be the most robust to violations of the inter-individual model. On the other hand, when the true inter-individual model is in fact multivariate normal, then the nonparametric method will be less efficient than the parametric method based on the correct model.

We shall conduct Monte Carlo simulation studies to measure the relative efficiency and robustness of all our proposed estimators . More specifically:

For parametric methods the standard benchmark program is the NONMEM system (Beal and Sheiner, 1992). Our algorithm of NONMEM type should behave similarly to the "first order linearization" method in the NONMEM system. Our algorithm of Lindstrom-Bates type should behave similarly to the "first order conditional" method in the NONMEM system. Our group is a licensed user of the NONMEM system, so that these comparisons can be verified. Then we plan to do head-to-head comparisons of our proposed GTS, NONMEM, and Lindstrom-Bates type algorithms.

For the nonparametric methods, we have proposed a random effects model and mixed effects model. For the random effects model, we have two comparison programs for our use. The first is a FORTRAN program graciously provided by A. Mallet which implements his algorithm (Mallet, 1986). The second program is NPEM2 developed by Schumitzky (1991a) which is part of the USC PC package. We emphasize that the nonparametric algorithm in NPEM2 is different from the nonparametric algorithm in this proposal.

To our knowledge, there is no other program available to analyze nonparametric mixed effects models. Here we will compare our proposed method with the parametric programs described above. In this case, comparisons will be made of only the first and second moments. Note: There are new Bayesian methods in development

which accommodate nonparametric mixed effects models. For example, see (Wakefield and Walker 1994). We will watch these developments closely.

Examples.

Since the nonparametric approach is not so widely known, we conclude with two simple examples to illustrate the random and mixed effects algorithms.

Example 1. (One Compartment Model - Random Effects)

This simple problem has been used as a benchmark for many methods, see (Steimer, Mallet, *et al.*, 1984). The model for i^{th} subject, j^{th} measurement is:

$$c_{ij} = \frac{100}{V_i}\exp(-K_i t_{ij}), \quad Cl_i = K_i V_i$$

where (V_i , Cl_i) are the volume of distribution and clearance, respectively, for the i^{th} subject; the $\{t_{ij}\}$ are the times at which measurements will be taken; and the $\{c_{ij}\}$ are the corresponding concentrations. Noisy observations are then measured:

$$y_{ij} = c_{ij} + e_{ij}, \quad e_{ij} \sim N(0, R_{ij}), \qquad R_{ij} = 0.0225 * (c_{ij})^2$$

In this case $x_i = (V_i , Cl_i)$ are the random effects assumed to be joint normal:

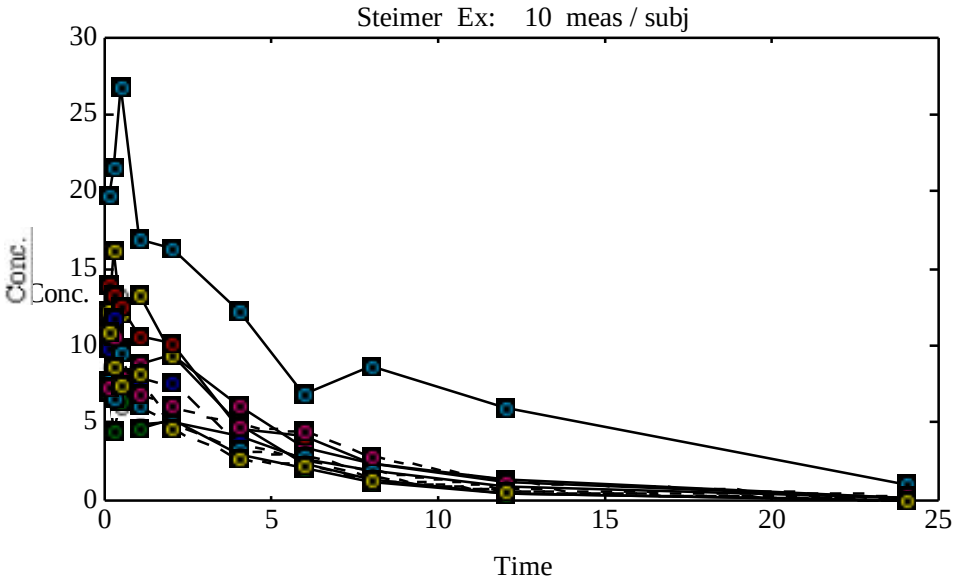
$$m_{Cl} = 2, \; m_V = 10, \; s_{Cl} = .60 \text{ (CV 30\%)}, \; s_V = 3.0 \text{ (CV 30\%)}, \; r = .80$$

Monte Carlo simulations are performed with 10 subjects and two measurement scenarios: "data rich" and "data poor". Namely,

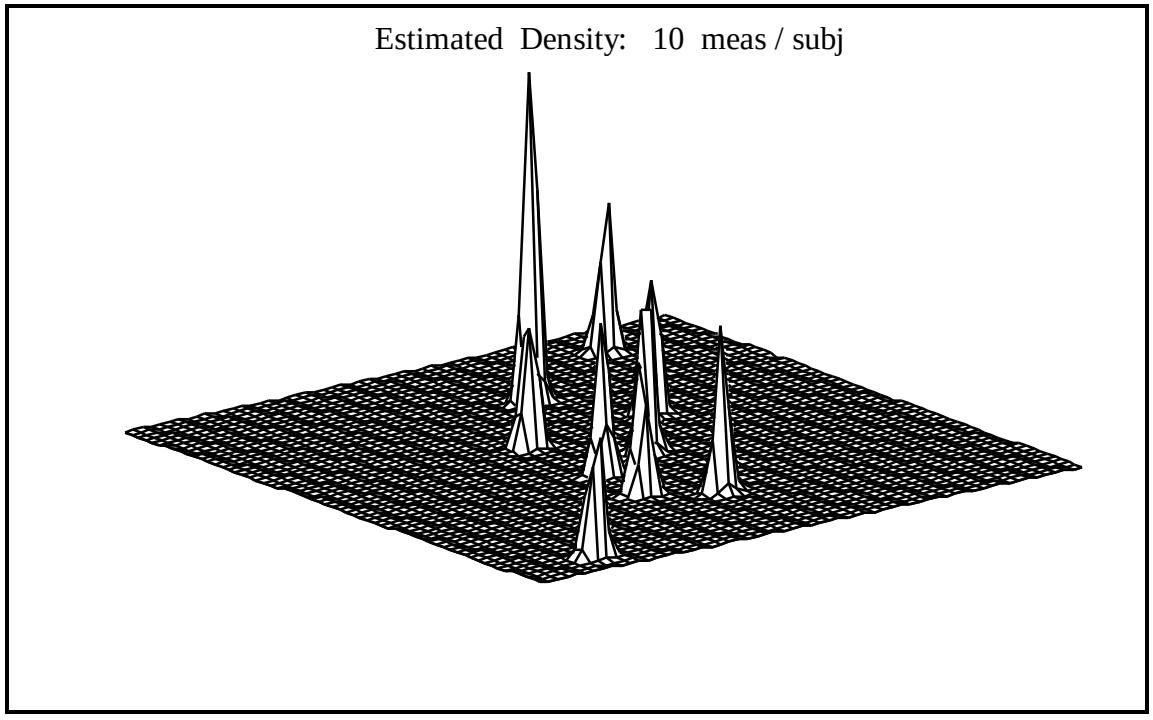
a) 10 measurements/subject at times {.1, .25, .5, 1, 2, 4, 6, 8, 12, 24}
b) 2 measurements/subject - measurements suboptimally selected

The figure below shows the Concentration versus Time graph for the simulation in case a).

RESULTS - Steimer Example



The next figure shows the (smoothed) estimated density of (V, Cl) for case a) using our Mallet type algorithm for random effects models. Cl is on the x-axis; V is on the y-axis. The correlation between V andCl is apparent.



Moments of this estimated distribution are easily obtained. The Table below shows the results. Note that there is practically no loss of accuracy in going from the 10 measurements per subject case to the 2 measurements per subject case.

Results from Steimer Example

	m_{Cl}	m_V	σ_{Cl}	σ_V	ρ
True	2.00	10.0	.60	3.0	.80
10 meas/subj	2.02	10.0	.67	3.1	.83
2 meas/subj	2.07	10.7	.63	3.2	.78

Example 2. (Common Means Problem - Mixed Effects)

The next example is an important problem in the history of maximum likelihood estimationIt was first posed by Neyman and Scott (1948). The problem is to estimate a fixed parameter when there are additionally an infinite number of "nuisance" parameters". This problem also motivated the nonparametric approach of Kiefer-Wolfowitz (1956).

The model for i^{th} subject, j^{th} measurement is:

$$y_{ij} = v + x_i e_{ij}, \quad e_{ij} \sim N(0, 1), \quad j = 1,..., 4; \quad i = 1,..., 12$$

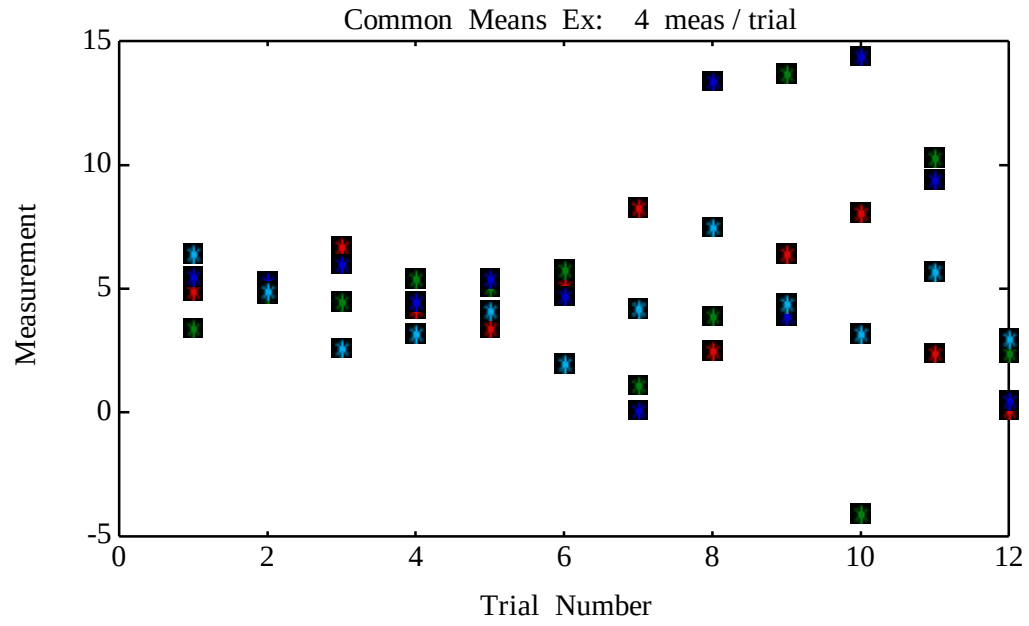
where v (the common mean) is the fixed parameter, and the standard deviations $\{x_i\}$ are the nuisance parameters. In our notation: v is the fixed effect and the $\{x_i\}$ are the random effects.

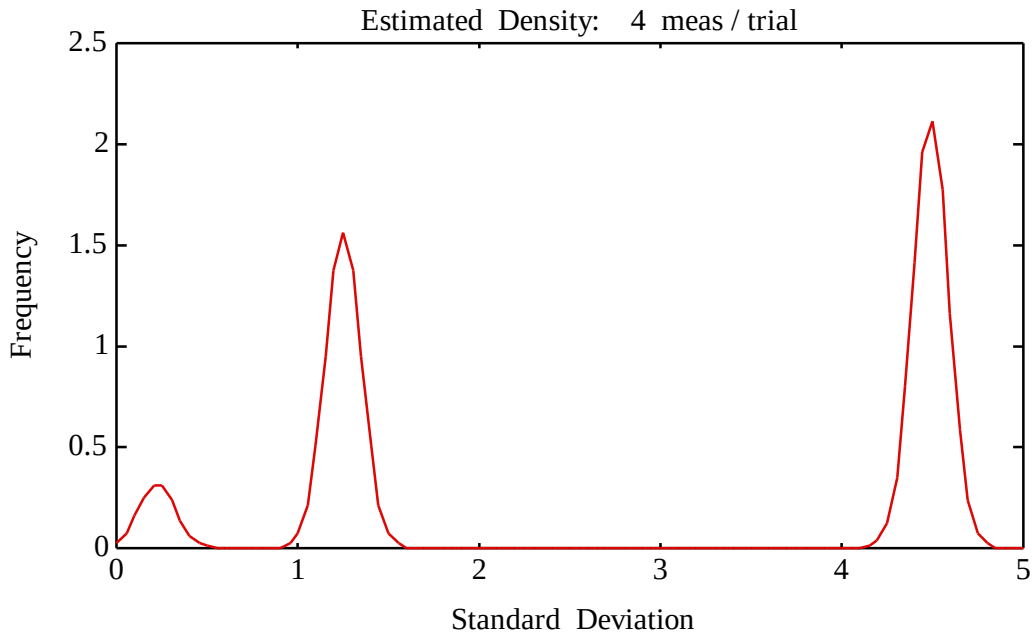
This problem was posed in Lesperance and Kalbfleisch (1992) to test their nonparametric method. In the simulation: v = 5, x_i = 1, i = 1-6; x_i = 4, i = 7-12. Using our Mallet type algorithm for mixed effects models, our results agreed exactly with theirs namely:

$$\hat{v} = 4.66, \quad \hat{F} = (\hat{Q}, \hat{W}), \quad \hat{Q} = \{.22,1.25,4.49\}, \quad \hat{W} = \{.08,.39,.53\}$$

The next two figures show the simulated data and the (smoothed) estimated density of the $\{x_i\}$. (The smoothing in Examples 1 and 2 was performed by placing a normal density with small variance at each of the support points of the discrete F^{ML}.)

RESULTS - Common Means Example


Common Means Ex: 4 meas / trial
Measurement
Trial Number
15
10
5
0
-5
0
2
4
6
8
10
12


Estimated Density: 4 meas / trial
Frequency
Standard Deviation
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5

References

Bayard D., M. Milman and A. Schumitzky. (1994) Design of Dosage Regimens: A Multiple Model Stochastic Control Approach. *Int. J. Biomedical Computing*, 36: 103-115.

Beal, S.L. and L.B. Sheiner. (1982) Estimating Population Kinetics. *CRC Critical Reviews, Bioengineering* 8: 195-222.

Beal, S. and L. Sheiner. (1992) *NONMEM User's Guide.* NONMEM Project Group, University of California, San Francisco.

Bell, B. M. (1990) Global Convergence of a Semi-Infinite Optimization Method. *Appl Math and Opt* . 21: 69-88.

Bell, B. M. (1995) The Iterated Kalman Smoother as a Gauss-Newton Method. *SIAM J. Optimization.* 4: 626-636.

Bell, B. M., J. Burke and A. Schumitzky. (1996) An Algorithm for Estimating Parameters and Variances in Multiple Data Sets. *Computational Statistics and Data Analysis.* 22: 119-135.

Bell, B. M., B. Paisley and D. Trippel. (1994) *O-Matrix for Windows Users Guide.* Harmonic Software, Inc., Seattle, WA.

Bell, B. M. and A. Schumitzky. (1997a) An Algorithm that Simultaneously Fits Mean and Variance Parameters in Nonlinear Models. Submitted.

Bell and Schumitzky (1997b) Asymptotic Properties for Estimators that Simultaneously Fit Mean and Variance Parameters in Nonlinear Models. Submitted.

Bohning, D. (1985) Numerical Estimation of a Probability Measure. *Journal of Statistical Planning and Inference* 11: 57-69.

Bohning, D. (1988) Likelihood Inference for Mixtures: Geometric and Other Constructions of Monotone Step-length Algorithms. *Biometrika* 76: 375-378.

Caines, P. E. (1988) *Linear Stochastic Systems.* Wiley, New York.

Clark. F. (1983) *Optimization and Nonsmooth Analysis.* Wiley-Interscience, New York.

Davidian, M. and A. Gallant. (1993) The Nonlinear Mixed Effects Model with a Smooth Random Effects Density. *Biometrika* 80: 475-488.

Davidian M. and D. Giltinan. (1995) *Nonlinear Models for Repeated Measurement Data.* Chapman and Hall, New York.

Deak , I. (1990). Multidimensional Integration and Stochastic Programming. In *Numerical Techniques for Stochastic Optimization*, eds. Y. Ermoliev and J.B. Wets. Springer- Verlag, New York, 1990.

Dempster, A.P., N. Laird and D. Rubin. (1977) Maximum Likelihood from Incomplete Data Via the EM Algorithm. *J. Roy. Statis. Soc. B.* 39: 1-38.

Fedorov, V. V. (1972) *Theory of Optimal Experiments.* Academic Press, New York.

Gallant, A. R. (1987) *Nonlinear Statistical Models*, Wiley, New York.

Geyer, C. (1996) Estimation and Optimization of Functions. In *Markov Chain Monte Carlo in Practice*, eds. W. Gilks, S. Richardson and D. Spiegelhalter. Chapman and Hall, London.

Jamshidian, M. and R. Jennrich. (1993) Conjugate Gradient Acceleration of the EM Algorithm. *Journal of the American Statistical Association* 88: 221-228.

Jelliffe, R., A. Hurst and B. Tahani. (1994) A 2-Compartment Population Model of Vancomycin made with the new Multicompartment NPEM2 Computer Program. *Clin. Pharmacol. Therap*. 55: 160.

Kaliski, J., D. Haglin, C. Roos, and T. Terlaky. (1996). Logarithmic Barrier Decomposition Methods for Semi-Infinite Programming. Report 96-51. Faculty of Technical Mathematics and Informatics. Delft, Holland.

Kiefer, J. and J. Wolfowitz. (1956) Consistency of the Maximum Likelihood Estimator in the Presence of Infinitely Many Incidental Parameters. *Annals of Math. Stat.* 27: 886-906.

Laird, N. (1978) Nonparametric Maximum Likelihood Estimation of a Mixing Distribution. *Journal of the American Statistical Association* 73: 805-811.

Lesperance, M. and J. Kalbfleisch. (1992) An Algorithm for Computing the Nonparametric MLE of a Mixing Distribution. *J. American Stat. Association* 87: 120-126.

Lindsay, B. (1983) The Geometry of Mixture Likelihoods: A General Theory. *Ann. Statistics* 11: 86- 94.

Lindstrom, M. J. and D.M. Bates. (1990) Nonlinear Mixed Effects Models for Repeated Measures Data. *Biometrics* 46: 673-687.

Lyne, A., R. Boston, K. Pettigrew and L. Zech. (1992) EMSA: A SAAM Service for the Estimation of Population Parameters Based on Model Fits to Identically Replicated Experiments. *Computer Methods and Programs in Biomedicine* 38: 117-151.

Mallet, A. (1986) A Maximum Likelihood Estimation Method for Random Coefficient Regression Models. *Biometrika* 73: 645-656.

Nesterov, Y. and A. Nemirovsky. (1994) *Interior-Point Polynomial Algorithms in Convex Programming.* SIAM, Philadelphia.

Neyman J. and E.L. Scott. (1948) Consistent Estimates Based On Partially Consistent Observations. *Econometrica* 16: 1-32.

Pettigrew, K.D. (1964) Estimation of Parameters from Observations with Unequal Precisions in the Presence of Nuisance Parameters. Master's Thesis, George Washington University.

Press W., B. Flannery, S. Teukolsky, and W. Vetterling. (1987) *Numerical Recipes.* Cambridge University Press, Cambridge.

Preston S. L. and G. L. Drusano. (1996) Nonparametric expectation maximization population modeling of ganciclovir. *J . Clin. Pharmacol.* 36: 301-310.

Racine-Poon, A. and A.F.M. Smith. (1990) Population Models. In *Statistical Methodology in the Pharmaceutical Sciences,* D.A. Berry, ed. Marcel Dekker, pp. 139-162.

Schumitzky, A. (1991) Nonparametric EM Algorithms for Estimating Prior Distributions.*Applied Mathematics and Computation* 45: 141-157.

Schumitzky, A. (1991b) Applications of Stochastic Control Theory to Optimal Design of Dosage Regimens. In *Advanced Methods of Pharmacokinetic and Pharmacodynamic Systems Analysis,* DZ.. D'Argenio, ed. Plenum Press, New York, 137-152.

Schumitzky, A. (1992) Nonlinear Population Modeling. In *Proc. 1992 Western Simulation Multiconference - Simulation in Health Carre.* Society for Computer Simulation, San Diego, pp. 57-64.

Schumitzky, A. (1993) The Non-parametric Maximum Likelihood Approach toPharmacokinetic Population Analysis. In*Proc. 1993 Western Simulation Multiconference-Simulation in Health Care*:. Society for Computer Simulation, San Diego, 95-100.

Schumitzky, A. (1995) EM Algorithms and Two Stage Methods in Pharmacokinetic Population Analysis. In *Advanced Methods of Pharmacokinetic and Pharmacodynamic Systems Analysis,* D. Z. D'Argenio, ed., Plenum Publ. Co. pp 145 - 160.

Schumitzky, A., B. Bell and D. Foster. (1995) SAAM II Population Analysis. In Proc. *1995 Western Simulation Multiconference - Simulation in Health Care* . Society for Computer Simulation, San Diego, 211-215.

Schumitzky, A., R. Jelliffe and M. Van Guilder. (1994) NPEM2: A Program for Pharmacokinetic Population Analysis. *Clin. Pharmacol. Therap.* 55: p. 163.

Silvey, S.D., D.M. Titterington and B. Torsney. (1978) An Algorithm for Optimal Designs on a Finite Design Space. *Commun. Statist.-Theor. Meth.* A7 (14): 1379-1389.

Spieler, G. and A. Schumitzky. (1993) Asymptotic Analysis of Extended Least Squares Estimators with Application to Population Pharmacokinetics. In *Proc. 1993 Biopharmaceutical Section.* American Statistical Society, 177-182.

Steimer, J.-L., A. Mallet and F. Mentre. (1985) Estimating Interindividual Pharmacokinetic Variability. In: *Variability in Drug Therapy: Description, Estimation, and Control,* M. Rowland, J.-L. Steimer and L. Sheiner, eds. Raven Press, New York, 65-111.

Steimer, J.-L., A. Mallet, J.-L. Golmard, and J.-F. Boisvieux. (1984) Alternative Approaches to Estimation of Population Pharmacokinetic Parameters: Comparison with the Nonlinear Mixed-Effect Model. *Drug Metabolism Reviews* 15: 265 - 292.

Vonesh, E. F. (1996) A Note on the Use of Laplace's Approximation for Nonlinear Mixed-Effects Models. *Biometrika* 83: 447-452.

Vonesh, E. F. and R. L. Carter. (1992) Mixed-Effects Nonlinear Regressions for Unbalanced Repeated Measures. *Biometrics* 48: 1-17.

Vonesh, E. F. and V. M. Chinchilla (1997) *Linear and Nonlinear Models for Analysis of Repeated Measurements.* Marcel Dekker, New York.

Wakefield J. and S. Walker. (1994) Population models with a nonparametric random coefficient distribution. Technical Report, Department of Mathematics, Imperial College, UK.

Wakefield J., A.F.M. Smith, A. Racine-Poon and A. Gelfand. (1994) Bayesian Analysis of Linear and Nonlinear Population Models. *Applied Statistics* 43: 201-222.

White, H. (1994) *Estimation, Inference and Specification Analysis*. Cambridge University Press, New York.

Wong, W. H. and T. A. Severini. (1991) On Maximum Likelihood Estimation in Infinite Dimensional Parameter Spaces. *Annals of Statistics* 19, 2: 603-632.

Project 2: Design and Development of Software Systems

Project Coordinator: Hugh Barrett and Brad Bell
Co-Investigators: Claudio Cobelli, David Foster, Hellmut Golde, Alan Schumitzky, Gianna Toffolo and Paolo Vicini
Collaborators: Roger Jelliffe, University of Southern California
David Bourne, University of Oklahoma

Introduction

The goal of this project is to produce software systems (deliverables) that address questions in population kinetics, to make them available to individuals and organizations, and to maintain them once they are released. These deliverables will be designed and developed to IEEE standards using state-of-the-art computer science methods. The software systems will be developed incrementally, and will be modularized so they can be used by other software developers upon request. Release information and service support for the modules and algorithms will be developed as part of our dissemination package. Each system will be described in more detail below.

Unlike the other project descriptions in this application, we will not follow the usual "specific aims -> background -> significance -> methods" paradigm. The reason is that the design and development of reliable software is a complex process; we feel the presentation is better accommodated using a paradigm for the life cycle of a software deliverable shown in Figure 2.1.

The narrative description of this project will be divided into the following sections.

- Introduction to the software design and development process. Here we will present the components of the life cycle for a software deliverable, and summarize how we will use this in the specification and design process.

- Deliverables. Here we will describe the software deliverables we propose to develop under the aegis of this proposal.

- Link between Projects 1 and 2. Here we will describe the link between our two core research and development projects, and give an initial projection (in time) for finishing the deliverables.

- General requirements. Here we will give examples of the more salient requirements that we will apply in our design, development and coding process.

- Development cycle for each deliverable. Here we will provide a more detailed description of our design and coding process for each deliverable. This section will be subdivided into 5 parts. These are:

  - Functional Specifications
  - Software Design Specification
  - Implementation and Testing
  - Deliverables Testing
  - Release and Maintenance

  Each of these sections will contain a description of what will be done, a set of specific aims, a description of the significance (i.e. why it is necessary), and methods (a paradigm for doing the work). Examples of each will be available for the Special Study Section.

Introduction to the Software Design and Development

In this section, we will review the life cycle of software and the software verification and validation process. We do this because it is the framework we will adopt in developing and designing our software deliverables. It also illustrates the complexity of the process both in terms of design and implementation, and the management of the entire RFPK team.

To begin, verification and validation can be defined as the process of creating documented evidence that a computer program executing on a particular system has done, is doing and will do, reliably, what it purports to do. In order for a computer application to be considered validated by the FDA, for example, the following evidence must be made available upon demand: standards for designing software and conducting a validation ("Standard Operating Procedures" - SOPs), a validation protocol for the application, comprehensive specifications for the application describing what the system purports to do, a test plan providing information about validation tests that have been conducted, and a validation report summarizing the results of the validation and containing all evidence (results) of the testing that took place.

We will use the FDA guidelines [1] modified using information from [2,3] to formulate, design and, when required, validation plans for all our deliverables. These will include the following documents or guidelines:

1. Standard operating procedures. This document specifies exactly what rules will be followed in designing and validating the software. RFPK will write an internal document specifying ourSOPs; all other documents written in association with the project will follow these guidelines. This means anyone who knows our SOPs should be able to read any other document produced by RFPK related to a deliverable.

2. Validation protocol for a deliverable. This document specifies how we will apply our SOPs to a specific deliverable. It will include: operational limits (software/hardware configurations under which the deliverable will be validated) a general statement about how the results produced by the deliverable will be checked, what will be reported, how validation documents and evidence will be maintained, and conditions under which revalidation is required (for example, when a change is made to the program or when the operating system is updated).

3. Specifications for the software at the user interface level. This document will prescribe everything that the user can input to the program (e.g. data, commands), and everything the user can obtain from the program (e.g. results of calculations, results of using model building tools).

4. Test plan for validating a deliverable. This document is the actual plan for testing each of the specific aspects of the deliverable for accuracy and correctness tests, and for proper behavior under limit conditions. It provides details of how each test result will be evaluated. It also provides details about how a revalidation will be performed.

5. Validation report. This document gives a summary of the validation process, describes any deviations from the test plan, provides results from accuracy and correctness tests, limit tests, reports any detected anomalies in the results, and indicates where the complete validation, including all evidence, is stored.

From the above, it would appear that much writing takes place before there is any coding. In fact, this is the case. Based upon our experience with developing other software deliverables, we have found that if we took the care to follow the above guidelines, the actual time to complete the whole project (i.e. coding and testing) took less time. The main reason was that misunderstandings of intent were worked out a priori instead of a posteriori.

In managing this project, we will follow the guidelines specified by ANSI/IEEE and summarized in a FDA Technical Report[1]. The following figure shows a modification of the software verification and validation plan overview as formulated by the IEEE [2]. This plan is consistent with a notion of a software "life cycle", i.e. the period of time that starts when the software is conceived at the "concept" level until it reaches the "maintenance" phase. Superimposed at the bottom of the figure are the steps in the design and validation processes described above. Notice that the writing of the SOPs and the formulation for the validation (test) plan are independent of the life cycle; in fact, they precede this process.

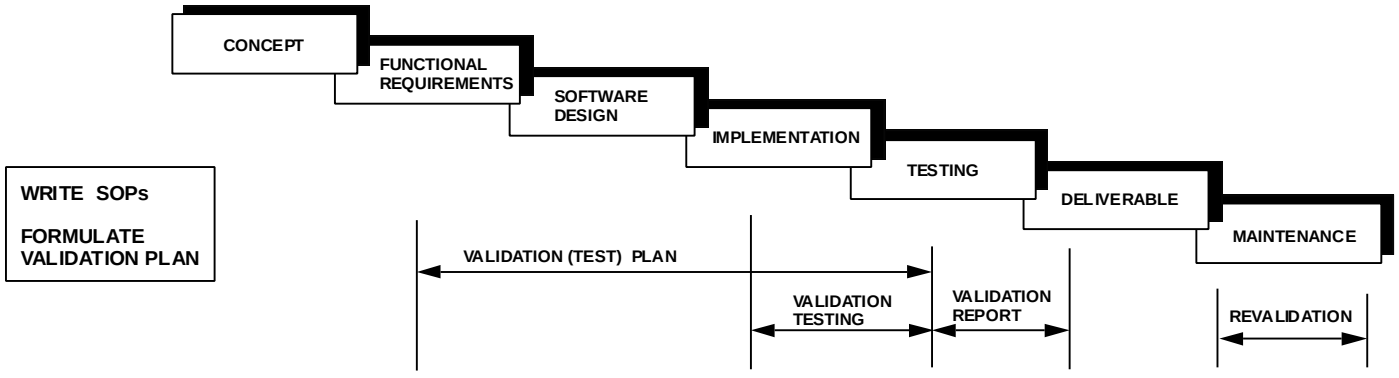


Figure 2.1*:* The software life cycle (adapted from IEEE). See text for explanation.

The software system, or components of the software system, starts with a concept of what the user wants. This concept is a written statement which should contain sufficient detail that the functional requirements can be written. The functional requirements are written specifications of what the software has to do, and can include information on performance requirements (e.g. speed or accuracy), interface requirements (how the components of the system communicate) and operational requirements (how the system will communicate with the user). For RFPK, this will be written in sufficient detail that the software engineers can write the software design document. This document will describe how the functional requirements can be implemented. Once this document is written, the engineers can implement the design, i.e. code the software. Included in this are internal (validation) test procedures. Once the software is coded and the results from the internal test procedures documented, it will undergo alpha (in house) and beta (outside test sites) testing as described in more detail below. When the testing is completed, the validation report will be prepared and the deliverable will be assembled. This is the software system and documentation (user and installation guide) that will be delivered to the user. Once released, the maintenance phase starts. As described in more detail below, this is a comprehensive plan to deal with performance reports that describe problems users have with the software and minor enhancements. As part of maintenance, revalidation will occur.

It should be noted that some of these activities will be iterative. For example, testing will most likely require implementation changes, and sometimes software design changes. These may affect the validation test plan and testing.

In providing the details as to how we will utilize this framework to develop our software deliverables (described in more detail in the section "Development Process for Each Deliverable), we will subdivide the above activities into five integrated parts:

- Part I. Functional Specification
- Part II. Software Design Specification
- Part III. Implementation and Testing
- Part IV. Deliverable Testing
- Part V. Release and Maintenance

At the end of the process, for each deliverable we will have a fully tested and documented (internal reference manuals and a user guide) software system. We should point out at this point that some deliverables will be verified and validated to FDA standards while others will have technical reports available for interested users. This distinction is made because some deliverables are almost impossible to fully validate according to FDA standards. An example is a complex graphical user interface; it is more time consuming than our proposal can admit to test every possible combination of users options in a graphical user interface.

We will also add a sixth part, Scheduling, since this is a management exercise which will make the whole process work. Scheduling is built around the organization structure of RFPK described in Part II of Section 3 of this application.

Deliverables

While the end goal of Project 2 will be comprehensive compartmental population kinetic deliverables with an MS-Windows interface, we will achieve this goal by sequentially developing a series of deliverables which can be used by our collaborators or other software developers seeking comparable computational capabilities.

As described in Project 1, there are five methodologies which will be developed: the global two-stage, the "NONMEM" method, Lindstrom/Bates, Maximum Likelihood, andnonparametric. Each will be implemented in the deliverables defined below. As summarized in Figure 2.2, there will be 25 deliverables produced under the aegis of this project.

The deliverables for each of the five methods are:

SPK	The basic population kinetic subroutine
SDPK	The subroutine that simulates population kinetic data
SCPK	The compartmental population kinetic subroutine
PCPK	A batch-mode compartmental population kinetic program
GCPK	A compartmental population kinetic program with an MS-Windows interface

In addition, two other deliverables will be developed. One will be a program for converting SCPK files into a NONMEM [4] format (for parametric methods), and the other for converting these files into an NPEM2 [5] format (for nonparametric methods). These will be made available to users wishing to compare the results of SCPK (the workhorse of GCPK) with NONMEM and NPEM2, and will also be used internally for testing and validation.

Computationally, specialized integrators and optimizers will be required. Internally, these will be created as modules. When deemed appropriate, these modules will be fully documented and made available to other investigators wishing to incorporate these specialized technologies in their software products.

Below we describe in detail the characteristics of each deliverable.

SPK: Subroutine for Population Kinetics

This will be a subroutine that performs population kinetic analysis using each of the five methods described in Project 1. This deliverable, since it is the computational workhorse for each method, will be scheduled for the earliest possible release for use by our collaborators. It will thus be tightly linked to Project 1 since all theoretical work for each method must be completed before the design can start. In Figure 2.2, we give our estimates for the finish date for each method. The subroutine will be portable to a supercomputer (of the type at UCSD); this design will be in collaboration with DrJelliffe.

The user will be required to program a specific model in order to use this routine. Thus this product will not be restricted to any specific type of modelling. Its input values will be checked and if an error is detected, an error code will be returned. This error code will specify the exact nature of the error from the user's point of view.

An example of SPK user defined input is:

- a model that describes the mean and variance for each data point as a function of all parameters;
- define the fixed and adjustable parameters
- define the covariates on an experimental time line (that also includes any perturbations, i.e. "changing the conditions" of the experiment;
- prescribe the measurement values (data); and
- provide initial parameter estimates for the adjustable parameters.

The output from SPK will be the population kinetic parameters as described in Project 1.

SDPK: Simulation of Data for Population Kinetics

This deliverable will be a subroutine that performs Monte Carlo simulations of population kinetic data. The input to this routine will include the values that SPK determines (i.e. a set of parameter values in a format similar to that used by SPK). The output of this routine will be a simulated version of the data that is required by SPK as input.

SCPK: Subroutine for Compartmental Population Kinetics

This deliverable will be a subroutine that performscompartmental population kinetic analysis. It will use SPK to perform the population kinetics part of its work. It will include multiple methods for solving the differential equations for compartmental models (e.g. Runge-Kutta and stiff integrators). The projected date of completion for each methodology for SCPK is summarized in Figure 2.2. The subroutine will have a standard interface, and be portable to the supercomputers described for SPK. The user will define the compartmental model using a data structure instead of having to program it directly. Thus this product will require less work, on the part of the user, than SPK. On the other hand, it will be restricted to thecompartmental modelling case. Its input values will be checked and, if an error is detected, an error code will be returned. This error code will specify the exact nature of the error from the user's point of view.

PCPK: Program for Compartmental Population Kinetics

This deliverable will be a program that both simulates and analyzes population kinetic data sets. It will use SDPK and SCPK to perform the majority of its work. It will be restricted to population kinetics of compartmental models. The user will define thecompartmental model and data values using a file structure. This product will require less work on the part of the user than the routine described above. Projected delivery dates are summarized in Figure 2.2. This program will be validated in accordance with FDA guidelines, and will be portable to supercomputers as described for SPK. Its input values will be checked and if, an error is detected, an error message will be printed. This message will specify the exact nature of the error including exactly where in the input file the error was detected.

GCPK: A GUI (graphical user interface) forCompartmental Population Kinetics

This deliverable will contain a graphical user interface to invoke SDPK to simulate data sets and SCPK to analyze data sets. The user will define a specific model using graphical drawing tools. This deliverable will have a context sensitive help system so that specific help can be obtained for each dialog, window, or error message. It will also read and write files that have the format used by the program PCPK.Its input values will be checked and, if an error is detected, an error message will be displayed. This message will specify the source of the error from the user's point of view.

It is our intent that GCPK will make the initial model development and testing process for population kinetic analysis easy, but that PCPK will be the deliverable which, because it operates in batch mode, will be used to generate final numerical values to FDA standards.

NONMEM and NPEM2 format converters

In addition, we will develop programs for NONMEM and NPEM2 format conversion. These deliverables will convert files between the NONMEM and NPEM2 format, and the format necessary for input to the program PCPK.

Link between Projects 1 and 2

Figure 2.2 shows the link between the theory and algorithmic development described in Project 1, and the software deliverables to be developed in Project 2.

METHOD

DELIVERABLE (Project 2 \ Project 1)	Global Two Stage	"NONMEM"	Lindstrom/ Bates	Maximum Likelihood	Nonparametric
SPK	01	01	01	02	03
SDPK	01	02	02	03	03
SCPK	01	02	03	04	04
PCPK	02	02	03	04	05
	nonmem converter: 02 / npem2 converter: -	nonmem converter: 02 / npem2 converter: -	nonmem converter: 04 / npem2 converter: -	nonmem converter: 04 / npem2 converter: -	nonmem converter: - / npem2 converter: 05
GCPK	02	02	04	05	05

Figure 2.2 The link between the methodologies developed in Project 1 and the software developed in Project 2; see text for explanation.

The five methodologies described in Project 1 are listed across the top of the figure, and the five software deliverables defined above are listed on the left hand side of the figure. The numbers in each box refer to the year of the grant when we anticipate the deliverable will be finished and released to users. As in any software development project, these projects reflect our best estimate at this time based upon our previous experience in

the development process; they will be subject to periodic updating as we progress through the different developmental stages.

Note that the "NONMEM" and "NPEM2" converter described above applies only to the deliverable PCPK.

The estimation of the year of release reflects both the importance we put on getting a specific methodology out to our user community, and the time when we believe the theory and algorithm development will be completed in Project 1.

General Requirements

The following list contains examples of the general requirements that will apply to our software design and development process.

Documentation and Reproducibility

The deliverable products described above will each have their own users guide, both in interactive and printed form, together with internal documentation. These products will be sufficiently documented so that their results can be independently reproduced by other computer programs. This will require a precise specification of the algorithms that are used by the products. For example, if a parameter in an algorithm is given a specific value, the value of that parameter will be specified.

Other subroutines that have general application will be made available to the biomedical research community. These routines will have specifications as comments within their source code but may not have any other documentation.

Analytic Derivatives

Tools that automatically differentiate algorithms have recently become available. For example, the program ADIFOR, which differentiates FORTRAN 77 algorithms, recently won the Wilkinson prize for numerical analysis [6]. This makes it possible to have analytic derivatives for model functions without the user having to program them. Analytic derivatives are a significant help during optimization procedures. They are also helpful when integrating stiff differential equations which sometimes occur in compartmental modelling. The software system will take full advantage of analytic derivatives without requiring extra work on the part of the user.

LAPACK and Level 3 BLAS

Fast machine specific implementations of the Basic Linear Algebra Subroutines (BLAS) are currently available on many machines. For example, Intel has a special library for its processors [7]. LAPACK [8] takes advantage of the BLAS and is the most highly regarded and tested linear algebra package. The software system will use LAPACK and the BLAS wherever it creates a speed advantage when the program is ported to a supercomputer.

Automated Internal Testing

As described below, each component of a deliverable will have a test suite that runs in an automated fashion and generates a report. This report will include a list of the tests that failed. These tests will exercise all the

possible paths in the corresponding routine. (Some paths that return a code corresponding to a program error may not be possible to exercise.)

Collaborator Input

Our collaborators will review the complete set of functional requirements for each of the deliverables. This will help ensure that the products solve the problems of interest to the collaborators. It will also ensure that the deliverables meet other needs of our users. For example, it will ensure that it is easy to convert the data from the form in which they are recorded into the form necessary for each product.

Windows PC Execution and Portability

The software system will run on a PC in a Windows environment. When required, the code will be written in an ANSI standard language, such as FORTRAN77, that will be easy to port to the supercomputer and possibly other operating systems.

Development Process for Each Deliverable

Each software deliverable will have an overall functional statement of what it will do; this is essentially what has been described above as "Deliverables". This functional statement will serve as the starting point for the development process for each deliverable.

The process itself takes advantage of the expertise of RFPK personnel. Our personnel are divided into "domain experts" and "software engineers". The domain experts are those who determine precisely what a deliverable should do; the software engineers then take the requirements and create a deliverable which will do what is required.

The development process for each deliverable starts with the overall functional statement. The deliverable is then broken down into its component parts realizing this will result in an organizational tree of how the deliverable is constructed internally. When all parts of the tree have been identified, the process described below will begin. The end result of this process is the desired deliverable with documentation. Dr. Bell will be responsible for the overall organization of each deliverable making sure each component part can be integrated properly into the deliverable. Figure 2.3 illustrates what we mean by the "tree" structure for the deliverable.

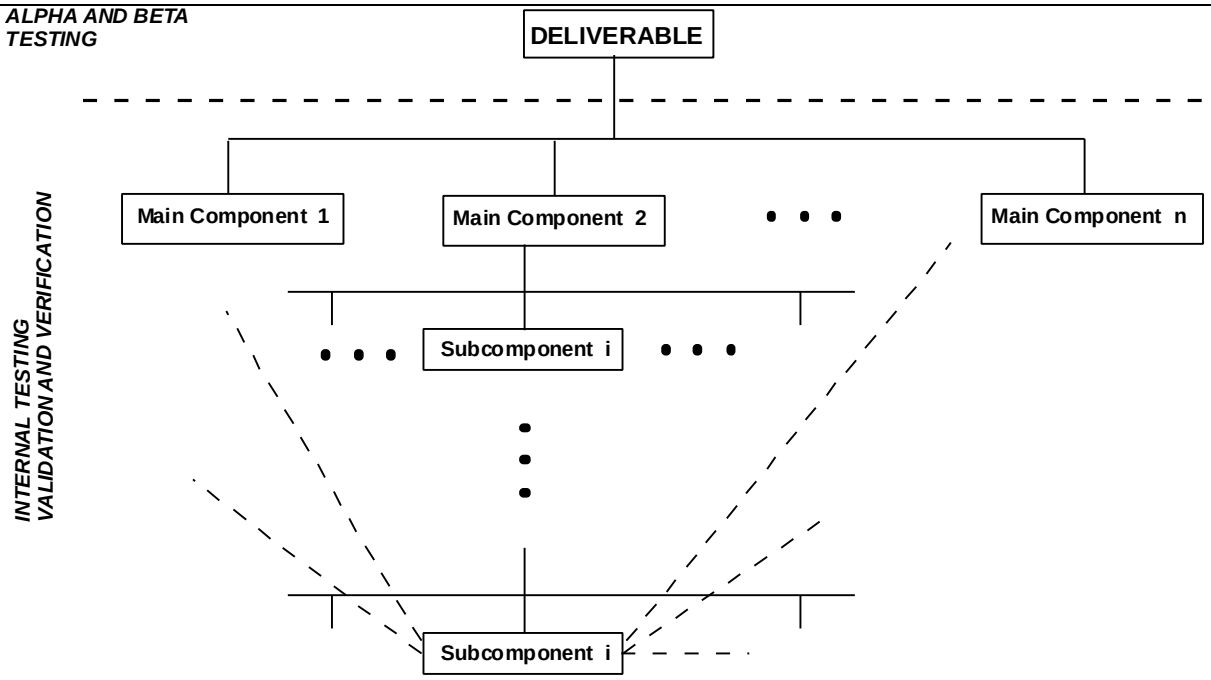


Figure 2.3. A schematic of the "tree" structure of defining a software deliverable. See text for additional explanation.

Examples of this tree structure that we have used in previous software development will be available for the Special Study Section.

It is important to note that the process described below applies to each box in this figure. What occurs in each box will be tested internally. The dashed lines indicate that each component has to fit in with the overall design for the deliverable. Thus when a component is incorporated into the developing deliverable, it will be tested again. This is part of the internal testing, and our validation and verification process. Finally, when the deliverable is assembled, it will again be tested internally. At this stage, it will move to the alpha and beta testing described below, and then be released.

The steps in the process will be explained in detail below in Parts I - V, and provide the details as to how the framework illustrated in Figure 2.1 will be implemented in RFPK.

Part I: Functional Specifications

Project Coordinators: Hugh Barrett, Brad Bell and Alan Schumitzky
Co-Investigators: Claudio Cobelli, David Foster, Gianna Toffolo, Paolo Vicini
Collaborator: Roger Jelliffe, University of Southern California

What will be done

For a given software deliverable, the aim for Part I of this project is to write the functional specification document for each component of the deliverable (as illustrated in Figure 2.3). This includes both computational and user interface functionality (i.e. input specification or a graphical user interface), and hence applies to all deliverables. This activity will be done in collaboration with Project 1 when computational specifications are required, and will serve as the vehicle to move the theory/algorithm development in Project 1 into the software deliverables of Project 2 (as illustrated in Figure 2.2).

Responsibility

- Hugh Barrett. Overall coordination (management); scientific requirements.
- Brad Bell. Numerical theory.
- Alan Schumitzky. Statistical theory.

Specific Aims

The specific aim for Part I is to produce sufficient specification for each component of a deliverable that the software design document (Part II) can be written. This will be accomplished in the following steps.

1. Specify precisely the concept of what is required (which can include the theory from Project 1 and/or user requirements.
2. If required, prototype algorithms (computational) or methodologies (e.g. GUI) to test the concept.
3. Write a functional requirements document which includes
   - a comprehensive statement of the requirements (including how the component fits in at the "global" level to the deliverable);
   - a summary of the prototyped algorithms or methodologies; and
   - a description of what new capabilities are needed;
4. Review by technical staff and collaborators of the functional requirements document, and modify as required.

Significance

The functional specifications, besides serving as the first step in the process of designing and then coding software, also lay the groundwork for designing tests to ensure the desired requirement(s) is(are) met. Documentation associated with this process will become a component of the technical reference associated with the deliverable.

Methods

For each component or subcomponent (shown in Figure 2.3), the integral parts will be identified. These will then be described in sufficient detail that (i) all computational capability, and (ii) input/output specifications can be identified. Following the paradigm we have used for recent software development, a list of the internal parts necessary to satisfy the computational capability and input/output specifications will be made. A domain expert will then be assigned to write the functional requirements document as indicated in step 3 of the Specific Aims.

Where appropriate, prototype algorithms will be developed and tested. For requirements related to input or output, specifications will be guided by staff depending upon the exact nature of the requirement. For example, if it is a GUI requirement, Dr. Barrett will work with a software engineer prototyping potential windows and their functionality.

As described in step 4, the document will be reviewed by software engineers, and sent to specific collaborators for their assessment. When all reviews have been received, the domain expert will write the final requirements document. This can then go to the next step, designing the software.

We would like to reiterate two points we have already made. First is that while this exercise is time consuming, we have found that it actually speeds up the time to finish coding because everyone involved understands what

is expected. Second, the written material will become part of the internal technical reference; this will be used in designing tests and, when necessary, aid in writing user manuals.

Examples

Examples of functional specifications that we have used in the past will be available for the Special Study Section.

Scheduling

Scheduling of this activity is part of the overall scheduling activity described below.

Part II: Software Design Specification

Project Coordinators: Hugh Barrett and Brad Bell
Co-Investigator. David Foster
Collaborator: Roger Jelliffe, University of Southern California

What will be done

The aim for Part II of this project is to produce a software design document from the functional specifications. The result of this step will be part of our internal documentation; this step will also include a description of all internal tests used for validation.

Responsibility

- Hugh Barrett. Human engineering
- Brad Bell. Software engineering

Specific Aims

The specific aims for Part II are:

1. write a requirements document specifying how the functional requirement will be implemented; and
2. write a design document encompassing
    - internal specification
    - internal documentation
    - internal testing

Significance

The software design specification for a specific component of a deliverable will be complete to the extent that a software engineer should be able to write code that meets the requirements. This document will become part of our internal documentation library, and will also serve as a basis to design specific tests for validation purposes.

Methods

The software engineers will work with the domain experts who authored the functional requirement for the specific component to write the documents called for by the specific aims. The RFPKSOPs will be followed.

Examples

Examples of software design specifications that we have used in the past will be available for the Special Study Section.

Scheduling

Scheduling of this activity is part of the overall scheduling activity described below.

Part III: Implementation and Internal Testing

Project Coordinators: Hugh Barrett and Brad Bell
Co-Investigators: David Foster and Paolo Vicini
Collaborators: Roger Jelliffe, University of Southern California

What will be done

Each component will be coded in accordance with the design specification, tested internally, and incorporated as an integral part of the software deliverable.

Responsibility

- Hugh Barrett. Human engineering
- Brad Bell. Independent comparisons and other tests of correctness

Specific Aims

The specific aims for Part III are:

1. to implement (code) a specified component for the deliverable from the design document produced in Part II;
2. to test the implemented functionality internally using the tests specified in the design document; and
3. to document steps 1 and 2.

Significance

The significance of implementation and internal testing is that RFPK will produce sound software code that can be verified as defined by our SOPs.

Methods

Implementation

Implementation relates primarily to scheduling. All RFPK software engineers are capable of designing and writing code given any specification document. Thus implementation of a specific component falls under the scheduling category since we first estimate the time required to implement a specific requirement, and then schedule this into our overall plan in order to maximize the time distribution of effort.

All RFPK's software engineers are highly skilled in implementing code.

Internal testing and validation

All software code will be tested according to RFPK's SOPs. When required by the specifications document, internal testing that has been designed to meet FDA requirements will be carried out as part of the implementation process. The full results of these tests will be collected in adocument which can be reviewed upon request.

Examples

Examples of implementation and internal testing that we have used in the past will be available for the Special Study Section.

Scheduling

Scheduling of the activity is part of the overall scheduling activity described below.

Part IV: Deliverable Testing

Project Coordinators: Hugh Barrett and Hellmut Golde
Co-Investigators: Claudio Cobelli, David Foster, Gianna Toffolo and Paolo Vicini
Collaborators: Roger Jelliffe, University of Southern California
David Bourne, University of Oklahoma

What will be done

Once a deliverable has been fully tested according to RFPK internal testingSOPs, it will be alpha (in house - RFPK personnel) and beta (outside users) tested for final assessment.

Responsibility

- Hugh Barrett. Coordinate beta testing.
- Hellmut Golde. Development and maintenance of the test library coordinating the alpha testing.

Specific Aims

The specific aims for Part III are:

1. to alpha test the deliverable;
2. to identify a beta test group, and beta test the deliverable;
3. to maintain performance reports from the alpha and beta tests;

4. to suggest changes to the deliverable as required; and
5. to document the results of steps 1 - 4.

Significance

Alpha and beta testing arerequired as a final step of ensuring the deliverable will perform as specified.

Methods

Testing

alpha testing

Alpha, or in house, testing, will be conducted using the paradigm we have used in the past.

Dr. Golde, working with Drs. Barrett, Foster and Vicini, will assemble a group of test problems that will test the individual component being tested and how it incorporates as an integral part of the deliverable. These tests can be run automatically using software such as Microsoft TEST. Results are stored and compared with known results (e.g. known analytic solutions of specific problems) or results from other software systems (for example NONMEM).

Alpha testing is an iterative process in that deliverables can be partially operable as they are being created, i.e. certain functionalities can be tested before the whole deliverable is complete. As the "prototype" deliverables are developed, RFPK personnel will test them and report back to the software engineer responsible for a particular component of the deliverable being tested.

beta testing

RFPK will develop and maintain a list of researchers who have agreed to test the deliverables we are producing; these include our collaborators and others with whom we interact. These individuals have told us they will exercise the software on a regular basis, and will provide us with "performance reports" at timely intervals. These performance reports describe their experiences with the deliverable, and include changes they would like to see in how different features are implemented.

Performance reports for each deliverable will be kept with the documentation for that deliverable. If a performance report contains a requested change in the deliverable, the RFPK development team will decide whether or not the change will be made.

At the completion of beta testing, i.e. when all performance reports associated with alpha and beta testing have been assessed and action taken when required, the deliverable will move to the release stage.

Example

Examples of alpha and beta testing that we have done in the past will be available for the Special Study Section.

Part V: Release and Maintenance of the Software System

Project Coordinators: Hugh Barrett and David Foster

Co-Investigator: Hellmut Golde
Collaborator: David Bourne, University of Oklahoma

What will be done

Release of the deliverable refers to writing and printing the user manual/guide, packaging the deliverable, announcing its availability (via, for example, the RFPK homepage), sending it out, and developing and maintaining a userdata base. Maintenance refers to the system we will install to provide support including user comments, and to make small changes for maintenance releases of the deliverable.

Responsibility

- Hugh Barrett. Coordination of the release date; coordinate all aspects of preparing the deliverable (including the homepage); work with UW Office of Technology Transfer (OTT) personnel to ensure compliance with the University's guidelines for intellectual property management.
- David Foster. Work with UW Office of Technology Transfer (OTT) personnel to ensure compliance with the University's guidelines for intellectual property management.
- Hellmut Golde. Design and coordinate software documentation.

Specific Aims

The specific aims for Part IV are:

1. set a release date for a software deliverable
2. coordinate activities related to a release including
   - writing user documentation
   - disk and user documentation duplication
   - announcement on the RFPK homepage
   - packaging
   - shipping
   - developing and maintaining a user data base.
3. Design and implement a comprehensive maintenance program.

Significance

A coordinated release plan will mean that RFPK software deliverables will be easily available to researchers wishing to use our products. Good documentation and information on our homepage will mean information to use our deliverables is accessible to all. A good maintenance plan is mandatory for the long-term viability of our software deliverables.

Methods

- Setting the release date for a deliverable. Once the component parts of a deliverable have been fully identified, an initial release date can be set using the scheduling program implemented for the SAAM II project by Mr. Reissig. Initially, we expect the list of component parts and the time estimates for the first three parts of the process described above to be only "reasonably" accurate.

Once alpha testing starts on the deliverable, we will have a good idea of what is left to do before the deliverable can go into beta testing. At that point, an accurate (to within 2-3 weeks) release date can be set.

- With a release date set, a schedule for the preparation of the user documentation can be formulated; any changes to the release date as a result of this schedule can be made. Disk and user documentation duplication can be scheduled. Packaging can also be scheduled. The userdata base can be developed, and an announcement on the RFPK home page prepared. Guidelines for release with OTT can be established so that when the deliverable is ready for shipping, it can be shipped.

- The maintenance program we will adopt for our deliverables will use the paradigm we have developed for SAAM II. We will deal with (i) user problems, and (ii) requests for enhancements. The latter will be divided into major (requiring a major time commitment) and minor.

  Performance reports for users providing us with feedback via telephone, fax, e-mail or otherwise will be maintained for each deliverable. The report "format" will be available on the RFPK homepage. Work related to minor enhancements will be scheduled, and as required, maintenance releases will be scheduled. The development of a maintenance release will follow the same five-part description above.

Part VI: Scheduling

Project Coordinators: Hugh Barrett and Brad Bell

In order to produce each deliverable on time and for the funds requested, RFPK must have an excellent organizational structure and scheduling guidelines. Scheduling refers to assigning RFPK personnel to (i) write the concept statement, (ii) write the functional requirements, (iii) write the software design specification, (iv) implement (code), test, release and maintain. We will use Microsoft PROJECT as we have done in the past to manage these activities. The overall areas of responsibility are indicated on the RFPK organizational structure in Part II of Section 3 of this application under the headings "Modeling Theory" and "Software Engineering". While "Modeling Theory" refers to Project 1, as noted in the above narrative, it impinges on Project 2 both in terms of total design and implementation and scheduling. That is, for a specific deliverable, the theories and algorithms have to be developed before the deliverable can be finished.

Examples of the scheduling software will be available at the site visit.

References

1. Software Development Activities. Reference Materials and Training Aids for Investigators. USPHS-FDA Technical Report, 1987.

2. IEEE Standard for Software Test Documentation. Institute of Electrical and Electronics Engineers, Inc., New York, 1983. Publication 829-1983

3. Software Control Procedures. Tecnomare Reference Manual PRO-115, Tecnomare, SpA, Venice, Italy

4. Beal, S. and L. Sheiner. (1992) *NONMEM User's Guide.* NONMEM Project Group, University of California, San Francisco.

5. Schumitzky, A., R. Jelliffe and M. Van Guilder. (1994) NPEM2: A Program for Pharmacokinetic Population Analysis. *Clin. Pharmacol. Therap.* 55: p. 163.

6. http://www.mcs.anl.gov/adifor/

7. http://www.intel.com/design/perftool/perflibst/mkl/sysreq.htm

8. http://www.netlib.org/lapack/index.html

COLLABORATIVE RESEARCH

Project 3: Application of Population Kinetics in Clinical Pharmacology Studies

Project Leader: Alan Schumitzky and David Foster
Collaborator: Roger Jelliffe, University of Southern California

Project 4: Application of Population Kinetics in PK/PD Studies

Project Coordinators: Alan Schumitzky and David Foster
Collaborators:
University of Washington: John Slattery
Other Institutions: George Drusano, Medical College of Albany

Project 4.1: Development of Integrated Population Pharmacokinetic/Pharmacodynamic Relationships for Antiviral Chemotherapy

Project Coordinators: Alan Schumitzky
Collaborator: George Drusano

Project 4.2: Population Concentration-Effect Relationships of CY Metabolites in Hematopoietic Stem Cell Transplantation

Project Coordinators: Alan Schumitzky and David Foster
Collaborator: John Slattery

Project 5 : Application of Population Kinetics to Lipid and Lipoprotein Metabolic Studies

Project Leader: Hugh Barrett
Co-Investigator: David Foster
Collaborators:
University of Washington: none
Other Institutions: Murray Huff, University of Western Ontario
Ernst Schaefer, Tufts University

Project 6 : Application of Population Kinetics to Studies in Intermediary Metabolism

Project Leader: Claudio Cobelli
Co-Investigators: David Foster, Gianna Toffolo and Paolo Vicini
Collaborators:
University of Washington: none

Other Institutions:	Angelo Avogaro	University of Padova
	Richard Bergman	University of Southern California
	Roberto Vettor	University of Padova

Project 7: Application of Population Kinetics to Environmental Toxicokinetic Studies

Project Coordinators:	Alan Schumitzky and David Foster
Collaborator:	Danny Shen
	Crispin Pierce