MATHEMATICAL MODELING IN DISSOLUTION KINETICS

AND APPLICATIONS OF A NEW NONLINEAR REGRESSION PROGRAM

Peter Veng Pedersen

M.Sc., Dip. Ind. Pharm., Cand. Pharm.

A thesis submitted to the University of Sydney
as a requirement for the degree of Doctor of
Philosophy in the Faculty of Science.

January 1977

ACKNOWLEDGEMENT

I would like to express my deep gratitude to Dr. K.F. Brown for his encouragement and continued interest in my work. In particular I want to direct my thanks to him for his suggestions and comments in the development of the papers published based on this work.

I would also like to thank Professor T.R. Watson for providing facilities and a tutorship during my studies in this department. The help I received to migrate to Australia and at the beginning of my postgraduate studies is acknowledged with particular gratitude.

My thanks are due to the members of the academic staff, technical staff and fellow postgraduate students, for the pleasant time I have enjoyed in their company. The encouraging discussions with Dr. J.J. Ashley who acted as my supervisor during Dr. K.F. Brown's sabbatical year is acknowledged.

I am indebted to Mr. K. Murray and M.J. Taylor for their help in construction of some items of equipment used for this work and to Mrs. Anita Auzins, Mrs. Anna Marie Pokorny and Ann Broughton for their help in typing my work.

Finally, I would like to express a very deep gratitude for the patience, love and understanding of my wife, Connie.

[REDACTION]

- o O o -

PUBLICATIONS

1. Peter Veng Pedersen & K.F. Brown, "Dissolution in Relation to Initial Particle Distribution", J. Pharm. Sci., 64, 1192 (1975).

2. Peter Veng Pedersen & K.F. Brown, "Size Distribution Effects in Multiparticulate Dissolution", J. Pharm. Sci., 65, 1981 (1976).

3. Peter Veng Pedersen & K.F. Brown, "Theoretical Isotropic Dissolution of Nonspherical Particles", J. Pharm. Sci., 65, * (1976).

4. Peter Veng Pedersen & K.F. Brown, "Experimental Evaluation of Three Single Particle Dissolution Models", J. Pharm. Sci., 65 * (1976).

5. Peter Veng Pedersen, M.J. Crooks & K.F. Brown, "A Method of obtaining Drug-Macromolecule Binding Parameters directly from Dynamic Dialysis Data", J. Pharm. Sci., 65, * (1976).

6. Peter Veng Pedersen & K.F. Brown, "A New Method of Characterizing the Dissolution Properties of Drug Powders", J. Pharm. Sci., 65, * (1976).

7. Peter Veng Pedersen, "Methods of Enhancing Dissolution". Proceedings of a Workshop in Formulation and Biopharmaceutics, University of Sydney, pp. 1-34 (1975).

8. Peter Veng Pedersen, "Dissolution of Powders as a Tool on Quality Control of Drugs". Proceedings of a Workshop on Automated Laboratory Techniques, University of Sydney, pp. 1-16 (1976).

(* = papers in press).

Papers recently submitted for Publication

9. Peter Veng Pedersen, "Modelling in Pharmacokinetics and some Practical Experiences with NONLIN and a New Program FUNFIT", J. Pharmacokin. Biopharm. (1976).

10. Peter Veng Pedersen, "General Treatment of Linear Pharmacokinetics",

J. Pharm. Sci. (1976).

11. Peter Veng Pedersen & K.F. Brown, "A Universal Standardization of Dissolution Rate Measurements", J. Pharm. Sci. (1976).

12. Peter Veng Pedersen & K.F. Brown, "A General Class of Multiparticulate Dissolution Models", J. Pharm. Sci. (1976).

13. S. Mason, P. Veng Pedersen, M.J. Crooks & K.F. Brown, "The Use of a New Nonlinear Regression Computer Program in Describing Multiple Equilibria", J. Pharm. Sci. (1976).

14. J. Whitlam, M.J. Crooks, K.F. Brown & P. Veng Pedersen, "Aspects of the Binding of Ibuprofen to Serum Albumin", Biochem. Pharmacol. (1976).

Technical Reports

15. Peter Veng Pedersen, "Fitting Nonlinear Equations to Data", 99 p. 2 copies, University of Sydney Library (1976).

16. Peter Veng Pedersen, "Guidelines for the use of FUNFIT Nonlinear Regression Program", 30 p. 10 copies, University of Sydney Library (1976).

- o O o -

CONTENTS

- o O o -

CHAPTER 1

INTRODUCTION

Historically pharmacy has been primarily concerned with the forms in which drugs are administered. Most effort has been directed to making formulations of drugs physically and chemically stable and acceptable to the patient with regard to appearance, colour, taste and smell and form of administration. It has become apparent, however, that other factors are important in drug formulation as techniques in pharmacology, medicine and analysis of drugs in the body improved. Emphasis has been directed toward studying the interactions between drug formulations and the organism and this resulted in the establishment of biopharmaceutics as a distinct field within the pharmaceutical sciences.

One of the major biopharmaceutical problems has been absorption in relation to drug formulation. Many very potent drugs have high lipid/water partition coefficients which facilitates their penetration through biological membranes. While this property lowers the biological barrier and favours absorption, it also paradoxically creates formulation problems because such drugs are frequently only very slightly soluble in water.

It is generally accepted that limited absorption after oral intake is frequently the result of slow dissolution. This kind of absorption problem occurs so frequently that bioavailability testing has become an important part of modern drug and dosage form design and production control. Many examples of differences in bioavailability resulting from differences in dissolution behaviour have been given in the literature (1, 2). Research into dissolution kinetics of drugs is therefore of great importance.

The numerous reviews published confirm the importance of the field (1-5).

The main purpose of most dissolution tests is to *describe* or *quantitate* the dissolution of a compound or dosage form. The dissolution process is most frequently described in the form of a dissolution profile. Although this is the most exact representation it is useful in practice to *summarize and quantitate* the dissolution kinetics in terms of one or more parameters e.g. the so-called dissolution rate constant. In order to do so, it is necessary to decide on some kind of mathematical description.

If the purpose is just to *summarize* dissolution data then a great number of arbitrary *approximating functions* can be used, e.g. polynomials and multiexponential expressions. Although the original data may be regenerated closely from parameters obtained in this way it is generally not reliable to make predictions or extrapolate on this basis. Furthermore, in the absence of mechanistic understanding and because of the substantial number of parameters involved (as in the case of polynomials), it is not possible to characterize the dissolution in a meaningful way. The use of arbitrary approximating functions is therefore of limited use. It is thus more meaningful to use a *mathematical model* which has some mechanistic significance at least conceptually and which adequately describes the dissolution behaviour. The extent to which the *true* mechanism of the system should be approximated by the mathematical model is a somewhat philosphic question, which depends on the purpose of the investigation. For practicability, the model chosen must involve a compromise between its correctness and its simplicity.

Dissolution is a heterogeneous process which is virtually impossible to describe rigorously because of the complexity of the mass transport phenomenon in an agitated heterogeneous system. It is apparent

from Chapter 4 that even when the dissolution mechanism of a free falling single spherical particle is analysed, using several simplifying assumptions, a "rigorous" mathematical model becomes too complex to be of *practical* significance. The description of the kinetics must therefore rely on a simpler model. As shown in Chapter 4 several such models can be derived on the basis of various assumptions about the mass-transport mechanism in the interfacial region.

The model for a *multiparticulate* system must be based on a model for the single particle dissolution behaviour together with a model for the particle size distribution.

Several multiparticulate dissolution models may be postulated. A decision about the model which best characterises the dissolution behaviour must be a compromise between the extent of experimental verification, its simplicity, and whether it is mechanistically meaningful. It may be possible to postulate very simple models which fit a particular set of dissolution data very well. However, oversimplification of the dissolution kinetics may lead to a lack of generality or flexibility. For example it may be possible to derive simple models for the dissolution of monodisperse systems. However, these will usually not fit dissolution data from polydisperse systems adequately; whereas more complex models for polydisperse systems may describe monodisperse systems as a special case.

The evaluation of how well a model fits dissolution data must rely on regression analysis. If the model is linear or can be transformed to linear form then the curve fitting procedure is generally simple. However, the multiparticulate dissolution models for polydisperse systems derived in this thesis are of a nonlinear form which can only be fitted and analysed

properly using a nonlinear regression computer program. As described in Chapter 9 several such programs are available. However, none of these programs are written for interactive time sharing and dedicated to mathematical modeling. The program FUNFIT described in Chapter 8 has been developed with these features. It is a *general* nonlinear regression program which can be used in any branch of science.

Methods for Studying Dissolution Behaviour *in vitro*.

The large number of dissolution rate measurement procedures and apparatus which have been described indicates that a universally acceptable method has not yet been developed. The many different techniques have been well reviewed by Hersey (6), Wagner (5) and Braun & Walker (7) and more recently by Hersey and Marty (8) and Groves (9).

The objectives of various tests are often very different. One worker may be interested in the fundamental dissolution behaviour, another in the effect of agitation and type of vehicle flow, or perhaps in the thermodynamics of the dissolution process, thus many kinds of dissolution apparatus are adapted to the study of fundamental principles of the process as for example the rotating-disc method, where the experimental conditions are better defined in terms of surface area of drug and agitation. Others are aimed at investigating dissolution behaviour of dosage forms *in vitro* to provide an estimate of their behaviour *in vivo*. The goal in these cases is to establish an *in vitro - in vivo* correlation such that a particular drug can be screened *in vitro* to provide an index of its expected behaviour *in vivo*. This requires that the composition of the drug formulation, the materials used and the production techniques are constant. Prediction of *in vivo* performance from *in vitro* data cannot be guaranteed but is only

intended to enable screening of dosage forms more readily and economically than *in vivo* procedures. Other reasons for performing dissolution tests include development purposes to guide the pharmaceutical formulator in the preparation of optimum dosage forms of drugs for clinical trial or for control purposes to ensure that a given pharmaceutical product is essentially uniform from lot to lot.

The various apparatus used for dissolution testing have been reviewed by Hersey (6), who classified them according to the type of agitation (free and forced convection) and the existence of sink or non-sink conditions.

In the past, the term sink conditions has most frequently referred to the situation where the concentration of dissolved drug is kept small, of the order of about 10% of the solubility, such that re-deposition of dissolved material onto the dissolving solid is considered negligible. This is somewhat vague and imprecise. In Chapter 4 of this work the sink condition is defined in more precise mathematical terms.

The following section of this chapter is not intended to be a comprehensivereview of apparatus used in dissolution testing, this has already been done by several authors (5-9), but is an attempt to outline briefly the basic principles of their design and discuss their advantages and limitations.

Apparatus with non-sink conditions and constant vehicle volume.

Most apparatus described in the literature belongs to this group because of the simplicity of design and operation. They are based on a constant volume and differ only in the shape of the dissolution container, the way agitation is supplied or in the position of the

dissolving sample. One of the most widely used techniques is the "beaker method" often credited to Levy & Hayes (11) although it was used earlier by Parrott & coworkers (12). It has been used widely for fundamental dissolution rate studies (13, 14) and in modified form for dissolution studies by Shefter & Higuchi (15) and other workers.

The rotating disc apparatus consists of a disc holder to which a tablet or compressed disc of the pure drug is fixed and rotated at suitable constant speed in a round bottomed flask. The method, proposed by Levy & Sahli (19), has been used chiefly for studies of intrinsic dissolution rate and studies concerned with heterogeneous reaction kinetics, diffusion layer theory, effect of agitation on dissolution rate and other fundamental problems and has therefore found wide use (20-22). In interpreting the results of this apparatus it is important to consider that the enormous pressures (for example 50.000 psi on a 0.5 inch diameter disc) often used to make these discs from pure drug powder may introduce changes in crystal form and consequently alter the physical properties of the drug. Clevely & Williams (23) reported that grinding of crystalline barbituric acid derivatives may produce changes in polymorphic form.

The static disc technique employs forced convection in the dissolution liquid and thus has less defined agitation than the rotating disc. Other disc methods include the solvometer technique (24, 25) and the hanging pellet method (13).

There are several disadvantages of the methods mentioned above. They will be discussed under following headings: (1) non-sink conditions, (2) agitation, (3) sampling, (4) introduction of drug sample.

Non-sink conditions

If the object of the dissolution test is to establish an *in vitro - in vivo* correlation or to provide a rough guide to the drug release rate of a dosage form *in vivo* then the methods above are of limited value according to Gibaldi & Feldman (26). These authors stated that unless sink conditions are maintained, *in vitro* results will bear little relationship to *in vivo* observations, for drugs that show dissolution rate limited absorption.

Dissolution testing is obviously most relevant for those drugs which represent the greatest dissolution problems. In general such drugs are the least soluble, which means it will frequently be necessary to use exceedingly large volumes of solvent to follow dissolution behaviour for more than just a small fraction of the drug sample used. For example, to follow the dissolution of a tablet containing 5 mg glibenclamide to completion in 0.1 M HCl, it is theoretically necessary to use 500 ml solvent (solubility of glibenclamide is 0.5 mg/100 ml). Dissolution will be very slow, however, because the process slow continuously as saturation is approached. In practice it is usually not convenient to exceed about 20% of saturation. Thus in the example above 2.5 l of solvent would be necessary. This is a rather inconvenient and unwieldy volume for handling and maintaining proper agitation and temperature control.

Agitation

The large volume of solvent required for apparatus operating under non-sink conditions causes problems in the maintenance of suitable agitation. If the drug is in the form of a powder or disintegrating dosage form a high degree of agitation will be required to suspend the drug

particles in the solvent so that they are all exposed to similar conditions of agitation. On the basis of *in vivo* data Hamlin *et.al.* (27) indicated that only low agitation rates could adequately differentiate rates of release from solid dosage forms. These authors studied the rates of dissolution of two polymorphic forms of methylprednisolone and found that sensitivity in distinguishing between the rates decreased at higher agitation intensities. This confirms the importance of using a dissolution apparatus where agitation is well defined and can be varied over a wide range.

Sampling.

Several problems are encountered with respect to sampling in the types of apparatus mentioned above. It is impossible to sample without disturbing the dissolution process to some extent. Sampling affects solvent volume, which in turn may affect agitation conditions. Replacement of removed solvent is not an entirely satisfactory solution to the problem because it results in a discontinuity or small drop in the concentration of dissolved drug. The agitation conditions may change during additions of replacement liquid and affect the homogeneity of the system by creating "pockets" of fresh solvent which mix only slowly with the bulk. The latter is particularly important under conditions of low agitation. The most serious problem in sampling these systems seems to be the fact that the most precise and detailed characterisation of the dissolution process requires the greatest frequency of sampling, which compounds the errors due to the sampling procedure. The time taken for sampling is difficult to define precisely. In most cases it is considered to be instantaneous rather than a time interval. Such errors may be quite substantial at the initial stages of the experiment, where the concentration is changing very rapidly.

Introduction of drug sample

It has been suggested that only a small amount of solid sample should be used for a complete dissolution test if the solubility is low. Furthermore, slightly soluble drugs are often ground finely or micronised which increases their surface-free energy so that particles may adhere and be difficult to disperse evenly in the solvent. They frequently adhere to the side of the dissolution container and a significant fraction of the powder may be washed up the side of the container where it is no longer exposed to the vehicle. The powder may also aggregate and float on the surface.

It can thus be seen that the most widely used type of dissolution apparatus (non-sink, constant, vehicle volume) has several serious design disadvantages which limits its usefulness in fundamental dissolution kinetics studies.

Dissolution apparatus with non-sink conditions, constant vehicle volume and automatic recording

Schroeter & Wagner (28) appear to be the first to describe an automatic recording dissolution apparatus. They combined a beaker type apparatus with a peristaltic pump. Filtered solvent was pumped to an automatic recording spectrophotometer and returned to the beaker. This procedure has several advantages. It provides unlimited data points, there is no discontinuity in the system due to sampling and the number of sources of error are reduced. Furthermore, it is fast and convenient.

Among the disadvantages are the facts that particles may be trapped in the filter system, the concentration range is limited by the spectrophotometer; however variable path length absorption cells can extend

the range. The system is only useful where there is little background absorption from solvent or excipients. The problems associated with the beaker method including introduction of sample, agitation, solvent volume, non-sink conditions are still present.

The technique described by Baum & Walker (7) represents a considerable improvement in the method used by Schroeter & Wagner. Agitation is performed by the solvent as it flows through a colum type dissolution chamber bounded at both ends by mesh screens. Solvent is recycled continuously through the column via a beaker or a flask. Sampling can be done from a collection reservoir or better by passing the filtered solvent through an automatic recording spectrophotometer. This system has several advantages compared to that of Schroeter & Wagner. Firstly, the solvent flow or agitation experienced by the particles is better defined and more easily adjustable. Entrapment of particles by the screen or filter should have little effect since they will still be exposed to almost the same solvent flow. Wetting and dispersion problems are reduced. In addition much larger solvent volumes can be handled readily so that sink conditions can be approached without increasing agitation rate. This system could be improved by altering to a non-recycled open system.

Dissolution apparatus with sink conditions: non-recycled open systems.

These types of apparatus include those most recently developed and represent the most suitable apparatus available for studies of dissolution kinetics. They consist essentially of a dissolution vessel with continuous input of fresh solvent and output of filtered solvent containing dissolved drug. The concentration of drug can be continuously monitored by some automated analytical procedure or solution sample may be

collected in fractions and assayed separately.

The various methods usually differ only in the design of the dissolution vessel and the mode of agitation, which may be produced either by the solvent flow or other means. The most common flow-through dissolution cells are cylindrical, with filters at both ends to enclose the sample and use the solvent flow as the only source of agitation (29-32).

Lapidus & Lordi (33) described a flow-cell which included a holder for compressed drug discs or tablets, enabling drug release measurements under sink conditions from a constant surface area. Their method represents a valuable alternative to the rotating disc technique for studying dissolution kinetics of pure drugs.

There are many advantages to using the continuous-flow column type apparatus. It is a flexible system that permits changes to be made readily, even within a test run, of important factors such as temperature, flow rate and vehicle composition including pH, viscosity, drug concentration and surfactant concentration.

Data obtained using a continuous recording technique with this system are in differential form. Thus a direct recording is made of the variation of release rate with time. This is superior to integral data (cumulative amounts dissolved with time) which tends to obscure small changes in dissolution rate particularly if only relatively few fractions are taken. A continuous recording yields unlimited data points which enable very precise characterisation of the dissolution behaviour. For this reason, it is also suitable for automatic data processing (34). Unlike most other systems, in particular the beaker method, sampling does not influence the dissolution process.

Normally the problem of wetting is minimal. However, when necessary

wetting can be accomplished by introduction of a surfactant solution for a short period initially and allowing it to be washed away immediately by fresh solvent (30).

The dissolution process can be followed to completion provided sufficient solvent is available and that the analytical method is sufficiently sensitive. All particles of a powder experience essentially the same intensity of solvent flow, including those particles which collect on the filters or screen at the ends of the column, thus agitation conditions are related to the solvent flow rate in a meaningful way and the latter is easy to define and control. Complete sink condition can be approximated very well. Thus results are more reproducible and instantaneous initial drug release measurements can be made, in contrast to most other methods. Furthermore, the continuous flow method is fast, convenient and very suitable for routine dissolution tests.

It is apparent that the continuous flow dissolution cell apparatus is a powerful and versatile tool in studies of dissolution kinetics.

The high precission dissolution apparatus described in Chapter 2 belongs to this category.

CHAPTER 2

EXPERIMENTAL

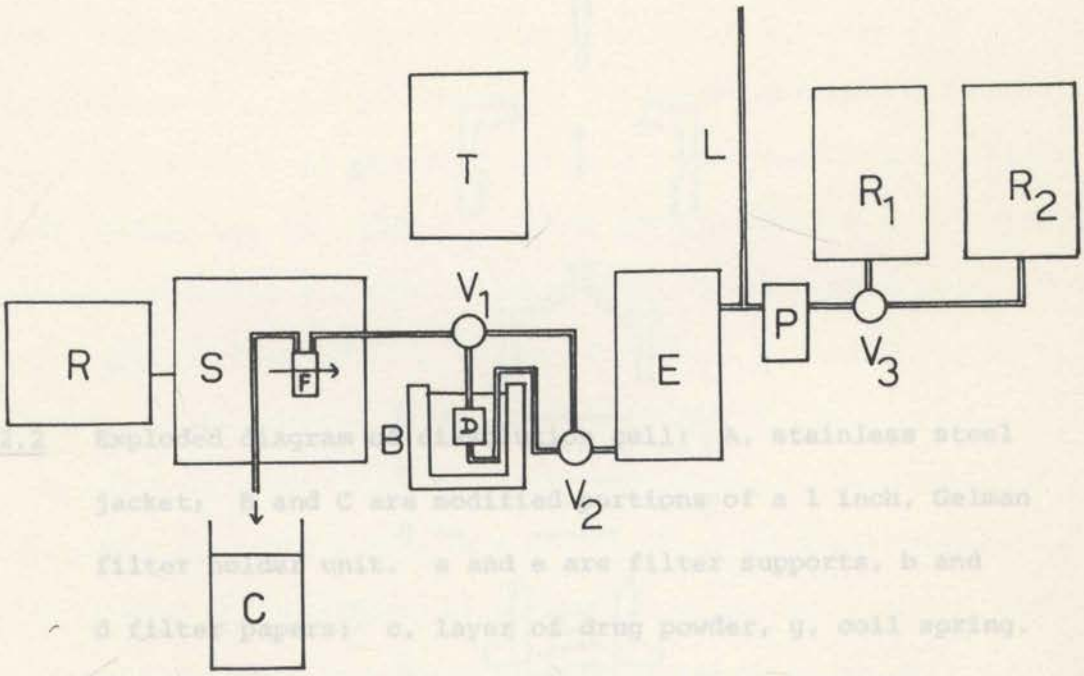
Dissolution Apparatus

A diagram of the continuous-flow recording dissolution apparatus is shown in Fig. 2.1. R_1 and R_2 are 20 litre reservoirs containing the dissolution media. P is a peristaltic pump (MHRE Watson-Marlow Ltd., England) transporting the liquid from R_1 or R_2 through a heat exchanger E, which adjusts the liquid to the required temperature before it reaches the dissolution cell D. This is immersed in a water filled, jacketed beaker, B, maintained at the same temperature as the dissolution liquid leaving E. Liquid from the dissolution cell, D, passes through a flow-cell, F, in the spectrophotometer, S, (Perkin-Elmer 124) which was fitted with chartrecorder, R, and finally accumulates in the container, C. V_1 and V_2 are two-way valves which enable by-pass of the dissolution cell, D, for zeroline adjustment of the spectrophotometer with blank liquid from the reservoir. V_3 is a similar valve by means of which liquid can be drawn from either reservoir R_1 or R_2. Flexible polyethylene tubing (i.d., 0.35 cm) was used throughout with the exceptions of that used in the pump, which was a silicon tubing (silastic, Dow Corning, i.d., 0.335 cm and o.d., 0.465 cm). L is an open tube type meter that monitors the pressure governing the flow rate of liquid.

The Dissolution Cell

Figure 2.2 shows a detailed diagram of the dissolution cell constructed for this work. Powder to be investigated is spread in a thin evenly distributed layer, c, in a sandwich-like arrangement between two

Figure 2.1 Diagram of continuous-flow dissolution apparatus: R_1 and R_2 are reservoirs containing the dissolution liquid; P, peristallic pump; E, head exchanger (Fig. 2.3); D dissolution cell (Fig. 2.2); B, water filled jacketed beaker; F, spectophotometer flow-cell (Fig. 2.5); S, spectophotometer with reorder R. V_1, V_2 and V_3 are two way valves; C, collection vessel; T, thermostat; L, pressure meter.


T
L
R1
R2
V1
R
S
F
P
V3
E
B
D
V2
C

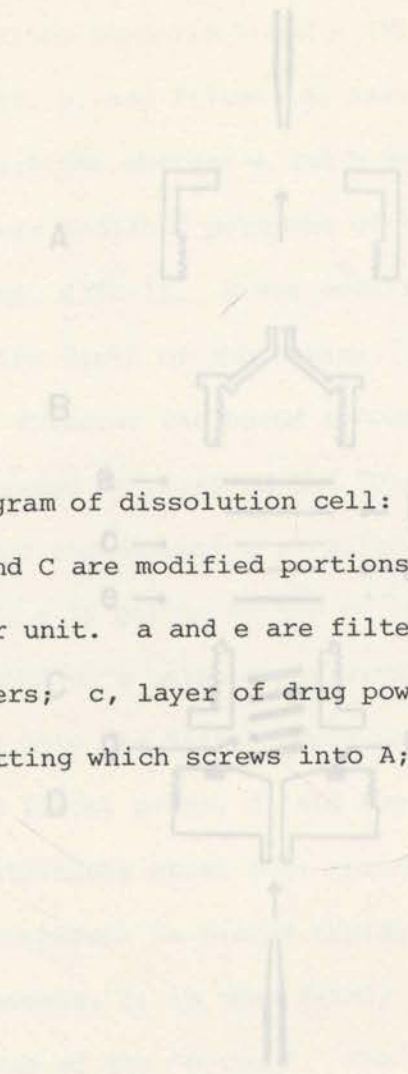
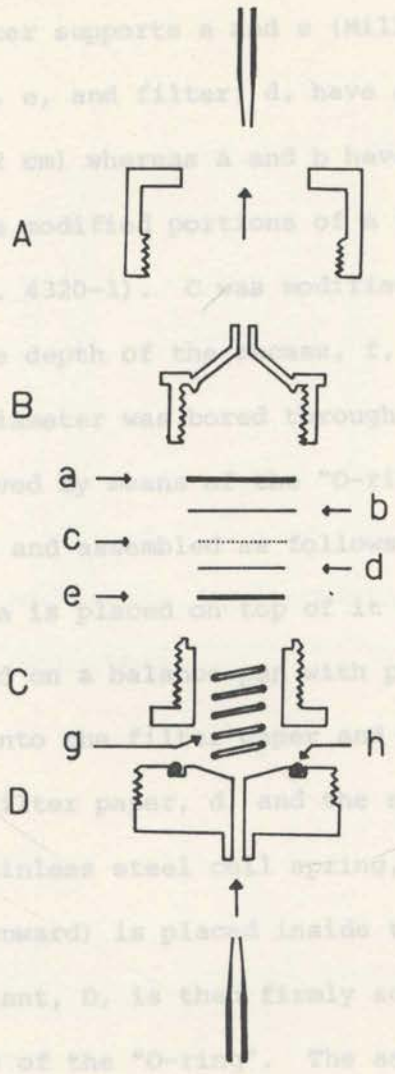


Figure 2.2 Exploded diagram of dissolution cell: A, stainless steel jacket; B and C are modified portions of a 1 inch, Gelman filter holder unit. a and e are filter supports, b and d filter papers; c, layer of drug powder, g, coil spring. D, teflon fitting which screws into A; h, "o-ring".


A
B
a
b
c
d
e
C
g
h
D

paper filters b and d (Whatman quantitative filter paper) supported on both sides by stainless steel filter supports a and e (Millipore cat. no. XX 3002503). Filter support, e, and filter, d, have a diameter equal to the inside diameter of C (2.2 cm) whereas a and b have the same diameter as the recess f. B and C are modified portions of a 1 inch Gelman, filter holder, unit (Gelman cat. no. 4320-1). C was modified from the commercial filter unit by increasing the depth of the recess, f, to contain both a and b. In addition a hole, 2.2 cm diameter was bored through the centre. A water tight pressure seal is achieved by means of the "O-ring", h.

The cell is loaded and assembled as follows: Filter paper b is positioned in the recess f, a is placed on top of it and B and C screwed together. The unit is placed on a balance pan with portion B as the base. Powder is weighed directly onto the filter paper and distributed as a thin, uniform layer. The second filter paper, d, and the support, e, are placed on top together with the stainless steel coil spring, g. The complete B - C unit (with B still downward) is placed inside the stainless steel jacket, A. The Teflon component, D, is then firmly screwed to A to form a tight seal with C by means of the "O-ring". The assembled unit is placed upright in a tripod stand with the hoses connected at both ends and the whole is immersed in the jacketed beaker (B on Fig. 2.1) for thermal equilibration 10 minutes before any run.

Heat Exchanger

The equipment for temperature regulation was also designed specially for this work, to overcome the problems of maintaining large volumes of dissolution fluid at constant temperature. A diagram is shown in Fig. 2.3. It consists of a cylindrical PVC container 40 cm in depth with an

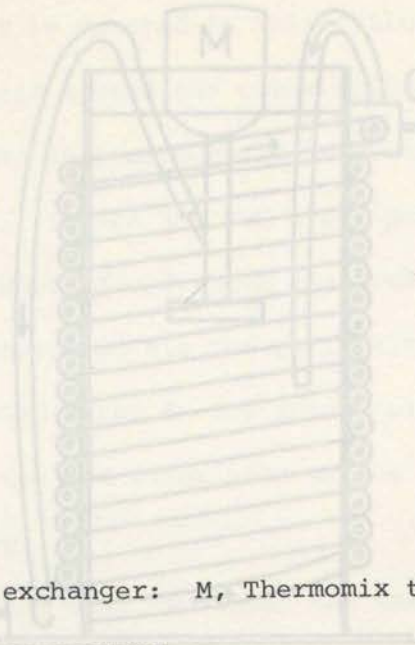
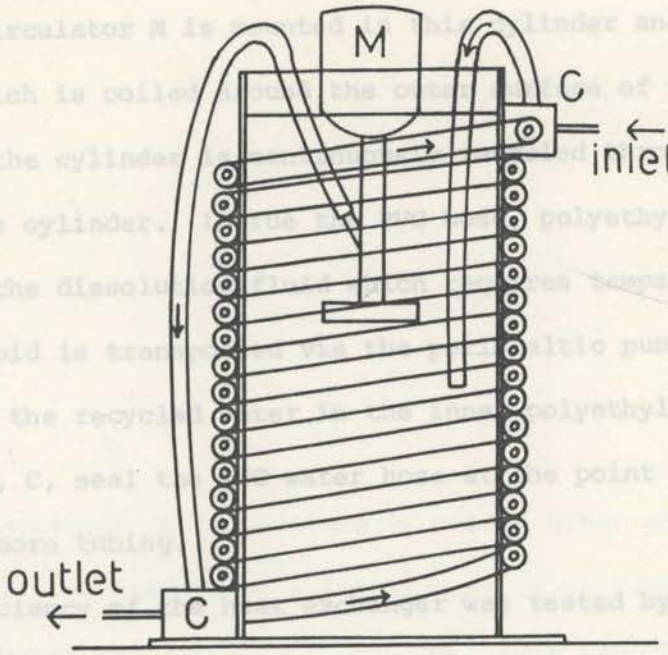
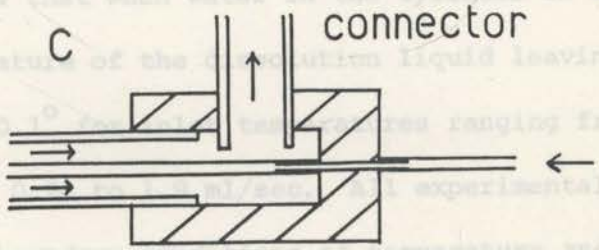


Figure 2.3 Diagram of heat exchanger: M, Thermomix thermostat water circulator; C, connectors.


M
C
inlet
outlet
C


C
connector

inside diameter of 15.5 cm filled with distilled water. A Thermomix thermostat water circulator M is mounted in this cylinder and connected to a soft PVC hose which is coiled around the outer surface of the cylinder. Heated water from the cylinder is continuously recycled through the hose and returned to the cylinder. Inside the PVC hose, polyethylene tubing is threaded carrying the dissolution fluid which requires temperature regulation. The dissolution fluid is transported via the peristaltic pump and flows counter-current to the recycled water in the inner polyethylene hose. Special connectors, C, seal the PVC water hose at the point of entry and exit of the small bore tubing.

The efficiency of the heat exchanger was tested by passing water of varying initial temperature through the apparatus at two flow rates and determining the temperature of the outlet. The results are graphed in Fig. 2.4. These data show that when water in the cylinder is maintained at $37.7 \pm 0.1^{\circ}$ the temperature of the dissolution liquid leaving the heat exchanger is $37.5 \pm 0.1^{\circ}$ for inlet temperatures ranging from between 5° and 30° and flow rates 0.24 to 1.9 ml/sec. All experimental work was carried out within these boundary conditions of temperature and flow rate. Equilibration of temperature was achieved within 10 minutes irrespective of starting temperature.

The heat exchanger is a very convenient piece of apparatus since it is highly efficient and eliminates the practical difficulties associated with maintaining large volumes of dissolution medium of constant temperature.

An inexpensive flow-cell for the spectrophotometer was also designed and is shown in schematic form in Fig. 2.5. It consists of a standard rectangular quartz cell of 1 cm path length to which is fitted

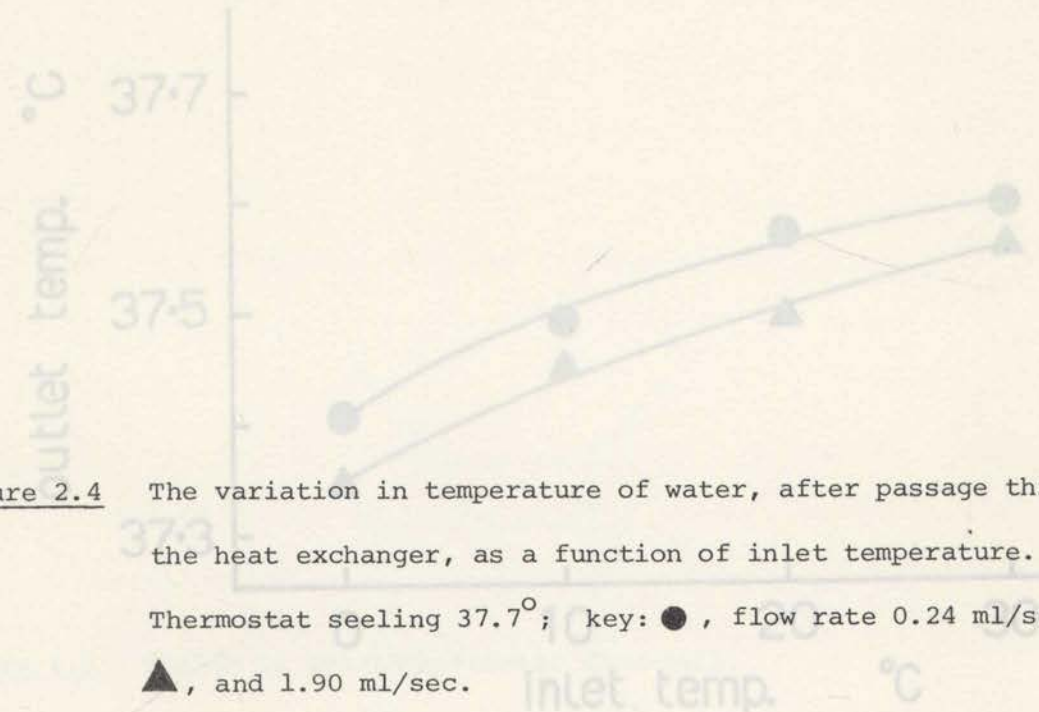
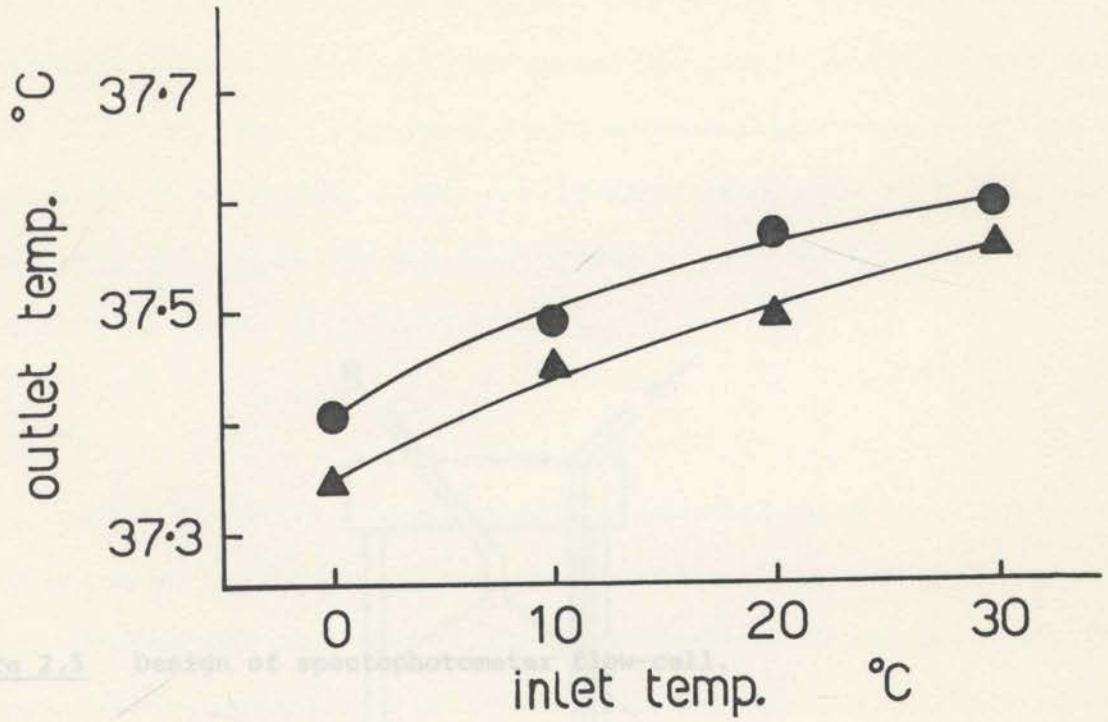


Figure 2.4 The variation in temperature of water, after passage through the heat exchanger, as a function of inlet temperature. Thermostat seeling 37.7°; key: ● , flow rate 0.24 ml/sec ▲, and 1.90 ml/sec.


°C
37·7
37·5
37·3
outlet temp.
0
10
20
30
inlet temp.
°C

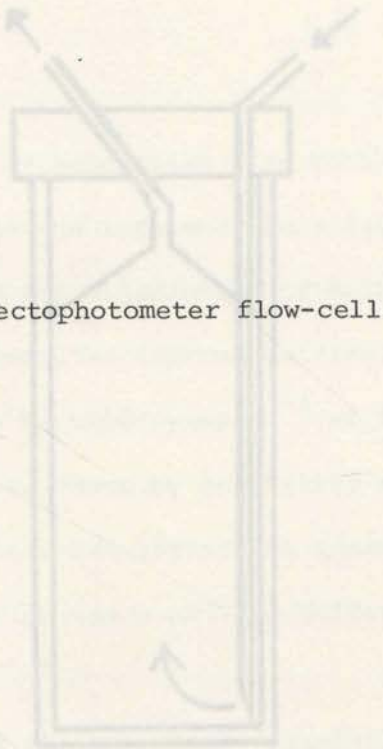
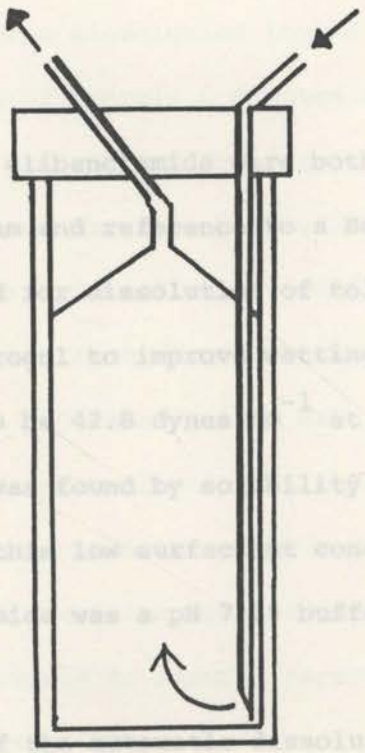


Figure 2.5 Design of spectophotometer flow-cell.



a specially designed rectangular teflon stopper with silicon rubber seal. A narrow bore stainless steel tube passes through the stopper and extends from one side of the cell. This tube is adjustable in length and does not obstruct the light path. Dissolution medium containing dissolved solute is introduced to the bottom of the cell by this means. The solution flows upward across the light path and exits through a cone shaped outlet which assists in the removal of gas bubbles.

Analytical

Tolbutamide and glibenclamide were both determined by ultraviolet spectrophotometry at 220 nm and reference to a Beer-Law curve.

The vehicle used for dissolution of tolbutamide was 0.1 M HCL containing 10^{-5} M cetomacrogol to improve wetting. The surface tension of this solution was found to be 42.8 dynes cm^{-1} at 37^{o} (mean of 10 measurements with s.d. 5%). It was found by solubility measurements, that no solubilization occurs at this low surfactant concentration. The dissolution liquid used for glibenclamide was a pH 7.25 buffer, I = 0.05, containing 10^{-5} M cetomacrogol.

The operation of the automatic dissolution apparatus (Fig. 2.1) is self-explanatory from the previous description. Values of absorbance were read from the chart recording at 2.5 minutes intervals from the beginning of the experiment. These values together with the volumetric flow rate, the Beer-law constant, the initial amount used, the amount of undissolved drug remaining in the dissolution cell at the end of the experiment and a code for the particular data treatment desired were fed into a digital computer and processed according to a FORTRAN program. The amount of undissolved drug remaining was found by disconnecting the

dissolution cell and transferring the filter paper/drug/filter paper "sandwich" to a volumetric flask. The drug was then dissolved in 95% ethanol and the solution assayed. This procedure increases the accuracy of the dissolution curves generated by the computer because it enables corrections to be made on the principle of mass balance.

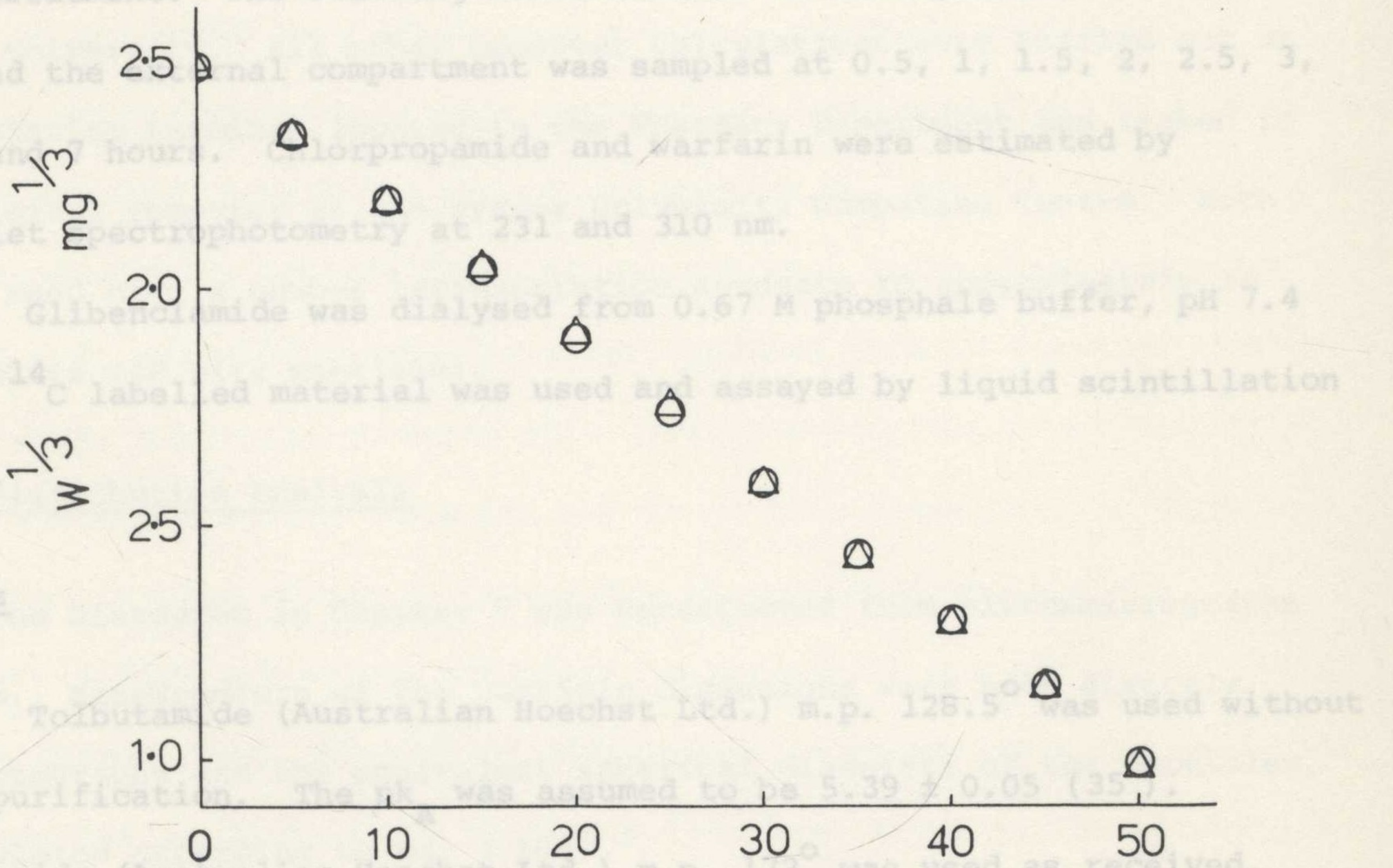
A test of this accuracy was made using 12.5 mg 60/85 mesh fraction of tolbutamide, a flow rate of 0.149 cm/sec and 0.1 M HCL + 10^{-5} M cetomacrogol as solvent. The dissolution liquid containing dissolved drug was collected at intervals of exactly 5 minutes as it left the spectrophotometer cell (F in Fig 2.1) and assayed for drug. The cube root of the calculated amount undissolved was then plotted versus time as seen in Fig. 2.6. On the same figure are plotted values generated by the computer from the absorbance readings taken from the chart recording. It appears from the plot that the computer generated points are almost coincidental with the points obtained from direct analysis. A chi-square test showed no significant difference between the two methods ($P > 0.99$).

The accuracy and precision obtained using this experimental technique have made it possible to examine aspects of the dissolution kinetics which would be impossible with most other methods.

Dynamic Dialysis

The data used for the evaluation of the new method of obtaining drug-macromolecule binding parameters described in Chapter 11 was kindly supplied by Dr. M.J. Crooks (35). The dynamic dialysis technique used was that described by Meyer and Guttman (36). Chlorpropamide was dialysed from 1% bovine serum albumin in 0.067 M phosphale buffer pH 7.4 at 37^{o} in the presence of a fixed pre-concentration of warfarin (1.6×10^{-5} M).

Figure 2.6 Dissolution of 12.5 mg, 60/85 mesh fraction, tolbutamide in 0.1 μ HCl containing 10^{-5} μ cetomacrogol using dissolution cell at solvent flow rate 0.149 cm/sec. Key: ●, values calculated from intermittent samples collected at 5 min's intervals; ▲, values generated by digital computer from continuous chart recording of absorbance *versus* time.


2·5
2·0
2·5
1·0
$w^{1/3}$
$mg^{1/3}$
0
10
20
30
40
50
min

Although warfarin displaces some chlorpropamide, the latter still appears to bind to two classes of sites (37). This system was selected so that such a model of two classes of sites could be appraised by the new method of data treatment. The starting concentration of chlorpromide was 3.62 x 10^{-3} M and the external compartment was sampled at 0.5, 1, 1.5, 2, 2.5, 3, 4, 5, 6 and 7 hours. Chlorpropamide and warfarin were estimated by ultraviolet spectrophotometry at 231 and 310 nm.

Glibenclamide was dialysed from 0.67 M phosphale buffer, pH 7.4 at 37°. ^{14}C labelled material was used and assayed by liquid scintillation counting.

Materials

Tolbutamide (Australian Hoechst Ltd.) m.p. 128.5° was used without further purification. The pk_A was assumed to be 5.39 ± 0.05 (35). Glibenclamide (Australian Hoechst Ltd.) m.p. 172° was used as received. The pk_A was estimated as 6.50 ± 0.05 (35). Cetomacrogol was obtained from Glovers Chemicals Ltd. The concentration used the surfactant had negligible absorbance at 220 nm.

Data Treatment

The dissolution rate data (dw/dt *vs.* time) used for the accuracy test of the dissolution apparatus (Fig. 2.6) were integrated by means of a FORTRAN program (38). The dissolution rate data presented in Chapter 10 and the dialysis rate data in Chapter 11 were analysed according to the respective mathematical models by nonlinear least squares regression techniques, using the FORTRAN program FUNFIT listed in the appendix. The subroutine, MODEL, defining the mathematical models are listed in the

respective chapters.

The calculations and drawings of the theoretical dissolution curves in chapter 6 was done using a CDC digital computer equipped with a calcomp plotter installed at the National Standards Laboratory, C.S.I.R.O., at Sydney University. All other computer calculations were carried out at the time-sharing terminal located in the Pharmacy Department and linked to the CDC digital computer at the Sydney University Computing Centre. Both computers used have a number representation accurate to approximately 14 decimal digits (48 bits mantissa).

Particle Distribution Analysis

The histogram in Chapter 7 was constructed from eltronmicrographs 30 x 30 cm. Measurements of the particle dimensions were made directly on the photographs and the equivalent spherical diameters of the particles were calculated according to Eq. 4.104.

Pharmacokinetic Data

The data used for the comparison of NONLIN and FUNFIT in Chapter 9 was kindly supplied by Mr. A. Somogyi. Although artificially generated data containing random noise just as well could have been used in such a comparison it was felt that the biological data would give a more realistic basis for comparison.

CHAPTER 3

BIOLOGICAL ASPECTS AND METHODS OF ENHANCING DISSOLUTION

Potent drugs very often have a high lipid/water partition coefficient which facilitates their penetration through biological membranes. While this property lowers the biological barrier and favours absorption it also paradoxically creates formulation problems, since these drugs are frequently only slightly soluble in water, causing them to dissolve slowly.

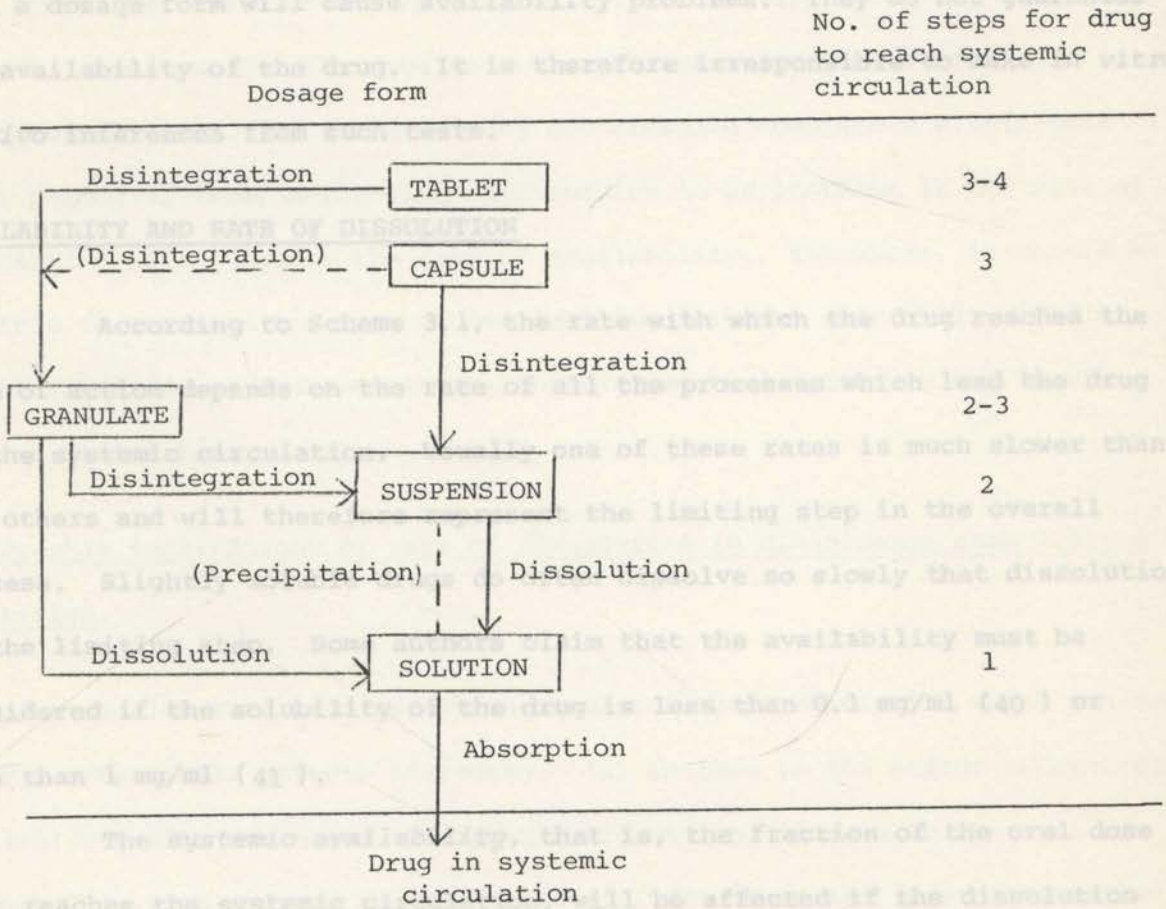
It is generally accepted that limited absorption after oral intake of poorly soluble drugs is often the result of slow dissolution. This kind of absorption problem occurs so frequently that bioavailability and dissolution testing has become important in evaluation and control of slowly dissolving drugs. Many examples of differences in bioavailability resulting from differences in dissolution behaviour have been given in the literature (1, 39). Research into dissolution kinetics of drugs is therefore of great importance and a better understanding of the influence of the formulation on the dissolution process is necessary to overcome these problems. The numerous reviews published recently confirm the importance of the field (2,3-5).

Dissolution as a step in the pathway to systemic circulation.

When a drug is given orally it has to go through a number of steps before it reaches the systemic circulation. Scheme 3.1 illustrates this for various oral dosage forms. It is expected from the number of steps, that the availability of the dosage forms should be of the order solution > suspension > granulate > capsule > tablet.

Scheme 3.1

Pathways to systemic circulation for oral dosage forms



Solid dosage forms are often tested *in vitro* for their disintegration or dissolution behaviour. Such tests essentially provide only information about these processes in relation to the particular experimental conditions. They are primarily useful in evaluating how the disintegration or dissolution is influenced by the drug formulation and by factors such as agitation, pH, vehicle composition etc. and are mainly of value in the development of better formulations.

The aim has often been to use the tests as a measure or indication of how the dosage form will perform *in vivo*. Most conclusions of this nature will however be unreliable because these *in vitro* tests in their very simple

forms cannot properly reflect the rather complex *in vivo* system.

The *in vitro* tests are only able to tell in some but not all cases when a dosage form will cause availability problems. They do not guarantee the availability of the drug. It is therefore irresponsible to make *in vitro - in vivo* inferences from such tests.

AVAILABILITY AND RATE OF DISSOLUTION

According to Scheme 3.1, the rate with which the drug reaches the site of action depends on the rate of all the processes which lead the drug to the systemic circulation. Usually one of these rates is much slower than the others and will therefore represent the limiting step in the overall process. Slightly soluble drugs do often dissolve so slowly that dissolution is the limiting step. Some authors claim that the availability must be considered if the solubility of the drug is less than 0.1 mg/ml (40) or less than 1 mg/ml (41).

The *systemic availability*, that is, the fraction of the oral dose that reaches the systemic circulation, will be affected if the dissolution rate is too slow. Much more easily affected, however, is the *rate of availability*, that is, the rate with which the drug enters the systemic circulation. This availability for drugs absorbed by passive diffusion will theoretically be affected *in any case where the drug is not completely dissolved when it reaches the main absorption site*. This implies that the dissolution rate of even relatively rapidly dissolving drugs will affect the rate of availability.

The stomach is the first place where absorption of a swallowed dosage form takes place. The absorption rate at this site is rather poor compared to the intestine due to the relatively small mucosal surface area,

the limited agitation and the longer diffusion pathway. The absorption rate first becomes substantial when the drug enters the intestine. Studies have shown that the stomach emptying half-life can vary from 7 to 22 minutes (42,43). The time for the drug to enter the intestine should be significantly less. If the drug does not dissolve completely within this short period of time, any action that results in an increase in the rate of dissolution will increase the rate of availability. Therefore, it should be possible to increase the rate of availability for even relatively rapidly dissolving drugs by a change in the formulation that increases the *in vivo* rate of dissolution.

Therapeutic implications of rate of dissolution in dissolution rate limited absorption.

A slow rate of dissolution results in poor absorption which can cause several undesirable effects including: (a) changes in the extent of systemic availability with decreased therapeutic effect, (b) changes in the rate of availability with delayed onset of activity, (c) increased inter- and intra-subject variability and unpredictability of response, (d) increased residency of drug in gastrointestinal tract with increased damage to mucosal tissue (Aspirin, KCl, steroids) (39). It is therefore nearly always advantageous to formulate both slowly and readily dissolving drugs so they dissolve as fast as possible in the organism. The only exceptions are drugs with a short biological half-life where a slow release dosage form is desirable to avoid the inconvenience of too frequent drug intake. The effect of dissolution rate on pharmacological activity has been discussed by Levy (44).

Factors affecting drug dissolution from solid dosage forms.

The dissolution of a drug from a solid dosage form is influenced by several factors that can be summarised as shown in Scheme 3.2.

Scheme 3.2

Factors affecting drug dissolution from solid dosage forms

I Disintegration

II Environmental factors

(1) agitation

(2) Dissolution media

(a) drug concentration and gradient

(b) pH

(c) viscosity

(d) interfacial tension

(e) complexation

(f) solubilization

III Factors related to the drug itself

(a) solubility

(b) polymorphism

(c) solvation

(d) salt form

(e) particle size

(f) particle size distribution

It is possible by proper formulation techniques to influence the majority of these factors such that a higher *in vivo* dissolution rate can be achieved.

Disintegration

A rapid disintegration of the dosage form (tablet, capsule) is necessary in order to achieve fast dissolution. The factors influencing

disintegration have been reviewed by Wagner (45). The aim of disintegration is to make the drug particles available for dissolution as *quickly* and *efficiently* as possible. Therefore, a disintegration test should include a test for both disintegration time and disintegration efficiency. This efficiency is measured in terms of the availability of disintegrated particles for dissolution. A conventional disintegration (time) test is therefore actually only useful when it is combined with a dissolution test.

The factors influencing dissolution (Scheme 6.2) can be divided into environmental factors and intrinsic factors, the latter being factors related to the drug itself. The environmental factors can be influenced either by additives in the dosage form or by the administration of the drug e.g. whether it is given before, or after a meal or with or without liquid etc. The intrinsic factors can be manipulated by physical or chemical means only e.g. by change in particle size or by making a salt-form of the drug.

Environmental factors

Agitation

It is well known that dissolution rate increases with increasing agitation (46-48). There does not seem to be any way the *in vivo* agitation can be influenced by drug formulation.. It is possibly too drastic to include other drugs that increase gastric emptying rate and gastrointestinal motility. It would be more worthwhile to consider the dependence of agitation on the composition of the food. It has been shown that meals of low viscosity are emptied more rapidly than meals of high viscosity (49) and that fats and fatty acids inhibit gastric motility (50). The physical activity of the patient also plays a role. Walking generally produces a higher gastrointestinal motility than lying (1).

pH

Noyes and Whitney have shown that the rate of dissolution of a drug is proportional to its solubility (51). Most drugs are weak acids or bases and their solubility is dependent on the pH of the vehicle. By use of the Henderson-Hasselbalch equation it can be shown that the solubility, C, of a weak base having solubility, C_o, in unionised form is dependent on the hydrogen ion concentration as follows:

$$C = C_o(1 + K_b(H^+)/K_w) \qquad (3.1)$$

where K_w is the ion product of water and K_b is the dissociation constant of the base. The corresponding expression for a weak acid is:

$$C = C_o(1 + K_a/(H^+)) \qquad (3.2)$$

The addition of small amounts of alkaline buffer substances to formulations of weak acids results in a higher pH in the vicinity of the drug particles which can enhance the dissolution of the drug (52). Buffers can in this way help to increase the dissolution rate, but their effect on drug absorption should also be considered. It is unfortunate that the effect of pH on the dissolution rates of acids and bases is opposite to the effect of pH on the intrinsic absorption rates of these weak electrolytes. The absorption of a weak acid from solution is optimal at low pH where the rate of dissolution of the weak acid is lowest. On the other hand, the rapid dissolution of a weak base in the stomach acid content is not so important because little absorption occurs in this part of the gastrointestinal tract.

Viscosity

It has been shown by several investigators (53-55) that the dissolution rate decreases with increasing viscosity of the vehicle. There

does not seem to be any way by which the formulation of a solid drug (tablet, capsule) can influence significantly the viscosity of the gastrointestinal contents. The implication of viscosity and viscosity-enhancing agents used in suspension on drug absorption has been discussed by Gibaldi (56).

Interfacial tension

It is well recognised that the dissolution rate is proportional to the effective surface area of the drug, i.e. the surface area available for dissolution. The effective surface area for hydrophobic drugs is usually considerably less than the real surface area because the interfacial tension between the solid and the liquid does not allow complete wetting. The effective surface area can however be increased by the addition of a surface active agent that facilitates wetting and hence results in an increase in the dissolution rate. It was shown by Finholdt and Solvang (57) that the gastric juice in most humans contains surface active agents and has a surface tension that is considerably less than the 0.1M HCl that is often used in *in vitro* experiments. *In vitro* investigations of the effect of surfactants on the dissolution rate will therefore probably result in overly optimistic expectations about the *in vivo* effect.

Complexation

The effect of complexation on the dissolution rate of drugs has been extensively investigated (58-60). Most soluble macromolecules such as high molecular weight polyols, gums and cellulose derivatives that are able to form complexes with drugs, can increase the water solubility of the drugs. There will not, however, be a proportional increase in the dissolution rate

since the drugs have to dissolve before complex formation can take place. One investigator explains that the increased dissolution rate observed in the presence of certain complex forming agents is possibly caused by a lowering of the energy change for transferring drug molecules from crystal to solution (58). The viscosity increasing properties of macromolecular complex forming agents will, however, reduce the diffusion rate of drugs and often counteract the former effect, causing a possible decrease in the dissolution rate. Complexing agents have primarily been used in dosage forms in order to increase the solubility or stability of a drug rather than to influence its dissolution behaviour. It is important to consider the implications of complex formation on drug absorption before any such agents are used (56,61).

Solubilization

The effect of solubilizing agents on the dissolution process has been investigated by several authors (62-64). The increase in the solubility due to micellar solubilization will usually not result in a proportional increase in the dissolution rate. The significant increase in the rate sometimes found is probably caused by the wetting effect rather than the solubilizing effect. It is therefore apparent considering Finholdt and Solvangs' investigations (57) that these agents are limited in their ability to increase the *in vivo* dissolution rate. They are mainly used for technical reasons in production but are also occasionally used for improving absorption, although there still seems to be controversy as to whether they actually enhance or retard drug absorption (56).

FACTORS RELATED TO THE DRUG ITSELF

Solubility

It was early recognised that the rate of dissolution in a diffusion-controlled dissolution process is proportional to the solubility (51). Several papers have discussed dissolution rate in relation to drug solubility (65-67). It has been pointed out that the solubility of very small particles increases with decreasing particle size because of an increase in vapor pressure of the solid (66).

The solubility of a drug is determined by the interaction between the solid drug and the solvent. There are therefore two ways in which the solubility and hence the dissolution rate can be manipulated: by changing the vehicle composition, for example by use of buffering agents to alter pH in the vicinity of the drug particles, as discussed previously; or by changing the crystal form.

Polymorphism

Most drugs can exist in at least two crystal forms. In certain classes of compounds the incidence of polymorphism is even greater. The metastable crystal forms have a lower melting point and a higher solubility than the stable forms. The increased solubility generally results in an increased dissolution rate. The amorphous form represents the highest energy level of the molecules in solid form. A pronounced difference in therapeutic activity between the amorphous and crystalline forms of drugs has been observed in several cases (67,68). This difference can only be due to a difference in the *in vivo* dissolution rate of the two forms, because a drug has to dissolve before it can be absorbed and the properties of the dissolved

drug do not depend on its original crystal structure. The above observations therefore strongly confirm the role of the dissolution rate in the absorption process. The exploitation of polymorphism often provides an effective way of enhancing the dissolution rate. Polymorphism of drugs has for this reason been subject to considerable investigations in recent years (17,40,65). The problem in using the most soluble crystal form of a drug to achieve faster dissolution is that this form is often metastable and can change to a stable and less soluble form, possibly producing severe consequences (67). Drugs in amorphous or other metastable forms should therefore be used only when it can be ensured that they will not change to less therapeutically active crystal forms.

Solvation

A more reliable approach for enhancing the dissolution rate may be to use the drug in a more soluble solvated or non-solvated form. Significant differences in the dissolution rate of anhydrous and hydrated forms of caffeine, theophylline and glutethimide have been observed (75). The anhydrous forms dissolved faster in these three cases. The n-pentanol and ethylacetate solvates of fluorocortisone dissolved faster, however, than the non-solvated forms (72). The use of solvated or non-solvated forms of a drug to increase the dissolution rate can be troublesome, since conversion between forms sometimes occurs during or subsequent to the formulation of a product. Such conversions are temperature-dependent and in the case of hydrates, humidity-dependent as well. Therefore a complete study of the forms should be done under different temperature and humidity conditions before they are used in pharmaceutical preparations.

Salt form

Several drugs can exist in the form of salts and it is well documented that these usually have faster dissolution and absorption rates than their parent compounds (73-75). Sodium salts of weak acids dissolve much more rapidly than the corresponding acid forms, regardless of the initial pH. Their fast dissolution in the low pH of the stomach can be explained in terms of the ability of these salts to increase the pH at the drug-liquid interface, causing a fast release and high concentration in the diffusion layer. Some drugs may subsequently re-precipitate in the bulk fluid, but usually in the form of very fine particles that readily dissolve through further dilution or absorption in the gastointestinal tract (44). Chemical stability can in certain cases preclude the use of the drug in salt form. The sodium salt of aspirin is very unstable in solution and even the solid form is rather unstable. The sodium or potassium salts may react with atmospheric carbon dioxide and water to precipitate out poorly soluble parent compounds. This occurs on the surface and thereby retards the dissolution and absorption rates (76,77). The alkalinity of some salts may furthermore cause epigastric distress after oral intake of the drug in a solid form.

Particle size

Particle size reduction is undoubtedly the most used and important method of increasing the dissolution rate. The surface area to weight ratio of a particle of any shape varies inversely with its diameter. Therefore the total surface area and hence also the effective surface area of a drug powder will increase substantially by particle size reduction, in particular for fine particles. The greater effective surface area of the drug in contact with the gastrointestinal fluid will result in more rapid dissolution and

absorption. This has led most drug firms to micronize (particle size<25 μ) very slightly soluble drugs for use in oral dosage forms. There are numerous reports of the better dissolution and availability of micronised drug compared to crystalline (78-82). Particle size reduction of a drug does not always influence its *systemic* availability. It was found for example that 50, 200, 400 and 800 μ powders of chloramphenicol were absorbed to the same extent. The 50 μ and 200 μ powders showed essentially the same absorption rate or rate of availability with a peak blood level at one hour, while the 400 μ and 800 μ powders were different with peak blood levels at two and three hours respectively (83). These findings indicate that the *in vivo* dissolution rate of the 50 μ and 200 μ powders has been fast enough to enable both powders to dissolve before they reach the main absorption site. However, this was not the case for the 400 μ and 800 μ powders that had different rates of availability. Therefore, it can be concluded that nothing has been gained in therapeutic efficacy by reducing the particle size to less than 200 μ. Such findings are of value in situations where it is better to use coarse rather than fine particles because of production or stability reasons. A particle size reduction to enhance dissolution should seriously be considered when the absorption site is in the stomach or upper region of the intestine. If, on the other hand, the site is in the last section of the intestine, drug absorption may be nearly independent of particle size. This results when dissolution occurs before this section is reached, and depends on the stomach emptying rate and the peristaltic activity (2).

Relatively large particle sizes are often needed to give local action in the terminal part of the gastrointestinal tract (84,85). A weak organic basic drug will often rapidly dissolve in the form of an hydrochloride in the acidic content of the stomach. However, when it passes into the slightly basic content of the intestine, it can precipitate out

as the un-ionized compound. The precipitation will be in the form of very fine, rapidly dissolving particles, so the absorption rate of the compound will be independent of the initial particle size (52). The absorption of a slightly soluble weak acid drug should, on the other hand, be much more dependent on particle size. A reduction in particle size will largely facilitate its dissolution in the acidic gastric juice, where it is only slightly soluble. It then forms the more soluble salt when it passes into the duodenum and small intestine. However, the salt is highly ionised and its absorption rate is decreased, since only a small fraction of the drug molecules is in the undissociated state.

Such considerations regarding the effect of particle size reduction in relation to the acidic or basic character of the drug are based on the pH-partition theory for absorption and should therefore be subject to discussion and further investigation. Particle size reduction can in certain cases reduce the therapeutic efficacy of a drug if it is unstable in gastric juice, as for example penicillin G or erythromycin, fast dissolution will enhance degradation in the stomach. The effect of particle size in relationship to absorption and activity has been reviewed by Fincher (2).

Particle size reduction can be performed in several ways: (a) trituration, (b) ball milling, (c) fluid energy micronization or (d) controlled precipitation including spray drying (86). There are, however, limitations as to how much the particles can be reduced in size and how suitable an extreme particle size reduction would be with respect to formulation and dissolution. Fine particles very often show a strong tendency to aggregate and agglomerate due to their increased surface energy and the stronger van der Waals attraction between non-polar molecules. Furthermore, electrostatic charges in fine hydrophobic powders can cause severe technical

difficulties in production. The problems associated with the wettability of fine powders have already been discussed. Drugs with plastic properties are difficult to subdivide by mechanical means (a-c), since they tend to stick together, even if produced by controlled precipitation (d).

Lin *et al.* (87) found that the *in vitro* dissolution rates of micronized griseofulvin and glutethimide were slower than those of their coarser particles. The opposite finding was reported by Chiou and Riegelman for griseofulvin (88). The results of Lin *et al.* can be explained by the strong agglomeration and reduced wettability of the micronized powder. The previously mentioned investigations of Finholdt and Solvang (57) indicate that Chiou and Riegelman's results better reflect what is expected *in vivo*. Several reports about the better availability of drugs in micronized form strongly support this (78-80).

Drugs can also be introduced to the gastrointestinal fluids in the form of very fine particles formed by precipitation *in vivo*. This can be done in several ways (52). The drug can be dissolved in a non-aqueous water-miscible solvent from which it precipitates out by dilution in the gastrointestinal tract. Solutions of sodium or potassium salts of an acidic drug will in the same way precipitate out at the low pH existing in the stomach. Formulations resulting in *in vivo* drug precipitation should have better availability than formulations of drugs in micronized form, because agglomeration and wettability problems are avoided and the particle size of the precipitate usually will be smaller than that which can be produced by micronization. However, the many disadvantages of liquid drug formulations compared to solid with respect to stability and convenience do not seem to counteract the advantage of the somewhat better availability.

Solid Dispersions

Drug formulations based on solid dispersions of slightly soluble drugs have received considerable attention in recent years. The pharmaceutical application of these systems has been extensively reviewed by Chiou and Riegelman (89). Solid dispersions are micro-crystalline or molecular dispersions of a poorly soluble drug in a solid water-soluble matrix. The drug either precipitates out in a fine particle form which is readily available for dissolution or becomes solubilized when the matrix rapidly dissolves in the gastrointestinal juice. Solid dispersions may be classified as: (a) simple-eutectic mixtures, (b) solid solutions, (c) glass solutions and glass suspensions, (d) amorphous dispersion in crystalline carriers and various combinations of these systems. Several reports indicate that the availability and dissolution rate of these formulations are superior to formulations containing micronized drugs (90,91). Solid dispersions appear to be as available as the liquid formulations from which the drug precipitates out *in vivo*, but they do not have the disadvantages of the latter with respect to their convenience. The stability also seems to be much better although very little investigation has been done in this field (89).

Solid dispersions appear, therefore, to be a very promising new approach in the formulation of slightly soluble, slowly dissolving drugs. The major problem in designing this formulation form lies in finding a suitable production method and in finding a matrix material with the right properties such as low toxicity, high water solubility, high solubilizing capacity for the drug, thermal and chemical stability and suitability for processing. Possibly the main hindrance to the introduction of this formulation form is the problem of aging. Solid dispersions often contain the drug in a high energy, metastable form which, under prolonged storage,

can be transformed into a stable but less soluble and more slowly dissolving form. Solid solutions (molecular solid dispersions) often contain the drug in a supersaturated form which, through the slow diffusion process occurring in solids, can precipitate out during storage, particularly at elevated temperatures (90). The aging of solid dispersions should be an important research subject for the pharmaceutical scientist, hopefully resulting in the commercial acceptance of this unique dosage form.

Particle size distribution

In recent years there has been increasing interest in the effect of particle size distribution on the dissolution behaviour of drugs (92-96). The reduced systemic availability observed for certain drug formulations can possibly be explained in terms of a "size distribution effect". In most drug powders there is a considerable difference between the sizes of the larger and smaller particles, which produces an important difference in their time for complete dissolution. This time is for a particle, dissolving according to Hixson-Crowell's cube root law (97), proportional to its initial diameter. The larger particles can because of their manifold longer dissolution time in such cases reach the distal part of the intestine and only be partly dissolved. The further dissolution in this section of the intestine with its content of increased viscosity, solid matter and low water content will be rather slow and the drug will possibly get eliminated from the body by defaecation before completion of its dissolution. Therefore in order to avoid such availability problems it would be good practice to have standards for the particle sizes of slightly soluble drugs.

- o O o -

CHAPTER 4

SINGLE PARTICLE DISSOLUTION

There are three dominant models describing the interfacial mass transport in a dissolving dynamic (agitated) system. These are:

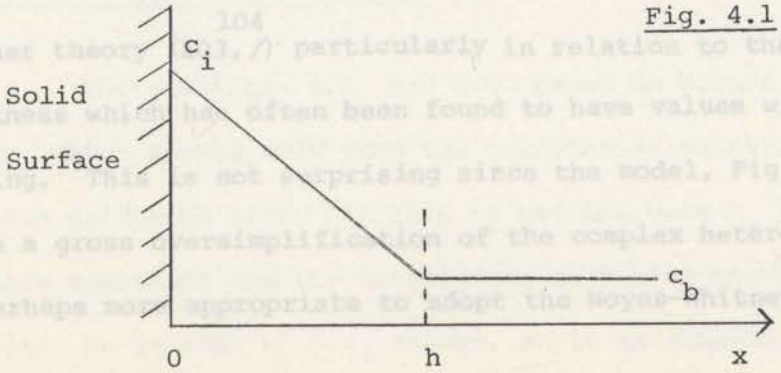
1. Nernst stagnant film theory (98).
2. Danckwerts surface renewal theory (99).
3. Interfacial solvation rate limited dissolution theory (100,101).

NERNST FILM THEORY

The theory of Nernst assumes a stagnant layer of solvent at the solid-liquid interface. The mass transport of solute through the layer is accomplished by simple molecular diffusion in a steady state fashion following Fick's law of diffusion. Once passed the stagnant layer the solute is then mixed quickly by convection and diffusion in the bulk of the liquid.

The Nernst model predicts a concentration profile from a plane surface in two dimensions as shown schematically in Fig. 4.1 , where h refers to the thickness of the so-called diffusion layer, c_i is the interfacial concentration and c_b the bulk concentration. The concentration gradient in the stagnant layer, $o \leq x \leq h$, is then constant because the diffusion flux,

Fig. 4.1



J_D, is constant during steady state:

$$J_D = -Ddc/dx = \text{constant} \qquad (4.1)$$

From geometrical considerations (fig. 4.1):

$$dc/dx = -(c_i - c_b)/h \qquad (4.2)$$

so that $$J_D = (D/h)(c_i - c_b) \qquad (4.3)$$

If it is assumed that the interfacial reaction rate is large compared to the rate of diffusion then c_i can be considered close to the solubility concentration c_o so 4.3 can be approximated by:

$$J_D \sim (D/h)(c_s - c_b) \qquad (4.4)$$

The dissolution rate from a plane surface of area, A, will then be:

$$dw/dt = -(DA/h)(c_s - c_b) \qquad (4.5)$$

If in 4.5 D/h is assumed constant, the equation reduces to the Noyes-Whitney equation:

$$dw/dt = -kA(c_s - c_b) \qquad (4.6)$$

after the workers who appear to be the first to have verified this equation experimentally (51). They did not attach any mechanistic significance to the quantity k. Nernst and Brunner (102) extended the Noyes-Whitney equation to include the concept of diffusion layer thickness and diffusion coefficient as presented in 4.5.

Over the years there have been several critics of the Nernst-Brunner theory (103, 104) particularly in relation to the diffusion layer thickness which has often been found to have values without any physical meaning. This is not surprising since the model, Fig. 4.1 , would appear to be a gross oversimplification of the complex heterogeneous process. It is perhaps more appropriate to adopt the Noyes-Whitney equation recognising

that the quantity k depends in some way on D, the hydrodynamic conditions and the geometry of the dissolving object.

It has been shown that k in Eq. 4.6 under fixed experimental conditions can often be considered constant and the equation has been successfully applied in several cases.

DANCKWERTS THEORY

Danckwerts (99) rejected the idea of a stagnant film and proposed a model based on the assumption that liquid motion is turbulent and extends to the surface of the dissolving crystal. The physical interpretation is that pockets of fresh solvent reach the interface by turbulence and renew parts of the surface, while 'old' pockets containing solute simultaneously leave the surfaces. The mathematical description of this system includes a parameter, γ, known as the mean rate of surface renewal:

$$dw/dt = A \sqrt{\gamma D} (c_s - c_b) \qquad (4.7)$$

For the dissolution of small particles Goyan (105) proposed the following model which combines 4.5 and 4.7 :

$$dw/dt = - A (D/r + \sqrt{\gamma D}) (c_s - c_b) \qquad (4.8)$$

When the radius of the particle, r, diminishes as $\gamma \rightarrow 0$ this equation reduces to 4.5 (in the case r = h).

LIMITED SOLVATION RATE THEORIES

The equations 4.5 and 4.6 based on Nernst film theory assume that $c_i \sim c_s$ which should hold when the interfacial reaction rate is large compared with the diffusion rate. If this is not the case c_s must be replaced by c_i in these equations and the dissolution mechanism becomes significantly more complex. In general $c_i < c_s$ always, so it is somewhat difficult to decide

between diffusion and reaction rate controlled dissolution.

Wurster and Taylor (104) proposed a method based on the temperature dependence of k. They claim that the process is diffusion-controlled if the temperature coefficient is approximately 1.3 and interfacial reaction rate controlled if it is close to 2.0. Higuchi (4) derived the following equation for an interfacial reaction rate and diffusion rate controlled process:

$$dw/dt = - \frac{AD}{h + D/k_i} (c_s - c_b) \qquad (4.9)$$

where k_i was termed the effective interfacial transport rate constant. This equation was presented in slightly modified form in a later publication (60). It is seen that when the effective interfacial transport rate constant is large i.e. when $k_i >> D/h$ the equation reduces to 4.5 and the process is diffusion rate limited. If $k_i << D/h$ equation 4.9 reduces to:

$$dw/dt = - A k_i (c_s - c_b) \qquad (4.10)$$

and the process is truly reaction rate controlled.

SPHERICAL DISSOLUTION

Nernst theory was derived for a plane interface. It is of interest to apply similar assumptions to a spherical particle, i.e. assume (Fig. 4.2):

1. Spherical symmetry in dissolution
2. $c = c_i \sim c_s$ as $r = r_t$
3. $c = c_b$ at $r = r_t + h$
4. Solute transported only by Ficks diffusion ($J_D = -D\partial c/\partial r$) in stagnant layer, $r_t \leq r \leq r_t + h$
5. quasi steady state conditions in the sense that the mass transport rate through a spherical surface in the diffusion

layer is constant at a fixed time, i.e.

$$(4\pi r^2 D\, dc/dr)_t = \text{constant} \quad r_t \leq r \leq r_t + h \tag{4.11}$$

or:

$$\left[d(r^2 dc/dr)/dr\right]_t = 0 \tag{4.12}$$

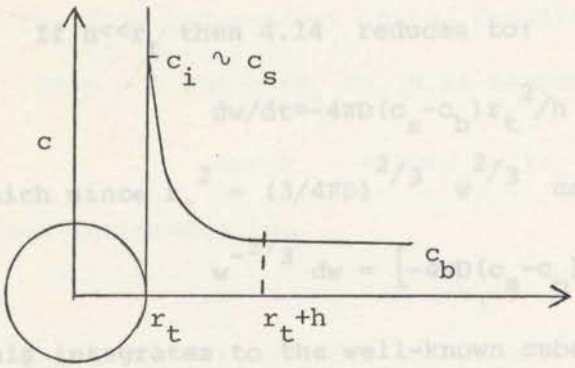


Fig. 4.2

Solving 4.12 under boundary conditions 2 and 3 above leads to:

$$(dc/dr)_t = -(c_s - c_b)\, r_t\, (r_t+h)/(hr^2) \quad r_t \leq r \leq r_t + h \tag{4.13}$$

From this expression, it is seen that the concentration gradient in the diffusion layer is *not* constant as it was in the case for the plane body (Fig. 4.1). The dissolution rate of the spherical particle according to assumption 4 is thus given by:

$$dw/dt = 4\pi r_t^2 D(dc/dr)_{r=r_t} = -4\pi D(c_s - c_b)(r_t+h) r_t/h \tag{4.14}$$

The diffusion layer thickness, h, is related to the intensity of the solvent flow or agitation in the vicinity of the spherical interface. This agitation will depend in some way on the size, r_t, of the particle. Therefore, some functional relationship must exist between the thickness of the diffusion layer and the radius of the dissolving particle.

Several functional relationships between h and r_t can be assumed. On this basis, integration of 4.14 will then lead to different models for single particle dissolution as demonstrated in the following section.

Constant Diffusion Layer Thickness.

Firstly, it would be of interest to investigate the case where h is constant independent of particle size, since this assumption has often been made in the literature. Two cases will be considered:

1. If $h \ll r_t$ then 4.14 reduces to:

$$dw/dt \simeq -4\pi D(c_s - c_b) r_t^2/h \qquad (4.15)$$

which since $r_t^2 = (3/4\pi\rho)^{2/3} w^{2/3}$ can be written:

$$w^{-2/3}\, dw = \left[-4\pi D(c_s - c_b)(3/(4\pi\rho))^{2/3}/h\right] dt \qquad (4.16)$$

this integrates to the well-known cube root law (97)

$$w^{1/3} = w_o^{1/3} - k_1 t \qquad (4.17)$$

$$\text{with:} \quad k_1 = (4\pi/3)^{1/3} D(c_s - c_b)\rho^{-2/3}/h \qquad (4.18)$$

2. If $h \gg r_t$ then 4.14 reduces to:

$$dw/dt \simeq -4\pi D(c_s - c_b) r_t \qquad (4.19)$$

which similarly to above integrates to:

$$w^{2/3} = w^{2/3} - k_2 t \qquad (4.20)$$

$$\text{where} \quad k_2 = 2(4\pi/3)^{2/3} D\rho^{-1/3} (c_s - c_b) \qquad (4.21)$$

It is interesting to note that in this case the diffusion layer thickness, h, is not a part of the rate parameter k_2.

Variable Diffusion Layer Thickness.

It is expected from hydrodynamic considerations that h decreases with increasing r. Any monotonically decreasing functional relationship between h and r could be considered. This investigation will be restricted to simple relationships based on the following power model:

$$h = k_p r_t^{-p} \qquad k_p, p > 0 \qquad (4.22)$$

where k_p and p are some positive constants. Several cases can be considered depending upon the value of p and the size of h relative to r_t. It should first be noted that in the cases where $h \gg r_t$ Eq. 4.14 still leads to Eq. 4.20 independent of the functional relationship between h and r_t.

1. If p = 1 and $h \ll r_t$ Eq. 4.14 becomes:

$$dw/dt = -4\pi D(c_s - c_b) r_t^3 / k_p \qquad (4.23)$$

which integrates to:

$$w = w_o \text{ EXP } (-k_3 t) \qquad (4.24)$$

where $$k_3 = 3D(c_s - c_b)/k_p \rho \qquad (4.25)$$

2. If p = 1 and h is not very different in magnitude from r_t Eq. 4.14 leads to the following expression:

$$w^{2/3} = (w_o^{2/3} + k_4) \text{ EXP } (-k_3 t) - k_4 \qquad (4.26)$$

where $$k_4 = k_p (4\pi\rho/3)^{2/3} \qquad (4.27)$$

and k_3 is given by 4.25.

3. If p = 2 and $h \ll r_t$ the following equation is obtained:

$$w^{1/3} = (w_o^{-1/3} + k_5 t)^{-1} \qquad (4.28)$$

where $$k_5 = 4\pi D(c_s - c_b)(4\pi\rho/3)^{4/3}/3k_p \qquad (4.27)$$

APPROACHES NOT BASED ON DIFFUSION LAYER ASSUMPTIONS

The Noyes-Whitney model applied to a spherical particle takes the form:

$$dw/dt = - kA (c_s - c_b) \qquad (4.28)$$

where k may or may not be constant or depend on the particle radius.

1. If k = constant 4.28 integrates to yield the model proposed by

Hixson and Crowell (97)[1]:

$$w^{1/3} = w_o^{1/3} - k_6 t \qquad (4.29)$$

where[1] $$k_6 = k(c_s - c_b)(4\pi/3)^{1/3}\rho^{-2/3} \qquad (4.30)$$

2. If k is inversely proportional to the particle radius, say, $k = k_7/r_t$ then 4.28 integrates to:

$$w^{2/3} = w_o^{2/3} - k_8 t \qquad (4.31)$$

where $$k_8 = 2k_7(c_s - c_b)(4\pi/3)^{2/3}\rho^{-1/3} \qquad (4.32)$$

3. If k is inversely proportional to the square root of the particle radius, i.e. $k = k_9 r_t^{-\frac{1}{2}}$, Eq. 4.28 integrates to yield:

$$w^{\frac{1}{2}} = w_o^{\frac{1}{2}} - k_{10} t \qquad (4.33)$$

$$k_{10} = k_9(2\pi)^{\frac{1}{2}}(2\rho/3)^{-1/3}(c_s - c_b) \qquad (4.34)$$

Sink Conditions.

The models for single particle dissolution presented above are derived for dissolution under sink condition. This condition can best be explained in a mathematical context to be a condition under which the change in c_b is so small that the errors, introduced by considering $(c_s - c_b)$ or $(c_i - c_b)$ to be constant, are acceptable. This definition is related to the concept of ignoring any time dependence of the concentration terms in the integration of the dissolution rate equation.

Comparing Models.

It is interesting to note that several of the dissolution models above could be derived both on the basis of the diffusion layer theory and

---

1. These workers did not specify the composition of k_6 but assumed it did not vary with the progress of the dissolution.

on the more simple model 4.28. In both approaches certain assumptions were made about the relationship between either the diffusion layer thickness, h, and the particle size or the rate parameter k (4.28) and the particle size.

In spite of oversimplification of the dissolution mechanism in the diffusion layer approach, it is appealing since it has greater conceptual value than the other approach. The flexibility of models based on a diffusion layer seems also to be greater.

Of the models considered above the cube root model (4.17 , 4.29), the square root model (4.33) and the 2/3-root model (4.20 , 4.31) have all been proposed in the literature (97,106,107).

There still seems to be some controversy as to which of these models best describes dissolution of the single particle. This is mainly because there appears to be no accurate experimental information directly concerning the dissolution of single particles.

The inferences made have been based on multiparticle dissolution from which it may be difficult to deduce single particle dissolution behaviour, particularly when the particle size distribution is not completely monodisperse (see Chapter 6).

In the experimental evaluation of the 1/3,1/2 and 2/3 root models in Chapter 7, the cube root model appears to be the best although further investigation is necessary to confirm this finding. It may well be that a more flexible model applies which is close to the 1/3 root model in the beginning and approaches the 1/2 and 2/3 root model as the dissolution process progresses. Such behaviour is expected if the dissolution follows Eq. 4.14. This equation reduces to the 1/3 model if $h \ll r_t$, which is likely to be true in the initial stages of dissolution if the particle size then is relatively large. Later when the particle is reduced in size the hydrodynamic activity in the vicinity of the interface will be considerably

smaller resulting in an increased h value so that for $h \gg r_t$ the dissolution reaction approaches the 2/3 root model. The intermediate state will result in a "square-root dissolution". This condition should occur when $(r_t+h)r_t/h \sim r_t^{3/2}$ i.e. when $h \sim r_t^2/(r_t^{3/2} - r_t)$.

STATIC DISSOLUTION MODELS

Study of single particle dissolution in which there is no agitation is of interest for dissolution in vehicles of extremely high viscosity in which the solute mass transport in most cases is completely diffusion-controlled. If a spherical particle is considered the system is mathematically well defined. Substantial theoretical studies have been performed in this area in chemical engineering (108-113). The many different mathematical approaches presented have been based on different assumptions. For example, the 112 assumption that the process is wholly diffusion controlled (109,/) or partly diffusion controlled (113). Most of the mathematical models have been described in terms of partial differential equations which for given boundary conditions can only be solved numerically using a digital computer.

It is interesting to note that the radius *vs.* time relationship computed in this way by Cable and Evans (109) and Ready and Cooper (112) is approximately linear except for an initial transient period. This means that the dissolution of a spherical particle under static conditions can be approximated (disgregarding the initial phase) by the cube root law. (The 1/3 model implies a linear decrease in radius with time.)

Of the equations presented for static spherical dissolution, the following equations given by Rosner (113) are appealing because of their simplicity:

$$dr/dt = (D\alpha/r) \ln ((1-c_\infty)/(1-c_i)) \quad (4.35)$$

$$k (c_s-c_i)^p = (D/r) \ln ((1-c_\infty)/(1-c_i)) \quad (4.36)$$

Where α is the ratio of solvent density/solute density, k is the rate constant governing the interface kinetics, p is the order of the solvation reaction. It was assumed that this reaction was of the ordinary Berthroud form, (100), i.e., that p = 1. The term c_∞ is the concentration infinitely far from the centre of the particle, which is the same as the initial concentration. In most cases therefore $c_\infty = 0$.

The above two equations define, in a computationally simple way, the variation of particle radius with time for given values of D, α, c_s and r_o. For example 4.36 can be solved numerically for c_i and this value substituted into 4.35 which by numerical integration provides the r_t-value for the chosen value of t. Such calculations were performed on a computer by Ridgway and Peacock (114) for various simple organic compounds.

The validity of the above two equations depends primarily on the following extended quasi steady-state assumption: "The instantaneous concentration field surrounding the dissolving sphere is approximated by the steady-state concentration field surrounding a hypothetical spherical solid of the same size, through which solute is being artificially forced at a mass rate equal to the instantaneous rate of dissolution".

If the interfacial concentration c_i is constant and close to the solubility concentration c_b the dissolution can be obtained directly by solving Fick's law:

$$\partial c/\partial t = D\nabla^2 c \quad (4.37)$$

under the following boundary conditions:

$$\text{B.C.1.} \quad c = c_s \quad r = r_t \quad t > 0 \quad (4.38)$$

$$\text{B.C.2.} \quad c = 0 \quad r > r_o \quad t = 0 \quad (4.39)$$

$$\text{B.C.3.} \quad c = 0 \quad r = \infty \quad (4.40)$$

This yields the following expression for the concentration profile in the liquid (115):

$$c(r,t) = c_s \frac{r_t}{r} \operatorname{erfc} \frac{r-r_t}{2\sqrt{Dt}} \qquad (4.41)$$

so that the mass flux per unit surface area at the interface is:

$$J_i = -D(\partial c/\partial r)_{r=r_t} = Dc_s\left[(\pi Dt)^{-\frac{1}{2}} + 1/r_t\right] \qquad (4.42)$$

Furthermore:

$$dw/dt = -4\pi r_t^2 J_i \qquad (4.43)$$

Now since $dw/dt = d(4\pi r_t^3 \rho/3)/dt = 4\pi\rho r_t^2 \, dr_t/dt$, 4.42 becomes:

$$dr_t/dt = -J_i/\rho \qquad (4.44)$$

i.e.

$$dr_t/dt = -\frac{Dc_s}{\rho}\left[(\pi Dt)^{-\frac{1}{2}} + 1/r_t\right] \qquad (4.45)$$

This equation can readily be integrated numerically to yield the time dependence of r_t and hence a numerical solution for the single particle dissolution for given values of ρ, D and c_s.

As dissolution proceeds the term $(\pi Dt)^{-\frac{1}{2}}$ in the bracket in 4.45 decreases while in the same time the term $1/r_t$ increases. Thus after the initial transition period $((\pi Dt)^{-\frac{1}{2}} + 1/r_t) \sim 1/r_t$ and 4.45 integrates to yield the 2/3-root law:

$$w^{2/3} = w_o^{2/3} - k_{11} t \qquad (4.46)$$

where

$$k_{11} = 2(4\pi/3)^{2/3} D\rho^{-1/3} c_s \qquad (4.47)$$

It is interesting and satisfactory to note that this model was also derived for the dynamic system (4.20), considering the diffusion layer theory for the case $h \gg r_t$ i.e. for a system with little agitation. In fact the rate parameters k_{11} and k_2 in 4.47 and 4.21 are identical for $c_b = 0$.

A DYNAMIC MODEL FOR A FREE FALLING SPHERICAL PARTICLE

It is very difficult to define, qualitatively or quantitatively, the agitation in the close vicinity of a particle in a mechanically stirred liquid. A better defined experimental system would be the examination of a spherical particle which falls freely through an unstirred liquid. Analysis of such a system should be particularly valuable in the study of single particle dissolution.

In the following section an attempt is made to describe this system in a mathematically rigorous way. In order to do so the following assumptions are made and will be referred to where necessary in the derivation:

A1. The particle remains spherical during dissolution.

A2. The interfacial concentration, c_i, is constant and very close to the saturation concentration, c_s.

A3. The temperature and densities remain constant.

A4. There is no c_s/r dependence.

A5. The effect of Brownian motions is negligible.

A6. The particle falls without producing turbulence, i.e. the Reynold's number, Re, is less than 0.1:

$$Re = 2r_t v_\infty \rho_l / \mu < 0.1 \qquad (4.48)$$

(The velocity of the particle in the direction of gravity is denoted v_∞, the viscosity of the liquid μ and its density ρ_l.)

A7. The fluid is incompressible.

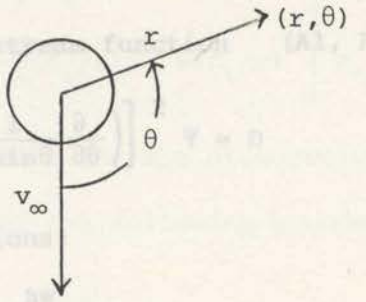
A8. The fluid flow behaves in a Newtonian way.

A9. Fick's laws of diffusion are obeyed with D=constant.

Spherical symmetry makes it convenient to choose a spherical coordinate system with the particle centre as origin and the θ coordinate

starting counter-clockwise from the direction of the gravity (Fig. 4.3)

Fig. 4.3



First it is necessary to derive an extension of Fick's law valid for incompressible liquids (A7.) in motion: A shell mass balance on an arbitrary volume element, p, in the bulk of the liquid yields (A9.):

$$\iiint_p \partial c/\partial t \, dp = \iint_s (-\underline{v}\cdot\underline{n}c + D\underline{\nabla}c\cdot\underline{n})\,ds \qquad (4.49)$$

where s is the surface of the element and $\underline{n}$ is the normal to the surface directed outward and $\underline{v}$ is the fluid velocity vector. The divergence theorem applied on r.h.s. of 4.49 yields:

$$\iiint_p \partial c/\partial t \, dp = \iiint_p (-\underline{\nabla}\cdot\underline{v}c + D\nabla^2 c)\,dp \qquad (4.50)$$

which also may be written:

$$\iiint_p \partial c/\partial t \, dp = \iiint_p (-\underline{v}\cdot\underline{\nabla}c - c\underline{\nabla}\cdot\underline{v} + D\nabla^2 c)\,dp \qquad (4.51)$$

Equating left and right integrand in 4.51, noting $\underline{\nabla}\cdot\underline{v} = 0$ (A7.) it follows that:

$$\partial c/\partial t = D\nabla^2 c - \underline{v}\cdot\underline{\nabla}c \qquad (4.52)$$

This is the extended form of Fick's second law which takes into account that the liquid in which diffusion takes place is in motion.

The fluid velocity vector function $\underline{v}$ above depends on the geometry of the particle and the velocity v_∞ with which the particle is falling through the liquid. This velocity can be related through Stokes' equation to the particle radius r_t.

The r and θ components of $\underline{v}$ can be found by solving the following differential equation for the stream function (A1, A3, A5-A8):

$$\left[\frac{\partial^2}{\partial r^2} + \frac{\sin\theta}{r^2}\frac{\partial}{\partial\theta}\left(\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\right)\right]^2 \Psi = 0 \qquad (4.53)$$

subject to the boundary conditions:

$$\text{B.C.1} \quad v_r = -\frac{1}{r^2\sin\theta}\frac{\partial\Psi}{\partial\theta} = 0 \quad \text{at} \quad r=r_t \qquad (4.54)$$

$$\text{B.C.2.} \quad v_\infty = \frac{1}{r\sin\theta}\frac{\partial\Psi}{\partial r} = 0 \text{ at } r=r_t \qquad (4.55)$$

$$\text{B.C.3.} \quad \Psi \to -\tfrac{1}{2}v_\infty r^2\sin^2\theta \quad \text{for } r\to\infty \qquad (4.56)$$

The velocity components are then obtained

$$v_r = -v_\infty\left[1-\frac{3}{2}\left(\frac{r_t}{r}\right) + \tfrac{1}{2}\left(\frac{r_t}{r}\right)^3\right]\cos\theta \qquad 0\le\theta\le\pi \qquad (4.57)$$

$$v_\theta = v_\infty\left[1-\frac{3}{4}\left(\frac{r_t}{r}\right) - \tfrac{1}{4}\left(\frac{r_t}{r}\right)^3\right]\sin\theta, \qquad 0\le\theta\le\pi \qquad (4.58)$$

The r and θ components of $\underline{\nabla}c$ in 4.52 is $\partial c/\partial r$ and $r^{-1}\,\partial c/\partial\theta$ respectively thus:

$$\underline{v}\cdot\underline{\nabla}c = -v_\infty\left[1-\frac{3}{2}(r_t/r) + \tfrac{1}{2}(r_t/r)^3\right]\frac{\partial c}{\partial r}\cos\theta \qquad (4.59)$$

$$+v_\infty\left[1-\frac{3}{4}(r_t/r) - \tfrac{1}{4}(r_t/r)^3\right] r^{-1}\frac{\partial c}{\partial\theta}\sin\theta$$

When there is no ϕ-dependence the ∇^2 operator in spherical coordinates is:

$$\nabla^2 = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\right) + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial}{\partial\theta}\right) \qquad (4.60)$$

The velocity v_∞ is related to r_t by Stokes' law:

$$v_\infty = (2\Delta\rho g/9\mu)r_t^2 \qquad (4.61)$$

where g is the acceleration of gravity and $\Delta\rho$ the solid-liquid density difference. Equations 4.52 and 4.59-61 can then be summarized to give:

$$\partial c/\partial t = \frac{D}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial c}{\partial r}\right) + \frac{D}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial c}{\partial\theta}\right)$$
$$- (2\Delta\rho g/9\mu)\left[1-\frac{3}{2}(r_t/r) + \tfrac{1}{2}(r_t/r)^3\right] r_t^2\frac{\partial c}{\partial r}\cos\theta \qquad (4.62)$$
$$+ (2\Delta\rho g/9\mu)\left[1-\frac{3}{4}(r_t/r) - \tfrac{1}{4}(r_t/r)^3\right] r_t^2\frac{\partial c}{\partial\theta}\sin\theta$$

which is the differential equation for the dissolution of spherical particle. The equation is to be solved for the following boundary conditions:

$$\text{B.C.1.} \quad c(r_t, \theta, 0) = c_s \qquad (4.63)$$

$$\text{B.C.2.} \quad c(\infty, \theta, t) = 0 \qquad (4.64)$$

$$\text{B.C.3.} \quad c(r, \theta, 0) = 0 \qquad r > r_t \qquad (4.65)$$

Because 4.62 is a nonlinear partial differential equation, it seems impossible to solve analytically using conventional techniques. The equation can, however, be solved numerically using a digital computer. It would then be convenient first to transform the variables involved to dimensionless quantities such as:

$$c^* = c/c_s \qquad 0\leq c^*<1 \qquad (4.66)$$

$$t^* = Dt/r_o^2 \qquad 0\leq t^*<\infty \qquad (4.67)$$

$$r^* = r/r_o \qquad 1\leq r^*<\infty \qquad (4.68)$$

$$u = \cos\theta \qquad -1\leq u\leq 1 \qquad (4.69)$$

It can then be shown that 4.62 becomes:

$$\partial c^*/\partial t^* = \frac{\partial}{r^{*2}\partial r^*}\left(r^{*2}\frac{\partial c^*}{\partial r^*}\right) + \frac{\partial}{\partial u}\left((1-u^2)\frac{\partial c^*}{\partial u}\right) \qquad (4.70)$$
$$- \tfrac{1}{2}\,Sc\,Re^o\left[1-\frac{3}{2}(r_t^*/r^*) + \tfrac{1}{2}(r_t^*/r^*)^3\right] r_t^{*2}\,u\frac{\partial c^*}{\partial r^*}$$
$$+ \tfrac{1}{2}\,Sc\,Re^o\left[1-\frac{3}{4}(r_t^*/r^*) - \tfrac{1}{4}(r_t^*/r^*)^3\right] r_t^*\,(1-u^2)\,\frac{\partial c^*}{\partial u}$$

where Sc is the Schmidt number ($Sc = \mu/\Delta\rho D$) and Re^{o} is the initial Reynold number ($Re^{o} = 2\, r_o v_\infty^{o}\, \rho_l/\mu$). The transformed equation is then to be solved under the following conditions:

$$\text{B.C.1.} \quad c^*(1, u, 0) = 1 \qquad (4.71)$$

$$\text{B.C.2.} \quad c^*(\infty, u, t^*) = 0 \qquad (4.72)$$

$$\text{B.C.3.} \quad c^*(r^*, u, 0) = 0 \qquad r^* > 1 \qquad (4.73)$$

The numerical solution obtained by solving 4.70 for various values of $ScRe^{o}$ will then be general in the sense that it encompasses an infinite number of combinations of values for the parameters μ, ρ_l, $\Delta\rho$, D and c_s.

NONSPHERICAL ISOTROPIC DISSOLUTION

The models for single particle dissolution considered previously in this chapter are all derived for spherical particles. Application of such models to real particle systems is complicated by the fact that pure drug particles are not spherical. The usual approach has been to treat the real particles as if they were spherical having the same surface area or volume. Such approximations may introduce substantial errors.

The influence of shape factors on dissolution kinetics of drugs has been discussed for tablets and controlled release tablets (116,117), but little attention has been given to single drug particles (118,119).

This section presents exact isotropic single-particle dissolution equations for several nonspherical forms and formulas enabling calculation of the diameters of hypothetical spherical particles which closely approximate the dissolution of these forms.

Assume that dissolution takes place istropically, that is, that the rate of dissolution per unit surface area, J, is constant so that the following equation can be written:

$$dw/dt = -JA \tag{4.74.}$$

where w is the amount undissolved and A is the surface area. This equation implies that the boundary of a *plane* interface retreats with constant speed during dissolution such that:

$$ds/dt = -J/\rho \tag{4.75.}$$

where ρ is the solid density, and s is the distance perpendicular to the interface from some fixed reference point in the dissolving solid.

Equation 4.75. integrates to:

$$s = s_o - Jt/\rho \tag{4.76.}$$

where s_o is the initial (t=o) distance to the reference point. When this equation is applied to the istropic dissolution of a spherical particle, the following equation arises:

$$w/w_o = (1 - Jt/r_o\rho)^3 \tag{4.77.}$$

or

$$(w/w_o)^{1/3} = 1 - Jt/r_o\rho \tag{4.78.}$$

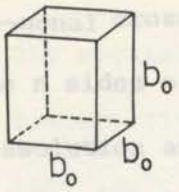
where r_o is the initial particle radius.

Therefore, when a spherical particle dissolves istropically (J=constant), it obeys the Hixson and Crowell (97) cube root law, that is, a plot of $(w/w_o)^{1/3}$ *vs.* is linear.

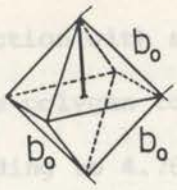
Dissolution of Prismatic Particles.

Structures I-VI (Fig. 4.4.) are 10 simple forms of the six crystal systems and illustrate the dimensional quantities b_o, c_o, l_o, h_o and α used in the following derivations. It is assumed, without loss of generality, that $b_o < c_o < l_o$.

Figure 4.4 Illustration of the six basic crystalforms for which spherical approximations to the theoretical isotropic dissolution are considered.

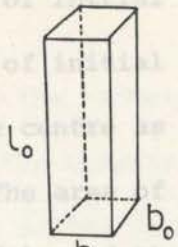


prism

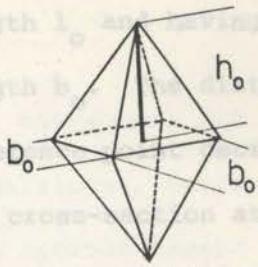


pyramid

I. ISOMETRIC

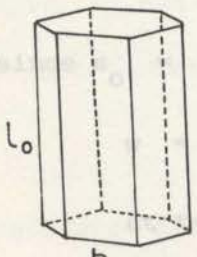


prism

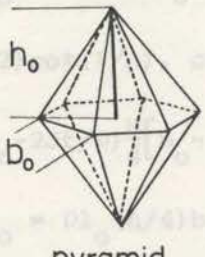


pyramid

II. TETRAGONAL

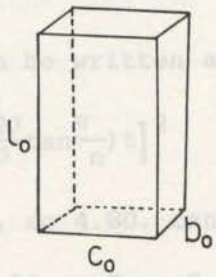


prism

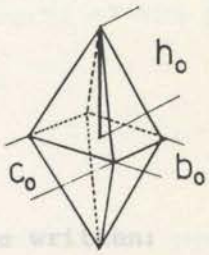


pyramid

III. HEXAGONAL

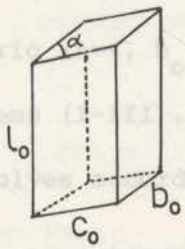


prism

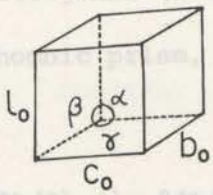


pyramid

IV. RHOMBIC



V. MONOCLINIC



VI. TRICLINIC

Consider a prismatic particle of initial length l_o and having a regular n-gonal cross-section with side of initial length b_o. The distance, s from the n sides of the polygan to the centre as reference point decreases during dissolution according to 4.76. The area of the cross-section at any time is then equal to $n\,(s_o - Jt/\rho)^2 \tan(\pi/n)$, and the length of the prism is equal to $l_o - (2Jt/\rho)$, so that particle weight at time t is:

$$w = \rho\,(l_o - 2Jt/\rho)\, n (s_o - Jt/\rho)^2 \tan(\pi/n) \qquad (4.79.)$$

which, since $s_o = (b_o/2)\cot(\pi/n)$, can be written as:

$$w = \rho\,(l_o - 2Jt/\rho)\frac{n}{4}\left[b_o - \left(\frac{2J}{\rho}\tan\frac{\pi}{n}\right)t\right]^2 \qquad (4.80.)$$

at t=o, $w_o = \rho l_o (n/4) b_o^2$, so 4.80. can be written:

$$w/w_o = \left(1 - \frac{2J}{l_o \rho}t\right)\left[1 - \left(\frac{2J}{b_o \rho}\tan\frac{\pi}{n}\right)t\right]^2 \qquad (4.81.)$$

Equation 4.81. comprises the exact dissolution equation for the prismatic forms of the isometric (n=4, $b_o = l_o$), tetragonal (n=4, $b_o = l_o$), and hexagonal (n=6), crystal systems (I-III). The rhombic prism, having a rectangular cross-section, dissolves according to:

$$w = \rho(b_o - 2Jt/\rho)\,(c_o - 2Jt/\rho)\,(l_o - 2Jt/\rho) \qquad (4.82.)$$

because of the istropic retreat of all surfaces. Similarly to 4.81., this equation can be written:

$$w/w_o = (1 - 2Jt/b_o\rho)\,(1 - 2Jt/c_o\rho)\,(1 - 2Jt/l_o\rho) \qquad (4.83.)$$

The monoclinic prismatic particle, having a parallelogram cross-section with an acute angle α, at any time has a cross-sectional area equal to bc sin α, where $b = b_o - \left[(2J/\rho)\sin\alpha\right]t$ and $c = c_o - \left[(2J/\rho)\sin\alpha\right]t$, so it dissolves according to:

$$w = \rho\,(l_o - 2Jt/\rho)\left[b_o - \left(\frac{2J}{\rho}\sin\alpha\right)t\right]\left[c_o - \left(\frac{2J}{\rho}\sin\alpha\right)t\right]\sin\alpha \qquad (4.84.)$$

or:
$$w/w_o = \left(1 - \frac{2Jt}{l_o\rho}t\right)\left[1 - \left(\frac{2J}{b_o\rho}\sin\alpha\right)t\right]\left[1 - \left(\frac{2J}{c_o\rho}\sin\alpha\right)t\right] \qquad (4.85.)$$

Dissolution of Pyramidal Particles.

The regular pyramidal forms of the isometric, tetragonal and hexagonal systems (I-III) all dissolve like spherical particles, following the "cube root law": all plane surfaces of the pyramid retreat toward its centre of symmetry with the same constant speed during istropic dissolution. Therefore, the shape of the pyramid remains the same while its size diminishes. For example, it can be shown geometrically that all lengths of the prism decrease by a factor of $(1-Jt/r_o\rho)$, where r_o, given by:

$$r_o = \tfrac{1}{2} h_o b_o (h_o^2 + b_o^2/4)^{-\frac{1}{2}} \qquad (4.86.)$$

is the radius of the largest sphere that can be contained in the pyramid initially. The weight of the regular n-gonal prism at time t is equal to $\frac{1}{6}\rho n h_t b_t^2 \cot(\pi/n)$, where the height, h_t, and side, b_t, are $h_t=h_o(1-Jt/r_o\rho)$ and $b_t=b_o(1-Jt/r_o\rho)$, respectively, according to the above theory. Thus, its weight is:

$$w = \frac{1}{6}\rho n h_o b_o^2 (1-Jt/r_o\rho)^3 \cot(\pi/n) \qquad (4.87.)$$

from which it follows that:

$$w/w_o = (1-Jt/r_o\rho)^3 \qquad (4.88.)$$

This equation is identical to 4.77. Therefore, a regular pyramidal crystal form dissolves in identical manner to the largest (hypothetical) spherical particle that can be contained within its boundaries initially. This is also approximately true for an irregular pyramidal form such as the rhombic pyramid when the irregularity is not too extreme. It can be shown, using a double integration approach, that this crystal form dissolves according to:

$$w/w_o = (1-Jt/\rho r_1)^2 \left[1-(\frac{1}{4r_1} + \frac{3}{4r_2})\frac{J}{\rho} t\right] \qquad (4.89.)$$

where

$$r_1 = \tfrac{1}{2} h_o b_o (h_o^2 + b_o^2/4)^{-\frac{1}{2}} \qquad (4.90.)$$

$$r_2 = \tfrac{1}{2} h_o c_o (h_o^2 + c_o^2/4)^{-\frac{1}{2}} \qquad (4.91.)$$

The deviation from spherical particle dissolution (4.77.) arises from the fact $b_o \neq c_o$. If $b_o = c_o$, then 4.89. reduces to the special cases 4.88. as expected.

To evaluate these single-particle dissolution equations, it is convenient to present them in a transformed simplified form which better illustrates their *intrinsic dissolution profile* (96). For example 4.81. can be transformed to:

$$w/w_o = (1-Ft^*)\ (1-t^*)^2 \qquad (4.92.)$$

$$\text{or} \qquad (w/w_o)^{1/3} = (1-Ft^*)^{1/3}\ (1-t^*)^{2/3} \qquad (4.93.)$$

where:

$$t^* = \left(\frac{2J}{b_o \rho} \tan\frac{\pi}{n}\right) t \qquad (4.94.)$$

is denoted *time length* and:

$$F = \frac{b_o}{l_o} \cot(\pi/n) \qquad (4.95.)$$

is denoted the *shape ratio*. This form of the equation clearly shows that the intrinsic dissolution profile depends only on the value of the dimensionless shape ratio, F, which defines the particle shape. Furthermore, the transformation makes it more convenient to evaluate the extent to which dissolution of the prismatic particle deviates from spherical particle dissolution (i.e. from the cube root law). For F=1 i.e. when $(w/w_o)^{1/3} = 1-t^*$, there is no such deviation; however, as F decreases, the deviation becomes more significant, i.e., when the length of the particle relative to its side length or "diameter" becomes more extreme.

It is seen (Fig. 4.5.) that as F decreases, the deviation from the cube root law becomes larger. Dissolution then approaches "the square root law", that is, a linear relationship between $(w/w_o)^{\frac{1}{2}}$ and time length (or time). This is in agreement with the fact that, for small F values, 4.92. approximates $(w/w_o)^{\frac{1}{2}} = 1-t^*$.

The cube root law and the square root law were each postulated previously as a model for the dissolution of spherical particles under sink conditions (97,106). Pure drug particles are not spherical, however, but are often prismatic in shape. Therefore, the particle shape effect should be considered in any experimental evaluation of such models.

The dissolution equation for a rhombic pyramidal particle, 4.89., can also be transformed to 4.92. where:

then:
$$t^* = \frac{2J}{\rho}\left(h_o^2 + b_o^{2/4}\right)^{-\frac{1}{2}} t \qquad (4.96.)$$

and:
$$F = \frac{1}{4} + \frac{3}{4}\left(\frac{h_o^2 + c_o^{2/4}}{h_o^2 + b_o^{2/4}}\right)^{\frac{1}{2}} \qquad (4.97.)$$

It is seen (Fig. 4.6.) that the shape ratio, F, for this pyramidal particle form does not deviate much from 1 for most shapes, indicating that in most cases dissolution closely approximates that of a spherical particle.

Dissolution equations for either a rhombic (4.83.) or a monoclinic (4.85.) particle can similarly be written in a common transformed form as:

$$w/w_o = (1-F_1t^*)(1-F_2t^*)(1-t^*) \qquad (4.98.)$$

where $F_1 = b_o/c_o$, $F_2 = b_o/l_o$, and $t^* = 2Jt/b_o\rho$ for a rhombic particle and $F_1 = b_o/c_o$, $F_2 = (b_o/l_o)\sin\alpha$, and $t^* = (2J/b_o\rho)\sin\alpha\ t$ for a monoclinic particle.

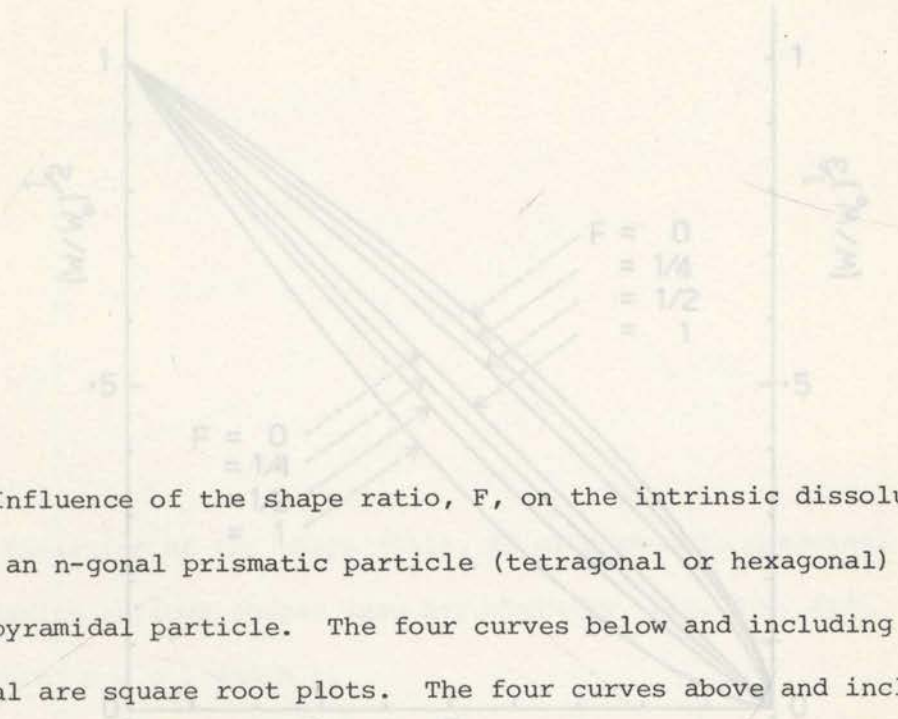
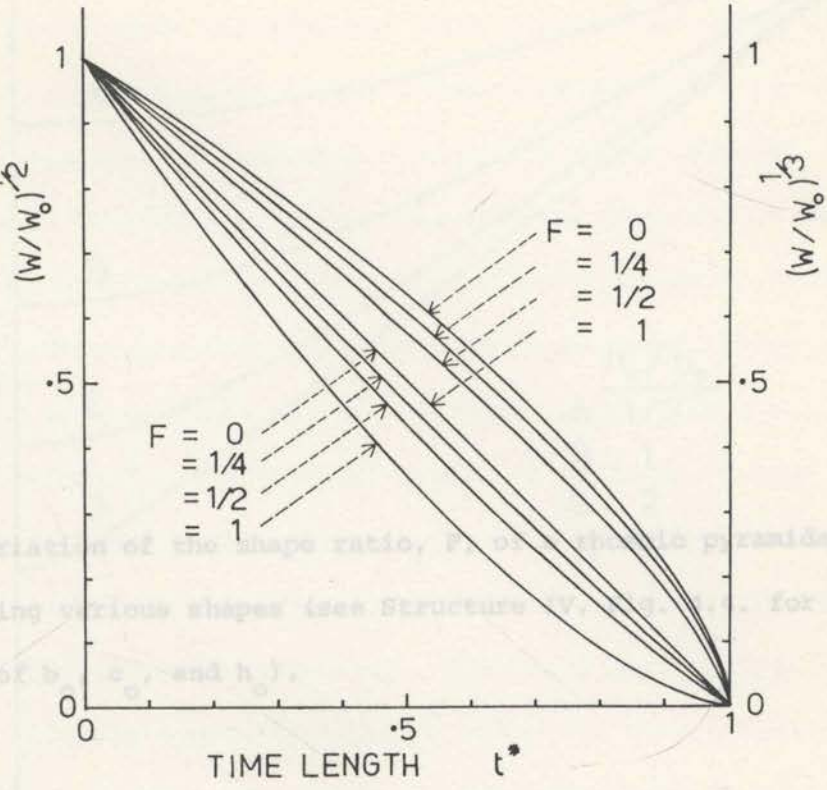


Figure 4.5

Influence of the shape ratio, F, on the intrinsic dissolution profile of an n-gonal prismatic particle (tetragonal or hexagonal) or a rhombic pyramidal particle. The four curves below and including the diagonal are square root plots. The four curves above and including the diagonal are cube root plots. The dissolution deviates increasingly from the $w/w_o^{1/3}$ versus t* linear relationship (the cube root law) as the shape ratio becomes less than 1 and approaches a linear $(w/w_o)^{1/2}$ versus t* relationship (the square root law).


1
·5
0
0
·5
1
$(W/W_o)^{1/2}$
$(W/W_o)^{1/3}$
F = 0
= 1/4
= 1/2
= 1
F = 0
= 1/4
= 1/2
= 1
TIME LENGTH t*

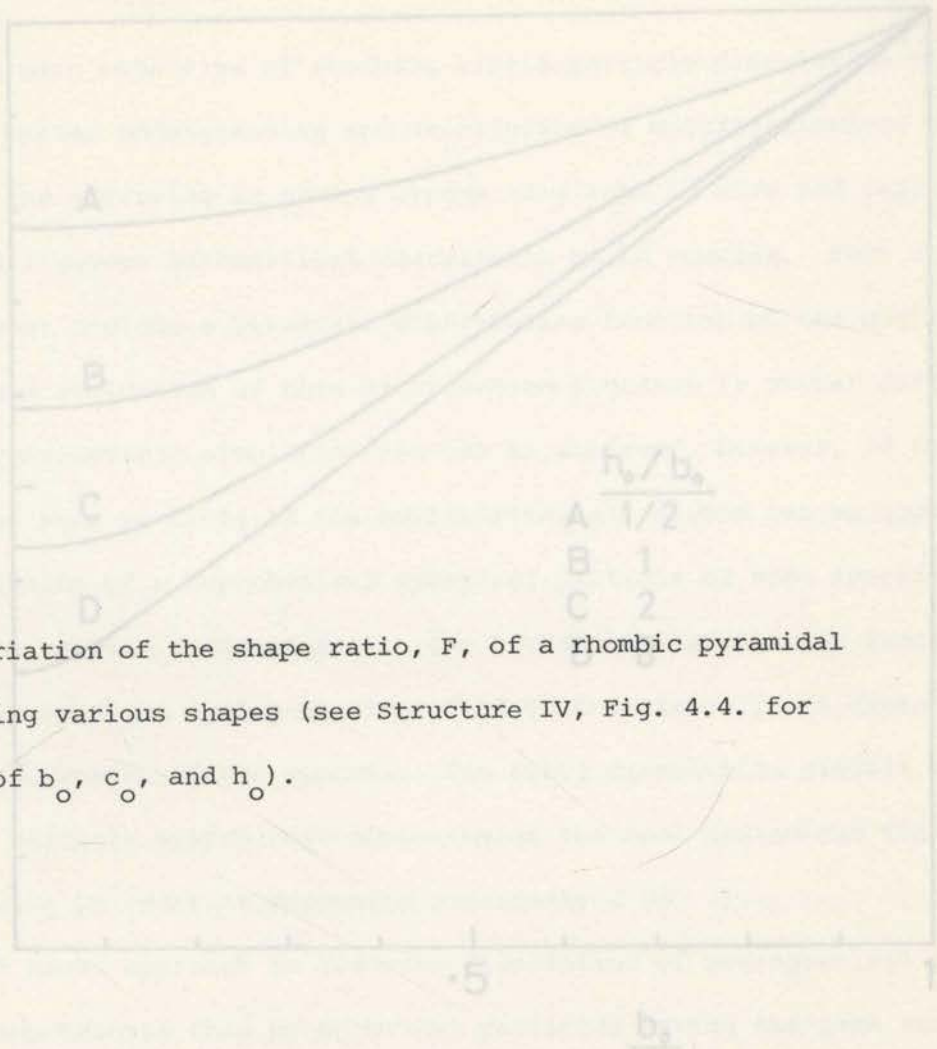
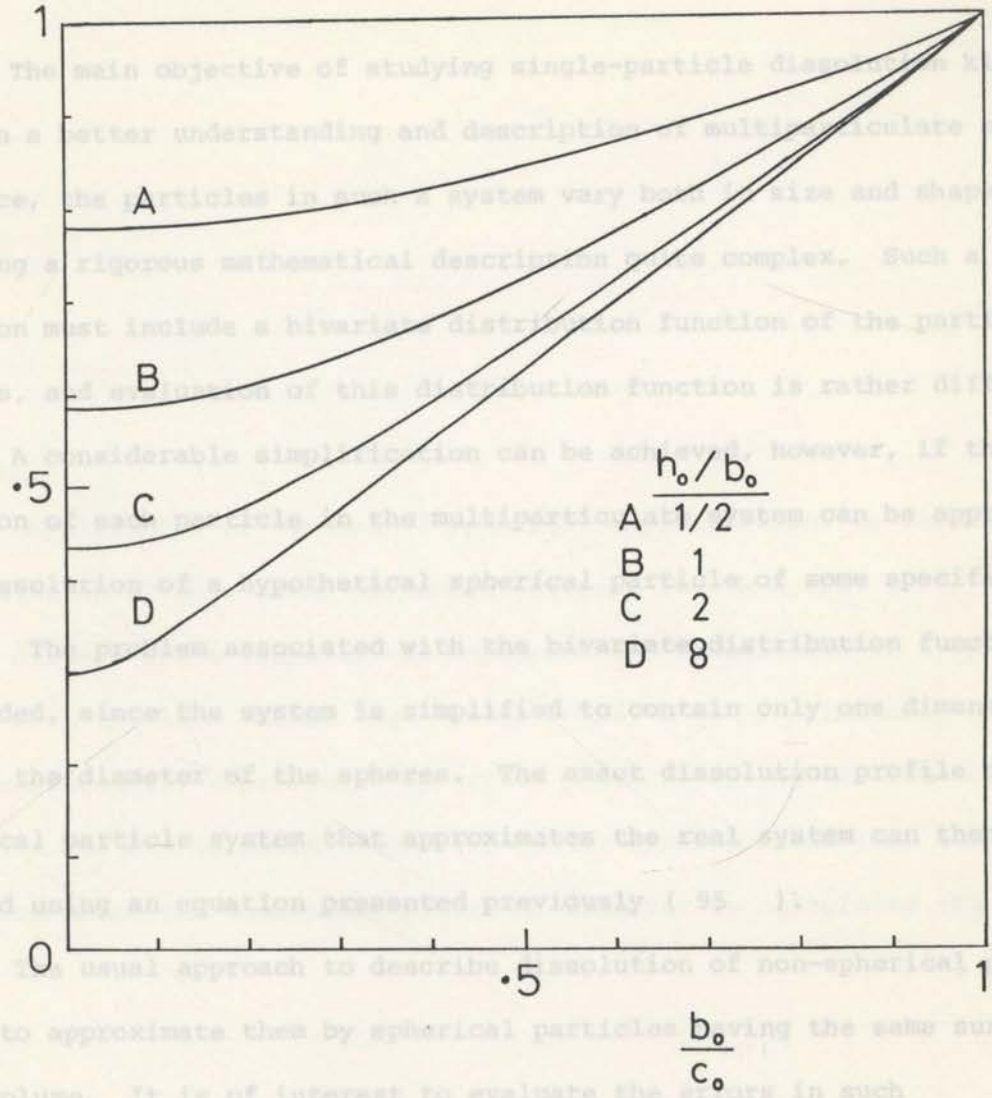


Figure 4.6.

Variation of the shape ratio, F, of a rhombic pyramidal particle having various shapes (see Structure IV, Fig. 4.4. for definitions of b_o, c_o, and h_o).


1
F
·5
0
·5
1
A
B
C
D
h_o/b_o
A 1/2
B 1
C 2
D 8
b_o/c_o

Spherical Approximations.

The main objective of studying single-particle dissolution kinetics is to gain a better understanding and description of multiparticulate systems. In practice, the particles in such a system vary both in size and shape, thus making a rigorous mathematical description quite complex. Such a description must include a bivariate distribution function of the particle dimensions, and evaluation of this distribution function is rather difficult.

A considerable simplification can be achieved, however, if the dissolution of each particle in the multiparticulate system can be approximated by the dissolution of a hypothetical *spherical* particle of some specified diameter. The problem associated with the bivariate distribution function is then avoided, since the system is simplified to contain only one dimensionless variable, the diameter of the spheres. The exact dissolution profile of the hypothetical particle system that approximates the real system can then be calculated using an equation presented previously (95).

The usual approach to describe dissolution of non-spherical particles has been to approximate them by spherical particles having the same surface area or volume. It is of interest to evaluate the errors in such approximations. A spherical particle, having the same surface area as an n-gonal prismatic particle with shape ratio F, dissolves according to:

$$(w/w_o)^{1/3} = 1 - \left(\frac{2\pi\cot\frac{\pi}{n}}{n(1+2/F)}\right)^{\frac{1}{2}} t^* \qquad (4.99.)$$

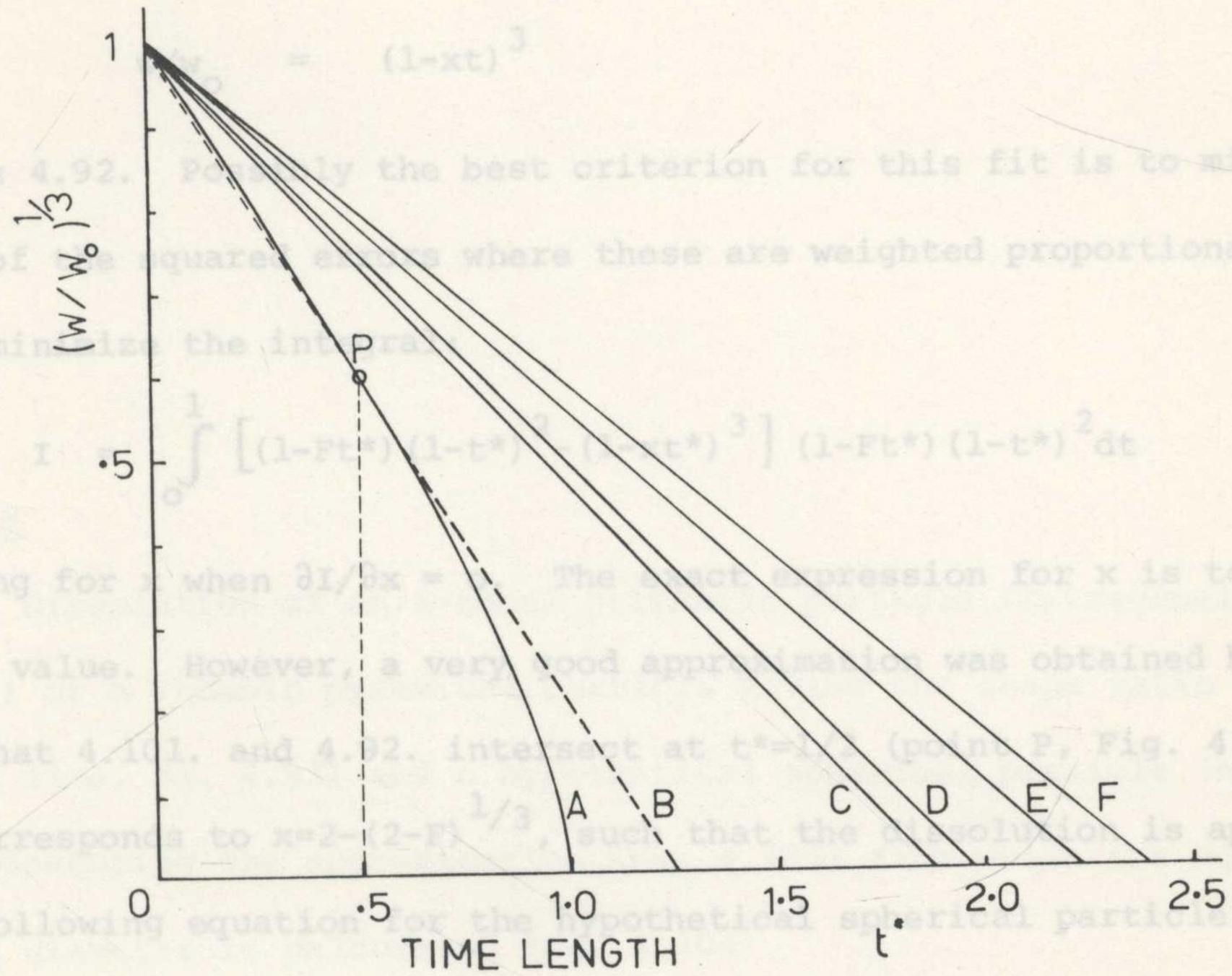
Or if it has the same volume, it dissolves according to:

$$(w/w_o)^{1/3} = 1 - \left(\frac{2\pi F\cot\frac{\pi}{n}}{3n}\right)^{1/3} t^* \qquad (4.100.)$$

where t* and F are defined by 4.94. and 4.95. Figure 4.7. shows the substantial errors introduced by such approximations based on equal surface area or volume. This is not only the case for F = 1/4 but for all other values of the shape

Figure 4.7

Application of the use of spherical approximations to describe the dissolution of an n-gonal prismatic particle (curve A, Eq. 4.93, F=1/4). Curve C and E respectively represent the dissolution of a spherica particle with the same volume (Eq. 4.100, F=1/4, n=4,6). Similarly curve D and F respectively represent a spherical particle having the same surface area (Eq. 4.99, F=1/4, n=4,6). Curve B is the approximation (Eq. 4.103, F=1/4) from which the equivalent spherical diameter is calculated (Eq. 4.104).


1
·5
0
·5
1·0
1·5
2·0
2·5
$(W/W_o)^{1/3}$
P
A
B
C
D
E
F
TIME LENGTH
t*

ratio less than 1.

The problem of finding the diameter of the spherical particle that best approximates the dissolution of an n-gonal prismatic particle is mathematically the same as finding a quantity, x, such that:

$$w/w_o = (1-xt)^3 \qquad (4.101.)$$

best fits 4.92. Possibly the best criterion for this fit is to minimize the sum of the squared errors where these are weighted proportional to w/w_o, i.e. to minimize the integral:

$$I = \int_o^1 \left[(1-Ft^*)(1-t^*)^2-(1-xt^*)^3\right](1-Ft^*)(1-t^*)^2 dt \qquad (4.102.)$$

by solving for x when $\partial I/\partial x = o$. The exact expression for x is too complex to be of value. However, a very good approximation was obtained by choosing x such that 4.101. and 4.92. intersect at $t^*=1/2$ (point P, Fig. 4.7.), which corresponds to $x=2-(2-F)^{1/3}$, such that the dissolution is approximated by the following equation for the hypothetical spherical particle:

$$(w/w_o)^{1/3} = 1 - \left[2-(2-F)^{1/3}\right] t^* \qquad (4.103.)$$

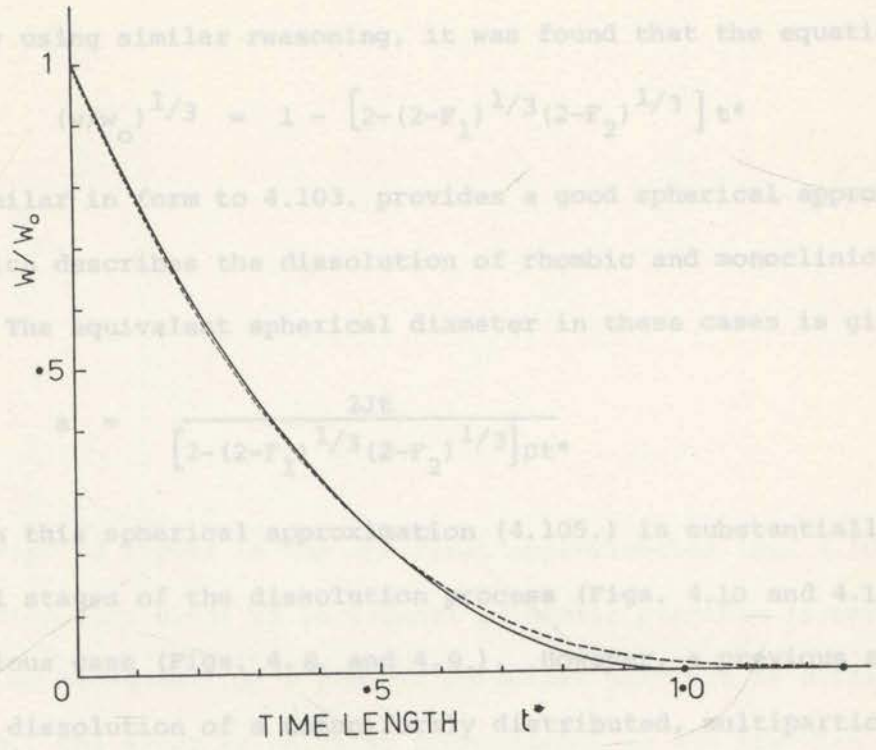
The equivalent spherical diameter, a, i.e., the diameter of this spherical particle, can then be obtained by equaling the right-hand sides of 4.103. and 4.78. from which it follows that:

$$a = \frac{2Jt}{\left[2-(2-F)^{1/3}\right]\rho t^*} \qquad (4.104.)$$

Although the approximation given by 4.103. (curve B, Fig. 4.7.) does not seem to be a particularly good fit to the exact dissolution curve calculated for the n-gonal particle (curve A), when the same two curves are plotted as w/w_o *vs.* time length (Fig. 4.8.) instead of as $(w/w_o)^{1/3}$, it is clear that this is because of the nonlinear scaling in the cube root plot. The stippled curve (Fig. 4.8.) representing the spherical approximation shows

Figure 4.8

Dissolution of an n-gonal prismatic particle (tetragonal or hexagonal) or a rhombic pyramidal particle having the shape ratio F=1/4 (unbroken line, Eq. 4.93) and a hypothetical spherical particle (broken line) representing the approximation (Eq. 4.103) from which the equivalent spherical diameter is calculated (Eq. 4.104).


1
W/W₀
•5
0
•5
1•0
TIME LENGTH t*

excellent fit to the true dissolution. The weighted errors of the spherical approximation were calculated for various values of the shape ratio F and showed (Fig. 4.9.) that this choice for the approximation was satisfactory.

By using similar reasoning, it was found that the equation:

$$(w/w_o)^{1/3} = 1 - \left[2-(2-F_1)^{1/3}(2-F_2)^{1/3}\right] t^* \qquad (4.105.)$$

which is similar in form to 4.103. provides a good spherical approximation to 4.98, which describes the dissolution of rhombic and monoclinic prismatic particles. The equivalent spherical diameter in these cases is given by:

$$a = \frac{2Jt}{\left[2-(2-F_1)^{1/3}(2-F_2)^{1/3}\right]\rho t^*} \qquad (4.106.)$$

The error in this spherical approximation (4.105.) is substantially greater in the final stages of the dissolution process (Figs. 4.10 and 4.11.) than in the previous case (Figs. 4.8. and 4.9.). However, a previous study (96) showed that dissolution of a nonuniformly distributed, multiparticulate system is only slightly affected by the dissolution behaviour of the smallest particles. Substantial truncation at the lower end of the particle-size distribution had very little effect on the dissolution profile calculated (95). Thus, approximation error in the later stage of the single-particle dissolution does not introduce the same degree of error when applied to a nonuniformly distributed, multiparticulate system. The approximation (4.105.) should, therefore, yield considerably better results when applied to a multiparticulate system than might be judged from Fig. 4.10. This explains the choice of the particular weighting of the errors in the approximation procedure.

Table 4.1 summarizes the dissolution of the particle forms shown in Structures I-VI and gives formulas for the calculation of the equivalent spherical diameter in each case. The dissolution of these 10 crystal shapes can be described by three basic transformed equations of cubic form in time-

Figure 4.9

Graph of errors in the spherical approximation (Eq. 4.103) of the dissolution (Eq. 4.93) of an n-gonal prismatic particle (isometric, tetragonal, or hexagonal) or a rhombic pyramidal particle of different shape ratios. The error is weighted proportional to the fraction undissolved (w/w_o). Key: 1, F = 1; 2, F = ½; 3, F = $\frac{1}{3}$, and 4, F = ¼.

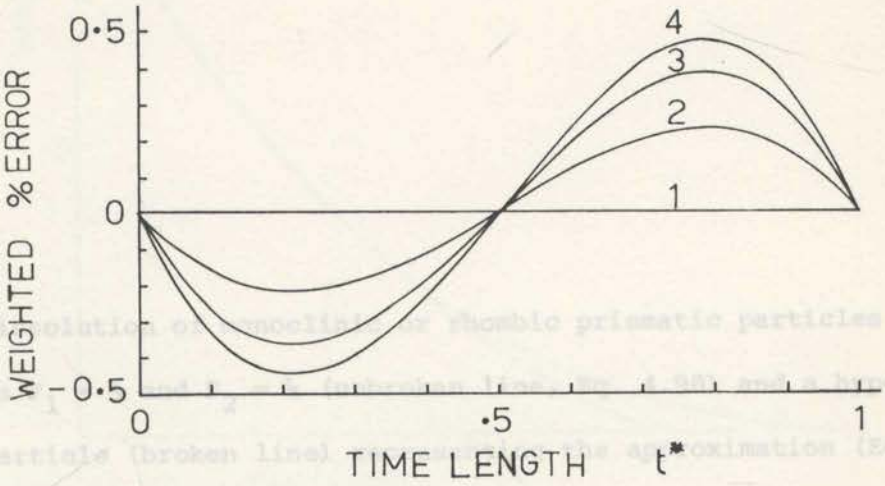
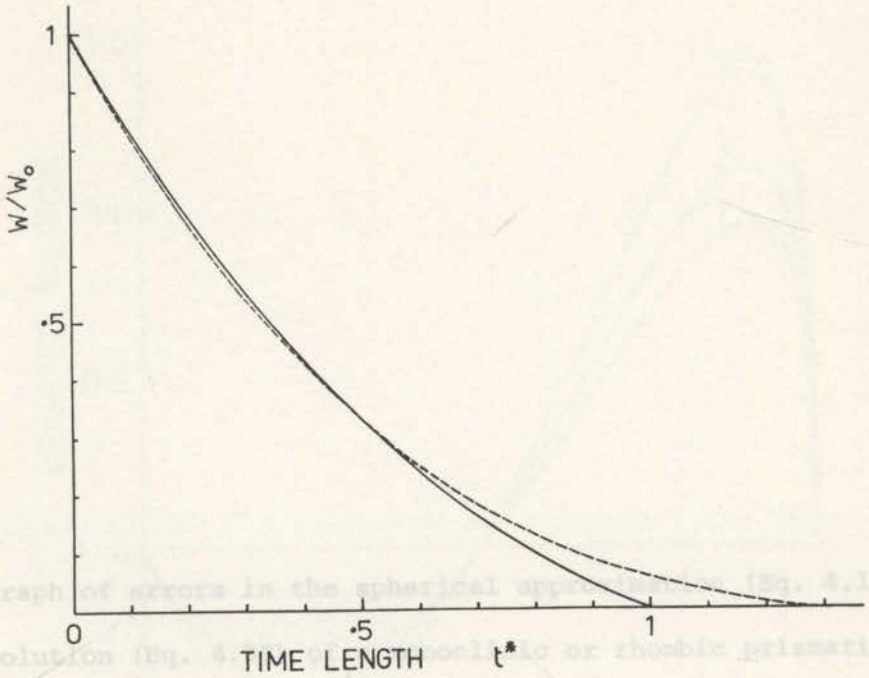

0·5
0
-0·5
WEIGHTED %ERROR
0
·5
1
TIME LENGTH t*
4
3
2
1

Figure 4.10

Dissolution of monoclinic or rhombic prismatic particles with shape ratios $F_1 = \frac{1}{2}$ and $F_2 = \frac{1}{4}$ (unbroken line, Eq. 4.98) and a hypothetical spherical particle (broken line) representing the approximation (Eq. 4.103) from which the equivalent spherical diameter is calculated (Eq. 4.104).


1
W/W₀
·5
0
·5
1
TIME LENGTH t*

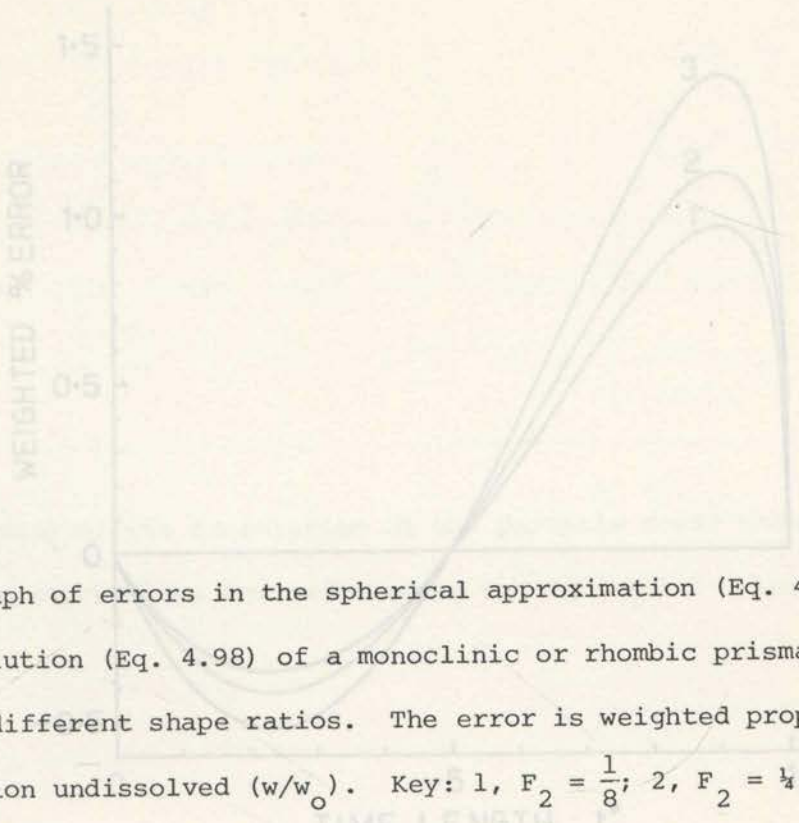


Figure 4.11

Graph of errors in the spherical approximation (Eq. 4.103) to the dissolution (Eq. 4.98) of a monoclinic or rhombic prismatic particle of different shape ratios. The error is weighted proportional to the fraction undissolved (w/w_o). Key: 1, $F_2 = \frac{1}{8}$; 2, $F_2 = \frac{1}{4}$; and 3, $F_2 = \frac{1}{3}$. $F_1 = \frac{1}{2}$.

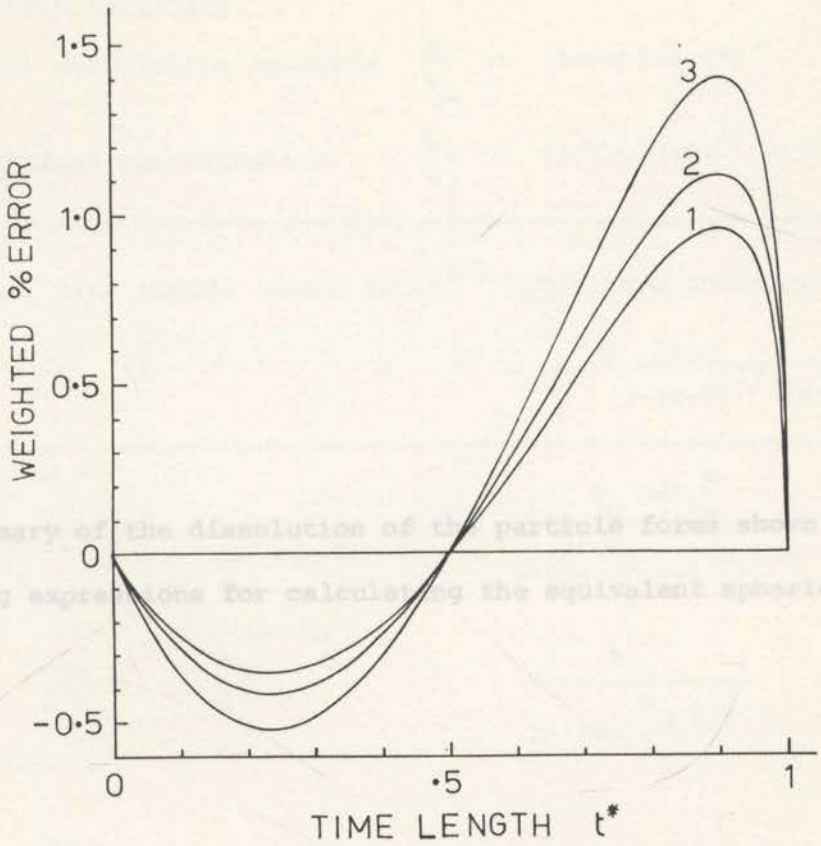

1·5
1·0
0·5
0
-0·5
WEIGHTED %ERROR
3
2
1
0
·5
1
TIME LENGTH t*

Table 4.1

Summary of the dissolution of the particle forms shown in Fig. 4.7. including expressions for calculating the equivalent spherical diameter.

DISSOLUTION OF PRISMATIC PARTICLES [a]

n-Gonal Prismatic Particles

Exact dissolution equation $\frac{w}{w_o} = (1-Ft^*)(1-t^*)^2$

Spherical approximation $\frac{w}{w_o} = (1-(2-(2-F)^{1/3})t^*)^3$

CRYSTAL SYSTEM	TIME LENGTH t^*	SHAPE RATIO [b] F	EQUIVALENT SPHERICAL DIAMETER $a = \frac{2Jt}{(2-(2-F)^{1/3})\rho t^*}$
n-Gonal	$(\frac{2J}{b_o\rho}\tan\frac{\pi}{n})t$	$\frac{b_o}{l_o}\cot\frac{\pi}{n}$	$\frac{b_o \cot\frac{\pi}{n}}{2-(2-\frac{b_o}{l_o}\cot\frac{\pi}{n})^{1/3}}$
Tetragonal (n=4)	$\frac{2J}{b_o\rho}t$	$\frac{b_o}{l_o}$	$\frac{b_o}{2-(2-\frac{b_o}{l_o})^{1/3}}$
Isometric (n=4, $b_o=l_o$)	$\frac{2J}{b_o\rho}t$	1	b_o
Hexagonal (n=6)	$\frac{2\sqrt{3}J}{3b_o\rho}t$	$\sqrt{3}\frac{b_o}{l_o}$	$\frac{\sqrt{3}\,b_o}{2-(2-\sqrt{3}\frac{b_o}{l_o})^{1/3}}$

Rhombic and Monoclinic Particles

Exact dissolution equations $\frac{w}{w_o} = (1-F_1t^*)(1-F_2t^*)(1-t^*)$

Spherical approximation $\frac{w}{w_o} = (1-(2-(2-F_1)^{1/3}(2-F_2)^{1/3})t^*)^3$

CRYSTAL SYSTEM	TIME LENGTH t^*	SHAPE RATIOS F_1	F_2	EQUIVALENT SPHERICAL DIAMETER $a = \frac{2Jt}{(2-(2-F_1)^{1/3}(2-F_2)^{1/3})\rho t^*}$

Rhombic	$\frac{2J}{b_o}$	$\frac{b_o}{c_o}$	$\frac{b_o}{l_o}$	$\frac{b_o}{2-(2-\frac{b_o}{c_o})^{1/3}(2-\frac{b_o}{l_o})^{1/3}}$
Monoclinic	$\frac{2J}{b_o \rho \sin\alpha}\, t$	$\frac{b_o}{c_o}$	$\frac{b_o}{l_o}\sin\alpha$	$\frac{b_o \sin\alpha}{2-(2-\frac{b_o}{c_o})^{1/3}(2-\frac{b_o}{l_o}\sin\alpha)^{1/3}}$

DISSOLUTION OF PYRAMIDAL PARTICLES

n-Gonal Pyramidal Particles

(Isometric, Tetragonal, Hexagonal)

Exact dissolution equation $$\frac{w}{w_o} = (1-t^*)^3$$

Time length $$t^* = \frac{2J\sqrt{h_o^2 + b_o^2/4}}{\rho}\, t$$

Equivalent spherical diameter[c] $$a = \frac{h_o b_o}{\sqrt{h_o^2 + b_o^2/4}}$$

Rhombic Pyramidal Particle

Exact dissolution equation $$\frac{w}{w_o} = (1-Ft^*)(1-t^*)^2$$

Spherical approximation $$\frac{w}{w_o} = (1-(2-(2-F)^{1/3})t^*)^3$$

Time length $$t^* = \frac{2J\sqrt{h_o^2 + b_o^2/4}}{\rho}\, t$$

Shape ratio $$F = \frac{1}{4} + \frac{3}{4}\sqrt{\frac{h_o^2 + c_o^2/4}{h_o^2 + b_o^2/4}}$$

Equivalent spherical diameter $a = \dfrac{1}{(2-(2-F)^{1/3})\sqrt{h_o^2 + b_o^2/4}}$

(a) Figure 1 defines the quantities b_o, c_o, l_o and α used.

(b) When F=1, i.e. $\dfrac{b_o}{l_o} = \tan\dfrac{\pi}{n}$ then the equivalent spherical diameter is equal to the biggest sphere that can be contained in the prismatic body. The spherical approximation of the dissolution will then become exact.

(c) This diameter is equal to the biggest sphere the pyramid can contain.

---

length or time. Dissolution can thus be described exactly by a third degree polynamial in time in all cases.

The basic assumptions behind these derivations is that the rate of dissolution per unit surface area, J, remains constant during dissolution and is the same everywhere at the interface of the dissolving crystal. This assumption can only be approximately true in practice under complete sink conditions. The higher activity at the crystal edges results in a larger J value in these areas and, therefore, a "rounding off" of the shape, so that during the later stages dissolution is slower than calculated. However, this should result in an improvement in the fit of the spherical approximation and could result in a closer fit to the real dissolution than the exact expression given for isotropic conditions. Thus, the approximating curve (stippled line, Fig. 4.8. and 4.10) is above the calculated dissolution curve in the later stages. The *true* dissolution curve, because of the rounding effect, is above the calculated curve and hence closer to the approximation.

Excellent agreement between experimental and calculated results was obtained for the dissolution of a multiparticulate system of particles, approximately tetragonal prismatic in shape, when the respective spherical approximations were applied (120).

- o O o -

CHAPTER 5

MULTIPARTICULATE DISSOLUTION

The dissolution of a multiparticulate system is considerably more complex to characterize than a single particle system. There are two main reasons for this. One reason is that it may be difficult to account for the particle size distribution effect (see Chapter 6) which is present when the powder is not monodisperse. Another reason is that there may be significant interaction between the dissolving particles; the dissolution of a particle in the powder will, in general, influence the dissolution of other particles and often to varying degrees. In evaluating the intrinsic dissolution properties of a powder it is therefore necessary to establish experimental conditions which minimize such interactions. This is made possible by the "single layer-flow through" principle employed in the dissolution cell used in these experiments (see Chapter 7 and 10).

Under such sink conditions it is possible to establish a multiparticulate dissolution model based on a single particle dissolution model and the initial particle size distribution. Such a model should describe the intrinsic dissolution properties of a powder.

Monodisperse powders.

If the initial particle size distribution can be considered to be infinitely narrow, i.e. if all the particles are of the same initial size, r_o, then the multiparticulate model becomes identical to the simple particle model when considered on a "fraction undissolved *versus* time basis".

For example if the *single* particle dissolution follows the cube root model:

$$w^{1/3} = w_o^{1/3} - kt \qquad (5.1.)$$

Then the multiparticulate model becomes:

$$W^{1/3} = W_o^{1/3} - n^{1/3}\, kt \qquad (5.2.)$$

where $$n = W_o/w_o \qquad (5.3.)$$

is the number of particles in the powder. Eqs. 5.1. and 5.2. are thus identical in the sense that:

$$W/W_o = w/w_o = \left(1-\frac{k}{w_o^{1/3}}t\right)^3 \qquad (5.4.)$$

Polydisperse powders.

-94

Several investigators (107,92) have considered the problem of exactly describing the dissolution profile of powders in relation to their particle size distribution. These authors have been concerned with powders initially consisting of particles with log-normal size distribution. Earlier attempts made use of approximations (107) or computer simulations (92). More recently, Brooke (93,94) developed an equation that permits calculation of the dissolution profile of such powders without the aid of a computer. This equation was later presented in a form to account for trimcated log-normal distributions.

A general equation that exactly describes the entire dissolution profile of powders under sink conditions is derived in the following section. The equation is valid for particles having any initial size distribution and dissolving according to any explicit single particle model. The equation is then applied to obtain an exact expression for a log-normal powder considering the cube root model.

Derivation, General Case.

Consider a powder consisting of particles which initially (t=o) have a weight density (probability distribution) $f_o(w_o)$. Let the particles dissolve independently of each other according to:

$$w = g(w_o, t, A) \tag{5.5.}$$

where w and w_o are the particle weights at time t and t=o, respectively and A collectively represents dissolution parameters such as solubility, particle density and particle shape factors. The inverse dissolution function is defined as:

$$w_o = g^{-1}(w, t, A) \tag{5.6.}$$

Using the rules of transformation of independent variables (121) the particle weight density function at time t becomes:

$$h(w) = f_o\left[g^{-1}(w,t,A)\right] \frac{d}{dw} g^{-1}(w,t,A) \tag{5.7.}$$

For 5.7. to hold, the following conditions for g must be satisfied: (a) g^{-1} must be a strictly increasing function of w for all t values, (b) g must decrease strictly with time until equal to zero, (c) g must remain equal to zero beyond that time. The latter two conditions ensure that the dissolution function reflects the actual physical conditions of the dissolution process. The first condition will rarely be violated because in application, g is nearly always a strictly increasing function of w_o for all t values. It is obvious that the second condition must be met by any dissolution equation.

The third condition is not satisfied for most equations in the literature (97,106,[107]/). To overcome this problem, it is necessary to redefine the particle weight density function such that it is generally applicable:

$$f(w) = \frac{h(w)}{\int_o^\infty h(w)\,dw} \quad \text{for } w > o \tag{5.8.}$$

$$f(w) = o \qquad \text{for } w \leq o \tag{5.9.}$$

where division by the integral is necessary to satisfy the condition that the total integral (from $-\infty$ to $+\infty$) of f(w) must be equal to 1.

The weight of undissolved powder, W, at any time, t, is equal to the product of the number of particles remaining, N_t, and the mean particle weight, which for a large number of particles is the same as the expected value of w, $E_t(w)$. Therefore, the following general equation can be written:

$$W = N_t E_t(w) \tag{5.10.}$$

The number of particles remaining at time t is:

$$N_t = N_o \int_o^\infty h(w)\,dw \tag{5.11.}$$

where the initial numbers of particles N_o is equal to the initial powder weight, W_o, divided by the initial mean particle weight:

$$N_o = \frac{W_o}{\int_o^\infty w_o f_o(w_o)\,dw_o} \tag{5.12.}$$

The mean particle weight as time t is given by:

$$E_t(w) = \int_{-\infty}^{\infty} wf(w)\,dw \tag{5.13.}$$

which according to 5.8. can be written:

$$E_t(w) = \frac{\int_o^\infty wh(w)\,dw}{\int_o^\infty h(w)\,dw} \tag{5.14.}$$

substituting Eqs. 5.11., 5.12. and 5.14. into Eq. 5.10. yields:

$$\frac{W}{W_o} = \frac{\int_o^\infty wh(w)\,dw}{\int_o^\infty w_o f_o(w_o)\,dw_o} \tag{5.15.}$$

This equation relates to unbounded particle weight distributions. In practice, the distribution is always bounded, so the limits of the integration must be changed accordingly.

Let m_o and M_o denote the initial weights of the smallest and largest particle, respectively.[1] These values then represent the lower and

1. These values are not intended to be absolute but rather represent limits giving the best fit when the actual particle distribution is approximated by any particular function. Therefore, they also represent truncation limits of the function.

upper boundaries, respectively, of $f_o(w_o)$; f(w) will be correspondingly bounded by $Pg(m_o,t,A)$ and $Pg(M_o,t,A)$. The operator P is introduced to ensure that the limits are never negative. Therefore, by definition, P is equal to 1 in the time period before the operand becomes zero and is equal to zero beyond that time. When these limits are introduced into 5.15. and h(w) is written according to 5.7., the following expression is obtained:

$$\frac{W}{W_o} = \frac{\int_{w=Pg(m_o,t,A)}^{w=Pg(M_o,t,A)} w f_o\left[g^{-1}(w,t,A)\right] \frac{d}{dw} g^{-1}(w,t,A)\,dw}{\int_{m_o}^{M_o} w_o f_o(w_o)\,dw_o} \qquad (5.16.)$$

If $g^{-1}(w,t,A)$ and w_o in the integrals are considered dummy variables and called x, 5.16. simplifies to:

$$\frac{W}{W_o} = \frac{\int_{L_1}^{L_2} g(x,t,A) f_o(x)\,dx}{\int_{m_o}^{M_o} x f_o(x)\,dx} \qquad (5.17.)$$

Because of this transformation and the properties of the operator P, the limits of integration, L_1 and L_2 have the following values:

$L_1 = m_o$ for t such that $Pg(m_o,t,A) > 0$ (5.18.)

$L_1 = g^{-1}(o,t,A)$ " " " " " $= 0$ (5.19.)

$L_2 = M_o$ for t such that $Pg(M_o,t,A) > 0$ (5.20.)

$L_2 = g^{-1}(o,t,A)$ " " " " " $= 0$ (5.21.)

The time at which $g(m_o,t,A) = 0$ is the critical time, that is, the time when the dissolving particles begin to disappear. When $g(M_o,t,A) = 0$ all particles are dissolved. Beyond that time, $L_1 = L_2$, thus making W/W_o equal to zero.

Equation 5.17. requires knowledge of the initial weight distributing $f_o(w_o)$, but more often the initial size distribution is of greater interest. Consider particles that are spherical and remain so during dissolution; in this case $w = \rho\pi a^3/6$, where ρ and a are the particle density and diameter, respectively.[2] Transformation in 5.16. from the initial weight distribution to the initial size distribution is then easily achieved; and by a similar procedure to that used to obtain 5.17. from 5.16., the following equation is derived:

$$\frac{W}{W_o} = \frac{\int_{R_1}^{R_2} g(\frac{\rho\pi}{6}x^3,t,A)\,l_o(x)\,dx}{\int_{d_o}^{D_o} \frac{\pi\rho}{6}x^3 l_o(x)\,dx} \qquad (5.22.)$$

where:

$R_1 = d_o$ for t such that $Pg(\frac{\rho\pi}{6}d_o^3,t,A) > 0$ (5.23.)

$R_1 = \left[\frac{6}{\rho\pi}g^{-1}(o,t,A)\right]^{1/3}$ " " " " " $= 0$ (5.24.)

$R_1 = D_o$ for t such that $Pg(\frac{\rho\pi}{6}D_o^3,t,A) > 0$ (5.25.)

$R_1 = \left[\frac{6}{\rho\pi}g^{-1}(o,t,A)\right]^{1/3}$ " " " " " $= 0$ (5.26.)

In Eq. 5.22, d_o and D_o are initial diameters of the smallest and largest particles, respectively,[1] l_o is the initial particle-size density function.

Eqs. 5.17. and 5.22. rigorously describe the entire dissolution profile of any powder if its initial particle weight density function, f_o, or initial particle-size density function, l_o, is known, together with the particle dissolution function, which can be any explicit expression. With appropriate choice of limits of integration, the equations are applicable to truncated as well as "ideal" distributions. The time-dependent integral

---

2. The derivation is valid for particles of other shapes as long as they remain unchanged during dissolution and an appropriate shape factor is used in place of $\pi/6$.

in the numerator of Eqs. 5.17. and 5.22. reduces to the constant integral in the denominator of zero time, thus making the ratio W/W_o equal to 1 as expected.

The general mathematical models expressed by 5.17. and 5.22. require the use of a computer for numerical evaluation because they are in integral form. However, if the initial particle weight or particle-size distribution can be approximated by some simple function, then the model can often be solved in terms of an expression suitable for evaluating without the use of a computer.

Special Case: Log-Normal Powders and the Cube Root Model.

Carstensen and Musa (92) have pointed out in their review of the literature that procedures such as milling, grinding and precipitation, which are based on random processes, produce particles having skewed distribution functions which often can be approximated by a log-normal distribution. It is therefore of interest to examine this special case.

Consider a powder consisting of spherical particles of initial diameters a_o, distributed such that $\ln a_o$ approximates a normal distribution with mean μ and standard deviation σ, truncated at $\ln d_o = \mu - i\sigma$ and $\ln D_o = \mu + j\sigma$, (Fig. 5.1.) where i and j are trimcation parameters. The density function of $\ln a_o$ is then given by:

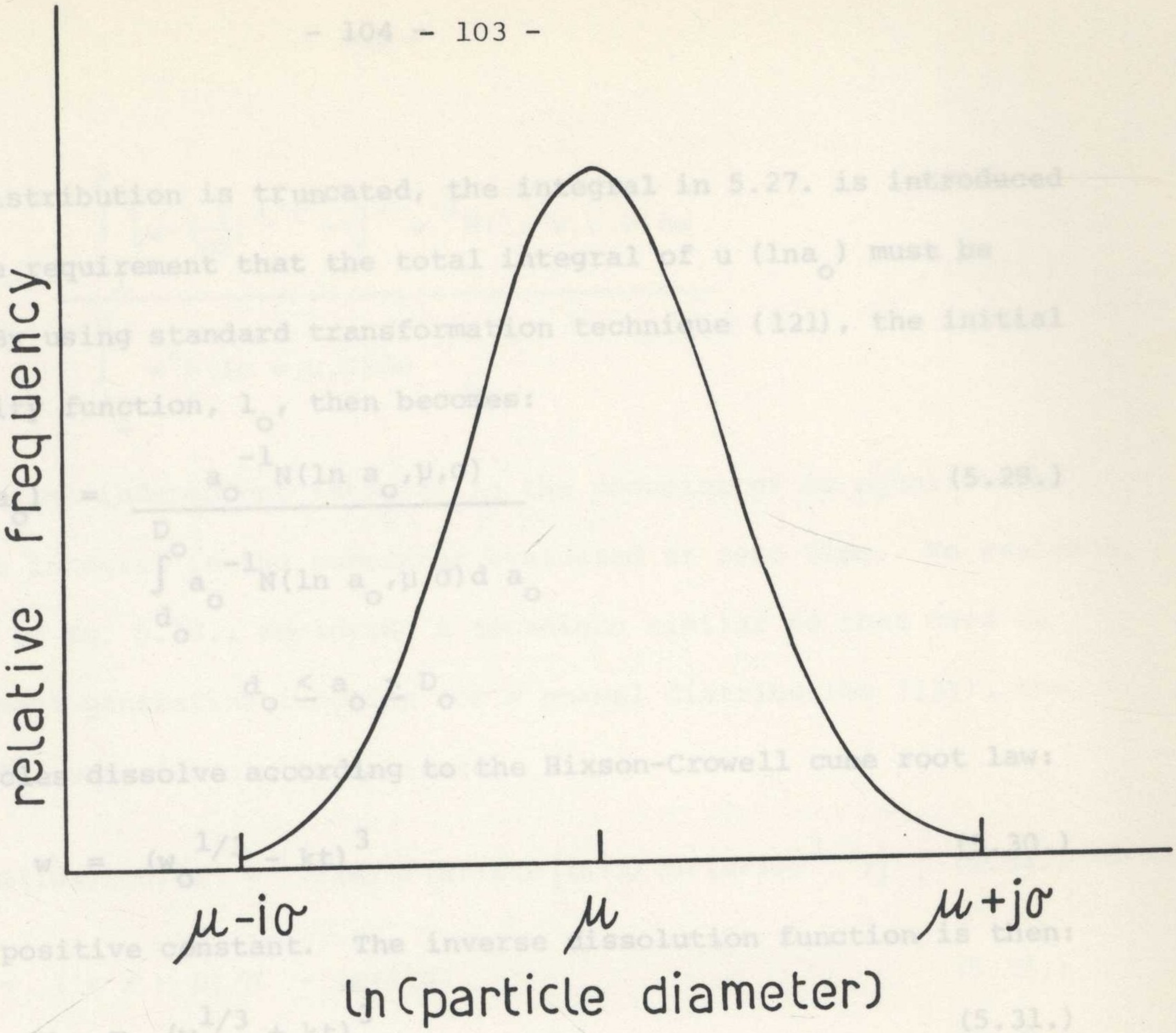
$$u(\ln a_o) = \frac{N(\ln a_o, \mu, \sigma)}{\int_{\ln a_o = \mu - i\sigma}^{\ln a_o = \mu + j\sigma} N(\ln a_o, \mu, \sigma)\, d \ln a_o} \qquad (5.27.)$$

$$\mu - i\sigma \leq \ln a_o \leq \mu + j\sigma$$

where the normal density function N is defined as:

$$N(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \text{EXP}\left[-(x-\mu)^2/2\sigma^2\right] \qquad (5.28.)$$

Figure 5.1. Illustration of the parameters in a truncated log-normal distribution.


relative frequency
μ-iσ
μ
μ+jσ
ln(particle diameter)

Because the distribution is truncated, the integral in 5.27. is introduced to satisfy the requirement that the total integral of u ($\ln a_o$) must be equal to 1. By using standard transformation technique (121), the initial particle density function, l_o, then becomes:

$$l_o(a_o) = \frac{a_o^{-1} N(\ln a_o, \mu, \sigma)}{D} \qquad (5.29.)$$

$$\int_{d_o}^{o} a_o^{-1} N(\ln a_o, \mu, \sigma) d a_o$$

$$d_o \leq a_o \leq D_o$$

Let the particles dissolve according to the Hixson-Crowell cube root law:

$$w = (w_o^{1/3} - kt)^3 \qquad (5.30.)$$

where k is a positive constant. The inverse dissolution function is then:

$$w_o = (w^{1/3} + kt)^3 \qquad (5.31.)$$

After the initial particle-size distribution (5.29.) and the particle dissolution equation (5.30. and its inverse 5.31.) are defined, the relationship giving the dissolution profile can then be derived by means of 5.22. in the following way:

After inserting Eqs. 5.29.-5.31. into Eq. 5.22., the integral in the numerator, I_n of 5.22. becomes:

$$I_n = \frac{\int_{R_1}^{R_2} \left[\left(\frac{\rho\pi}{6}\right)^{1/3} w - kt \right]^3 w^{-1} N(\ln w, \mu, \sigma) dw}{D} \qquad (5.32.)$$

$$\int_{d_o}^{o} w^{-1} N(\ln w, \mu, \sigma) dw$$

It was indicated previously that $W/W_o = I_n/(I_n)_{t=o}$. By using this fact, W/W_o can be written:

$$\frac{W}{W_o} = \frac{\int_{R_1}^{R_2} \left[w-\left(\frac{6}{\rho\pi}\right)^{1/3} kt\right]^3 w^{-1} N(\ln w,\mu,\sigma)\,dw}{\int_{d_o}^{D_o} w^2 N(\ln w,\mu,\sigma)\,dw} \qquad (5.33.)$$

In 5.33., the time-independent integral in the denominator is equal to the time-dependent integral in the numerator evaluated at zero time. To evaluate the numerator of Eq. 5.33., employing a technique similar to that used to derive the moment-generating function for a normal distribution (121), the following useful equation can be obtained:

$$\int_q^r x^s N(\ln x,\mu,\sigma)\,dx = (F(A)-F(B))\,EXP\left[(s+1)(\mu+(s+1)\sigma^{2/2})\right] \qquad (5.34.)$$

where $A = (\ln r - \mu)/\sigma - (s+1)\sigma$ (5.35.)

$B = (\ln q - \mu)/\sigma - (s+1)\sigma$ (5.36.)

and the function F( ) is the area under the standard normal curve function given by:

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-x^2/2}dx \qquad (5.37.)$$

Thus, to evaluate the numerator, I, in 5.33., the term $(w-Kt)^3 w^{-1}$ is expanded as $w^2-3(Kt)^2+3(Kt)^3 w^{-1}$ (letting $K = (6/\rho\pi)^{1/3}k$ for simplicity). The above formula is then applied term by term by putting s = 2,1,0, and -1; $r = R_2$ and $q = R_1$ and by treating t as a constant for the purpose of integration. The following equation is then derived using $W/W_o = I_n/(I_n)_{t=o}$ and noting that $\ln d_o = \mu-i\sigma$ and $\ln D_o = \mu+j\sigma$:

$$W/W_o = \left[\frac{F\left(\frac{T_2-\mu}{\sigma} - 3\sigma\right) - F\left(\frac{T_1-\mu}{\sigma} - 3\sigma\right)}{F(j-3\sigma) - F(-i-3\sigma)} \right]$$

$$-3(Kt) \left[\frac{F\left(\frac{T_2-\mu}{\sigma} - 2\sigma\right) - F\left(\frac{T_1-\mu}{\sigma} - 2\sigma\right)}{F(j-3\sigma) - F(-i-3\sigma)} \right] EXP\left[-\mu-5\sigma^2/2\right] \quad (5.38.)$$

$$+3(Kt)^2 \left[\frac{F\left(\frac{T_2-\mu}{\sigma} - \sigma\right) - F\left(\frac{T_1-\mu}{\sigma} - \sigma\right)}{F(j-3\sigma) - F(-i-3\sigma)} \right] EXP\left[-2\mu-4\sigma^2\right]$$

$$-(Kt)^3 \left[\frac{F\left(\frac{T_2-\mu}{\sigma}\right) - F\left(\frac{T_1-\mu}{\sigma}\right)}{F(j-3\sigma) - F(-i-3\sigma)} \right] EXP\left[-3\mu-9\sigma^2/2\right]$$

where:

$$T_1 = \mu - i\sigma \text{ for } \ln Kt \leq \mu - i\sigma \quad (5.39.)$$

$$T_1 = \ln Kt \text{ for } \ln Kt > \mu - i\sigma \quad (5.40.)$$

$$T_2 = \mu + j\sigma \text{ for } \ln Kt \leq \mu + j\sigma \quad (5.41.)$$

$$T_2 = \ln Kt \text{ for } \ln Kt > \mu + j\sigma \quad (5.42.)$$

and $$K = (6/\rho\pi)^{1/3} k \quad (5.43.)$$

The change in T, at timelength Kt = EXP $(\mu-i\sigma)$ (critical time) coresponds to the time when the smallest particles, initially having a diameter d_o = EXP $(\mu-i\sigma)$, begin to disappear. The change at Kt = EXP $(\mu+j\sigma) = D_o$ signifies the end of the dissolution process, so W/W_o becomes zero after that timelength.

Equation 5.38. describes the complete dissolution profile of log-normal powders and any sieve fraction of such powders. It assumes that the particles dissolve according to the cube root law (5.30.) which also can be written in the form $a = a_o - Kt$ for spherical particles, where a_o and a are the particle diameters at time zero and t, respectively. This relationship is similar to the equation $a = a_o - \tau$ used previously (92-94) in which τ denotes the timelength.

Brooke (94) derived an equation similar to 5.38 for log-normal powders. Directions were given for the changes required in the equation at the critical time. However, in his equation the first term is constant, equal

to 1. Therefore, his equation is incorrect if applied to dissolution *after* the critical time. The error so introduced becomes quite substantial for large timelengths and large values of σ.

Brooke calculated values of W/W_o for various values of $\tau/EXP(\mu)$ (the latter terminology corresponds to $EXP(-\mu)Kt$ used here). This procedure represents an ingenious method of "scaling time" (by the factor $EXP(-\mu)K$) so the dissolution profile becomes independent of the parameters μ and k (Eq. 5.30.) and only depends on σ, enabling the effect of σ alone to be assessed. In the present work, however, cube root type plots of $(W/W_o)^{1/3}$ *versus* $EXP(-\mu)Kt$ are used for better comparison with the fundamental particle dissolution equation (5.30) which obeys the cube root law.

Figure 5.2. shows such a plot for powders truncated at $\mu \pm 2\sigma$ (i=j=2) and having σ=0.3. The curvature of the unbroken line, calculated according to Eq. 5.38. is logically expected. The stippled line represents the dissolution profile calculated according to Brooke's equation (Eq. 4 of Ref. 94). The two profiles are, as expected, identical until critical time ($EXP(-\mu)Kt$=0.5488), but the later part clearly demonstrates the limitation of his equation.

Figure 5.3. demonstrates the effect of σ on the dissolution profile for a powder initially having an ideal distribution (i=j=10). Powders of uniform particle size, i.e., σ=0, give linear cube root plots as expected, while the deviation from linearity is significant for larger σ values. An increase in σ results in a decrease in the initial slopes of the curves, which is consistent with calculations made by Brooke (94). Among powders having the same logarithmic mean diameter, μ, those with broadest distribution will have the slowest initial release rate.

---

3. A Cyber 76 digital computer equipped with Calcomp platter was used for calculations and plots. Numerical evaluations were tested to six digits.

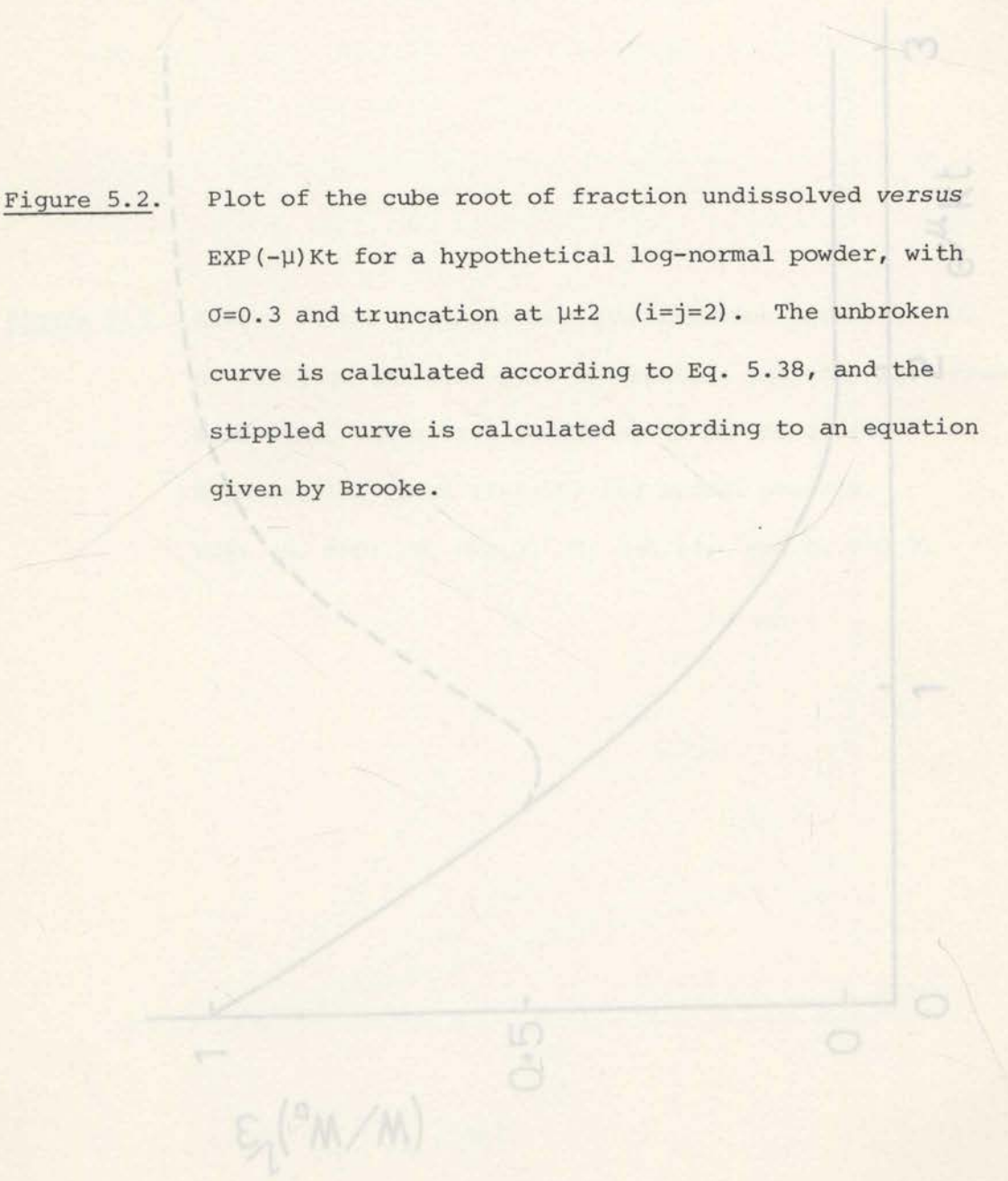
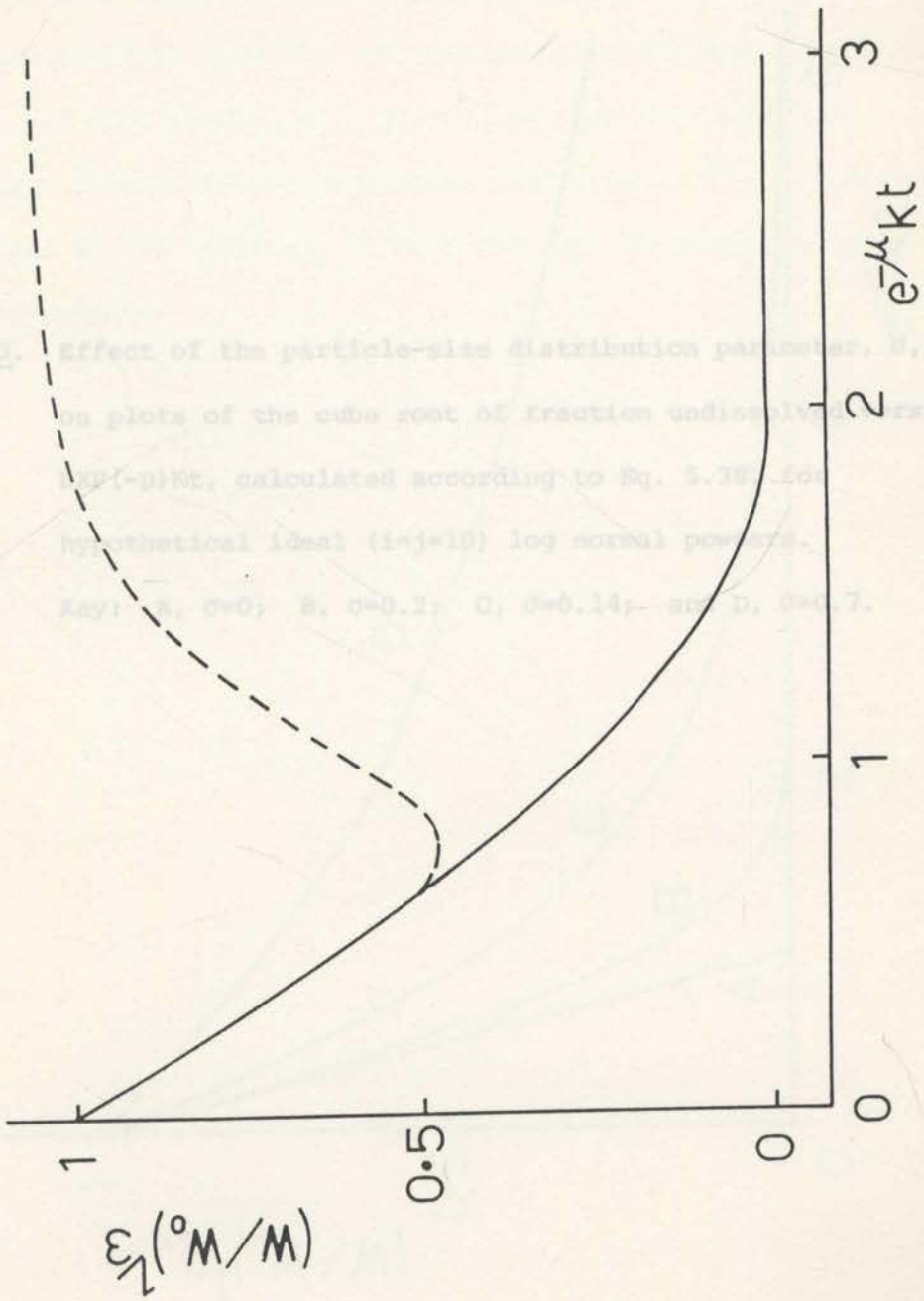


Figure 5.2. Plot of the cube root of fraction undissolved *versus* EXP(-μ)Kt for a hypothetical log-normal powder, with σ=0.3 and truncation at μ±2 (i=j=2). The unbroken curve is calculated according to Eq. 5.38, and the stippled curve is calculated according to an equation given by Brooke.


1
0.5
0
0
1
2
3
$e^{-\mu}Kt$
$\varepsilon_{\frac{1}{2}}(°M/M)$

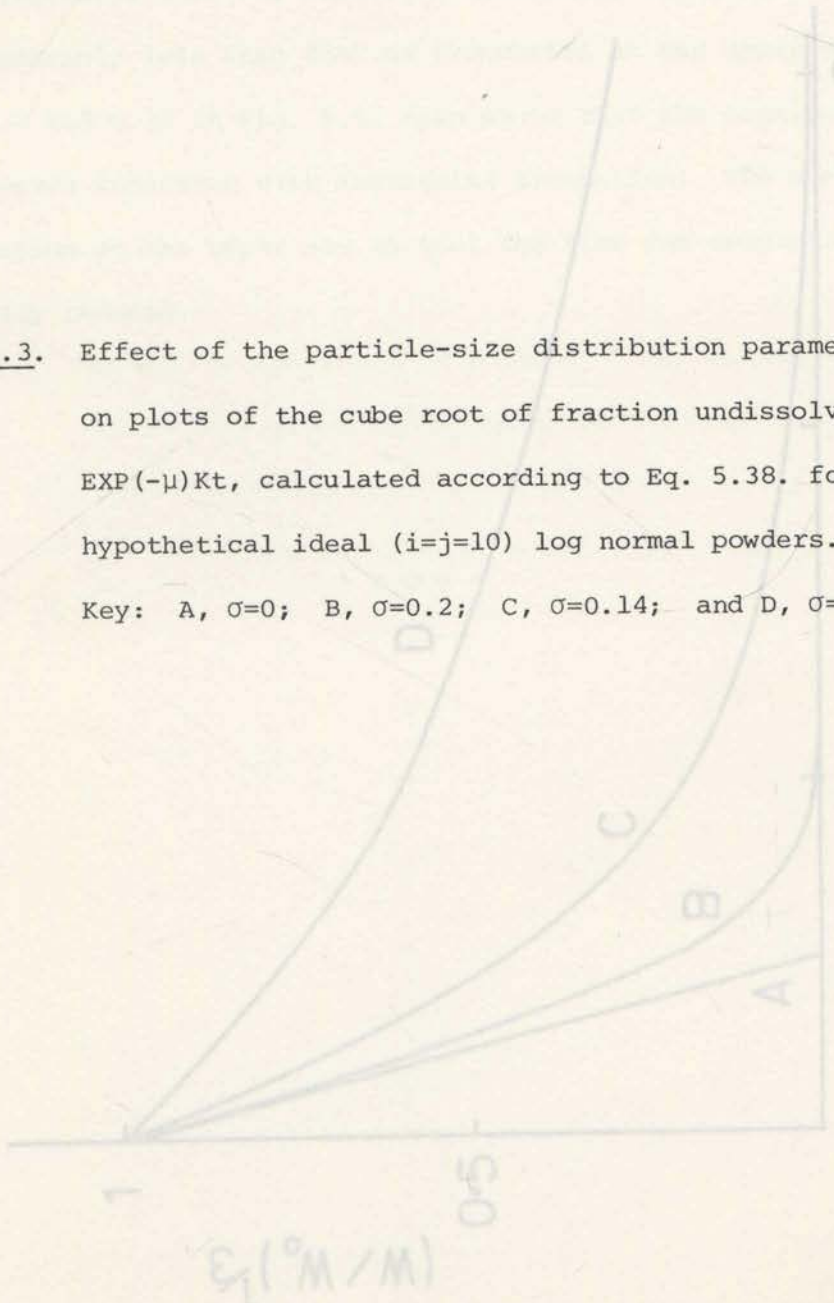


Figure 5.3. Effect of the particle-size distribution parameter, σ, on plots of the cube root of fraction undissolved *versus* EXP(-μ)Kt, calculated according to Eq. 5.38. for hypothetical ideal (i=j=10) log normal powders.
Key: A, σ=0; B, σ=0.2; C, σ=0.14; and D, σ=0.7.

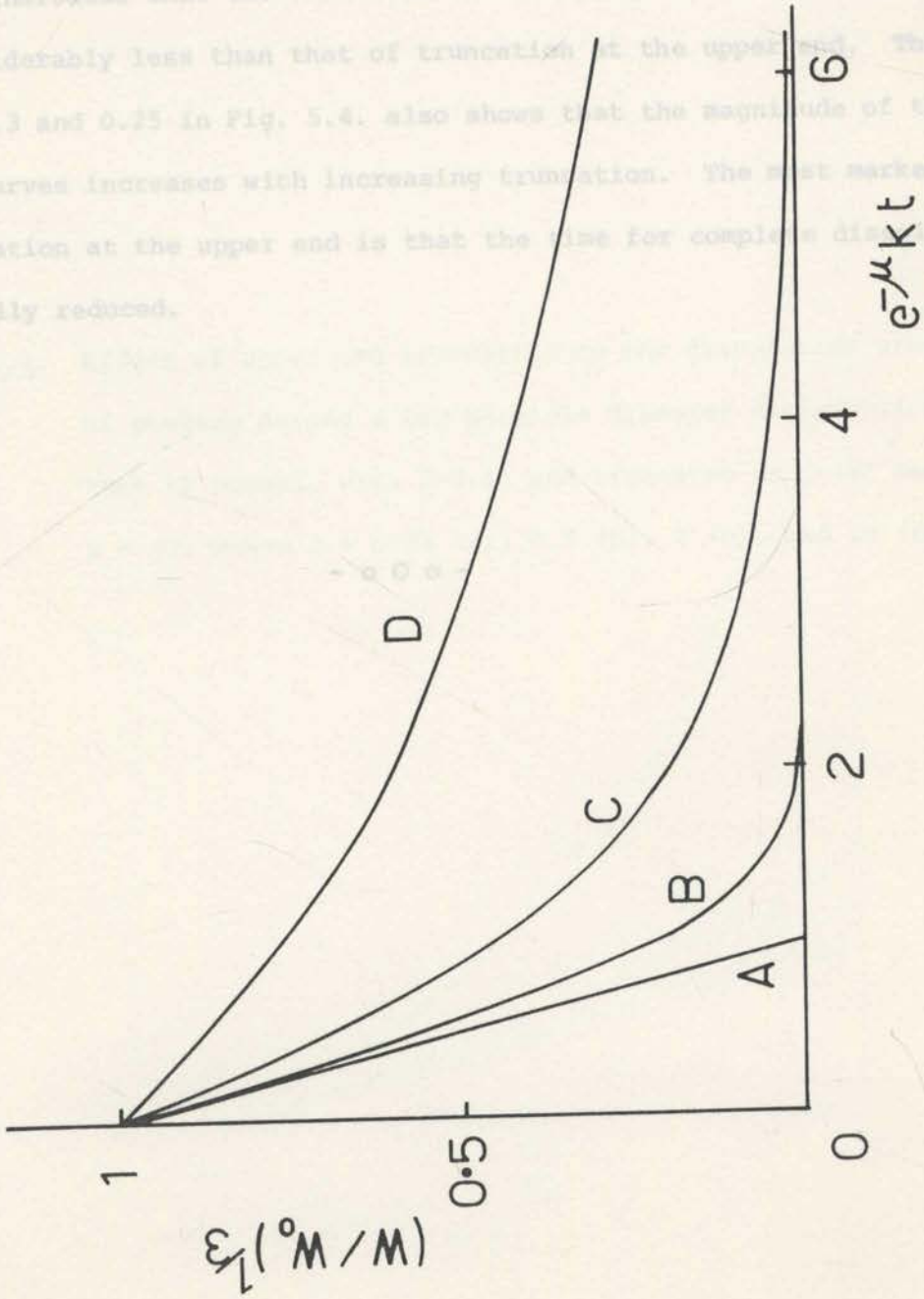

1
0·5
0
2
4
6
A
B
C
D
$(W/W_0)^{1/3}$
$e^{-\mu}kt$

Figure 5.4. and 5.5. show the effect of truncation on the dissolution profile when the initial ln (particle diameter) - distribution approximates a normal distribution (σ=0.5) with various degrees of upper (Fig. 5.4.) and lower (Fig. 5.5.) end truncation. Comparison of the two figures indicates that the effect of truncation at the low end is very small and considerably less than that of truncation at the upper end. The curves for j=10,1,0.3 and 0.25 in Fig. 5.4. also shows that the magnitude of the slopes of the curves increases with increasing truncation. The most marked effect of truncation at the upper end is that the time for complete dissolution is drastically reduced.

- o O o -

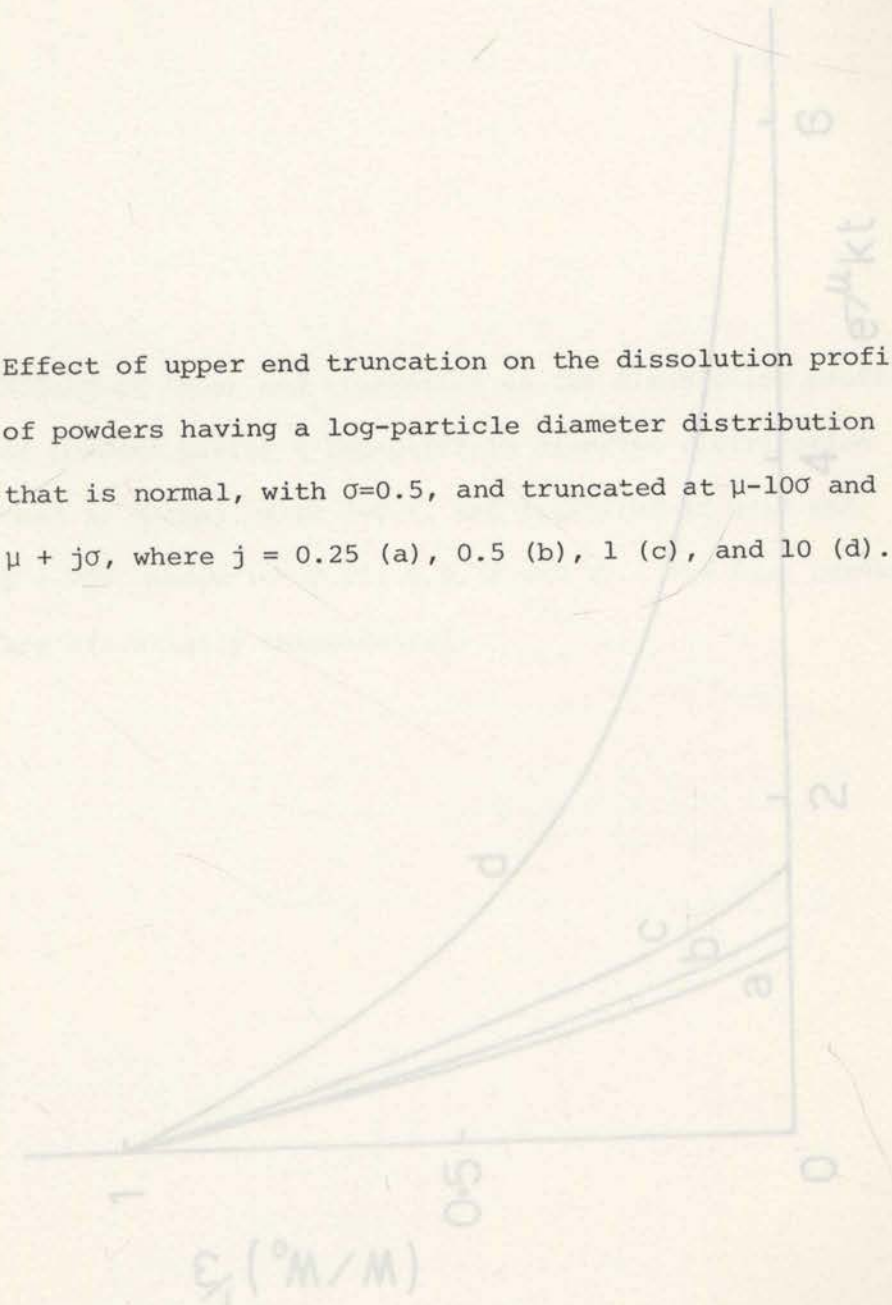
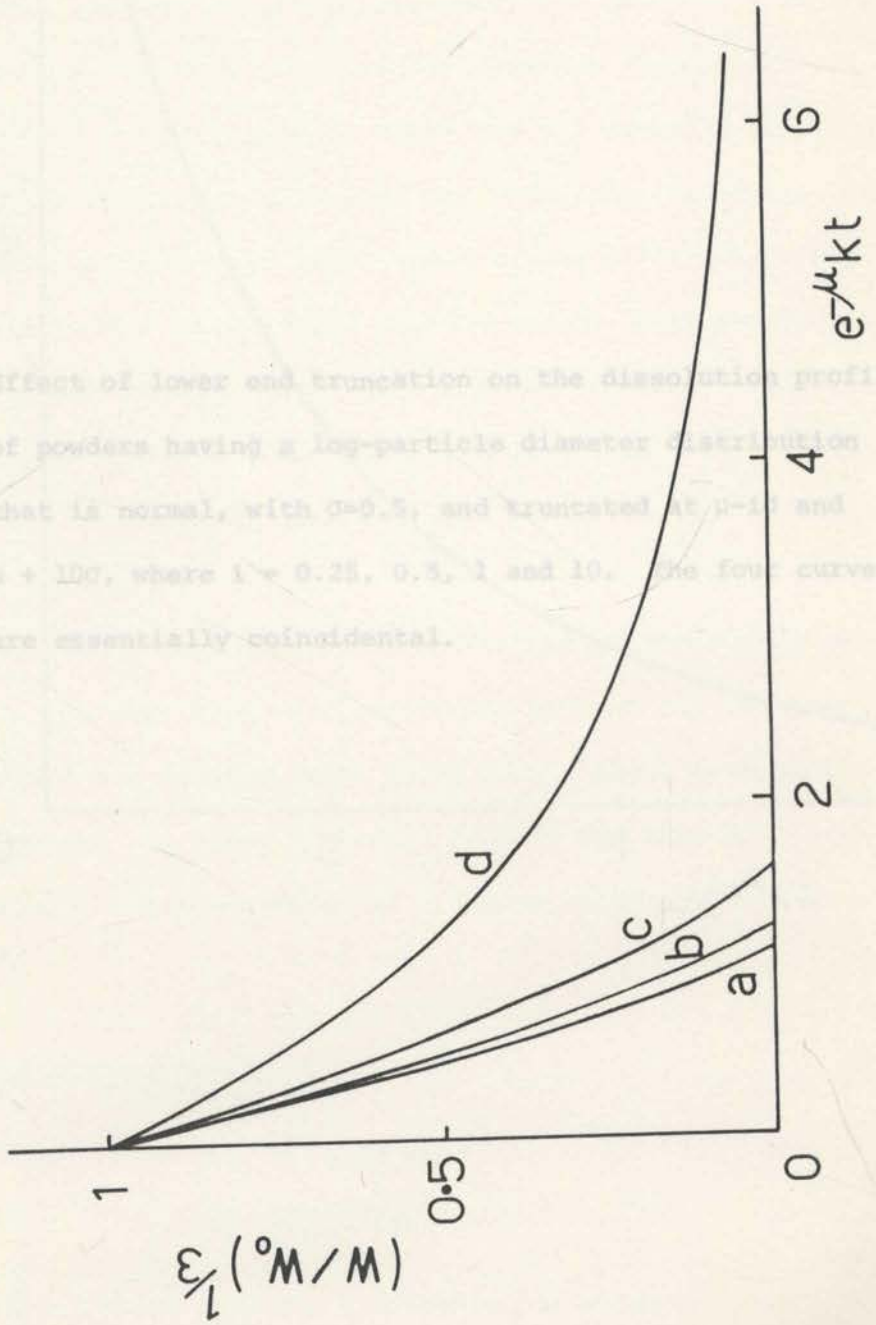


Figure 5.4. Effect of upper end truncation on the dissolution profile of powders having a log-particle diameter distribution that is normal, with $\sigma=0.5$, and truncated at $\mu-10\sigma$ and $\mu + j\sigma$, where $j = 0.25$ (a), 0.5 (b), 1 (c), and 10 (d).


1
0.5
0
2
4
6
$(W/W_o)^{1/3}$
$e^{-\mu kt}$
a
b
c
d

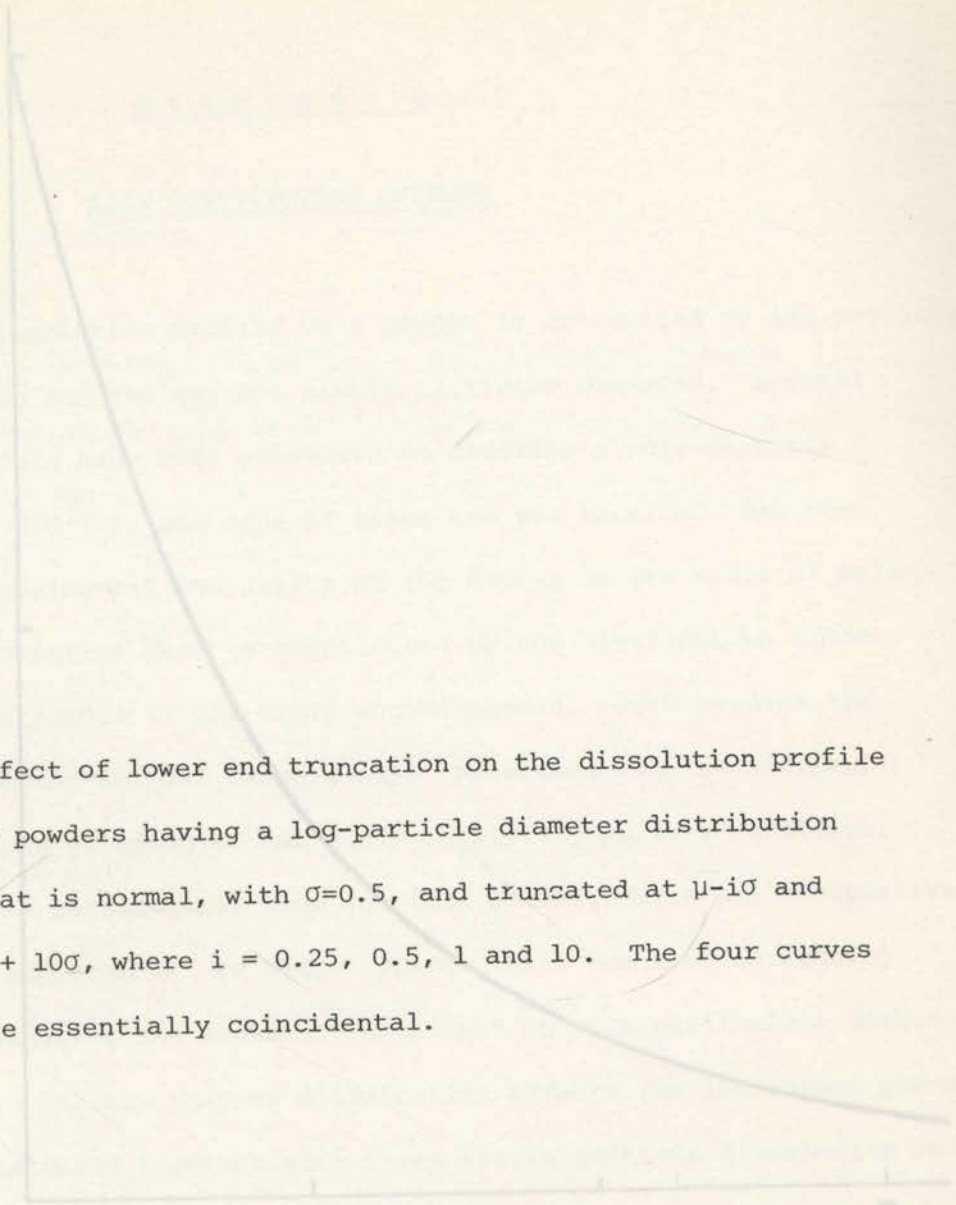
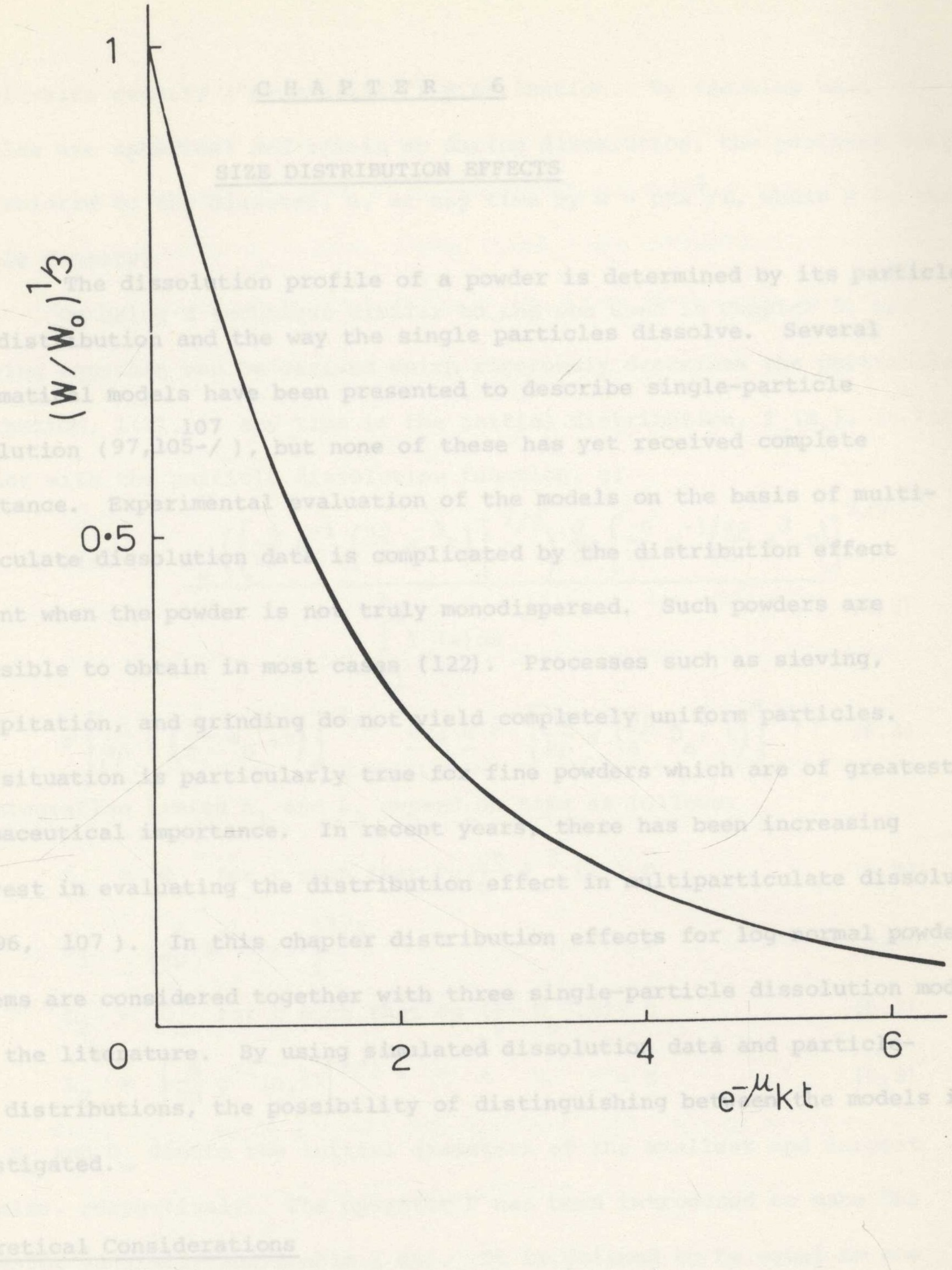


Figure 5.5. Effect of lower end truncation on the dissolution profile of powders having a log-particle diameter distribution that is normal, with $\sigma=0.5$, and truncated at $\mu-i\sigma$ and $\mu + 10\sigma$, where i = 0.25, 0.5, 1 and 10. The four curves are essentially coincidental.


1
0·5
$(W/W_o)^{1/3}$
0
2
4
6
$e^{-\mu_k t}$

CHAPTER 6

SIZE DISTRIBUTION EFFECTS

The dissolution profile of a powder is determined by its particle-size distribution and the way the single particles dissolve. Several mathematical models have been presented to describe single-particle dissolution (97,105-/ [107]), but none of these has yet received complete acceptance. Experimental evaluation of the models on the basis of multi-particulate dissolution data is complicated by the distribution effect present when the powder is not truly monodispersed. Such powders are impossible to obtain in most cases (122). Processes such as sieving, precipitation, and grinding do not yield completely uniform particles. This situation is particularly true for fine powders which are of greatest pharmaceutical importance. In recent years, there has been increasing interest in evaluating the distribution effect in multiparticulate dissolution (92-96, 107). In this chapter distribution effects for log-normal powder systems are considered together with three single-particle dissolution models from the literature. By using simulated dissolution data and particle-size distributions, the possibility of distinguishing between the models is investigated.

Theoretical Considerations

Let:
$$w = g(w_o,t) \tag{6.1}$$
and
$$w_o = g^{-1}(w,t) \tag{6.2}$$
describe the dissolution equation and inverse dissolution equation, respectively, for a single particle, where w and w_o are the particle weights at time t and o, respectively. Further, let l_o (a_o) denote the initial (t=o)

particle-size density ("probability") distribution. By assuming that particles are spherical and remain so during dissolution, the particle weight, w, is related to the diameter, a, at any time by $w = \rho\pi a^3/6$, where ρ is the particle density.

By using a technique similar to the one used in Chapter 5, the following equation can be derived which rigorously describes the particle-size distribution, l(a), at any time if the initial distribution, $l_o(a_o)$, is known together with the particle dissolution function, g:

$$l(a) = \frac{l_o\left(\left[\frac{6}{\pi\rho} g^{-1}\left(\frac{\pi\rho}{6} a,^3 t\right)\right]^{1/3}\right) \frac{d}{da}\left[\frac{6}{\pi\rho} g^{-1}\left(\frac{\pi\rho}{6} a^3, t\right)\right]^{1/3}}{\int_{L_1}^{L_2} l_o(a)\,da} \qquad (6.3)$$

For $$P\left[\frac{6}{\pi\rho} g\left(\frac{\pi\rho}{6} d_o^{\,3}, t\right)\right]^{1/3} \leq a \leq P\left[\frac{6}{\pi\rho} g\left(\frac{\pi\rho}{6} D_o^{\,3}, t\right)\right]^{1/3} \qquad (6.4)$$

The integration limits L_1 and L_2 depend on time as follows:

$$L_1 = d_o \quad \text{for t such that } Pg\left(\frac{\rho\pi}{6} d_o^{\,3}, t\right) > 0 \qquad (6.5)$$

$$L_1 = \left[\frac{6}{\pi\rho} g^{-1}(o,t)\right]^{1/3} \quad " \quad " \quad " \quad " \quad " = 0 \qquad (6.6)$$

$$L_2 = D_o \quad \text{for t such that } Pg\left(\frac{\rho\pi}{6} D_o^{\,3}, t\right) > 0 \qquad (6.7)$$

$$L_2 = \left[\frac{6}{\pi\rho} g^{-1}(o,t)\right]^{1/3} \quad " \quad " \quad " \quad " \quad " = 0 \qquad (6.8)$$

where d_o and D_o denote the initial diameters of the smallest and largest particles, respectively. The operator P has been introduced to make the expression generally applicable (95). It is defined to be equal to one in the time period before the operand becomes zero and is equal to zero beyond that time. The lower integration limit L_1 changes value at $g\left[(\rho\pi/6) d_o^{\,3}, t\right] = 0$, that is the critical time when the particles start to disappear. The time at which $g\left[(\rho\pi/6) D_o^{\,3}, t\right] = 0$ corresponds to the disappearance of the last particle and marks the completion of the dissolution process.

Many powders have size distributions that are approximately log-normal (122). Consider such a powder distributed such that ln a_o approximates a normal distribution with mean μ and standard deviation σ truncated at ln $d_o = \mu - i\sigma$ and at ln $D_o = \mu + j\sigma$, where i and j are truncation parameters. The initial particle-size distribution, $l_o(a_o)$, is then given by (96):

$$l_o(a_o) = \frac{a_o^{-1} N(\ln a_o, \mu, \sigma)}{\int_{d_o}^{D_o} a_o^{-1} N(\ln a_o, \mu, \sigma) d a_o} \qquad (6.9)$$

$$d_o \leq a_o \leq D_o$$

where N( ) is the normal distribution with ln a_o as the variable.

The change in the particle-size distribution during dissolution depends on the way the individual particles dissolve. Three widely known models for single-particle dissolution are considered. When written in the same form as 6.1, the cube root law (97) can be expressed as:

$$w = (w_o^{1/3} - k_3 t)^3 \qquad (6.10)$$

In a similar way, the equation presented by Niebergal *et al.* (106) can be written simply:

$$w = (w_o^{\frac{1}{2}} - k_w t)^2 \qquad (6.11)$$

and the model proposed by Higuchi and Hiestand (107) can be written:

$$w = (w_o^{2/3} - k_1 t)^{3/2} \qquad (6.12)$$

For simplicity and because their evaluation is not important to the theoretical discussion, the constants k_1, k_2 and k_3 are used in place of the original time coefficients which included parameters such as the shape factor, particle density, and diffusion coefficient. In the following section, Eqs. 6.12, 6.11 and 6.10 will be referred to as models 1, 2 and 3, respectively.

By having defined the initial size distribution $l_o(a_o)$ (Eq. 6.9),

and the dissolution function (Eqs. 6.10-12) the size distribution at time t, l(a), can be expressed applying 6.3:

$$l(a) = \frac{a^{(\frac{3}{m}-1)}\left(\frac{3}{a^m}+Kt\right)^{-1} N\left[\ln\left(a^{\frac{3}{m}}+Kt\right)^{\frac{m}{3}},\mu,\sigma\right]}{F\left(\frac{T_2-\mu}{\sigma}\right) - F\left(\frac{T_1-\mu}{\sigma}\right)} \qquad (6.13)$$

For $P(EXP\left[3(\mu-i\sigma)/m\right]-Kt)^{\frac{m}{3}} \leq a \leq P(EXP\left[3(\mu+j\sigma)/m\right]-Kt)^{\frac{m}{3}}$

and l(a) = 0 elsewhere. Also:

$$T_1 = \mu - i\sigma \quad \text{for } \frac{m}{3}\ln(Kt) < \mu - i \qquad (6.14)$$

$$T_1 = \frac{m}{3}\ln(Kt) \quad " \ " \ " \ " \geq \mu - i \qquad (6.15)$$

$$T_2 = \mu + j\sigma \quad \text{for } \frac{m}{3}\ln(Kt) < \mu + j \qquad (6.16)$$

$$T_2 = \frac{m}{3}\ln(Kt) \quad " \ " \ " \ " \geq \mu + j \qquad (6.17)$$

and $$K = (6/\rho\pi)^{\frac{1}{m}} k \qquad (6.18)$$

Equation 6.13 describes the size distribution for all three models. For Model 1, m = 3/2; for Model 2, m = 2; and for Model 3, m = 3. The constants k and K for each model should be k_1, k_2, k_3 and K_1, K_2, K_3, respectively. The function F( ) is the commonly tabulated area under the standard normal curve function defined earlier (5.37).

The main particle size (diameter), $\bar{a}$ can be obtained by applying the usual integration approach used in mathematical expectation:

$$\bar{a} = \frac{\int_R a^{\frac{3}{m}}\left(a^{\frac{3}{m}}+Kt\right)^{-1} N\left[\ln\left(a^{\frac{3}{m}}+Kt\right)^{\frac{m}{3}},\mu,\sigma\right] da}{F\left(\frac{T_2-\mu}{\sigma}\right) - F\left(\frac{T_1-\mu}{\sigma}\right)} \qquad (6.19)$$

The integration interval R in 6.19 is the same as the interval for a defined in 6.13. Equation 6.19 considers Model 1 and 2 (m = 3/2 and m = 2, respectively). The mean particle size for the third model (m = 3) simplies further to:

$$\bar{a} = \frac{F\left(\frac{T_2-\mu}{\sigma} - \sigma\right) - F\left(\frac{T_1-\mu}{\sigma} - \sigma\right)}{F\left(\frac{T_2-\mu}{\sigma}\right) - F\left(\frac{T_1-\mu}{\sigma}\right)} \text{EXP}(\mu+\sigma^2/2) - K_3 t \qquad (6.20)$$

Where T_1 and T_2 are still defined as in Eqs. 6.14-17. The exact dissolution profile of a log-normal powder with single particles dissolving according to each of the three models can be derived using 5.22 presented earlier(96):

$$\frac{W}{W_o} = \sum_{n=o}^{m} \binom{m}{n} (-Kt)^{(m-n)} \frac{F(A)-F(B)}{F(j-3\sigma)-F(-i-3\sigma)} \text{EXP}(C) \qquad (6.21)$$

For which

$$A = (T_2-\mu)/\sigma - 3n\sigma/m \qquad (6.22)$$

$$B = (T_1-\mu)/\sigma - 3n\sigma/m \qquad (6.23)$$

and

$$C = \frac{3}{m}(n-m)(\mu+\frac{3}{m}(n+m)\sigma^2/2) \qquad (6.24)$$

Where W and W_o are the amounts of undissolved powder at time t and 0, respectively. Equation 6.21 with m = 3, although presented in a more compact form, is identical to 5.38 derived earlier. Equation 6.21 is not defined for Model 1 (m = $^3/_2$) which must be considered separately:

$$\frac{W}{W_o} = \frac{\int_{R_1}^{R_2} (w^2-K_1t)^{3/2} w^{-1} N(\ln w,\mu,\sigma)dw}{F(j-3\sigma) - F(-i-3\sigma)} \text{EXP}(-3\mu-9\sigma^2/2) \qquad (6.25)$$

where

$$R_1 = \text{EXP}(\mu-i\sigma) \quad \text{for } (K_1t)^{\frac{1}{2}} < \text{EXP}(\mu-i\sigma) \qquad (6.26)$$

$$R_1 = (K_1t)^{\frac{1}{2}} \quad " \quad " \geq \text{EXP}(\mu-i\sigma) \qquad (6.27)$$

$$R_2 = \text{EXP}(\mu+j) \quad \text{for } (K_1t)^{\frac{1}{2}} < \text{EXP}(\mu+j\sigma) \qquad (6.28)$$

$$R_2 = (K_1t)^{\frac{1}{2}} \quad " \quad " \geq \text{EXP}(\mu+j\sigma) \qquad (6.29)$$

The derivation of these equations are based on two assumptions:

(a) that the particles in the multiparticulate system dissolve independently of each other, which will be approximated well under sink conditions; and

(b) that they dissolve according to the same single-particle dissolution

model having fixed parameters (for these cases, k_1, k_2 and k_3 are the same for all particles and do not vary during dissolution). If these conditions exist, then it is possible to propose some general rules concerning the dissolution process. These rules are explained in relation to what will be termed "the intrinsic dissolution profile", which can be defined in the following way: Dissolution curves have the same intrinsic dissolution profile if, by a suitable scaling of time, they can be brought into each other in a W/W_o *versus* time plot (Fig. 6.1).

It should be clear from observation of Eqs. 5.17 and 5.22 that the coefficient of time in an expression correctly defining the multiparticulate dissolution profile originates directly from the coefficient of time in the single-particle dissolution equation. Thus a different value of the rate parameter, that is, a different coefficient of time in the single-particle dissolution equation, has the same effect as a different scaling of time. Therefore, the intrinsic dissolution profile will still be the same. The following rule can thus be stated:

1. The intrinsic dissolution profile is independent of the value of the rate parameter, that is, the coefficient of time in the single-particle dissolution equation.

According to this rule, the rate parameter k_1, k_2 and k_3 (6.10-12) have no influence on the intrinsic dissolution profile. Furthermore, there will always be a proportional relationship between the coefficient of time in the multiparticulate dissolution equation and the rate parameter. The following rule can therefore be stated:

2. In two systems having identical particle-size distributions, the time-scaling factor that brings one dissolution curve into another is equal to the factor with which the rate parameters are proportionally related in the two systems. (Fig. 6.1)

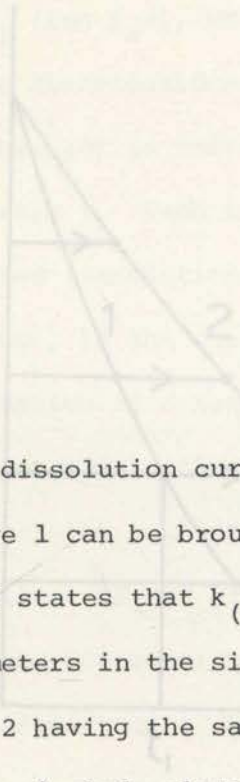
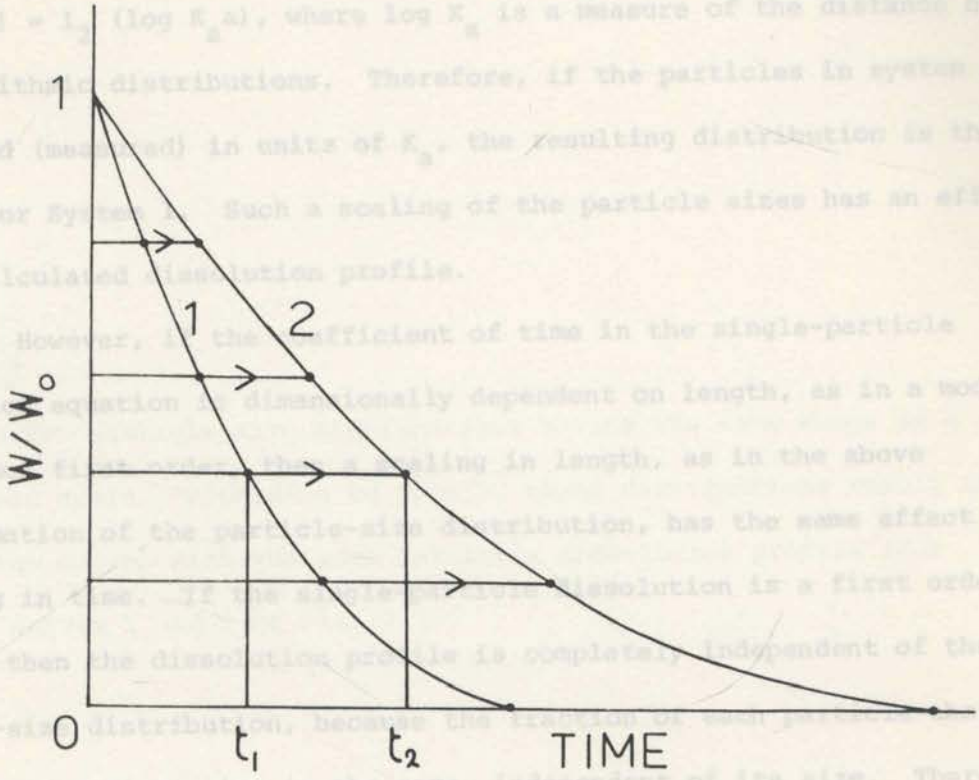


Figure 6.1

Two dissolution curves having the same intrinsic dissolution profile. Curve 1 can be brought into curve 2 by a time-scaling factor, t_2/t_1. Rule 2 states that $k_{(1)} = (t_2/t_1)k_{(2)}$, where $k_{(1)}$ and $k_{(2)}$ are the rate parameters in the single-particle dissolution equation for Systems 1 and 2 having the same particle-size distribution. These statements also include plots where W/W_o is raised to any other exponent.


1
W/W°
1
2
0
t₁
t₂
TIME

Consider two particle-size distributions that are distributed on a logarithmic scale $l_1(\log a)$ and $l_2(\log a)$ having the same shape. Then for any diameter, a (Fig. 6.2) $l_1(a) = l_2(\log a + \log K_a)$; i.e., $l_1(\log a) = l_2(\log K_a a)$, where $\log K_a$ is a measure of the distance between the logarithmic distributions. Therefore, if the particles in system 2 are scaled (measured) in units of K_a, the resulting distribution is the same as that for System 1. Such a scaling of the particle sizes has an effect on the calculated dissolution profile.

However, if the coefficient of time in the single-particle dissolution equation is dimensionally dependent on length, as in a model that is not first order, then a scaling in length, as in the above transformation of the particle-size distribution, has the same effect as a scaling in time. If the single-particle dissolution is a first order process, then the dissolution profile is completely independent of the particle-size distribution, because the fraction of each particle that dissolves in a given time is the same, independent of its size. Therefore, it can be concluded that system 1 and 2 have the same intrinsic dissolution profile, and the following rule can be given:

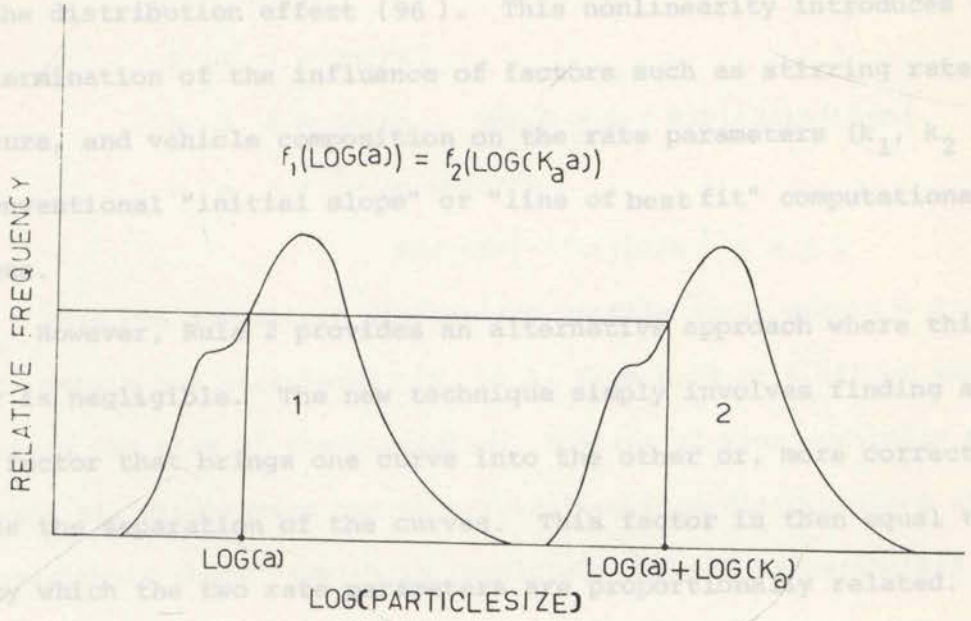
3. *Two powders dissolving according to the same single-particle dissolution model have the same intrinsic dissolution profile if their particle-size distributions are of the same shape on a logarithmic scale.* (Fig. 6.2)

It follows from this rule that it is not the "position" of the distribution, that is, not the actual size of the particles, but the shape of the distribution that affects the intrinsic profile. Thus it can be stated that:

4. *The intrinsic dissolution profile does not depend on the actual size of the particles but on the shape of their distribution.*

Figure 6.2

Two particle-size distributions having the same shape on a logarithmic scale. According to Rule 3, these distributions result in dissolution curves with the same intrinsic dissolution profile (for example, curves 1 and 2 of Fig. 7.1).


$f_1(LOG(a)) = f_2(LOG(K_a a))$
RELATIVE FREQUENCY
1
2
LOG(a)
LOG(a)+LOG(K_a)
LOG(PARTICLESIZE)

According to Rule 2, the concept of time scaling in dissolution studies should have some practical application. It has already been stated that it is very difficult to prepare completely monodispersed powders; therefore, some nonlinearity is always present in the dissolution profile due to the distribution effect (96). This nonlinearity introduces errors into determination of the influence of factors such as stirring rate, temperature, and vehicle composition on the rate parameters (k_1, k_2 or k_3) using conventional "initial slope" or "line of best fit" computational techniques.

However, Rule 2 provides an alternative approach where this type of error is negligible. The new technique simply involves finding a time-scaling factor that brings one curve into the other or, more correctly, minimizes the separation of the curves. This factor is then equal to the factor by which the two rate parameters are proportionally related. Perhaps the best criteria is to minimize the squared horizontal differences between the curves.

According to Rules 1 and 3, it should be possible to normalize the calculated dissolution profiles for log-normal powders by appropriate scaling of time to a form that does not depend on either the rate parameter (k_1, k_2 or k_3) or the actual sizes of the particles. One approach is to scale time in the W/W_o *versus* time plot as the time fraction ψ, defined as the fraction of the time necessary for complete dissolution. The expression defining the resulting normalized dissolution profile can be obtained in the following way, using Models 2 and 3 as examples. The time for complete dissolution, t_o, is given by

$(m/3)\ln(Kt_o) = \mu+j\sigma$; then since $\psi = t/t_o$ it follows that:

$$\psi = KtEXP\left[-3(\mu+j\sigma)/m\right] \qquad (6.30)$$

Accordingly, the Kt terms in 6.21 can be substituted by $\psi^{3(\mu+j\sigma)/m}$, which causes the μ term to cancel out. After rearrangement, 6.21 can be written:

$$\frac{W}{W_o} = \sum_{n=o}^{m} \binom{m}{n} (-\psi)^{(m-n)} \quad \frac{F(A) - F(B)}{F(j-3\sigma)-F(-i-3\sigma)} \quad EXP\ (C) \qquad (6.31)$$

where:

$$A = j - 3n\sigma/m$$

$$B = i - 3n\sigma/m \qquad \text{for } 0 \leq \psi \leq EXP\left[-3(i+j)\sigma/m\right]$$

$$B = j + \frac{m}{3} \ln \psi - 3n\sigma/m$$

$$\text{for } EXP\left[-3(i+j)\sigma/m\right] \leq \psi \leq 1$$

Thus the normalized dissolution profile (6.31 with m = 2 or m = 3) does not contain any rate terms (k_3 or k_2 from 6.10 or 6.11) or any term (μ) representing the size of the particles. Scaling of time according to 6.30 has brought all dissolution curves originating from distributions with the same "logarithmic shape" (which is completely defined by parameters σ, i and j) into one single curve (6.31) which does not depend on the size of the particles or the rate parameter from the single-particle equation. This confirms Rule 1 and 3.

The transformation has essentially normalized all possible systems having the same intrinsic dissolution profile into one single curve. This curve is unique in that it makes it possible to evaluate the isolated distribution effect. This evaluation is best done by plotting in a way that linearizes the underlying single-particle dissolution equation (6.10-12), by using $(W/W_o)^{1/m}$ instead of W/W_o in the plot. Such a plot will be linear with slope = -1 for a true monodisperse system. Any deviation from this linearity and slope will be due solely to the distribution effect.

Figure 6.3 shows such normalized dissolution profiles, calculated according to 6.21 and 6.31, for powders initially log-normal, having distribution (shape) parameters σ = 0.2 and i = j = 2, for particles dissolving

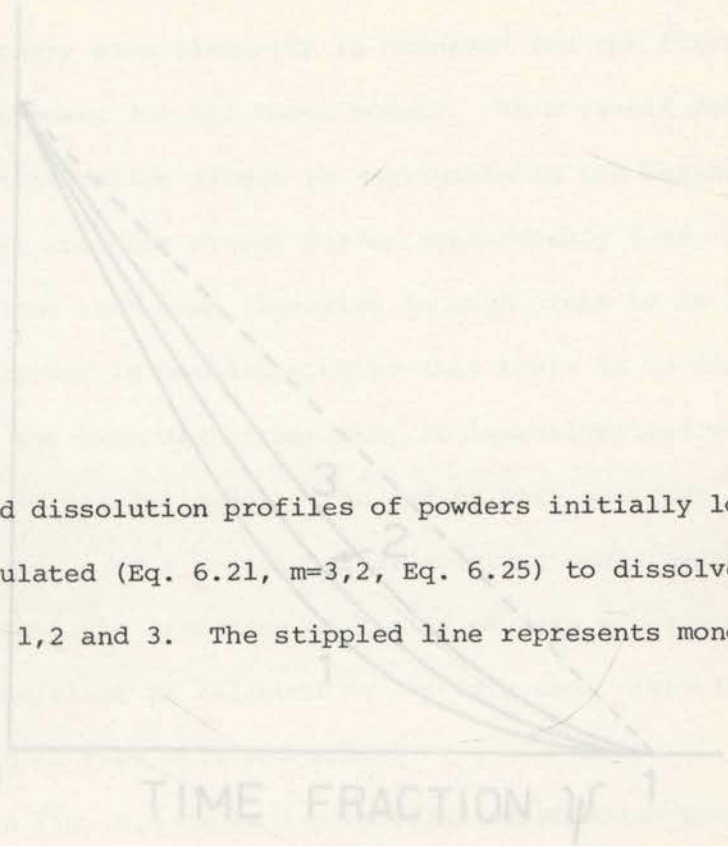
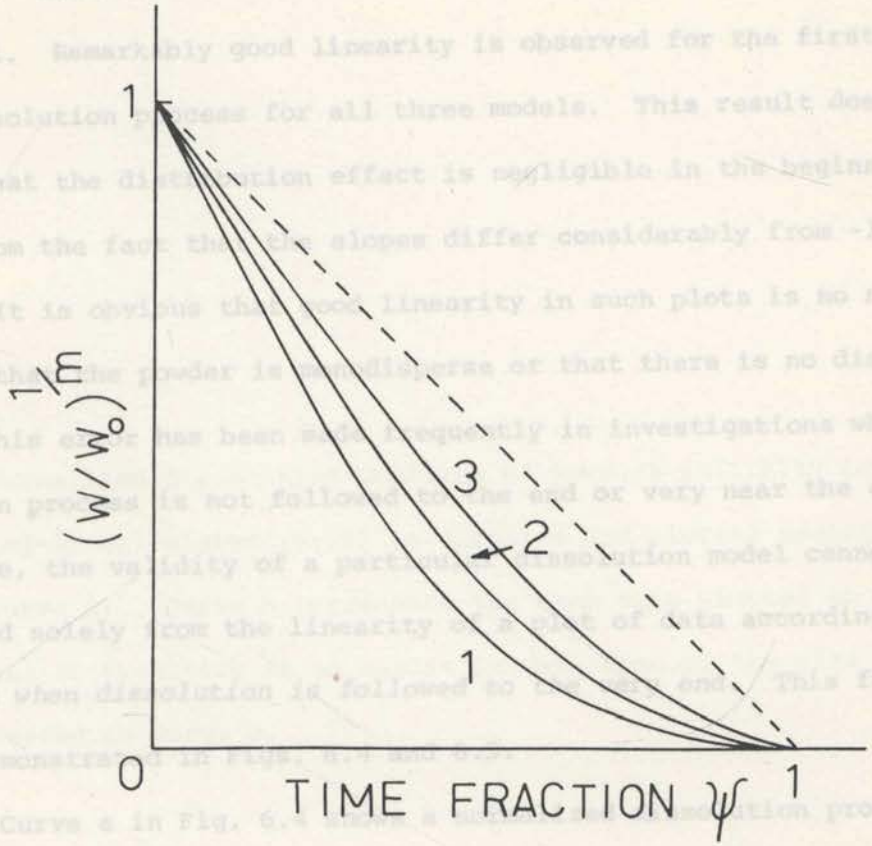


Figure 6.3

Normalized dissolution profiles of powders initially log-normal (σ=0.2, i=j=2) calculated (Eq. 6.21, m=3,2, Eq. 6.25) to dissolve and plotted according to models 1,2 and 3. The stippled line represents monodisperse powders (σ=o).


1
$(W/W_o)^{1/E}$
3
2
1
0
TIME FRACTION ψ 1

according to each of the three models (6.10-12). The fraction of undissolved powder, W/W_o, is raised to the powers of 1/3, 1/2 and 2/3 for the reason just given. The distribution effect is smallest for Model 3 and greatest for Model 1. Remarkably good linearity is observed for the first part of the dissolution process for all three models. This result does not mean, however, that the distribution effect is negligible in the beginning, as is seen from the fact that the slopes differ considerably from -1.

It is obvious that good linearity in such plots is no necessary criterion that the powder is monodisperse or that there is no distribution effect. This error has been made frequently in investigations where the dissolution process is not followed to the end or very near the end. Furthermore, the validity of a particular dissolution model cannot always be assessed solely from the linearity of a plot of data according to that model *even when dissolution is followed to the very end*. This fact is clearly demonstrated in Figs. 6.4 and 6.5.

Curve a in Fig. 6.4 shows a normalized dissolution profile of log-normal powders (σ=0.14, i=j=2) calculated (6.25) to dissolve and plotted (W/W_o to power of 2/3) according to Model 1. The size distribution effect is clearly reflect in the nonlinearity of the curve. By plotting the same data according to an incorrect model, Model 3 (W/W_o to power of 1/3), the size distribution effect is almost entirely cancelled and surprisingly good linearity is obtained *that extends to the very end of the dissolution process* (curve b).

Figure 6.5 shows the same phenomenon for powders (σ = 0.12, i = j = 2) where the particles are calculated to dissolve according to Model 2 (6.31, m = 2).

A judgment based solely on the linearity of such plots will often lead to false conclusions about the validity of the model, even where

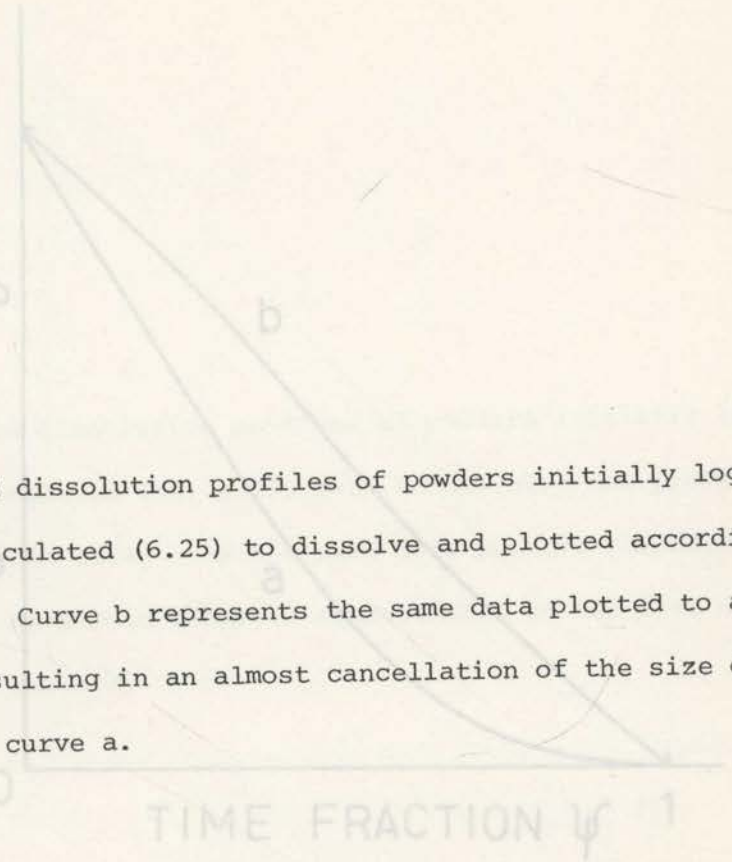
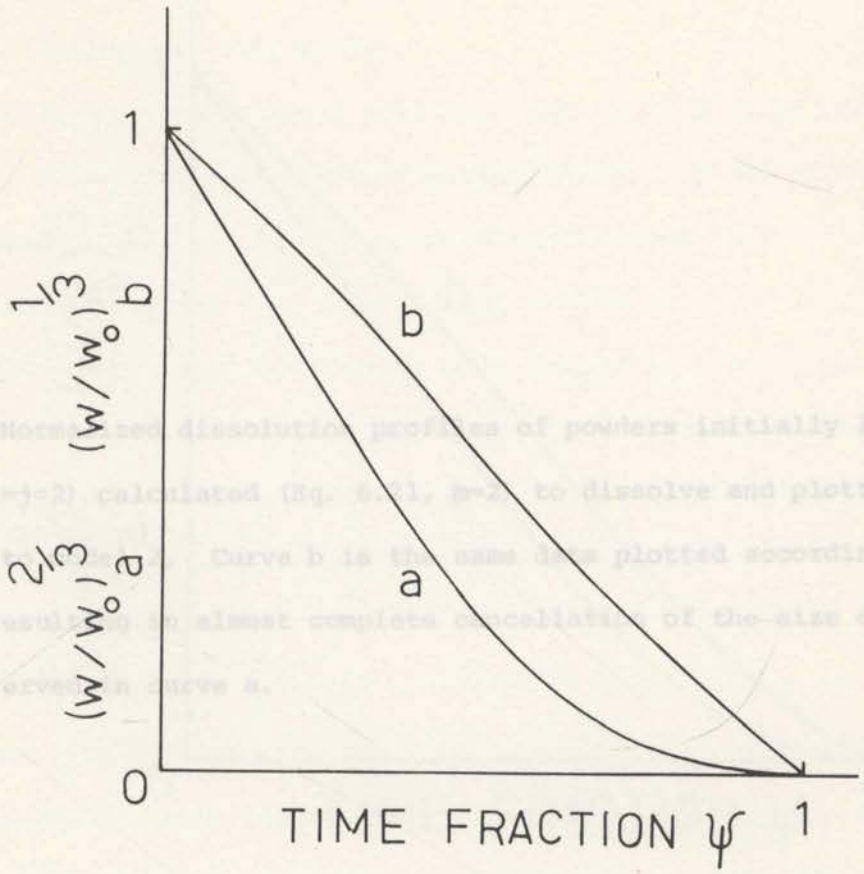


Figure 6.4

Normalized dissolution profiles of powders initially log-normal (σ=0.14, i=j=2) calculated (6.25) to dissolve and plotted according to model 1 (curve a). Curve b represents the same data plotted to an incorrect model (model 3) resulting in an almost cancellation of the size distribution effect observed in curve a.


1
$(W/W_o)^{1/3}$
b
$(W/W_o)^{2/3}$
a
0
b
a
TIME FRACTION ψ
1

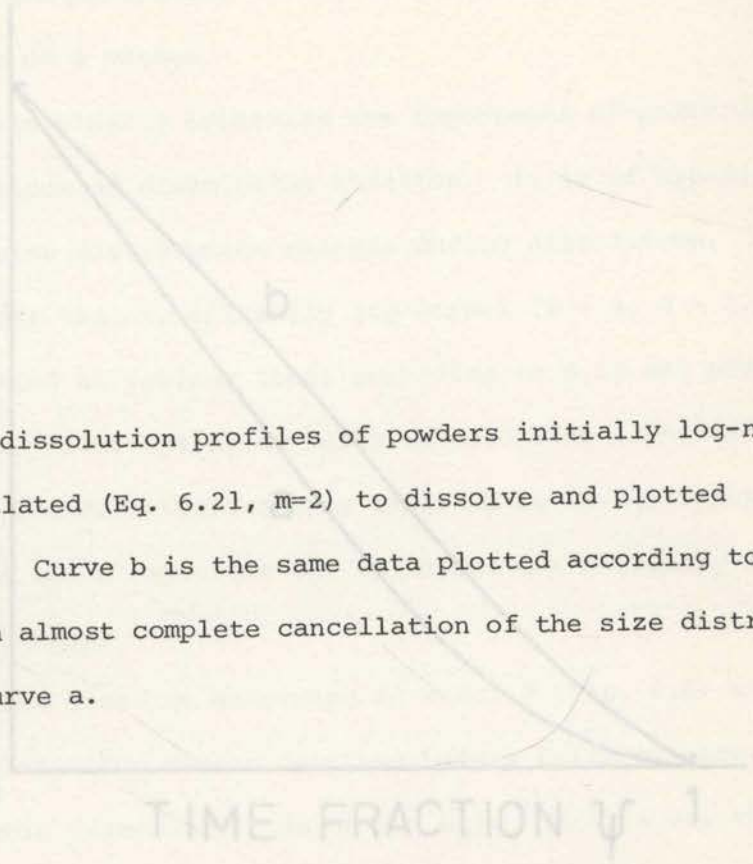
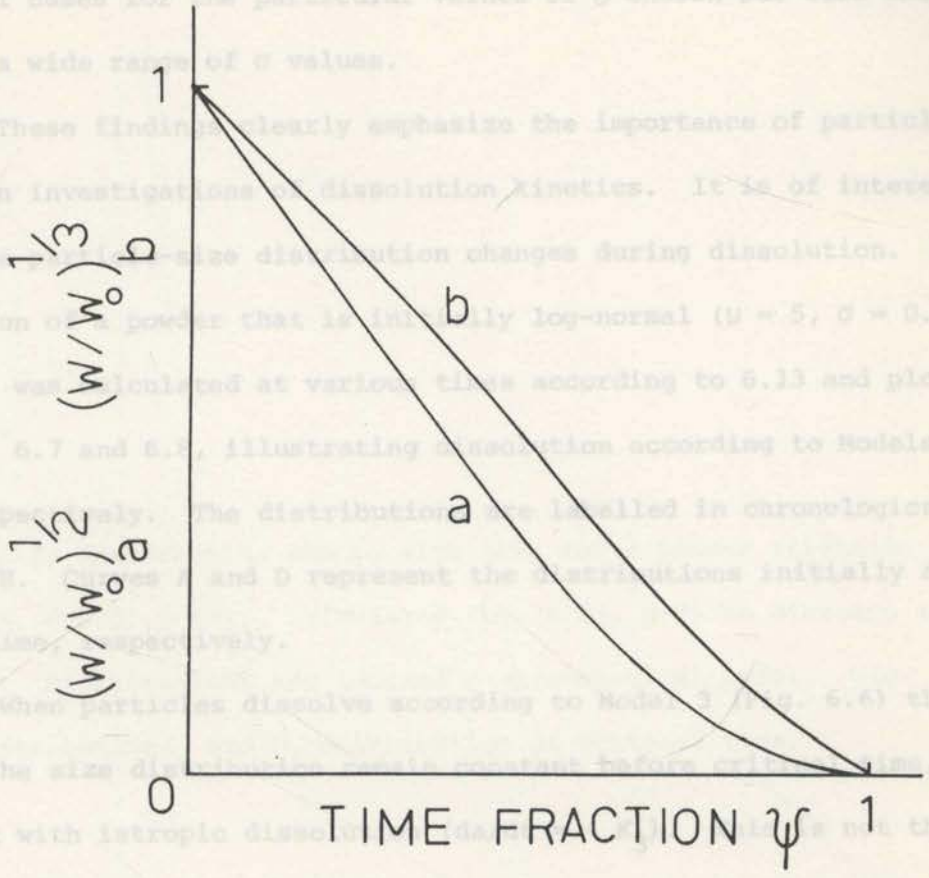


Figure 6.5

Normalized dissolution profiles of powders initially log-normal (σ=0.12, i=j=2) calculated (Eq. 6.21, m=2) to dissolve and plotted according to model 2. Curve b is the same data plotted according to model 3, resulting in almost complete cancellation of the size distribution effect observed in curve a.


1
$(W/W_o)^{1/3}_b$
$(W/W_o)^{1/2}_a$
b
a
0
TIME FRACTION ψ
1

dissolution is followed to competition, unless an analysis of particle size distribution is made. The phenomenon demonstrated in Figs. 6.4 and 6.5 are not special cases for the particular values of σ chosen but were observed to apply for a wide range of σ values.

These findings clearly emphasize the importance of particle-size analysis in investigations of dissolution kinetics. It is of interest to see how the particle-size distribution changes during dissolution. The distribution of a powder that is initially log-normal ($\mu = 5$, $\sigma = 0.2$, $i = j = 2$) was calculated at various times according to 6.13 and plotted in Figs. 6.6, 6.7 and 6.8, illustrating dissolution according to Models 3, 2 and 1, respectively. The distributions are labelled in chronological order from A to H. Curves A and D represent the distributions initially and at critical time, respectively.

When particles dissolve according to Model 3 (Fig. 6.6) then the shape of the size distribution remain constant before critical time, consistent with istropic dissolution ($da/dt = - K_3$). This is not the case for dissolution according to Models 1 and 2 where the absolute rate of change in size of the particles, da/dt, increases with time. For Model 1:

$$da/dt = - \frac{K_1}{2} (a_o^2 - K_1 t)^{-\frac{1}{2}} \qquad (6.32)$$

and for Model 2:

$$da/dt = - \frac{2}{3} K_2 (a^{3/2} - K_2 t)^{-1/3} \qquad (6.33)$$

As a result, the distribution broadens before critical time (Figs. 6.7 and 6.8) and is particularly affected at the small particle end as zero is approached where da/dt takes extreme values. Because of the latter effect, near the end of the dissolution process the relative frequency of the very small particles increases with increasing size for Models 1 and 2 (Fig. 6.8 and 6.7) while it decreases for Model 3 (Fig. 6.6). This information

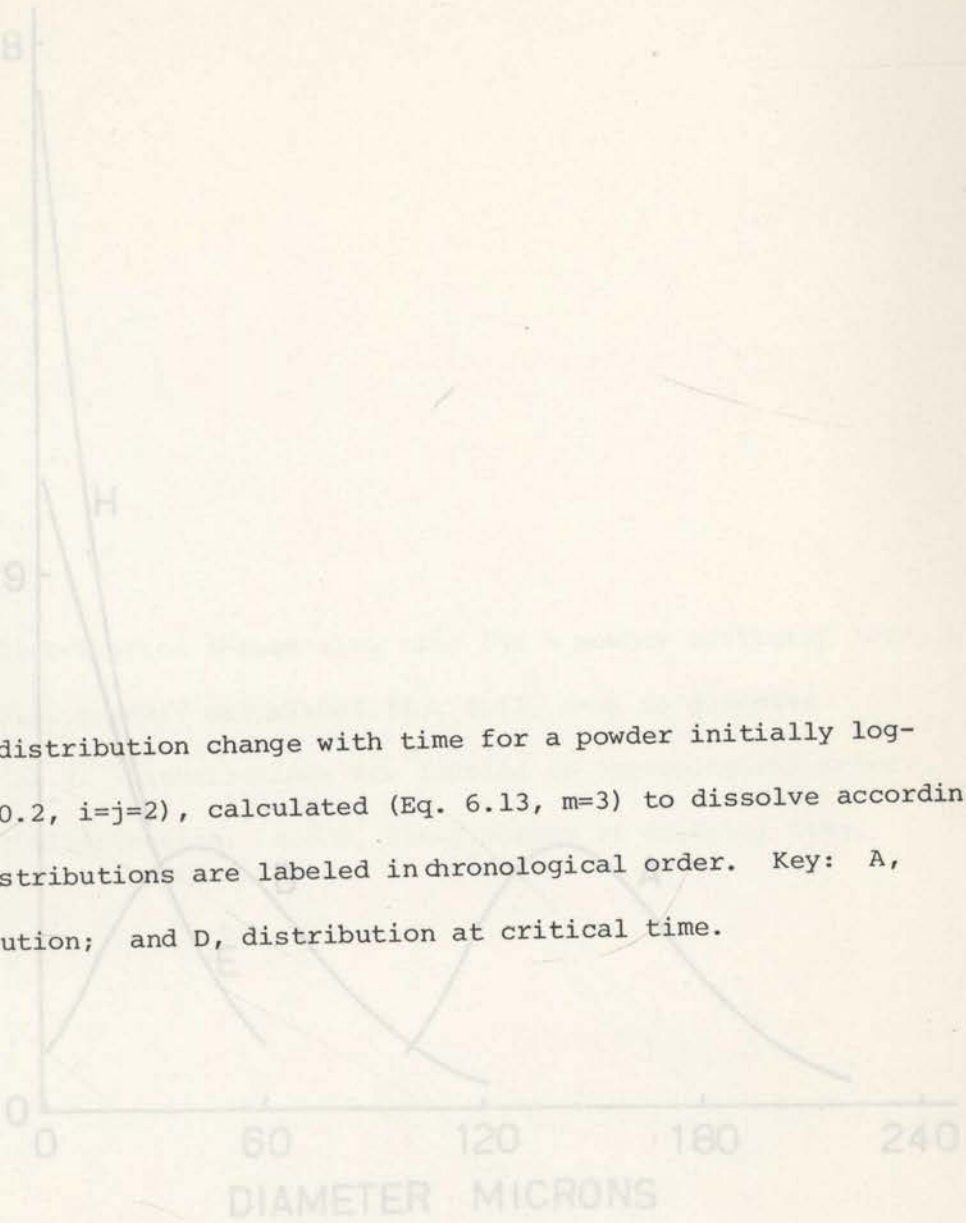
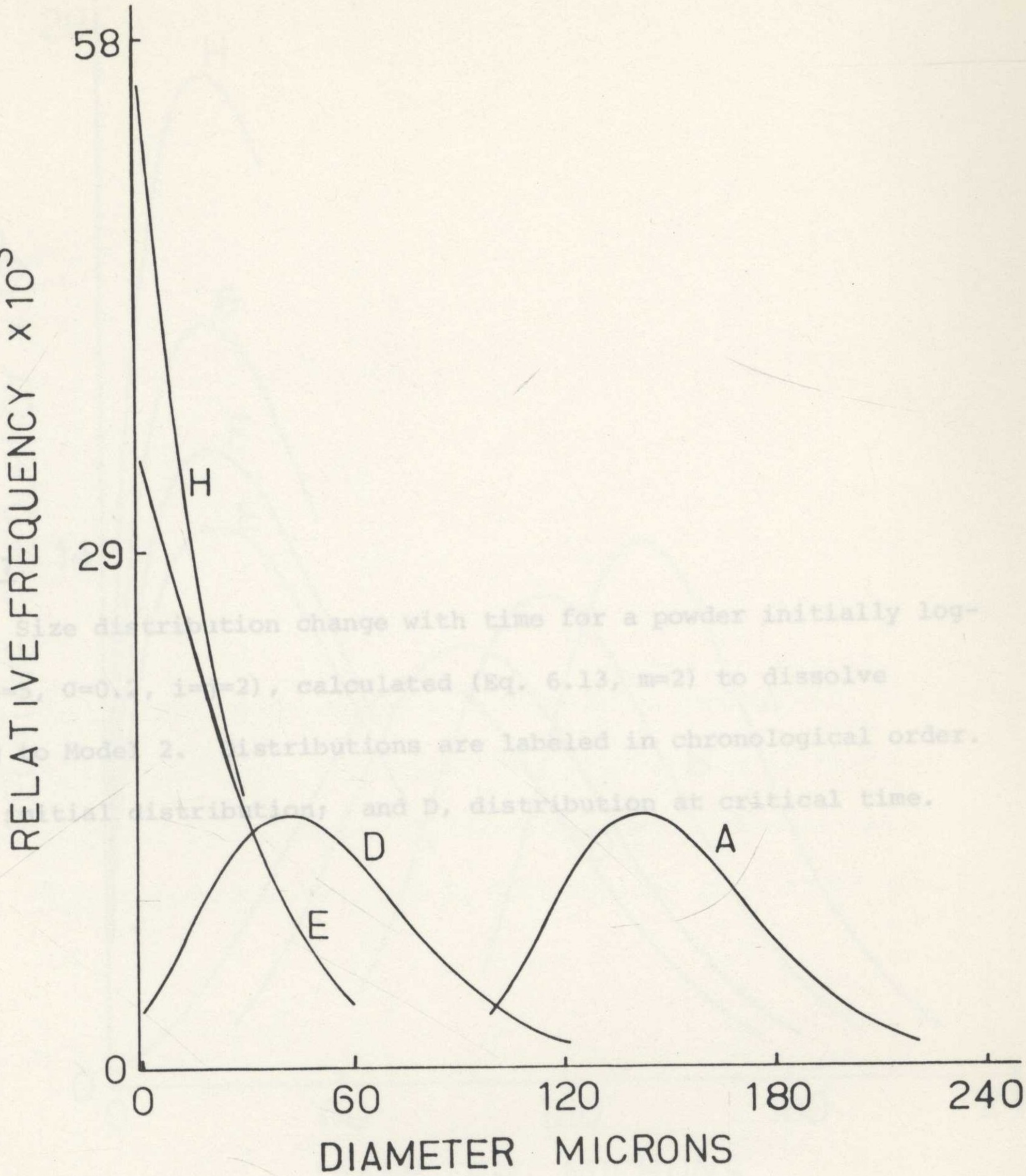


Figure 6.6

Size distribution change with time for a powder initially log-normal (μ=5, σ=0.2, i=j=2), calculated (Eq. 6.13, m=3) to dissolve according to Model 3. Distributions are labeled in chronological order. Key: A, initial distribution; and D, distribution at critical time.


58
29
0
RELATIVE FREQUENCY x 10^3
H
D
E
A
0
60
120
180
240
DIAMETER MICRONS

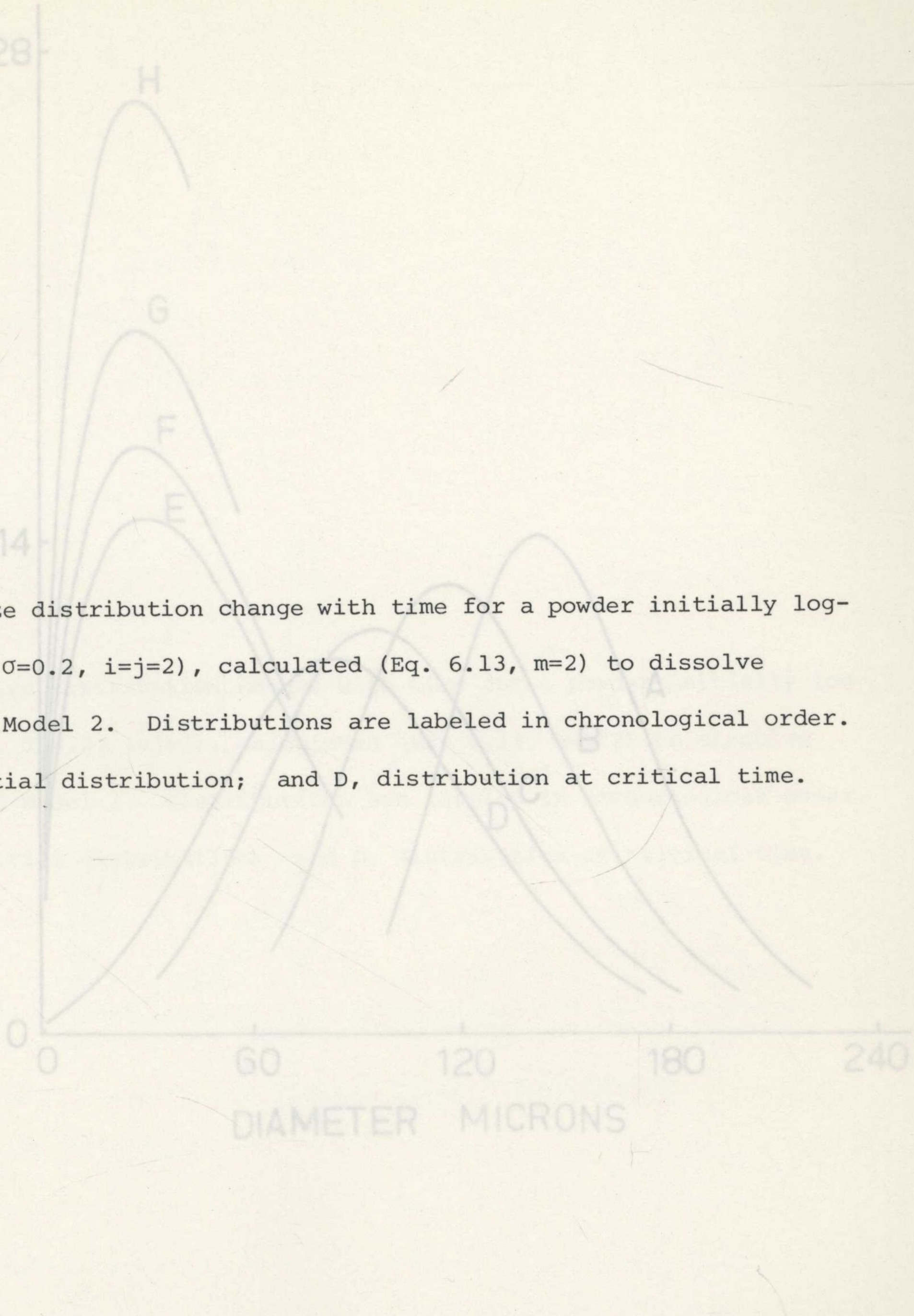
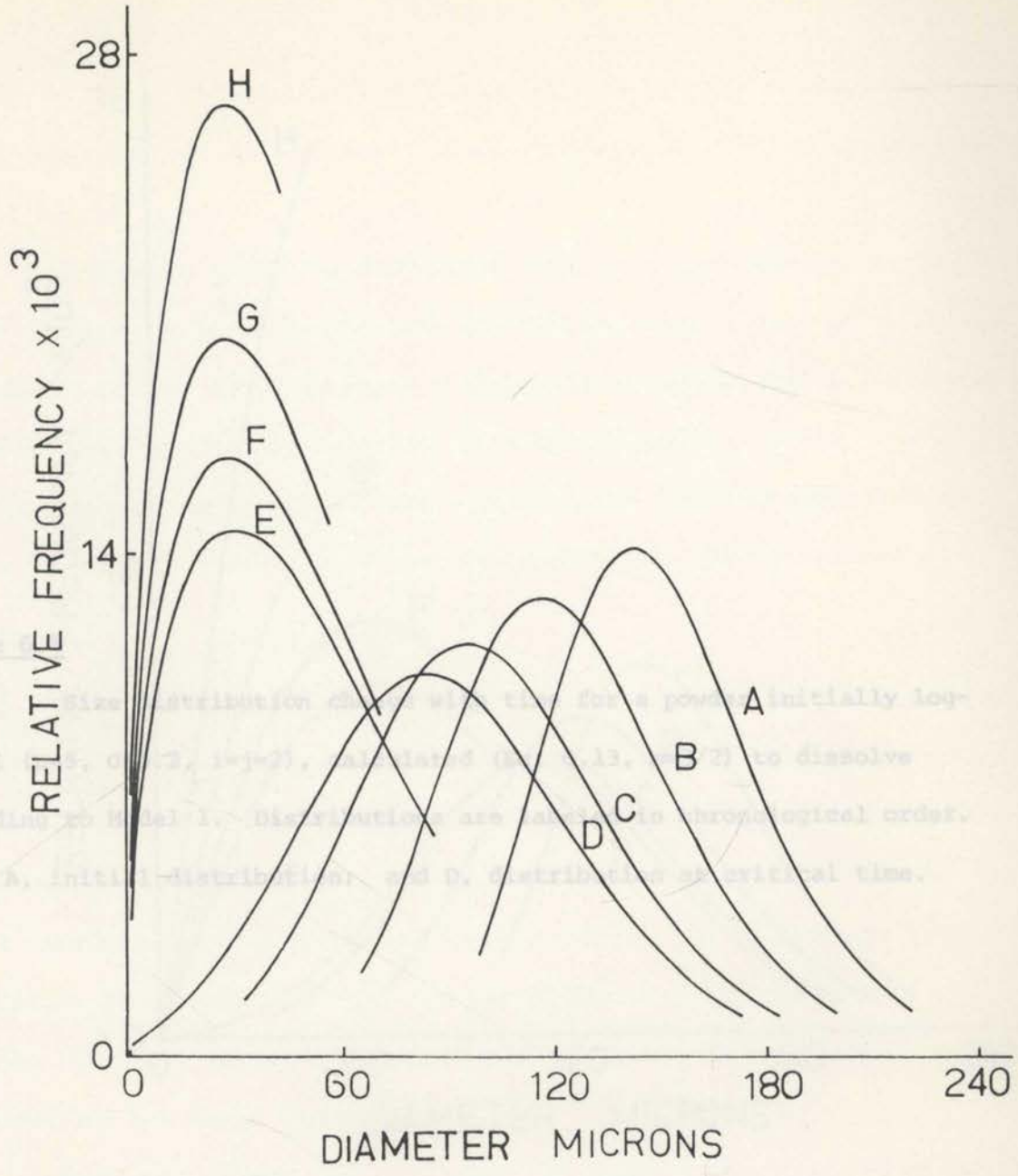


Figure 6.7.

Size distribution change with time for a powder initially log-normal (μ=5, σ=0.2, i=j=2), calculated (Eq. 6.13, m=2) to dissolve according to Model 2. Distributions are labeled in chronological order. Key: A, initial distribution; and D, distribution at critical time.


28
14
0
RELATIVE FREQUENCY x 10^3
H
G
F
E
A
B
C
D
0
60
120
180
240
DIAMETER MICRONS

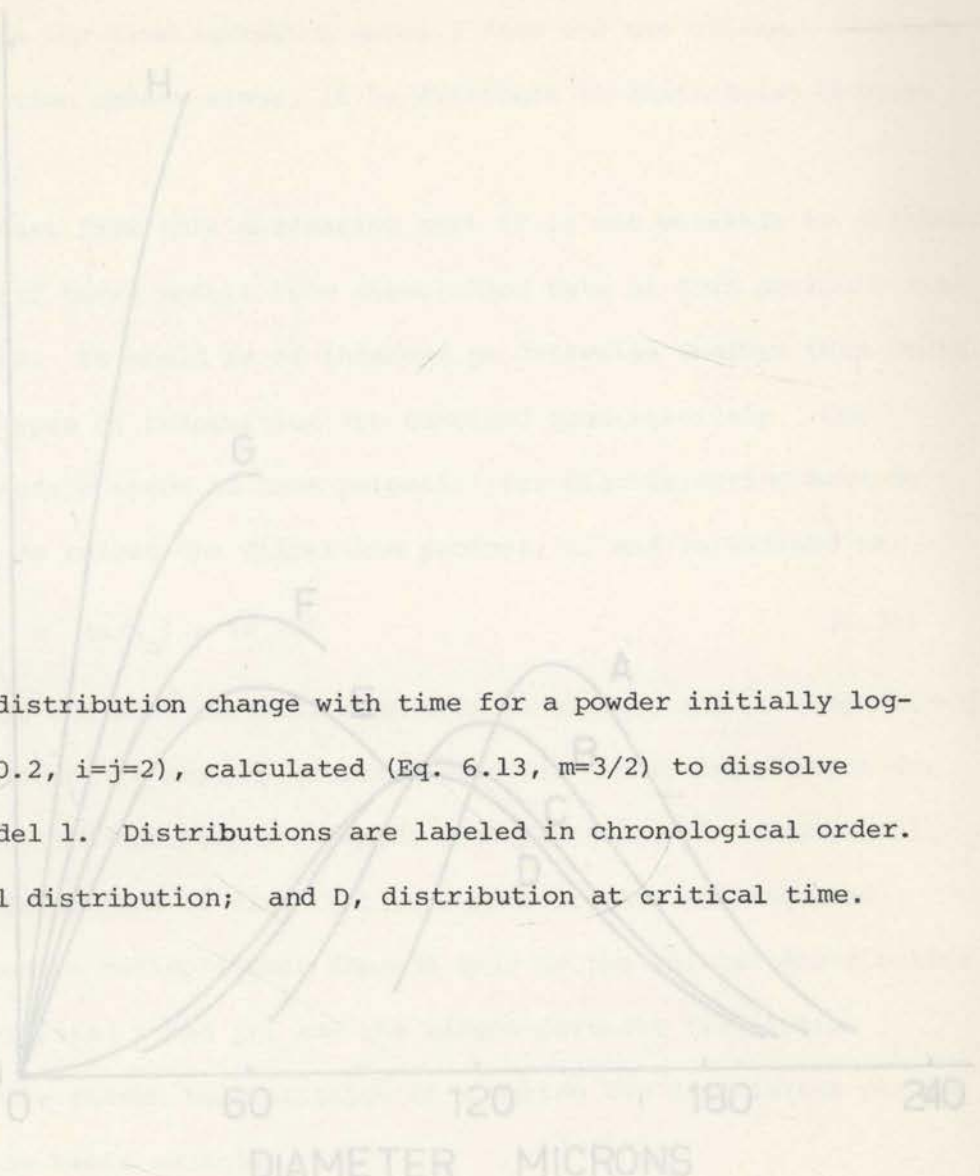
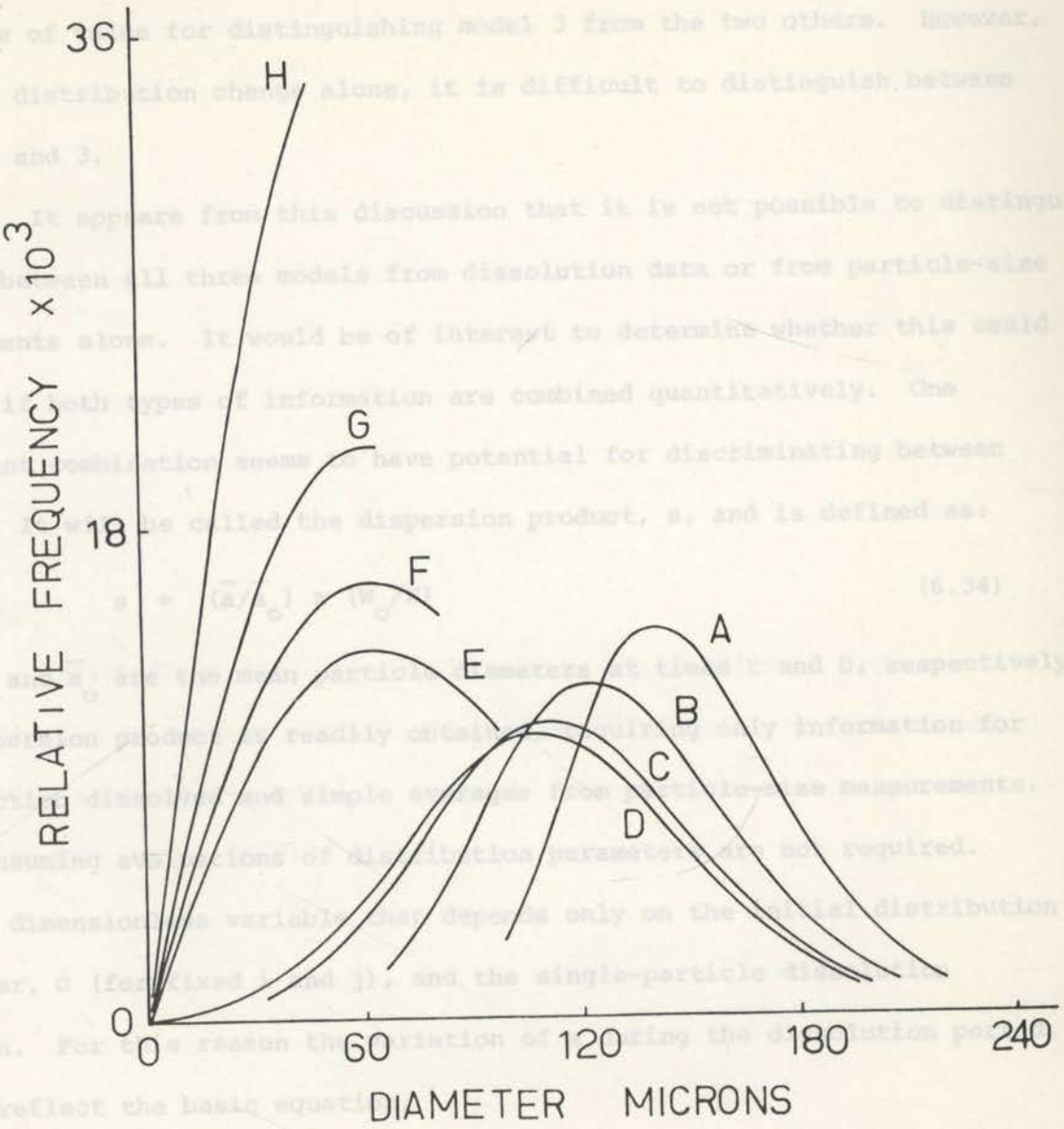


Figure 6.8

Size distribution change with time for a powder initially log-normal (μ=5, σ=0.2, i=j=2), calculated (Eq. 6.13, m=3/2) to dissolve according to Model 1. Distributions are labeled in chronological order. Key: A, initial distribution; and D, distribution at critical time.


36
18
0
RELATIVE FREQUENCY x10^3
H
G
F
E
A
B
C
D
0
60
120
180
240
DIAMETER MICRONS

should be of value for distinguishing model 3 from the two others. However, from the distribution change alone, it is difficult to distinguish between Models 2 and 3.

It appears from this discussion that it is not possible to distinguish clearly between all three models from dissolution data or from particle-size measurements alone. It would be of interest to determine whether this could be done if both types of information are combined quantitatively. One convenient combination seems to have potential for discriminating between models. It will be called the dispersion product, s, and is defined as:

$$s = (\bar{a}/\bar{a}_o) \times (W_o/W) \tag{6.34}$$

where $\bar{a}$ and $\bar{a}_o$ are the mean particle diameters at times t and 0, respectively. The dispersion product is readily obtained, requiring only information for the fraction dissolved and simple averages from particle-size measurements. Time-consuming evaluations of distribution parameters are not required. It is a dimensionless variable that depends only on the initial distribution parameter, σ (for fixed i and j), and the single-particle dissolution equation. For this reason the variation of s during the dissolution period should reflect the basic equation.

Figure 6.9 shows this variation for powders initially log-normally distributed ($\sigma = 0.2$, $i = j = z$). The curves representing the three dissolution models are significantly different. The basic shape of the curve remains the same for varying values of σ, although the minima shift to the right and to higher values for very narrow distributions. All three curves approach $s = 1$ (stippled line) when σ approaches zero as expected for a completely monodisperse powder. The values of the three minima remain approximately constant. When σ ranges from somewhat less than 0.1 to at least 0.2 (Fig. 6.10), which encompasses most fine powder distributions

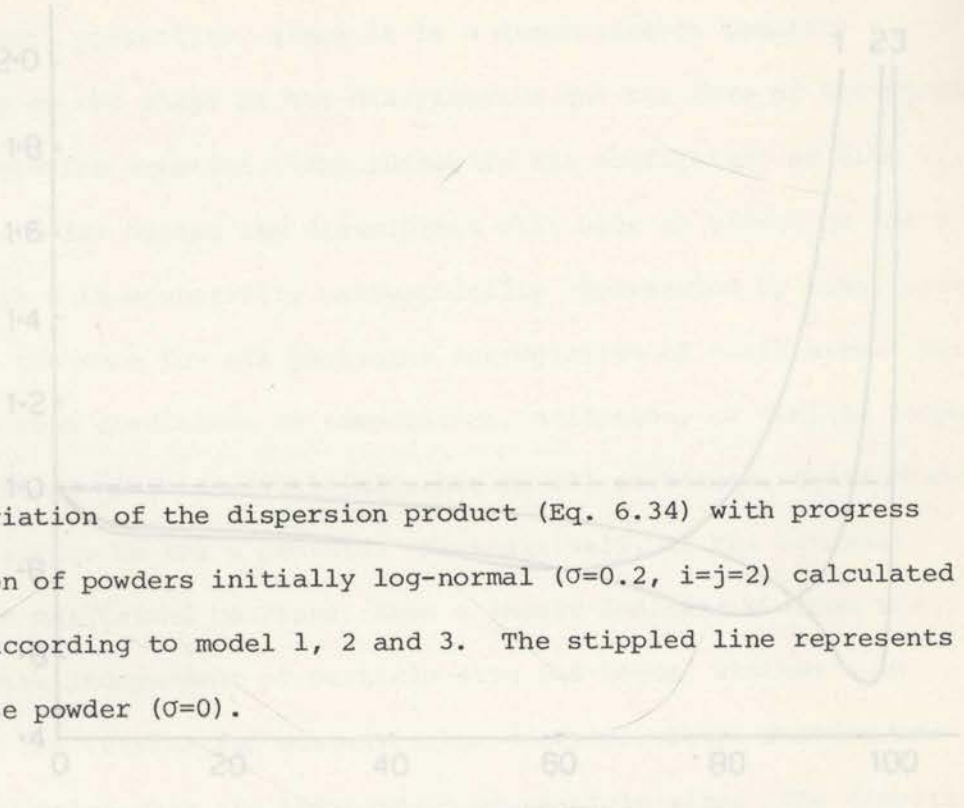
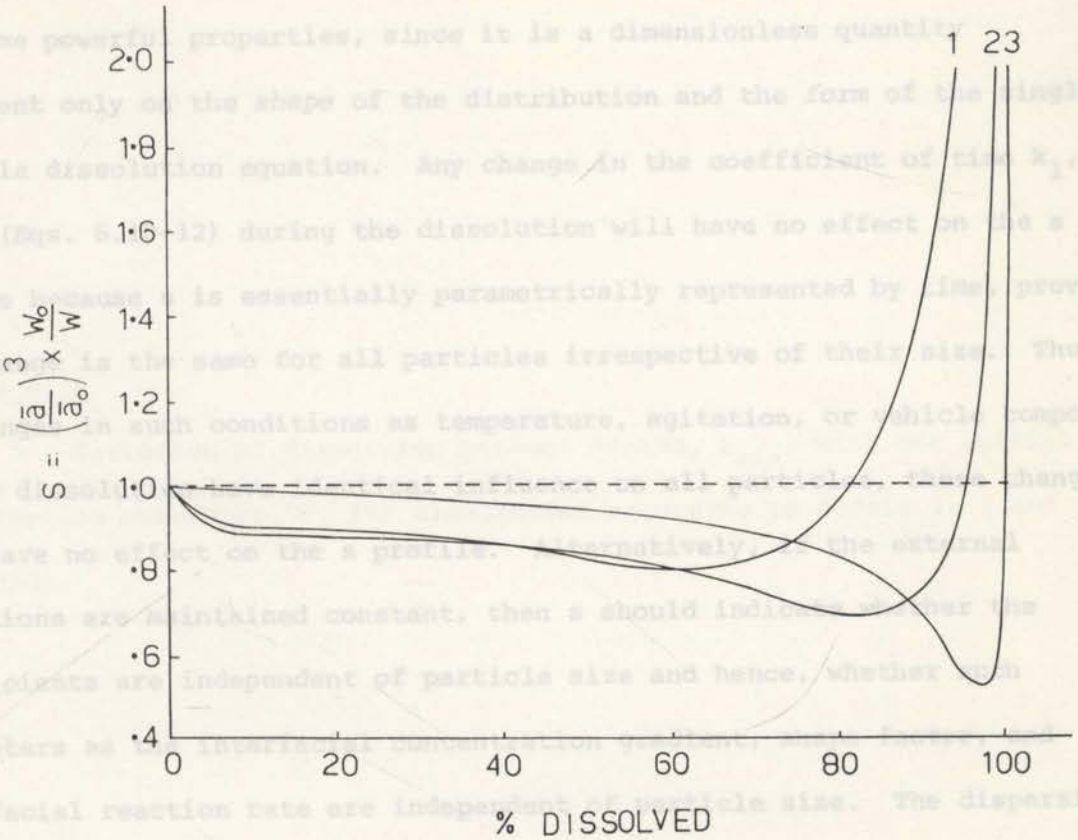


Figure 6.9

Variation of the dispersion product (Eq. 6.34) with progress of dissolution of powders initially log-normal (σ=0.2, i=j=2) calculated to dissolve according to model 1, 2 and 3. The stippled line represents a monodisperse powder (σ=0).


1 23
2·0
1·8
1·6
1·4
1·2
1·0
·8
·6
·4
$S = \left(\frac{\bar{a}}{\bar{a}_o}\right)^3 \times \frac{W_o}{W}$
0
20
40
60
80
100
% DISSOLVED

encountered in practice. Theoretically, the considerable difference between the s_{min} values should make it possible to distinguish between the three models.

The dispersion product profile (s *versus* W/W_o or percent dissolved has some powerful properties, since it is a dimensionless quantity dependent only on the *shape* of the distribution and the *form* of the single-particle dissolution equation. Any change in the coefficient of time k_1, k_2 or k_3 (Eqs. 6.10-12) during the dissolution will have no effect on the s profile because s is essentially parametrically represented by time, provided the change is the same for all particles irrespective of their size. Thus, if changes in such conditions as temperature, agitation, or vehicle composition during dissolution have identical influence on all particles, these changes will have no effect on the s profile. Alternatively, if the external conditions are maintained constant, then s should indicate whether the coefficients are independent of particle size and hence, whether such parameters as the interfacial concentration gradient, shape factor, and interfacial reaction rate are independent of particle size. The dispersion product should, therefore, be a valuable tool in dissolution kinetic studies.

The extent to which these mathematical models can be applied to describe the dissolution of a "real" powder depends on three assumptions.

1. As mentioned earlier, it was assumed that the particles dissolve independently of each other. This should be approximated well under sink conditions.

2. It was assumed that the dissolution of each particle in the powder can be described by an equation having the same parameter value (k) for all the particles. In practice, this assumption is rarely valid because of differences in individual particle shapes, crystal structure, and interaction with the vehicle. However, these types of effects probably can be averaged to produce a parameter value for the single-particle dissolution

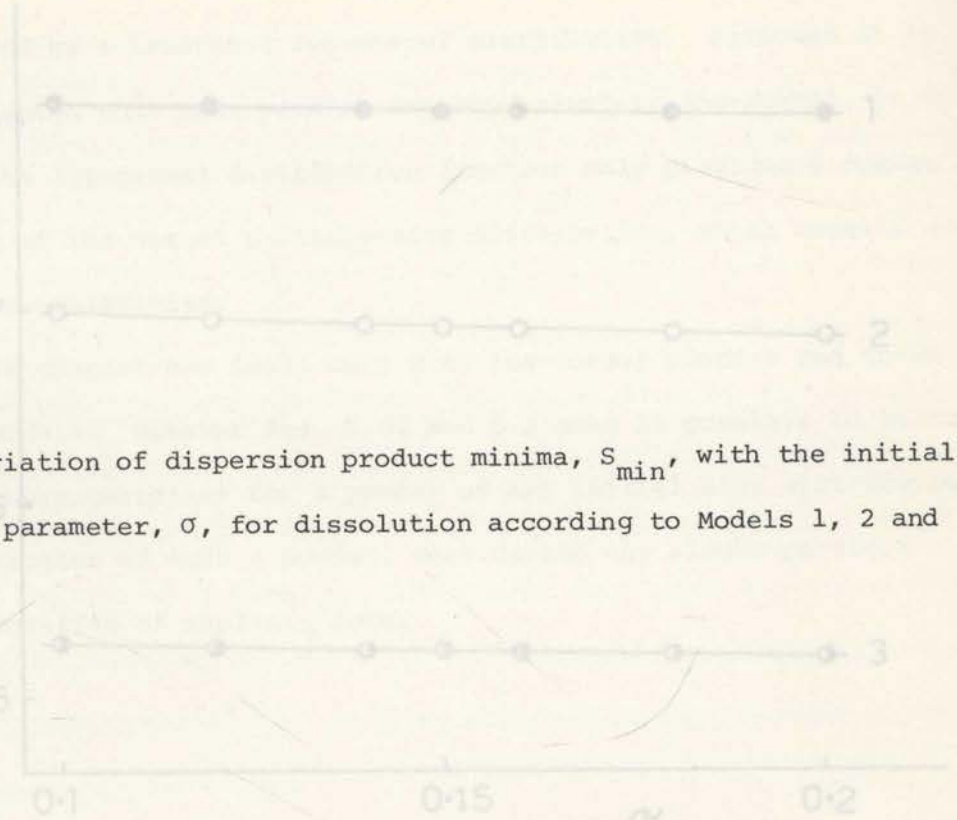
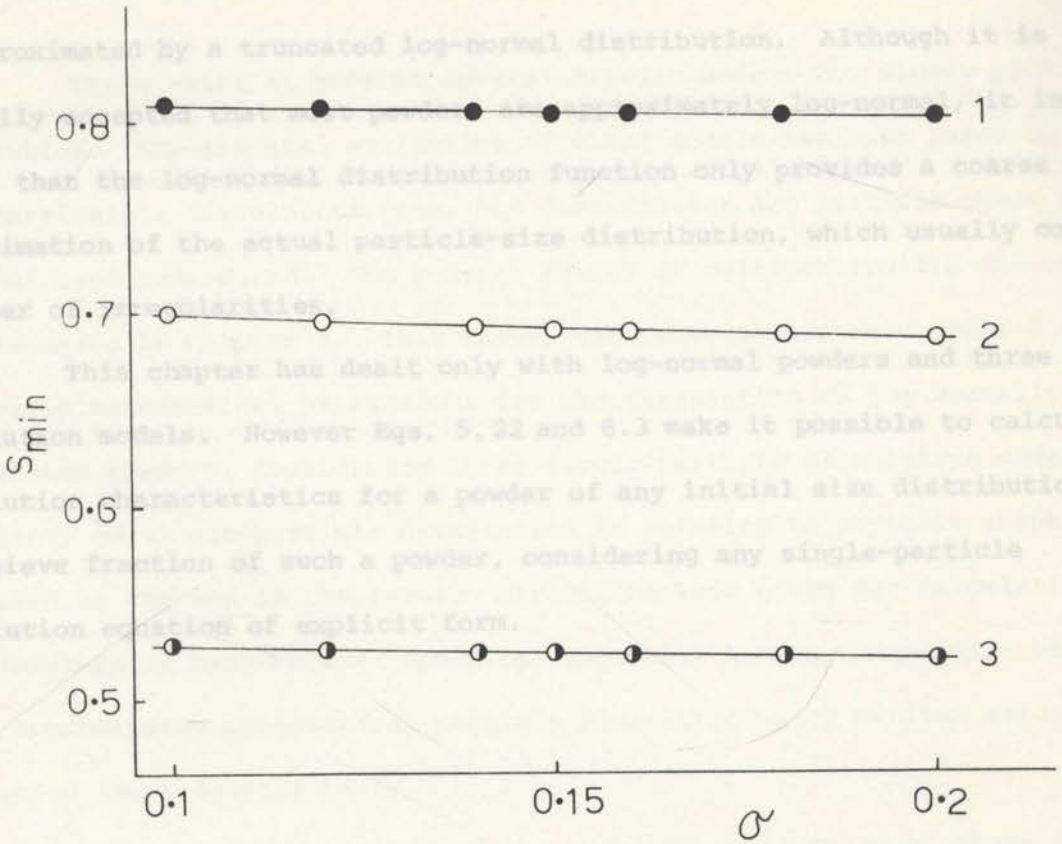


Figure 6.10

Variation of dispersion product minima, S_{min}, with the initial distribution parameter, σ, for dissolution according to Models 1, 2 and 3 (i=j=2).


0·8
0·7
0·6
0·5
S_{min}
1
2
3
0·1
0·15
0·2
σ

model which, when used in the multiparticulate dissolution equation, results in a good approximation of the actual multiparticulate dissolution behavior.

3. It was assumed that the initial particle-size distribution can be approximated by a truncated log-normal distribution. Although it is generally accepted that most powders are approximately log-normal, it is likely that the log-normal distribution function only provides a coarse approximation of the actual particle-size distribution, which usually contains a number of irregularities.

This chapter has dealt only with log-normal powders and three dissolution models. However Eqs. 5.22 and 6.3 make it possible to calculate dissolution characteristics for a powder of any initial size distribution, or a sieve fraction of such a powder, considering any single-particle dissolution equation of explicit form.

CHAPTER 7

EXPERIMENTAL EVALUATION OF THREE DISSOLUTION MODELS

There exist at present several kinetic models for single particle dissolution. Experimental evaluation of these models has been based on multiparticulate dissolution data, but distribution and particle shape effects have not been considered. The general theory of multiparticulate dissolution was discussed in Chapter 5. This theory was subsequently in Chapter 6 used to develop mathematical expressions for the dissolution of log-normally distributed powders, considering three single-particle dissolution models. The theory of single-particle dissolution in relation to particle shape was discussed at the end of Chapter 4. Directions were given for calculating the diameters of hypothetical spherical particles having dissolution behaviour which approximates nonspherical particle dissolution with minimum error on a weighted least squares basis.

This chapter demonstrates the combined application of these theories to explain the dissolution kinetics of 60/85-mesh fraction of tolbutamide such that both size distribution effects and particle shape effects are considered. By using a time-scaling approach, the three single-particle dissolution kinetic models are evaluated.

Theoretical Considerations

The three equations for dissolution of spherical particles under sink conditions considered can be written in common form as:

$$w = (w_o^{\frac{1}{m}} - kt)^m \qquad m = 3,\ 2,\ \frac{3}{2} \qquad (7.1)$$

$$k = k_1,\ k_2,\ k_3$$

representing the cube root, the square root and the 3/2-root models, respectively (97,106,107). The positive constants k_1, k_2 and k_3 replace the original coefficients of time. These contained quantities such as density, diffusion coefficient and shape factor. This simplification is made because the aim is not to evaluate the theoretical basis of the three equations but solely to assess them as models for describing the dissolution kinetic data.

These equations (7.1) do not strictly describe the dissolution correctly in their present form since w does not vanish for $t \to \infty$. A more correct formulation would therefore be:

$$w = (w_o^{\frac{1}{m}} - kt)^m \quad \text{for } t \leq w_o^{\frac{1}{m}}/k \qquad (7.2)$$

$$w = 0 \quad \text{for } t > w_o^{\frac{1}{m}}/k$$

The general multiparticulate dissolution equation presented earlier (5.22) however was developed such that it accepts the single particle dissolution function in the forms of both 7.1 and 7.2. Equation 7.2 will be used for simplicity.

In Chapter 6 the following equations were presented to describe the dissolution profile of a multiparticulate log-normal system where the spherical particles dissolve according to each of the three models (7.1):

$$\frac{W}{W_o} = \sum_{n=o}^{m} \binom{m}{n} (-Kt)^{(m-n)} \frac{F(A)-F(B)}{F(j-3\sigma)-F(-i-3\sigma)} \text{ EXP } (C) \qquad (7.3)$$

where $A = (T_2-\mu)/\sigma - 3n\sigma/m$

$B = (T_1-\mu)/\sigma - 3n\sigma/m$

$C = \text{EXP}\left[\frac{3}{m}(n-m)\left(\mu+\frac{3}{m}(n+m)\sigma^2/2\right)\right]$

$T_1 = \max\left[\frac{m}{3} \ln (Kt),\ \mu-i\sigma\right]$

$T_2 = \max\left[\frac{m}{3} \ln (Kt),\ \mu+j\sigma\right]$

$$\frac{W}{W_o} = \frac{\int_{R_1}^{R_2} (w^2-Kt)^{3/2}\, w^{-1}\, N(\ln w,\mu,\sigma)\, dw}{F(j-3\sigma)-F(-i-3\sigma)} \; EXP(-3\mu-9\sigma^2/2) \qquad (7.4)$$

where $R_1 = \max\left[(Kt)^{\frac{1}{2}},\ EXP(\mu-i\sigma)\right]$

$R_2 = \max\left[(Kt)^{\frac{1}{2}},\ EXP(\mu+j\sigma)\right]$

The function F( ) above is the area under standard normal curve function defined earlier (5.37). The function N(ln w,μ,σ,) is the normal distribution function with ln w as variable, defined by:

$$N(\ln w,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\; EXP\left[-\tfrac{1}{2}\left(\frac{\ln w-\mu}{\sigma}\right)^2\right]$$

The constant K is related to k_1, k_2 and k_3 by:

$$K = (6/\rho\pi)^{\frac{1}{m}} k_i \qquad i = 1, 2, 3 \qquad (7.6)$$

where ρ is the particle density.

It is assumed that the particle diameter (a) distribution is "log-normal", that is, ln a can be approximated by a normal distribution (mean = μ, standard deviation = σ) truncated at the lower end at μ-iσ and at the upper end at μ+jσ, where i and j are truncation parameters (Fig. 5.1).

The multiparticulate dissolution equations above consider spherical drug particles. Such particles are only encountered when the drug exists in liquid form as an emulsion. In solid form the particles are not spherical. The drug used for the dissolution tests was a 60/85 mesh fraction of Tolbutamide consisting of particles approximately tetragonal prismatic in shape. It was shown in Chapter 4 that the dissolution of such particles can be approximated well by the dissolution of hypothetical spherical particles. The equivalent spherical diameter, a, is the diameter of the spherical particle that best approximates the dissolution of the non-spherical particle, and is given by:

$$a = \frac{b_o}{2-(2-\frac{b_o}{l_o})^{1/3}} \qquad (7.7)$$

where l_o and b_o are the length and side respectively of the tetragonal particle. In this way dissolution of the non-spherical particle system can be suitably described by the dissolution of a hypothetical spherical particle system that can be rigorously treated using the equations above.

It is evident from these that k, μ, σ, i, j and ρ must be known to calculate the dissolution profile. The distribution parameters μ, σ, i and j can be obtained from micrographs and ρ by a standard method, but the single particle rate parameter k is unknown. It is possible however to calculate the exact intrinsic dissolution profile with much less information. According to the rules given previously (Chapter 6) for multiparticulate dissolution only the *shape* of the initial distribution, that is, of the above six parameters only σ, i and j are required to calculate the intrinsic dissolution profile when the single particle dissolution model is known. The concept of time scaling was discussed in Chapter 6. By such an approach it is possible to evaluate quantitatively the difference between the actual dissolution data and the calculated intrinsic dissolution profile.

Figure 7.1 illustrates this application of time scaling. Curve B through the experimental data points represents the dissolution curve ($(w/w_o)^{\frac{1}{m}}$ *versus* t) and curve A the corresponding calculated normalised intrinsic dissolution profile ($(w/w_o)^{\frac{1}{m}}$ *versus* ψ).

Let, N, be the number of data points and, f, the time scaling factor that brings curve B "into" curve A such that the sum of the squared deviations between the curves given by:

$$ss = \sum_{i=1}^{N} (ft_i - \psi_i)^2 \qquad (7.8)$$

is a minimum. This means f is obtained from $\partial ss/\partial f = 0$ which gives:

$$f = \frac{\sum_{i=1}^{N} \psi_i t_i}{\sum_{i=1}^{N} t_i^2} \qquad (7.9)$$

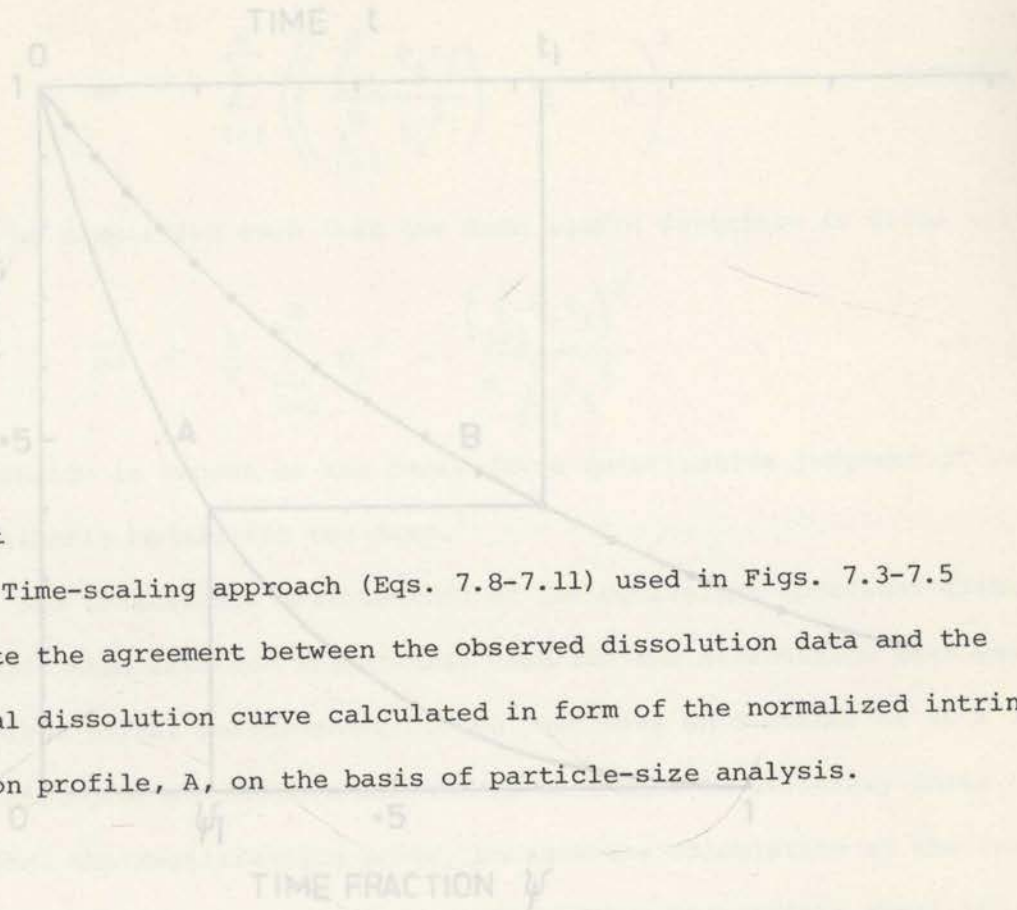
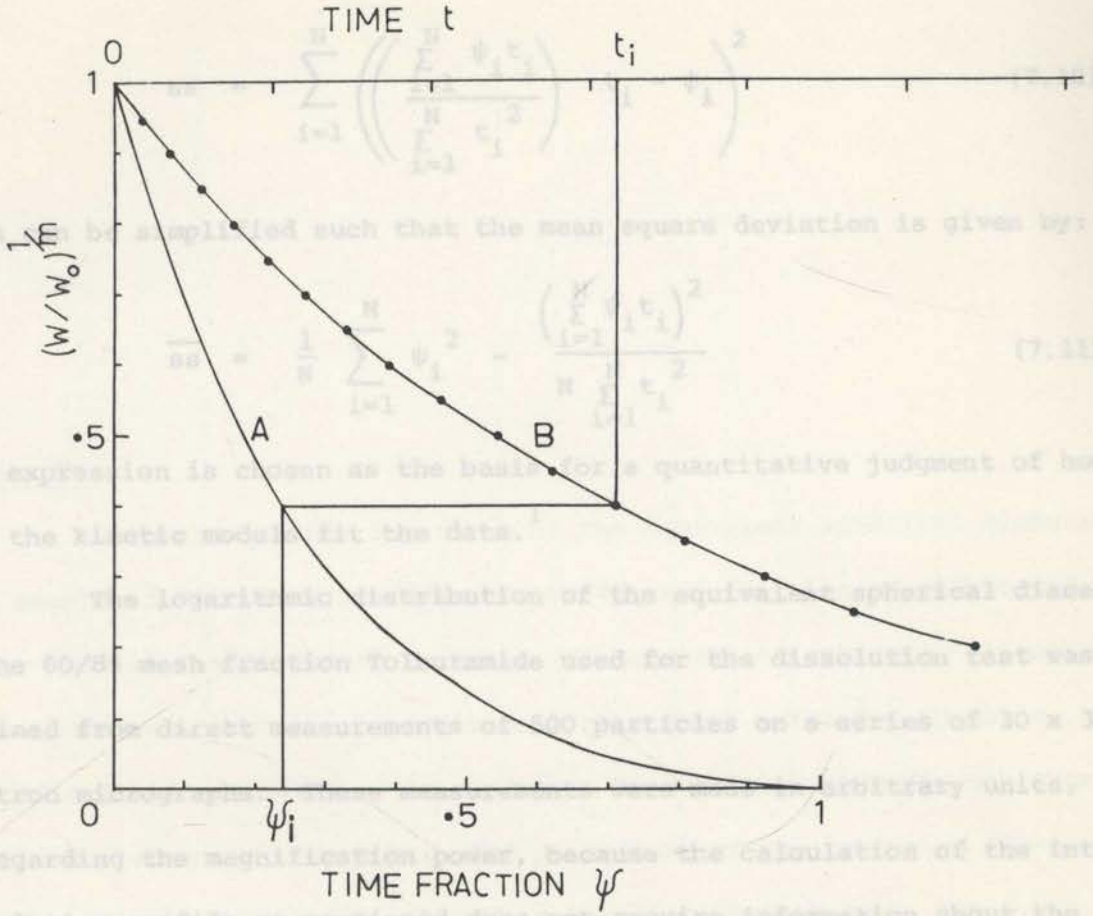


Figure 7.1

Time-scaling approach (Eqs. 7.8-7.11) used in Figs. 7.3-7.5 to evaluate the agreement between the observed dissolution data and the theoretical dissolution curve calculated in form of the normalized intrinsic dissolution profile, A, on the basis of particle-size analysis.


TIME t
0
1
t_i
$(W/W_o)^{\frac{1}{m}}$
•5
A
B
0
ψ_i
•5
1
TIME FRACTION ψ

The minimum of the sum of squares is therefore:

$$ss = \sum_{i=1}^{N} \left(\left(\frac{\sum_{i=1}^{N} \psi_i t_i}{\sum_{i=1}^{N} t_i^2} \right) t_i - \psi_i \right)^2 \qquad (7.10)$$

which can be simplified such that the mean square deviation is given by:

$$\overline{ss} = \frac{1}{N} \sum_{i=1}^{N} \psi_i^2 - \frac{\left(\sum_{i=1}^{N} \psi_i t_i \right)^2}{N \sum_{i=1}^{N} t_i^2} \qquad (7.11)$$

This expression is chosen as the basis for a quantitative judgment of how well the kinetic models fit the data.[1]

The logarithmic distribution of the equivalent spherical diameters of the 60/85 mesh fraction Tolbutamide used for the dissolution test was obtained from direct measurements of 500 particles on a series of 30 x 30 cm electron micrographs. These measurements were made in arbitrary units, disregarding the magnification power, because the calculation of the intrinsic dissolution profile as mentioned does not require information about the actual sizes of the particles but the shape of their distribution (that is, σ, i and j for a log-normal powder). Each particle was approximated by the tetragonal prismatic body which fitted best, and its equivalent spherical diameter was calculated using Eq. 7.7.

The histogram of the logarithm of these diameters (Fig. 7.2) shows a good fit to a normal distribution with standard deviation $\sigma = 0.395$ and mean

---

1. The alternative approach, to bring curve A "into" curve B, that has the character of a curve fitting to the data points would yield the same result in a comparison of the models. The above time scaling of the *data* is used for the convenience of plotting and to better illustrate the predicted time for complete dissolution.

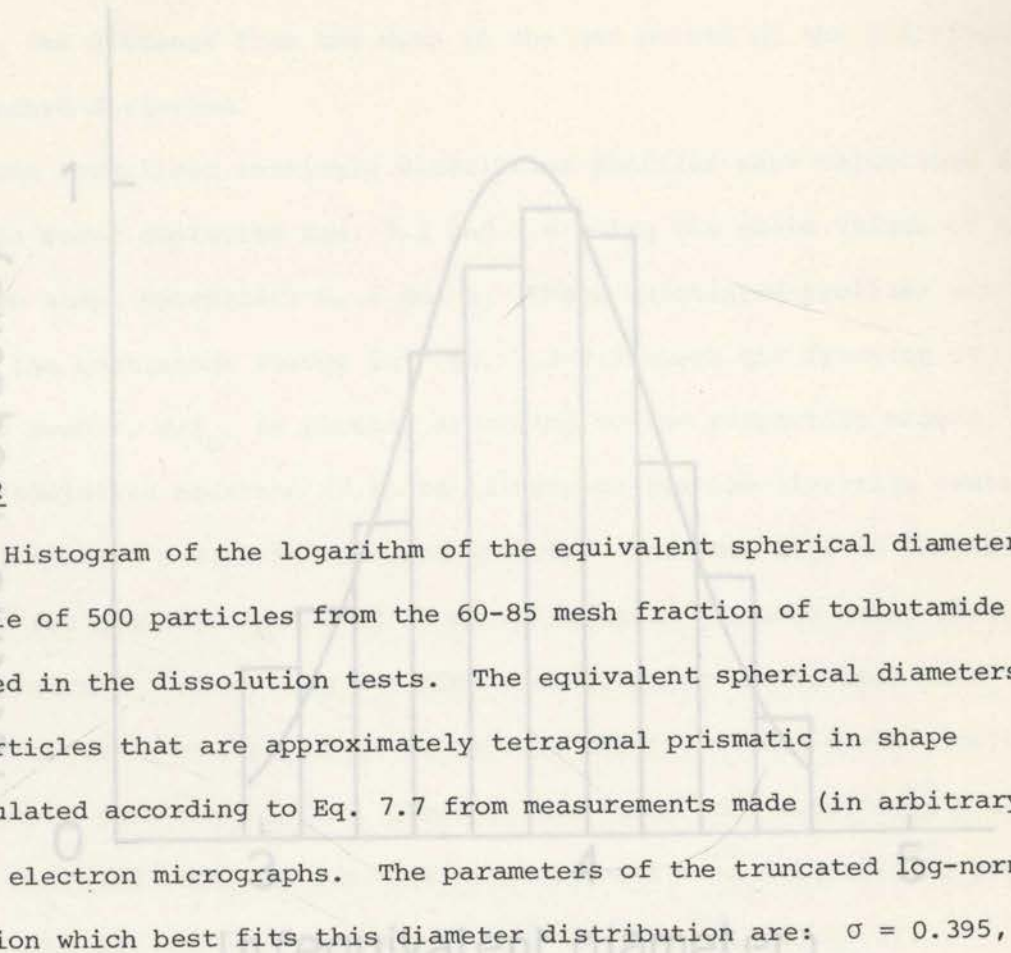
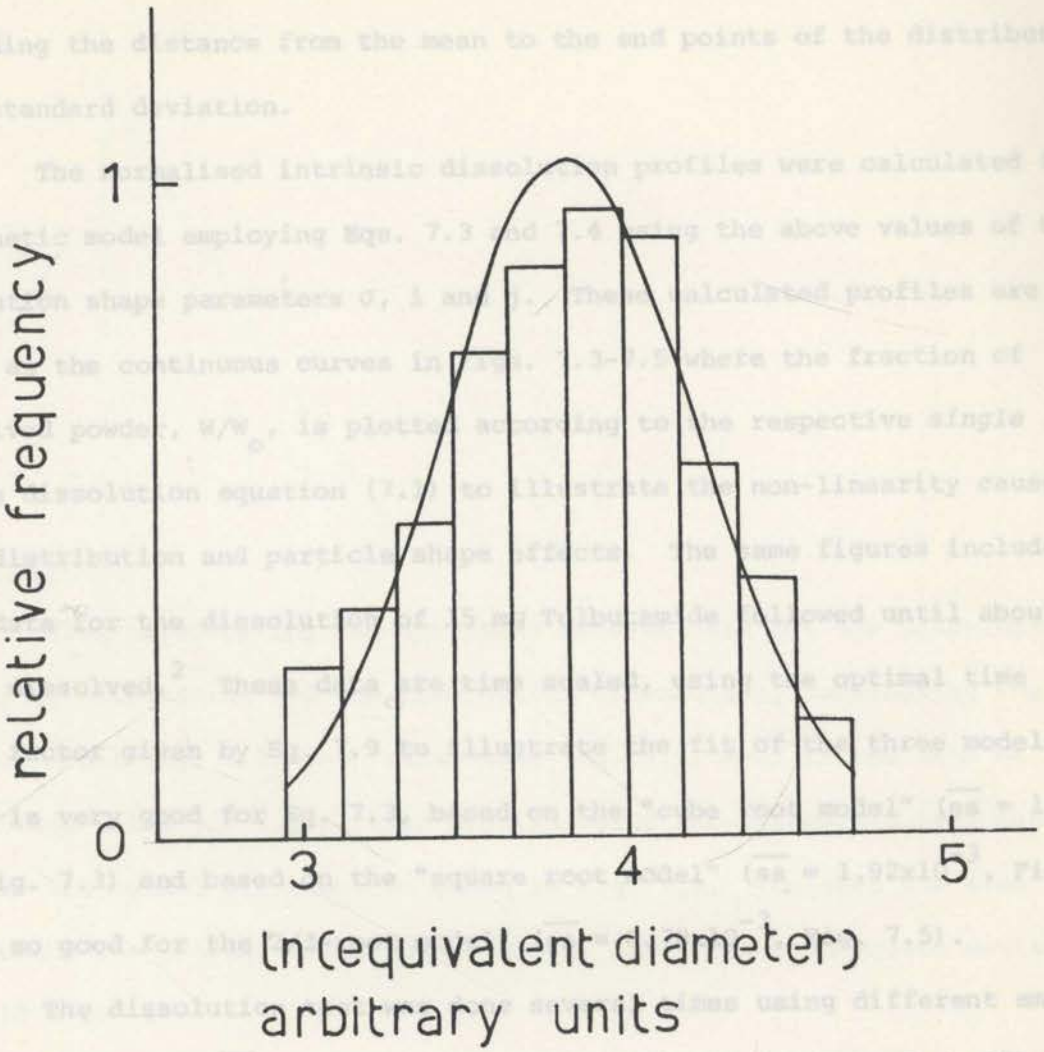


Figure 7.2

Histogram of the logarithm of the equivalent spherical diameters of a sample of 500 particles from the 60-85 mesh fraction of tolbutamide powder used in the dissolution tests. The equivalent spherical diameters of the particles that are approximately tetragonal prismatic in shape were calculated according to Eq. 7.7 from measurements made (in arbitrary units) on electron micrographs. The parameters of the truncated log-normal distribution which best fits this diameter distribution are: $\sigma = 0.395$, $\mu = 3.82$, $i = 2.25$, and $j = 2.20$.


relative frequency
1
0
3
4
5
ln (equivalent diameter)
arbitrary units

$\mu = 3.82$. The truncation parameters, $i = 2.25$, and $j = 2.20$, were obtained by dividing the distance from the mean to the end points of the distribution by the standard deviation.

The normalised intrinsic dissolution profiles were calculated for each kinetic model employing Eqs. 7.3 and 7.4 using the above values of the distribution shape parameters σ, i and j. These calculated profiles are graphed as the continuous curves in Figs. 7.3-7.5 where the fraction of undissolved powder, W/W_o, is plotted according to the respective *single* particle dissolution equation (7.1) to illustrate the non-linearity caused by the distribution and particle shape effects. The same figures include a set of data for the dissolution of 15 mg Tolbutamide followed until about 90% had dissolved.[2] These data are time scaled, using the optimal time scaling factor given by Eq. 7.9 to illustrate the fit of the three models. The fit is very good for Eq. 7.3, based on the "cube root model" ($\overline{ss} = 1.10 \times 10^{-3}$, Fig. 7.3) and based on the "square root model" ($\overline{ss} = 1.92 \times 10^{-3}$, Fig. 7.4) but not so good for the "2/3-root model" ($\overline{ss} = 6.70 \times 10^{-3}$, Fig. 7.5).

The dissolution test was done several times using different amounts of Tolbutamide. The $\overline{ss}$ - values (Table 7.1) indicate (F-test, $P < .05$) that the "cube root model" describes the single particle dissolution best with the "square root model" almost as good, however, the "2/3-root model" is relatively poor.

In order to make the above evaluations of the three models by comparing dissolution data with theoretical calculations it is necessary that the experimental conditions are consistent with the assumptions behind these calculations. The three assumptions on which Eqs. 7.3 and 7.4 are based are:

---

2. The absorbance after 90% dissolution was so small that substantial error would have been introduced if the process were followed much further.

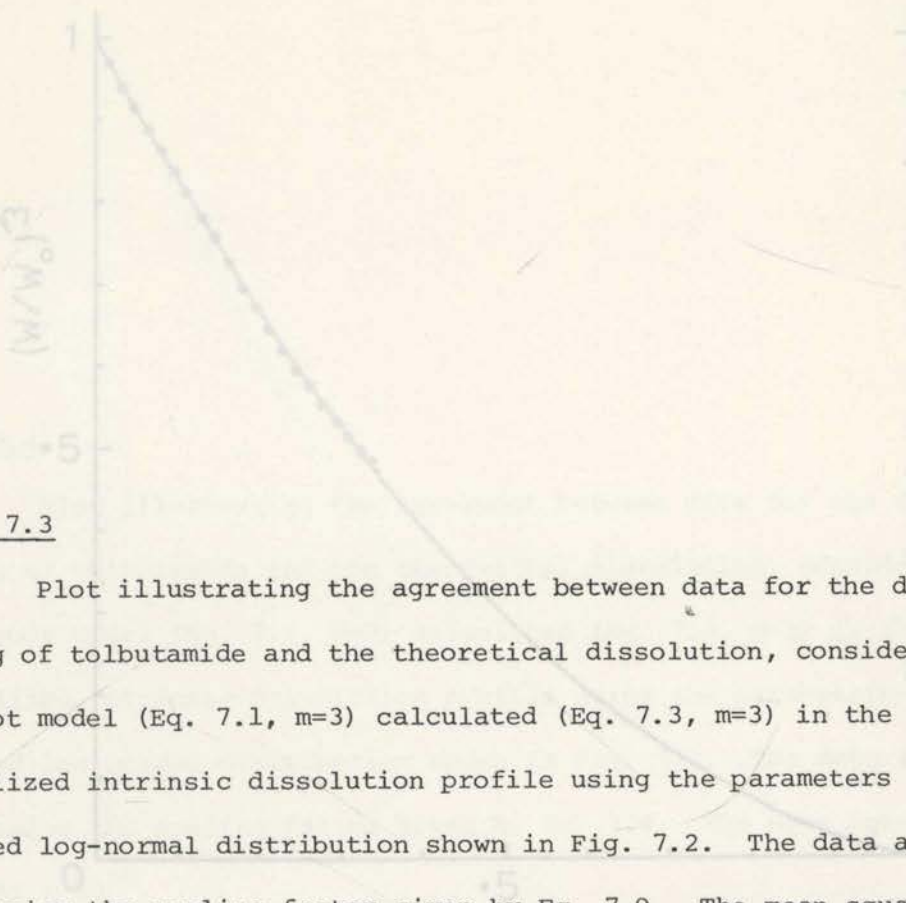
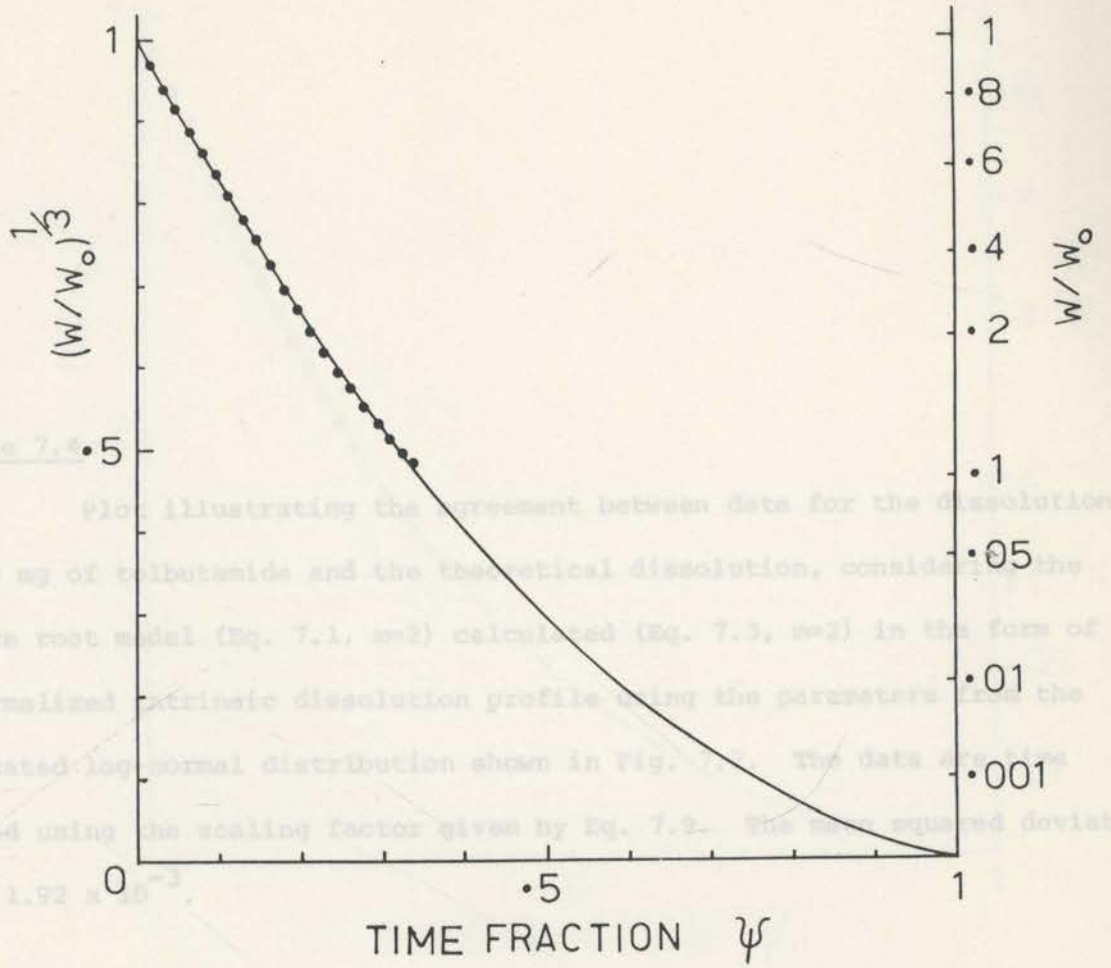


Figure 7.3

Plot illustrating the agreement between data for the dissolution of 15 mg of tolbutamide and the theoretical dissolution, considering the cube root model (Eq. 7.1, m=3) calculated (Eq. 7.3, m=3) in the form of a normalized intrinsic dissolution profile using the parameters from the truncated log-normal distribution shown in Fig. 7.2. The data are time scaled using the scaling factor given by Eq. 7.9. The mean square deviation $\overline{ss} = 1.10 \times 10^{-3}$.


1
•5
0
•5
1
$(W/W_o)^{1/3}$
1
•8
•6
•4
•2
•1
•05
•01
•001
W/W_o
TIME FRACTION ψ

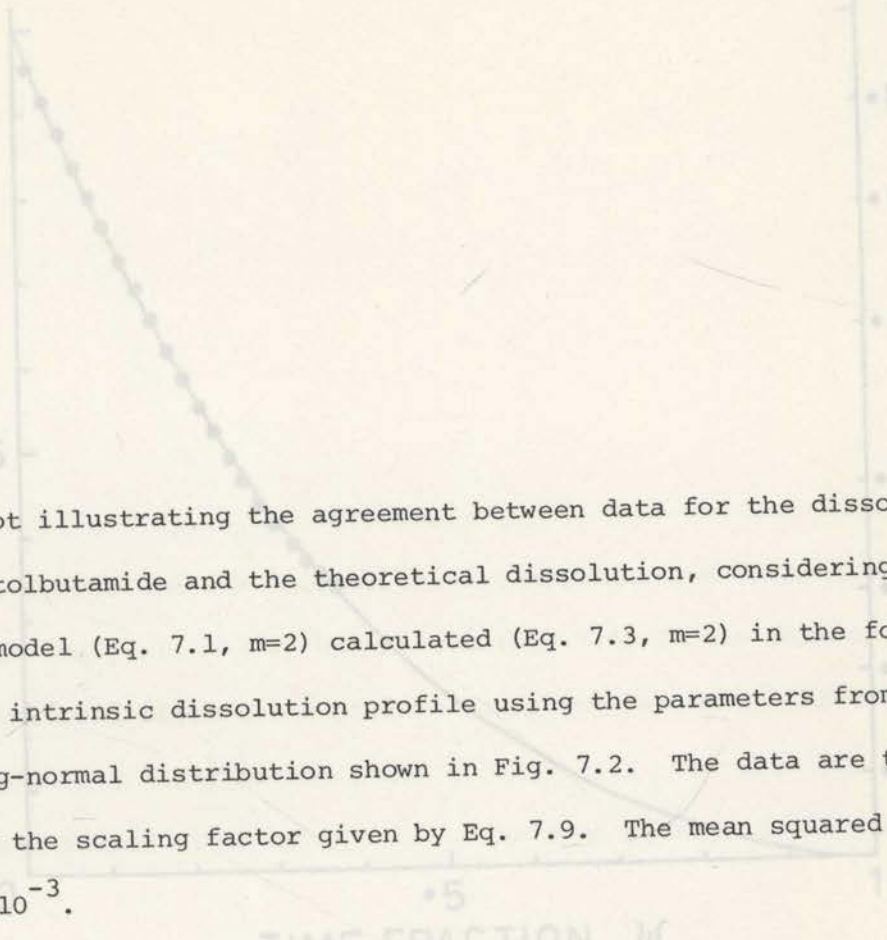
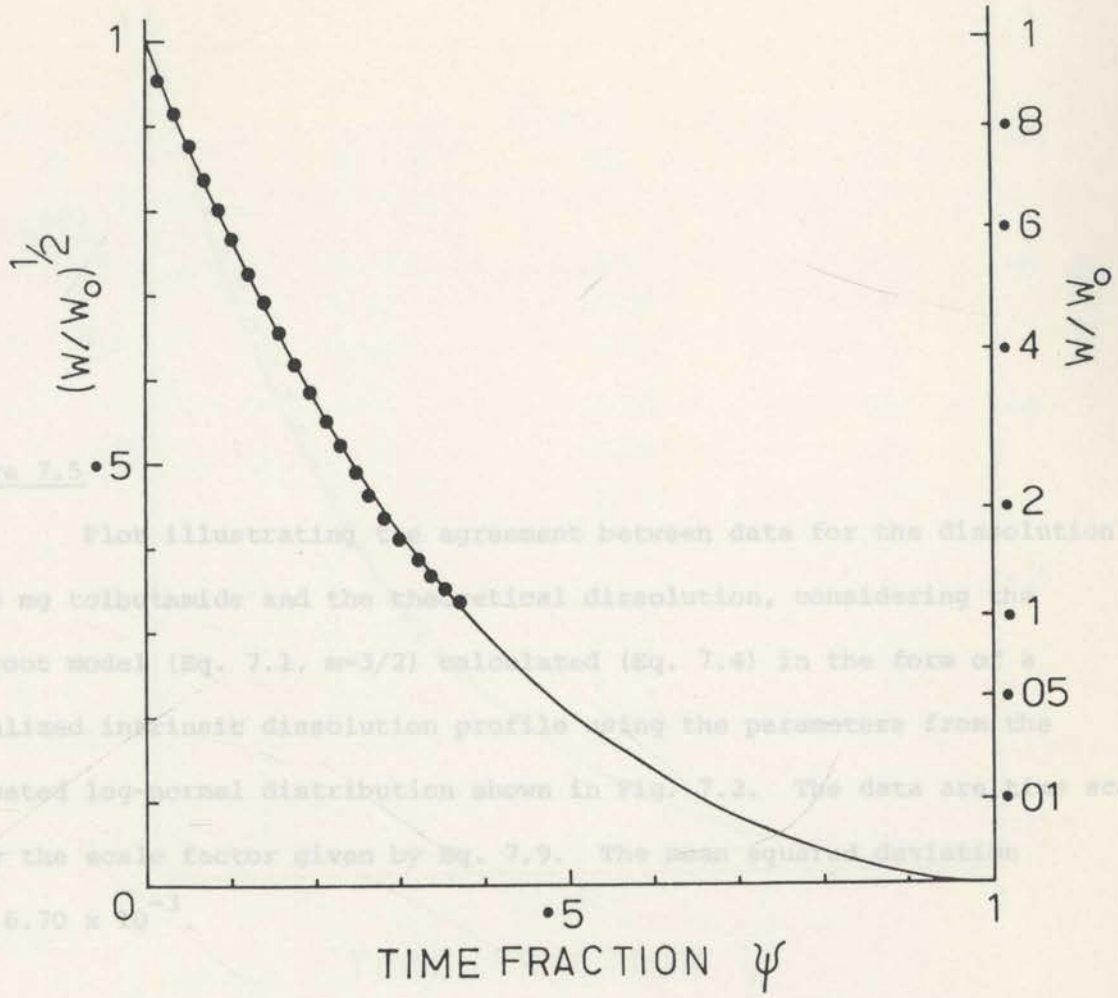


Figure 7.4

Plot illustrating the agreement between data for the dissolution of 15 mg of tolbutamide and the theoretical dissolution, considering the square root model (Eq. 7.1, m=2) calculated (Eq. 7.3, m=2) in the form of a normalized intrinsic dissolution profile using the parameters from the truncated log-normal distribution shown in Fig. 7.2. The data are time scaled using the scaling factor given by Eq. 7.9. The mean squared deviation $\overline{ss} = 1.92 \times 10^{-3}$.


1
•5
0
•5
1
$(W/W_o)^{1/2}$
1
•8
•6
•4
•2
•1
•05
•01
W/W_o
TIME FRACTION ψ

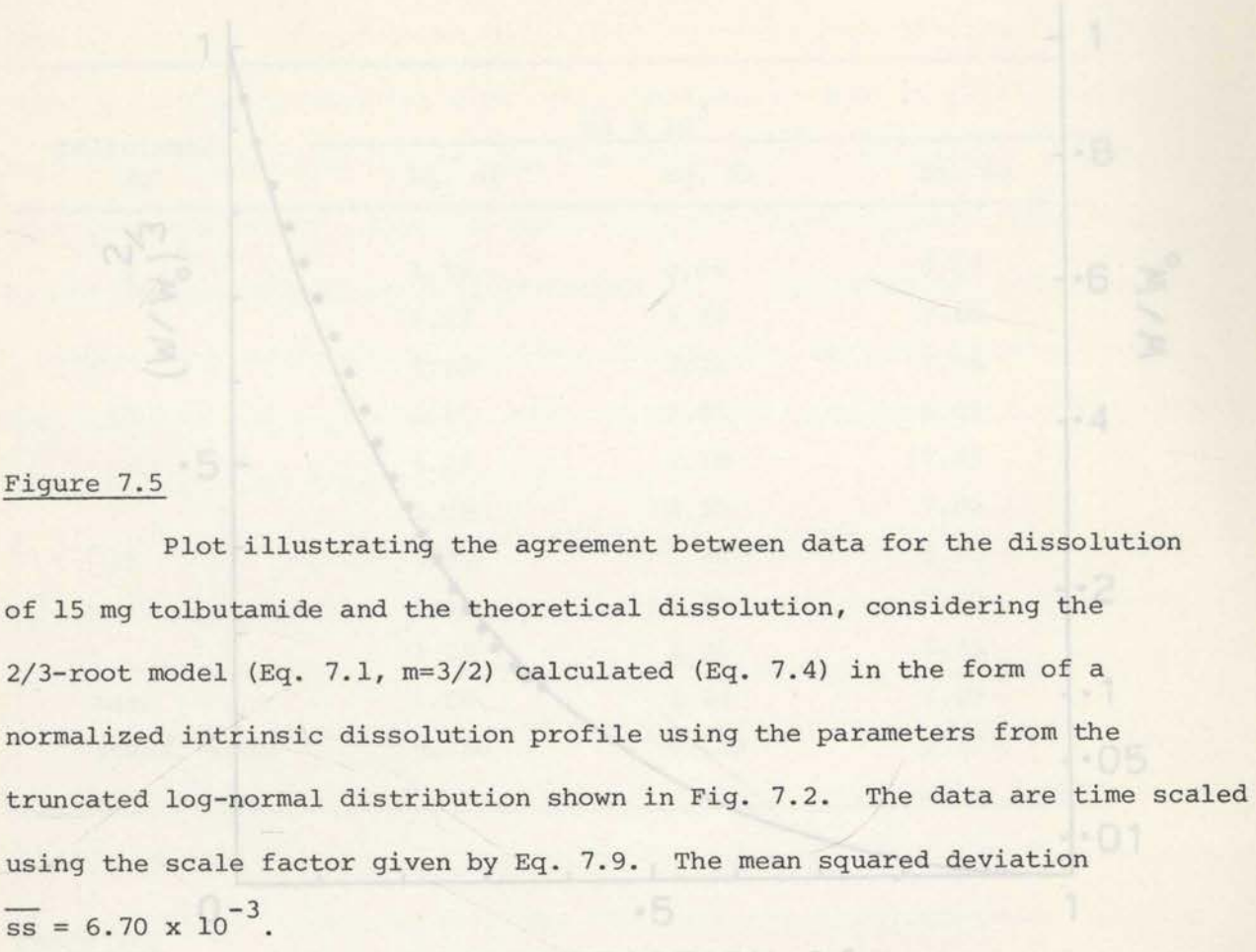
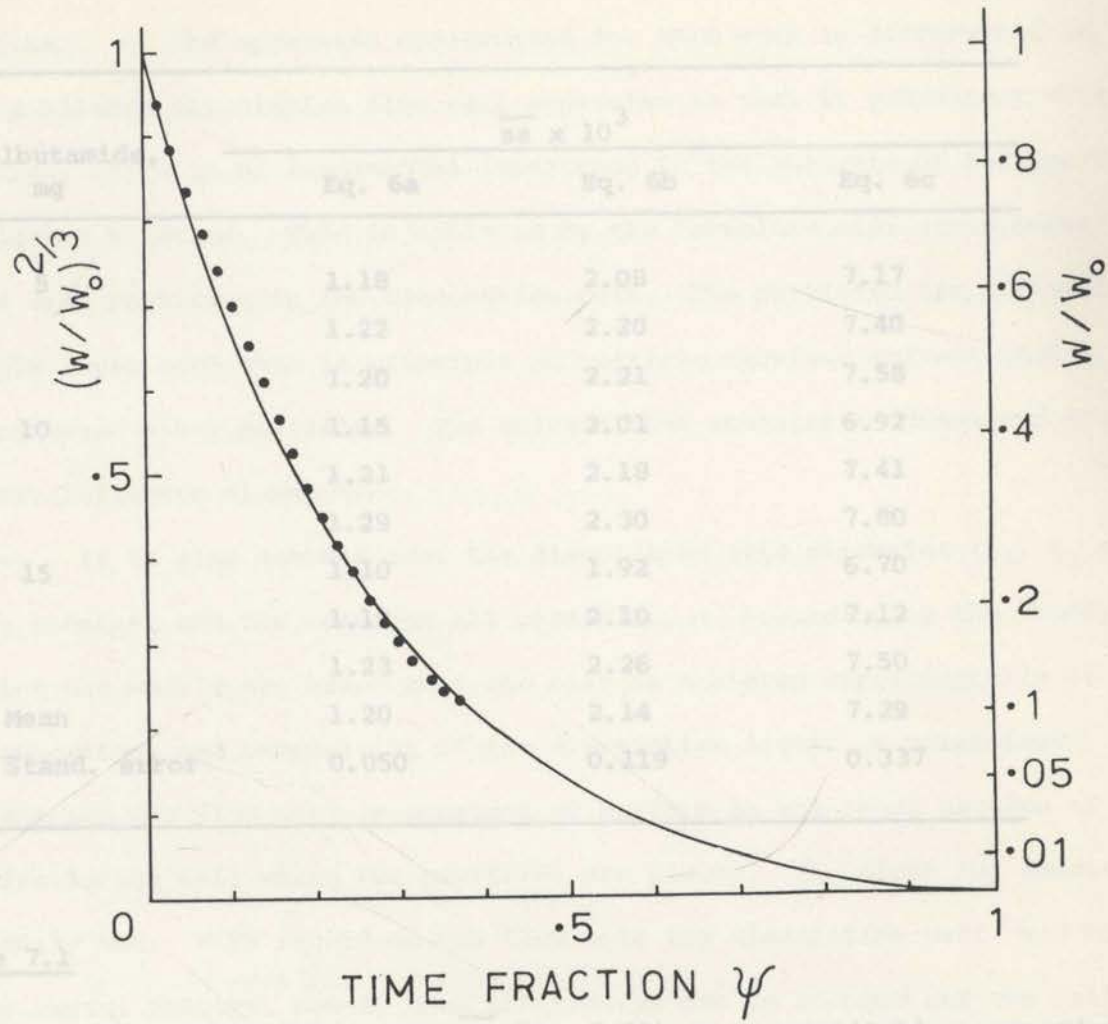


Figure 7.5

Plot illustrating the agreement between data for the dissolution of 15 mg tolbutamide and the theoretical dissolution, considering the 2/3-root model (Eq. 7.1, m=3/2) calculated (Eq. 7.4) in the form of a normalized intrinsic dissolution profile using the parameters from the truncated log-normal distribution shown in Fig. 7.2. The data are time scaled using the scale factor given by Eq. 7.9. The mean squared deviation $\overline{ss} = 6.70 \times 10^{-3}$.


1
·5
0
·5
1
$(W/W_o)^{2/3}$
1
·8
·6
·4
·2
·1
·05
·01
W/W_o
TIME FRACTION ψ

Tolbutamide, mg	$\overline{ss} \times 10^3$		
	Eq. 6a	Eq. 6b	Eq. 6c
5	1.18	2.08	7.17
	1.22	2.20	7.40
	1.20	2.21	7.58
10	1.15	2.01	6.92
	1.21	2.18	7.41
	1.29	2.30	7.80
15	1.10	1.92	6.70
	1.19	2.10	7.12
	1.23	2.26	7.50
Mean	1.20	2.14	7.29
Stand. error	0.050	0.119	0.337

Table 7.1

Mean square deviation, $\overline{ss}$ (Eq. 7.11) as a quantitative comparison of the fit of the three multiparticulate dissolution equations. Eqs. 7.3 (m=2,3) and Eq. 7.4 (based on the single-particle dissolution models of Eq. 7.1 (m=3,2, 3/2) to the data from the dissolution of various amounts of 60/85 mesh fraction of tolbutamide.

1. —— It is assumed that the particles dissolve independently of each other. - The apparatus constructed for this work is different from other published dissolution flow cell apparatus in that it guarantees this condition, which is of fundamental importance in the analysis of multiparticulate dissolution kinetics. This is achieved by the "absolute sink arrangement" of the drug particles in the dissolution cell. The particles are placed in a single layer such that in principle no particle receives solvent that has contacted other particles. The solvent thus contains no dissolved drug that may influence dissolution. -

2. —— It is also assumed that the dissolution rate parameter (k_1, k_2 or k_3) is constant and the same for all particles. - According to the theory on which the models are based this can only be achieved experimentally if the temperature and composition of the dissolution liquid is maintained constant and the flow rate is constant or uniform in the cross section of the dissolution cell where the particles are placed. The first two conditions are easily met. With regard to the flow rate the dissolution cell used has a very useful feature, namely that the process can be stopped and the cell rapidly disconnected allowing the particles to be inspected at any stage of the dissolution process. Such inspections showed (after microscopic measurements) uniform dissolution over the whole particle layer indicating a uniform flow rate. The fact that the particles in the dissolution cell can be inspected in this way makes it possible for the dissolution data to be combined with particle size measurements.

3. —— It is further assumed that the initial particle size (diameter) distribution can be approximated by a truncated log-normal distribution function. - Figure 7.2 shows that this is a good approximation for the 500 particles *measured*, however, it does not guarantee the correctness of the

assumption that this small sample represents the particle size distribution in the samples used for the dissolution tests, although the uniformity of the powder supported this assumption.

These investigations indicate that it is possible to describe mathematically, the dissolution of a multiparticulate system with a high degree of accuracy by considering both the particle size distribution effect and the particle shape effect discussed earlier. It is evident from the dissolution data obtained, that among the three models investigated, the cube root model describes the kinetics best.

It is possible that more complex and flexible models for single particle dissolution could describe the dissolution more adequately. The fact that the $\overline{ss}$ values for the cube root and the square root model are almost the same suggests a model with properties between these two. The Danckwerts model as discussed by Goyan (105) is given by:

$$-dw/dt = A((Dp)^{\frac{1}{2}} + D/a)C_a \qquad (7.12)$$

where w = weight undissolved

A = the surface area

D = the diffusion coefficient

p = a quantity related to stirring

a = the radius of the particle

C_a = the steady state concentration.

This model is very flexible. When $(Dp)^{\frac{1}{2}}$ predominates, the apparent model would be the cube root model. As the quantity D/a becomes more important, then the square root model will become the apparent model. Finally, as the quantity D/a predominates the "squared cube root model" will become the apparent model. However, the Danckwerts model, when applied to the log-normal case, results in a mathematical expression which is much more complex than Eqs.

7.3 and 7.4.

The fit of the dissolution data to the cube root model is excellent. Therefore, if an application of the Danckwerts model results in an even better fit, this improvement will likely be statistically insignificant considering the magnitude of the experimental errors. In such a case the Hixson-Crowell model should be preferred because of its simplicity.

CHAPTER 8

- FUNFIT - A TIME SHARING PROGRAM FOR GENERAL NONLINEAR REGRESSION AND CURVEFITTING

Many investigators must deal with the problems of evaluating how well one or more mathematical models describe a certain physical system or of obtaining estimates of the parameters in a particular model. If the mathematical model is nonlinear in its parameters and cannot be transformed to a linear form[1] it is necessary to use a computer to cope with the complexity of computations. Several computer programs are available for the treatment of nonlinear regression (123-132). Considerable time and effort is often spent in applying the more powerful and versatile of these programs and their complexity frequently inhibits the less experienced. Even for the experienced user their inflexible and strict input structure may often lead to errors that can be difficult to find because of the lack of specific error messages. The time spent on correcting such errors is of considerable inconvenience, particularly when the program used is not written for interactive time-sharing.

This chapter describes a powerful and versatile FORTRAN computer program, FUNFIT, for nonlinear regression and curve fitting that does not have the disadvantages described above. It is written for the *maximum convenience of the user* and utilizes the many advantages of interactive

---

1. The expression $y = ae^{bx}$ is linear in the parameter a and nonlinear in b. By transforming to $\ln y = \ln a + bx$ it is linear in ($\ln a$) and b. The expression $y = ae^{bx} + ce^{dx}$ is nonlinear in b and d and cannot be transformed to a linear form by standard means.

time-sharing. The input occurs essentially as a communication with the computer. Questions are asked about which particular data treatment is desired and instructions are given how to enter data. Every input is extensively checked for numerical, logical and typing errors, so these can be corrected immediately. A special command 'BACK' makes it easy to edit previous inputs so multiple runs under various conditions can be made quickly. The program offers an extensive analysis of residuals that most other programs neglect, and a lattice search to obtain suitable initial parameter estimates. A lattice search combined with contour maps makes it possible to investigate whether a better solution of the nonlinear regression problem may exist.

THEORY

The Central Limit Theorem of probability theory justifies the assumption that random errors in a set of observations are normally distributed. If the mathematical model is correct, the independent variable exact, and the errors are independent and normally distributed with zero mean and the same variance, then the method of least squares is the best choice for the estimation of the parameters because the estimates obtained will be maximum likelihood estimates.

For example the equation to be fitted to some observations might be:

$$y = p_1 e^{p_2 x} + p_3 e^{p_4 x} = f(\underline{P},x) \qquad (8.1)$$

where $\underline{P} = \left[p_1, p_2, p_3, p_4\right]^T$ is the parameter vector and x the independent variable. The problem is then to minimize the sum of the squared residuals, that is:

$$SS = \sum_{i=1}^{NOBS} e_i^2 = \sum_{i=1}^{NOBS} \left[y_i - f(\underline{P},x_i) \right]^2 \qquad (8.2)$$

where $\left[(y_i, x_i),\ i = 1,2,\ldots,NOBS\right]$ are the observations. There may also be reasons to weight the observations in a certain way. The weighted residual sum of squares is then defined as[2]

$$SS = \sum_{i=1}^{NOBS} e_{w,i}^2 = \sum_{i=1}^{NOBS} w_i \left[y_i - f(\underline{P}, x_i)\right]^2 \qquad (8.3)$$

where w_i is the weight of the i-th observation. Unweighted data are in fact data having the same weight i.e., $w_i = 1$. The relationship between weighted and unweighted residuals is

$$e_{w,i} = \sqrt{w_i}\, e_i \qquad (8.4)$$

The above example represents a simple case with only 1 independent variable, x. The function to be fitted could, in fact, have 2 independent variables i.e., $\underline{X} = \left[x_1, x_2\right]^T$. The observations are then $\left[(y_i, x_{1,i}, x_{2,i}),\ i = 1,2,\ldots, NOBS\right]$ (e.g. concentration, time temperature) and are represented in 3 dimensions. The problem is then no longer a least squares *curve*-fitting but a least squares *surface*-fitting. 'FUNFIT' can fit 'hypersurfaces' with up to 9 independent variables.

SIMULANTEOUS FITTING OF SEVERAL RESPONSE SYSTEMS

The system under investigation can sometimes be measured for more than one response or dependent variable, y. For example blood levels and urinary excretion of a drug could be measured in the same pharmacokinetic experiment or, in an experiment in chemical reaction kinetics more than one reaction product could be followed. In general, if there is a functional

---

2. In FUNFIT the weights are normalized i.e. scaled so that their sum is equal to the number of observations, NOBS. This is done to achieve a better comparison of weighted and unweighted sum-of-squares values.

expression for each kind of measured response and these expressions contain one or more *common* parameters it is likely that a simultaneous least squares fit of all the functions would provide more reliable parameter estimates than would be obtained by fitting each function individually. If, for example, 2 different kinds of response are measured in a system, the simultaneous least squares fitting problem is to minimize 'the sum of the sum of squared residuals' given by:

$$SS = \sum_{i=1}^{NOBS_1} w_{1,i}\left[y_{1,i} - f_1(\underline{P}_1, \underline{X}_{1,i})\right]^2 + \sum_{i=1}^{NOBS_2} w_{2,i}\left[y_{2,i} - f_2(\underline{P}_2, \underline{X}_{2,i})\right]^2 \qquad (8.5)$$

where the parameter vectors $\underline{P}_1$ and $\underline{P}_2$ contain one of more common parameters. The symbols are used in analogy with equation 8.3 and subscripts 1 and 2 denote response systems 1 and 2. Hence, functions f_1 and f_2 are to be fitted simultaneously to $NOBS_1$ observations from response system 1 and $NOBS_2$ observations from response system 2 respectively. It is possible using 'FUNFIT' to fit simultaneously up to 10 functions (response systems) each containing up to 20 parameters and 9 independent variables, with or without weighting of the single observations in each response system. The sum-of-squares function can therefore be summarized by a general extension of Eq. 8.5:

$$SS = \sum_{j=1}^{NFUNC} \sum_{i=1}^{NOBS_j} w_{j,i}\left[y_{j,i} - f_j(\underline{P}_j, \underline{X}_{j,i})\right]^2 \qquad (8.6)$$

The total number of observations is

$$NOBS = \sum_{j=1}^{NFUNC} NOBS_j \qquad (8.7)$$

where $NOBS_j$ is the number of observations for the j-th function.

WEIGHTING OF RESPONSE SYSTEMS

The problem of weighting is of particular importance in simultaneous fitting for two reasons: 1. There may be a large difference in the orders of magnitude of the values of the dependent variables in each response system. 2. The variances of the errors may differ considerably between the systems.

The effect of condition 1. can be reduced by the proper choice of units for the dependent variables. Equation 8.6 assumes that the residual variance between each individual response system is the same. However this is seldom the case (i.e. condition 2. applies) and each response system must be given its own weight. It is generally accepted that, in a statistical sense, the best weighting scheme is to make the weight of each observation inversely proportional to the variance of the error (as estimated by the residual variance). Therefore the weight of each response system should be made proportional to the reciprocal of its residual variance which is obtained when the system is fitted individually by least squares (i.e. not simultaneously with other systems). Therefore the weight for the j-th response system should be:

$$W_j = \frac{NOBS_j - NPAR_j}{\left[\sum_{i=1}^{NOBS_j} w_{j,i}\left[y_{n,i} - f_j(\underline{P}_j, \underline{X}_{j,i})\right]^2\right]_{min}} \qquad (8.8)$$

where $NPAR_j$ is the number of parameters in the function f_j and $(NOBS_j - NPAR_j)$ is the residual degrees of freedom. Equation 8.6 can therefore be written in an improved form:

$$SS = \sum_{j=1}^{NFUNC} W_j \sum_{i=1}^{NOBS_j} w_{j,i}\left[y_{j,i} - f_j(\underline{P}_j, \underline{X}_{j,i})\right]^2 \qquad (8.9)$$

where W_j signifies the weights of the j-th response system and $w_{j,i}$ the weights of the observations within that system. For example, if it is desired in a pharmacokinetic experiment to fit blood and urine data simultaneously, an individual fitting of each of the two systems should be done first. The weights to be used for each system in the simultaneous fitting are calculated as the residual degrees of freedom divided by the (weighted or unweighted) residual sum-of-squares obtained from the individual fittings.

Fitting implicit functions

It is necessary to apply a special approach if a function to be fitted is of implicit form; i.e., if the dependent variable, y, cannot be expressed explicitly as a function of the independent variable(s), $\underline{X}$ and the parameters, $\underline{P}$.

The function may be described by:

$$g(y, \underline{X}, \underline{P}) = 0 \qquad (8.10)$$

(e.g. $y^2 + X^{P1} + (X + y)^{P2} = 0$) from which y can be found by an iterative procedure only when numerical values of $\underline{X}$ and $\underline{P}$ are given. The calculated values of y are found by including in the user-supplied subroutine 'MODEL' a suitable algorithm for finding the root of g. This algorithm will serve the purpose of solving Eq. 8.10 for y so the subroutine indirectly defines the functional relationship between y and $\underline{X}, \underline{P}$.

MINIMIZATION

It should be noted that for a given set of observations the residual sum of squares, SS, depends only on the parameters because in Eq's

8.2, 8.6 and 8.9, the quantities y, $\underline{X}$, w, W, NFUNC and NOBS are all numbers that are given. These equations can therefore be written in a shorter form as

$$SS = SS(\underline{P}) \tag{8.11}$$

indicating that SS is a function of the parameters only. The least squares fitting and parameter estimation is then simply reduced to a function minimization problem, namely to find the particular values of the elements of the parameter vector, $\underline{P}$, that minimize SS.

If the sum-of-squares function is strictly convex in a specified convex parameter space then it will only have one minimum (133). This condition is guaranteed when fitting linear but not nonlinear functions for which there may exist more than one minimum. There is, in general, in the latter case no guarantee that the minimum found by any nonlinear regression program is the smallest, giving the best possible fit and parameter estimates. This problem can be illustrated in the case of two parameters where Eq. 8.11 describes a surface in 3 dimensions: Computer programs find the minimum by some iterative procedure and therefore require initial estimates of the parameters p_1 and p_2 together with their initial step sizes Δp_1 and Δp_2.[3] For example if the SS surface has two troughs and the initial starting point (p_1,p_2) lies on the slope of one of these and the same is the case for the points $(p_1+\Delta p_1,p_2)$, $(p_1,p_2+\Delta p_2)$ or $(p_1+\Delta p_1,p_2+\Delta p_2)$ then the strategy of the minimization algorithms employed is to 'run downhill' so that the parameter estimates in the next iteration step will usually be closer to the bottom of that trough while the other trough which may contain a smaller minimum, is overlooked. Mathematically this problem exists also for sum-of-

---

3. If the initial step sizes are not included in the input they are chosen in relation to the upper and lower limits given for the parameters in FUNFIT.

squares surfaces in more than 3 dimensions (hyper surfaces) but is then impossible to visualize.

The importance of good (i.e. close) initial parameter estimates is obvious. The choice of small initial parameter step sizes will, as seen above, result in an increased probability of overlooking a smaller minimum. Large initial step sizes require more iterations and computation time but reduce this probability because a larger part of the parameter space is evaluated.

Most minimization algorithms in nonlinear regression programs are based on the gradient method of Gauss-Newton (134). These are particularly efficient in finding a minimum for a sum-of-squares function but are more likely to overlook a better minimum than non-gradient methods that usually take longer to converge. The simplex method of Nelder and Mead (135) used in 'FUNFIT' appears to be a particularly suitable non-gradient method. It has statistical application, is rapidly convergent and, on the way to the minimum, covers a large section of the parameter space, reducing the probability of missing a global (smallest) minimum. It has proven to be a very robust method that will always find a minimum and further has the advantage over gradient methods that nonlinear parameter constraints can more easily be employed. The economic disadvantage of the somewhat slower minimization procedure is insignificant in most cases.

LATTICE SEARCH

The most reliable way to search for a minimum smaller than that found by some minimization procedure is to evaluate SS ($\underline{P}$) at each point in a grid or lattice that spans over the whole parameter space. The computation time for such a procedure increases exponentially and very rapidly with the number of parameters and divisions per parameter. For example a lattice

consisting of just 6 parameters with 10 divisions requires 10^6 SS-evaluations each of which requires NOBS function evaluations. If SS ($\underline{P}$) is a surface in 3 dimensions (2 parameter case) its approximate shape can be visualized from a contour plot constructed from the SS-values of the points in a grid of the two parameters. If there are more than two parameters a composite picture of the sum-of-squares surface can be built up by "slicing" the parameter space, that is by fixing all except two parameters at a time. Contour plots help greatly in finding a global minimum and considerably reduce the number of function evaluations that otherwise would be required in such a search.

CONVERGENCE CRITERION

The accuracy with which the exact position of the minimum of SS ($\underline{P}$) can be determined is always limited. After a certain number of iteration steps the SS-values of the points used in calculating the position of the next $\underline{P}$ estimate differ so little from each other that further iterations will introduce round-off errors. It is because of this and for reasons of economy, necessary to specify a stopping criterion for the minimization process. A necessary (but not sufficient) condition for $\underline{P}$ to be close to a minimum is that the SS-surface in the proximity of $\underline{P}$ is "sufficiently flat". The minimization process is often considered to have converged when the SS ($\underline{P}$) value does not change more than a certain amount from one iteration step to the next. This guarantees a certain degree of 'flatness' as long as there is a significant change in the parameter vector $\underline{P}$ between the iterations. However such a stopping criterion evaluates the 'flatness' in a somewhat imprecise way and since SS ($\underline{P}$) is not usually dimensionless, depends on the units used for the experimental data. The stopping criterion

used in 'FUNFIT' checks the 'flatness' in terms of the coefficient of variation (percent) of the SS-values of some points closely surrounding $\underline{P}$. The flatness is in this way defined more strictly and does not depend on the units used. Results can therefore be compared on a more consistent basis than in the former case.

The stopping criterion determines how 'accurately' the parameters are determined. Its value should be chosen in relation to the variability of the parameters. There is no reason to choose a very small stopping criterion to get very 'accurate' parameters if their values are statistically rather uncertain. The variability of the parameters is related to the 'variability' of the SS-surface in the region of the minimum because a flat surface allows the parameters to have a wide range of values resulting in only a small variation of the SS ($\underline{P}$) value. The stopping criterion used in 'FUNFIT' therefore has an advantage over other commonly used criteria because it is based on the 'variability' of the surface and hence determines the 'accuracy' of the parameters in relation to their variability. To avoid termination on a flat plateau of the SS-surface or near a 'saddle point' it must further be ensured that the surface around the point found at convergence is convex. In gradient methods this is often done by checking whether the matrix of the second partial derivatives (the Hessian matrix) is positive definite as is the case if all its eigenvalues are positive (133). In 'FUNFIT' the minimization process is allowed to continue a certain number of iteration steps after the 'flatness criterion' has been satisfied and the process is considered to have converged if the new parameter values and SS-values are sufficiently close to the previous values.

STATISTICAL ESTIMATION

For many applications it is important to get statistical estimates of the variability of the parameters obtained. As indicated above, this information is found in the shape of the SS-surface near the minimum. The complexity of the $SS(\underline{P})$ expression in the nonlinear case does not allow exact statistical estimates to be obtained. These are obtained instead by approximating the surface in the region of the minimum with a more simple surface for which exact, i.e. unbiased, statistical estimates can be calculated. The $SS(\underline{P})$ surface is approximated in the region of the final parameter estimates, $\hat{\underline{P}}$, by a quadratic surface from a Taylor series expansion where third- and higher order terms are neglected; i.e.,

$$SS(\underline{P}) \approx SS(\hat{\underline{P}}) + \nabla^T SS(\hat{\underline{P}})(\underline{P}-\hat{\underline{P}}) + \tfrac{1}{2}(P-\hat{P})\nabla^2 SS(\hat{P})(P-\hat{P}) \qquad (8.12)$$

where $\nabla^T = (\partial/\partial p_1 + \partial/\partial p_2 + .. + \partial/\partial p_n)$ and n is the number of parameters. The statistical information is found in the term $\nabla^2 SS(\hat{\underline{P}})$, that is the square matrix of the second partial derivatives of SS(P) evaluated at $\hat{\underline{P}}$, the so-called Hessian matrix:

$$H(\hat{\underline{P}}) = \nabla^2 SS(\hat{\underline{P}}) = \begin{bmatrix} \dfrac{\partial^2 SS(\hat{\underline{P}})}{\partial p_1^2} & \cdots & \dfrac{\partial^2 SS(\hat{\underline{P}})}{\partial p_1 \partial p_n} \\ \cdots\cdots\cdots & & \cdots\cdots\cdots \\ \dfrac{\partial^2 SS(\hat{\underline{P}})}{\partial p_1 \partial p_n} & & \dfrac{\partial^2 SS(\hat{\underline{P}})}{\partial p_n^2} \end{bmatrix} \qquad (8.13)$$

The sample variance-covariance matrix of the parameters, V, is the inverse of the Hessian matrix multiplied by the residual variance estimate:

$$V = \frac{SS(\hat{\underline{P}})}{NOBS-n} H^{-1}(\hat{\underline{P}}) \qquad (8.14)$$

It can be seen from this equation that in order to decrease the variances of the parameters, the number of observations, NOBS, should be relatively

large compared to the number of parameters, n. It is recommended that NOBS should not be less than about 3n in statistical estimations. The covariance between the i-th and the j-th parameter is the (i,j)-th element of the variance-covariance matrix:

$$COV(\hat{p}_i,\hat{p}_j) = v_{i,j} \tag{8.15}$$

and the variance of the i-th parameter is the i-th diagonal element:

$$var(\hat{p}_i) = COV(\hat{p}_i,\hat{p}_i) = v_{i,i} \tag{8.16}$$

The elements of the correlation matrix, $\rho_{i,j}$, are calculated from the variance-covariance matrix by:

$$\rho_{i,j} = \frac{COV(\hat{p}_i,\hat{p}_j)}{\sqrt{Var(\hat{p}_i)\ Var\ (\hat{p}_j)}} \tag{8.17}$$

The joint probability distribution of the estimated parameters is multivariate normal and is given by (136).

$$g(\underline{p}) = \frac{1}{(2\pi)^{n/2}\sqrt{|V|}} \exp\left[-\frac{(\underline{p}-\hat{\underline{p}})^T V^{-1} (\underline{p}-\hat{\underline{p}})}{2}\right] \tag{8.18}$$

ASSUMPTIONS MADE IN THE STATISTICAL EVALUATIONS

The reliability of the statistical estimates above depends on the following assumptions:

1. The independent variable(s) is without error.
2. The errors are independent, have zero mean and the same variance, σ^2, i.e.:

$$COV(\varepsilon_i,\varepsilon_j)_{i\neq j} = 0 \tag{8.19}$$

$$E(\varepsilon_i) = 0 \tag{8.20}$$

$$Var(\varepsilon_i) = 0 \tag{8.21}$$

For the F- test, t- test and confidence limits to be valid the errors must furthermore be normally distributed with zero mean and the same variance:

$$\varepsilon_i \sim N(0,\sigma^2) \qquad (8.22)$$

3. The mathematical model is correct.

4. Equation 8.12 is exact.

When conditions 1., 2. and 3. are satisfied the least squares estimate of $\underline{P}$ is also the maximum likelihood estimate of $\underline{P}$ because the likelihood function, L, can be written (137):

$$L(\underline{P},\sigma^2) = (2\pi\sigma^2)^{\frac{-n}{2}} \exp[-SS(\underline{P})/2\sigma^2] \qquad (8.23)$$

which is maximized when $SS(\underline{P})$ is minimized (σ^2 = constant). Therefore the least squares parameter estimates are unbiased under these 3 conditions.

Assumption 4. is only true when linear but not nonlinear functions are fitted. The most crucial point in nonlinear regression appears to be how good an approximation Eq. 8.12 is, because the statistical estimates (variances, covariances) are based on formulae and analysis from linear regression theory. The estimates are therefore biased and often called 'asymptotic estimates' because of the asymptotic property of the Taylor series expansion (Eq. 8.12). It has been found in simulation studies that the standard deviations of parameters in nonlinear regression can be two- to three-fold different from their true values (138). Parameter estimates are usually less biased because they are less sensitive to violation of the above conditions.

WEIGHTING

Where assumption 2. is not true because the error variance, σ^2, is not constant, it is recommended that weights inversely proportional to

the individual variances be used, according to the following weighting scheme:

$$w_i = \frac{1}{\sigma_i^2} , \qquad i = 1,2,.., NOBS \tag{8.24}$$

Thus if,

$$\varepsilon_i \sim N(o,\sigma_i^2) \tag{8.25}$$

then the weighted errors will fulfil condition 2. because

$$\varepsilon_{w,i} = w\,\varepsilon_i = \frac{\varepsilon_i}{\sigma_i} \sim N(0,1) \tag{8.26}$$

so that this weighting will provide improved estimates. The error variances, σ_i^2, can only be estimated by repeated experiments. However in some weighting situations a certain functional relationship is assumed between the error and the dependent or independent variable. For example in pharmacokinetics it is often assumed that the standard deviation of the errors is proportional to the plasma concentration, so the data are assigned weights inversely proportional to the square of the concentration. It is not valid statistically to use weights just to get a better fit. There must be a sound basis for the weighting scheme used.

CONFIDENCE REGIONS AND CONFIDENCE LIMITS

When assumptions 1. to 4. are true it can be shown that $SS(\hat{\underline{P}})$ is distributed as chi-square with (NOBS-n) degrees of freedom:

$$SS(\hat{\underline{P}}) \sim \sigma^2 \chi^2_{NOBS-n} \tag{8.27}$$

and

$$SS(\underline{P}) - SS(\hat{\underline{P}}) \sim \sigma^2 \chi^2_n \tag{8.28}$$

Both $SS(\underline{P}) - SS(\hat{\underline{P}})$ and $SS(\hat{\underline{P}})$ follow a χ^2 distribution and are independently distributed. The ratio

$$\frac{\left[SS(\underline{P}) - SS(\hat{\underline{P}})\right]/n}{SS(\hat{\underline{P}})/(NOBS-n)} \sim F(n,NOBS-n) \qquad (8.29)$$

is therefore distributed as an F-distribution with n and (NOBS-n) degrees of freedom respectively (136). The term $SS(\underline{P})$ can be isolated from this equation so that the critical points of the F-distribution provide the exact (1-α) 100% confidence region of the parameters:

$$SS(\underline{P}) = SS(\hat{\underline{P}}) \left[1 + \frac{n}{NOBS-n} F_{\alpha,n,NOBS-n} \right] \qquad (8.30)$$

The set of parameter values, $\underline{P}$, for which this equation is satisfied forms a closed contour line (two parameter case) or a contour surface inside which the probability of *simultaneously* finding the true (population) parameter values is (1-α) 100%. In the nonlinear case where assumption 4. is violated, the confidence contours are still exactly defined by Eq. 8.30 but the (1-α) 100% confidence level is only approximated.

When linear equations are fitted the confidence region is ellipsoidal in shape and given by:

$$(\underline{P}-\hat{\underline{P}})^T H^{-1} (\underline{P}-\hat{\underline{P}}) \leq \frac{n}{NOBS-n} SS(\hat{\underline{P}}) F_{\alpha,n,NOBS-n} \qquad (8.31)$$

Because the parameters are correlated it is not possible to define confidence limits for each individual parameter. Limits are however often defined in terms of tangent points of 'support planes' i.e. planes parallel to the parameter coordinate axis and tangent to the ellipsoidal region defined by Eq. 8.31. The approximate (1-α) 100% 'support plane' confidence limits of the parameters are given by (123)

$$\hat{p}_i \pm SD_i \sqrt{F_{\alpha,n,NOBS-n} \times n} \qquad (8.32)$$

where SD_i is the standard deviation of the i-th parameter.

Assuming no correlation among the parameters the 'uni-plane' confidence limits are given by (123)

$$\hat{p}_i \pm t_{\alpha,NOBS-n} \times SD_i \tag{8.33}$$

where $t_{\alpha,NOBS-n}$ is the t-statistic with probability level α and (NOBS-n) degrees of freedom. Confidence limits are sometimes desired for the ratio of two parameters, for example p_1/p_2. This can be calculated using the following expression (139):

$$\frac{\hat{p}_1 \hat{p}_2 - t^2_{\alpha,NOBS-n} COV(\hat{p}_1,\hat{p}_2) \pm t_{\alpha,NOBS-n} A}{\hat{p}_2^2 - t^2_{\alpha,NOBS-n} Var(\hat{p}_2)} \tag{8.34}$$

where

$$A = \hat{p}_1^2 Var(\hat{p}_2) + \hat{p}_2^2 Var(\hat{p}_1) - 2\hat{p}_1\hat{p}_2 COV(\hat{p}_1,\hat{p}_2) - t^2_{\alpha,NOBS-n}\left[Var(\hat{p}_1) - \frac{COV(\hat{p}_1,\hat{p}_2)^2}{Var(\hat{p}_2)}\right] \tag{8.35}$$

STANDARD DEVIATION OF A FUNCTION OF THE PARAMETERS

Use of variance-covariance matrix

The variance-covariance matrix Eq. 8.14 that is printed out in most computer programs allows the standard deviation to be calculated for any quantity that is expressed as a function of one or more of the parameters. Let such a quantity be denoted

$$g = g(\hat{p}_1,\hat{p}_2,\ldots\hat{p}_k) \tag{8.36}$$

then its standard deviation is given by the following formula which is based on a Taylor series expansion (140)

$$(SD_g)^2 \approx \sum_{i,j=1}^{k} \frac{\partial g}{\partial p_i} \frac{\partial g}{\partial p_j} COV(\hat{p}_i, \hat{p}_j) \qquad (8.37)$$

where the summation extends over all k^2 choices of the two indices i and j. The formula is exact where g is linear in the parameters, but is otherwise an approximation which is sufficiently accurate provided the coefficient of variation of the parameters does not exceed about 20%.

Three special cases are of interest: 1. If g is a function of one parameter p_i only, then Eq. 8.37 reduces to

$$SD_g \approx \left|\frac{dg}{dp_i}\right| SD_i \qquad (8.38)$$

In particular if g is linear and of the form $a\hat{p} + b$ (a and b are constants) then $SD_g = aSD_p$ exactly. 2. If g is a linear function of the form:

$$g = \sum_{i=1}^{k} a_i \hat{p}_i \qquad (8.39)$$

then Eq. 8.37 becomes

$$(SD_g)^2 = \sum_{i=1}^{k} a_i^2 Var(\hat{p}_i) + \sum_{i \neq j}^{k} a_i a_j COV(\hat{p}_i, \hat{p}_j) \qquad (8.40)$$

3. If g is a ration of two parameters, for example $g = \hat{p}_1/\hat{p}_2$, Eq. 8.37 becomes

$$(SD_g)^2 \approx \left(\frac{\partial g}{\partial p_1}\right)^2 Var(\hat{p}_1) + \left(\frac{\partial g}{\partial p_2}\right)^2 Var(\hat{p}_2) + 2 \frac{\partial g}{\partial p_1} \frac{\partial g}{\partial p_2} COV(\hat{p}_1, \hat{p}_2) \qquad (8.41)$$

which in this case gives

$$(SD_g)^2 \approx \frac{Var(\hat{p}_1)}{\hat{p}_2^2} + \frac{\hat{p}_1^2 Var(\hat{p}_2)}{\hat{p}_2^4} - \frac{2\hat{p}_1 COV(\hat{p}_1, \hat{p}_2)}{\hat{p}_2^3} \qquad (8.42)$$

Use of the transformation technique

Equation 8.37 often leads to rather complex expressions if g is a nonlinear function of more than a few parameters. However, this problem can be avoided in most cases and the standard deviation of g obtained directly without the truncation errors that may be introduced when Eq. 8.37 is used. This can be done if, as is usually the case, Eq. 8.36 can be solved explicitly or numerically for one of its parameters, so that this parameter, $\hat{p}_i$, can be expressed as a function of g and the remaining parameters, i.e.

$$\hat{p}_i = h(g,\hat{p}_1,\hat{p}_2 \cdots ,\hat{p}_{k-1}) \qquad (8.43)$$

The function to be fitted to the data points can then be transformed so that it contains g as a parameter essentially replacing p_i. To evaluate the function, p_i is then simply calculated using Eq. 8.43 (where g and the (k-1) parameters are among the k input parameters). Initial estimates must be obtained for g since it acts as an input parameter. This is done by calculating it from Eq. 8.36 using the initial estimates of the k parameters or by using the final least squares estimates obtained from curve-fitting based on the original untransformed function.

This transformation technique is of considerable importance and highly recommended. For example, it allows the standard deviation to be calculated for essentially any pharmacokinetic quantity that is expressed in terms of the macro- or microparameters in the model.

EXAMINATION OF RESIDUALS

It is of fundamental importance to analyse the pattern of the residuals because this is essentially the only way of examining whether the

basic assumptions behind the nonlinear estimation are violated. Most general nonlinear programs seem to ignore this. Perhaps the best way to examine the residuals is to plot them against the independent and dependent variables (141). Significant systematic deviations can be visualized in this way. The assessment is somewhat complicated by the fact that there will always be a correlation between the residuals because NOBS residuals are only associated with (NOBS-n) degrees of freedom. 'FUNFIT' includes the Durbin-Watson statistic, given by (142):

$$d = \frac{\sum_{i=2}^{NOBS} (e_i - e_{i-1})^2}{\sum_{i=1}^{NOBS} e_i^2} \qquad (8.44)$$

to test for excessive serial correlation (systematic deviation) among the residuals. The statistic is compared with tabulated critical points at a given significance level.

The fundamental assumption of random errors is also tested using two nonparametric tests that will be called the 'group' statistic and the 'number' statistic. The 'group' statistic is based on an analysis of groups (runs) of residuals of equal sign. For example the sequence of residuals (+++) (--) (+) (--) (+) forms r = 5 groups. The least number of residuals with the same sign is L = 4. If NOBS residuals, with equal probability of being negative and positive, form a sequence with r groups, then the probability of getting $\leq$ r groups is (143)

$$P(\leq r) = \binom{NOBS}{L}^{-1} \sum_{i=2}^{r} f_r \qquad (8.45)$$

where $$f_r = 2\binom{L-1}{\frac{i}{2}-1}\binom{NOBS-L-1}{\frac{i}{2}-1} \text{ when i is even} \qquad (8.46)$$

and
$$f_r = \binom{L-1}{\frac{i-1}{2}}\binom{NOBS-L-1}{\frac{i-3}{2}} + \binom{L-1}{\frac{i-3}{2}}\binom{NOBS-L-1}{\frac{i-1}{2}} \qquad (8.47)$$

when i is odd. The 'number' statistic is defined as the probability of getting L or less residuals of same sign and is given by

$$P(\leq L) = 2^{-NOBS} \sum_{i=1}^{L+1} \binom{NOBS}{i-1} \qquad (8.48)$$

where the large brackets denote binomial coefficients i.e. $\binom{x}{y} = x!/(y!(x-y)!)$.

If $P(\leq r) < 0.05$ or $P(\leq L) < 0.05$ then the hypothesis that the residuals are random should be rejected (with α– error < 0.05). This can be used to reject the basic assumption of normal errors provided the model is correct. However, it can also mean that the mathematical model is wrong if it is assumed that the errors are in fact random with zero mean.

OUTLIERS

Several repeated experiments will be required to examine whether the errors are all normally distributed. When only one experiment is available a test must rely on the fact that if all the errors are deviates from the same normal distribution, they will then collectively be normally distributed. If the mathematical model is correct and it is assumed that $E(\varepsilon_i) = 0$ and $Var(\varepsilon_i) = \sigma^2$ then the residuals are unbiased estimates of the errors, i.e. $E(e_i) = \varepsilon_i$ and the residual variance given by

$$s^2 = \frac{\sum_{i=1}^{NOBS} (e_i - \bar{e})^2}{NOBS-n} \approx \frac{\sum_{i=1}^{NOBS} e_i^2}{NOBS-n} = \frac{SS(\underline{\hat{P}})}{NOBS-n} \qquad (8.49)$$

estimates σ^2. Further, if it is assumed that $\varepsilon_i \sim N(0,\sigma^2)$ it follows that $\varepsilon_i/\sigma \sim N(0,1)$. Now, if the 4 basic assumptions mentioned previously are true then e_i/s, the unit normal deviate form of the residuals, estimates ε_i/σ and is N(0,1).

The normal deviate form of the residuals can be used to test for outliers, that is, data-points which, in a statistical sense, are not typical of the rest of the data. The i-th observation is an outlier if the normal deviate e_i/s falls in the critical region of the N(0,1) distribution, that is if $|e_i/s| > 1.96$ ($\alpha<0.05$). If NOBS-n is smaller than 30 the normal deviate should be compared with the critical values of the t- distribution with (NOBS-n) degrees of freedom.

Outliers should be submitted to particularly careful examination since they may provide information of vital interest. They should only be rejected if they are caused by errors in the recording or the experimental technique (144). Outliers may also occur if the mathematical model is incorrect.

If there are many observations it may be useful to construct a half-normal plot of the residuals. The linearity of such a plot gives information about outliers and how normally the residuals are distributed (141). The above analysis extends also to weighted residuals.

COMPARISON OF PARAMETERS AND MODELS

It can be of interest to investigate whether there is a significant difference in a certain parameter, $\hat{p}_i$, between two experiments (1 and 2).

If it is assumed the parameter is normally distributed a t-test can then be applied.

$$t = \frac{\hat{p}_{i,1} - \hat{p}_{i,2}}{\sqrt{Var(\hat{p}_{i,1}) + Var(\hat{p}_{i,2})}} \qquad (8.50)$$

If t falls in the critical region $|t| > t_{\alpha/2}$ of a t- distribution with $(NOBS_1 + NOBS_2 - 2n)$ degrees of freedom, the null hypothesis that $\hat{p}_{i,1}$ and $\hat{p}_{i,2}$ are the same should be rejected at significance level α.

For any particular system there may be more than one mathematical model which could be appropriate. It is generally accepted that if two models give approximately "equal" fits, the simpler model should be chosen unless the more complex model can be justified on other criteria. At present there does not seem to be any simple, rigorously based statistical test which makes it possible to distinguish between alternative models. The following test, based on a comparison of residual variances, is no exception. It can however be used as a guide in the absence of other methods.

Let $SS(\hat{\underline{P}}_{(1)})/(NOBS_1-n_1) > SS(\hat{\underline{P}}_{(2)})/(NOBS_2-n_2)$ be the residual variances of models 1 and 2 respectively. If it is assumed $SS(\hat{\underline{P}}_{(1)})$ and $SS(\hat{\underline{P}}_{(2)})$ are independent random variables having chi-square distributions with $(NOBS_1-n_1)$ and $(NOBS_2-n_2)$ degrees of freedom, the variance ratio given by:

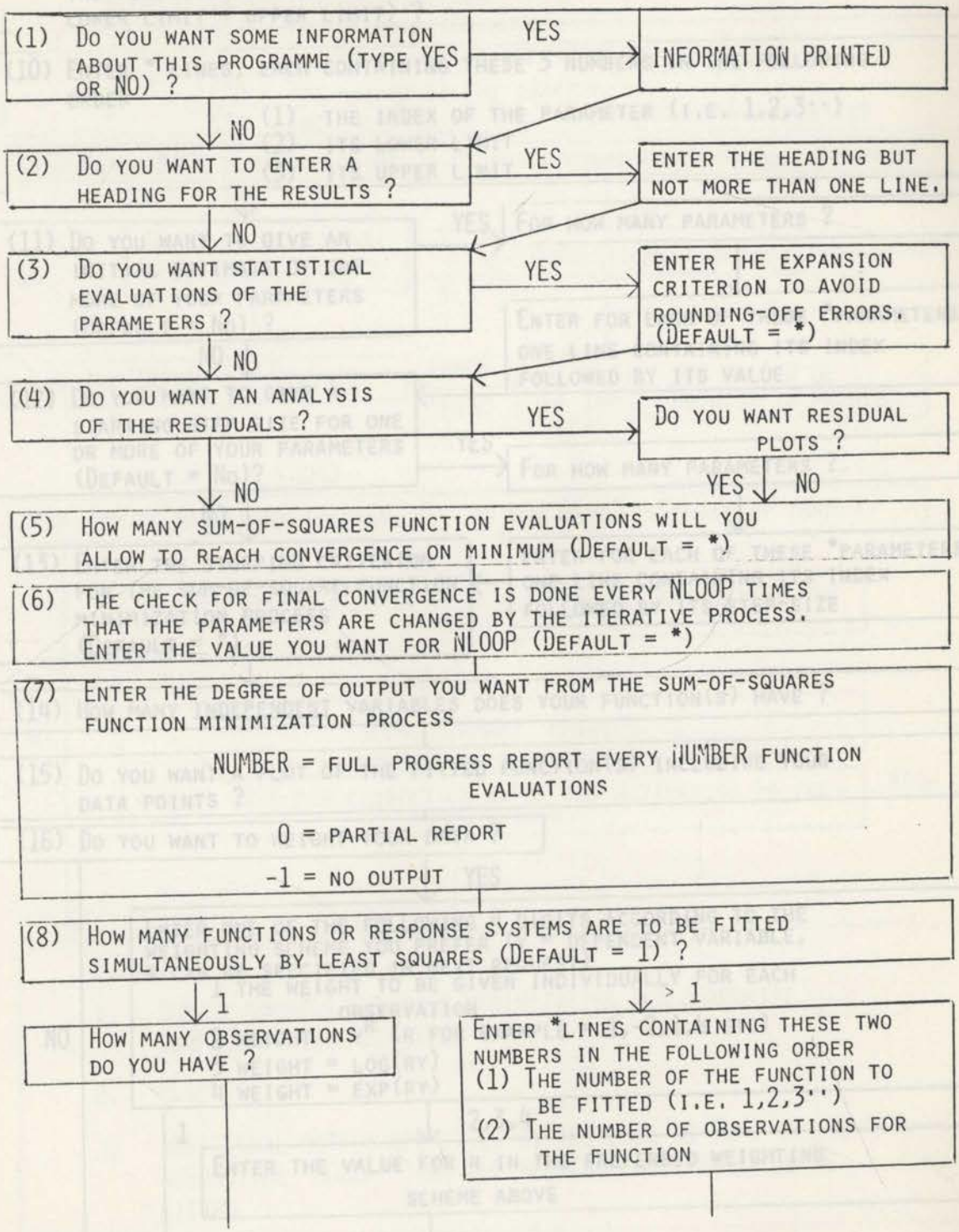
$$F = \frac{SS(\hat{\underline{P}}_{(1)})}{SS(\hat{\underline{P}}_{(2)})} \frac{NOBS_2-n_2}{NOBS_1-n_1} \qquad (8.51)$$

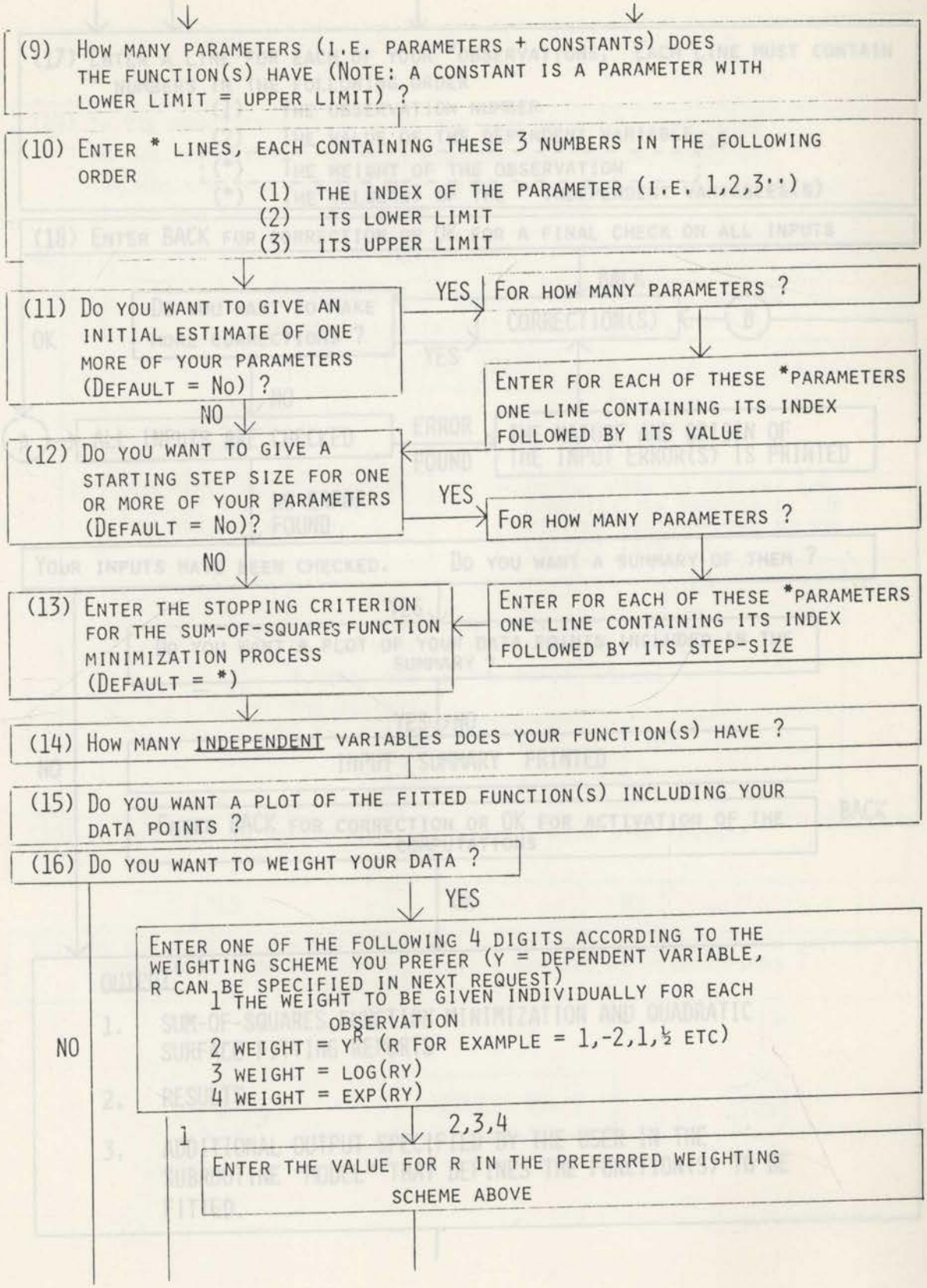
follows an F- distribution with $(NOBS_1-n_1)$ and $(NOBS_2-n_2)$ degrees of freedom. Therefore, if $F > F_{\alpha,NOBS_1-n_1,NOBS_2-n_2}$, the null hypothesis that the residual variances are identical should be rejected. The hypothesis that the residual variance of model 2 is significantly smaller than that of model 1, (i.e. that model 2 fits the experimental data better than model 1) can therefore be accepted. If F does not exceed the critical value, further investigations are required or the simpler of the two models should be chosen.

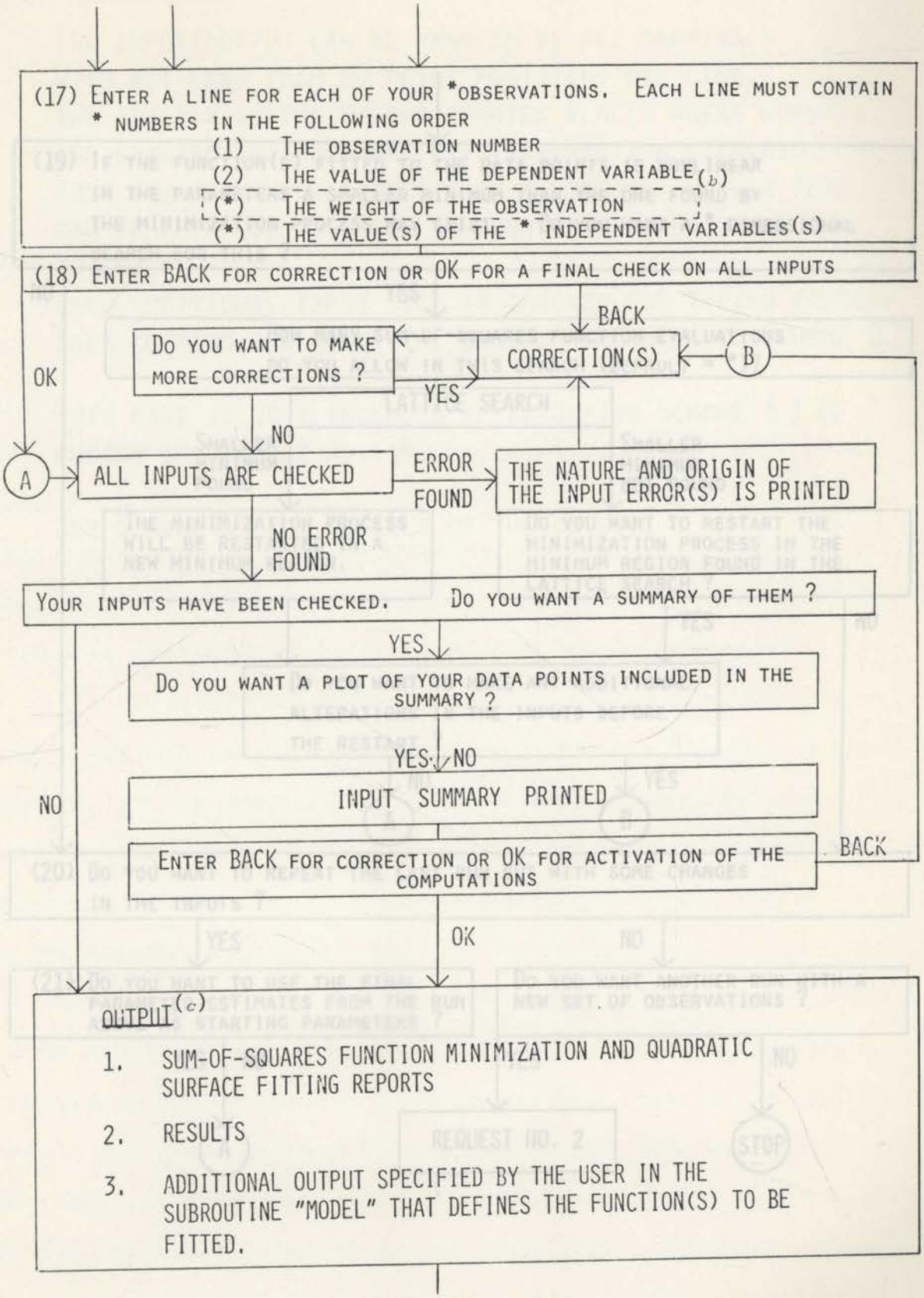
INPUT TO FUNFIT

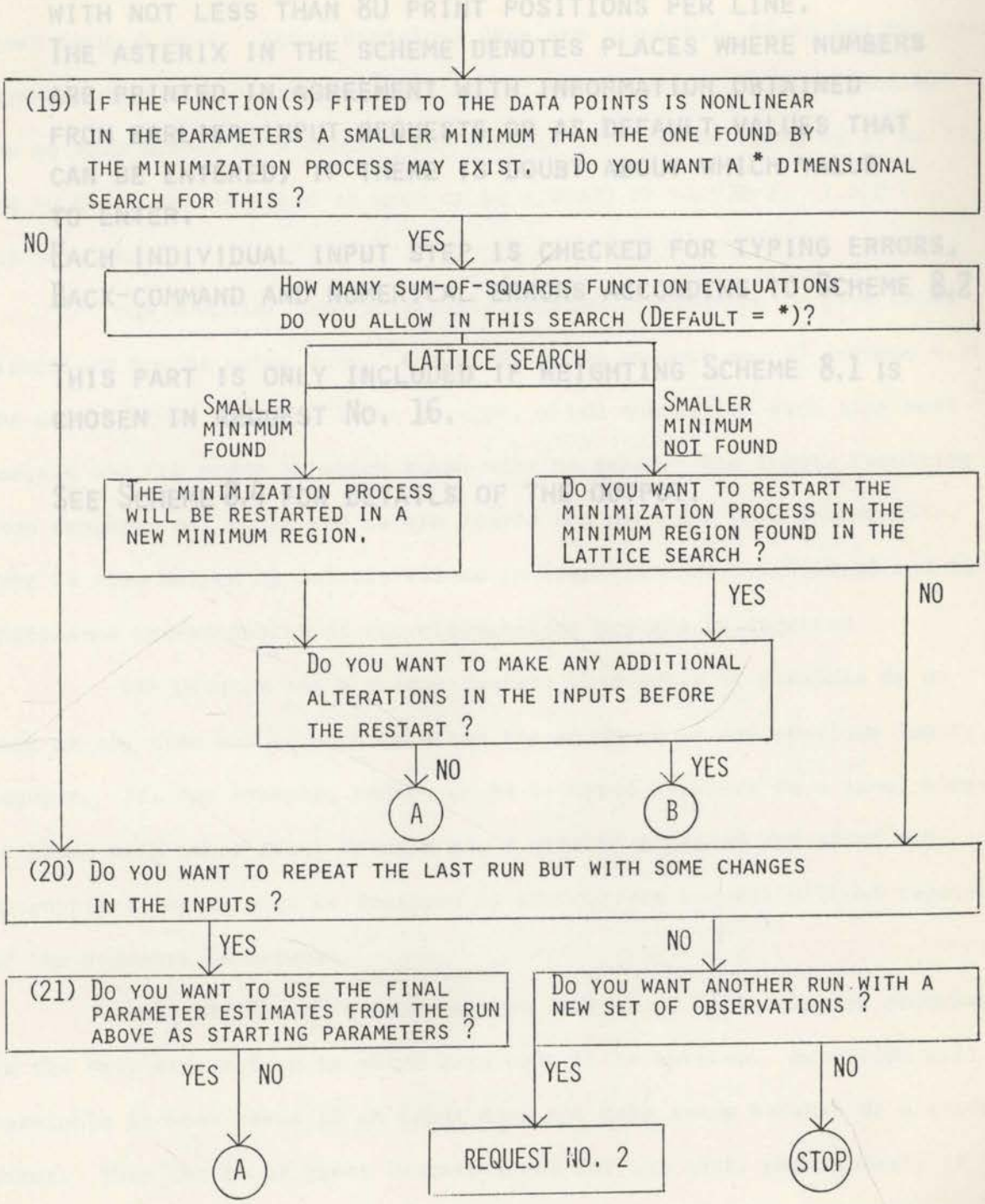
The input is provided in a communication between the user and the computer (scheme 8.1). Questions are asked about which data treatment is wanted and depending on the answers, the next questions are given in relation

SCHEME I USER - COMPUTER INTERACTION DURING EXECUTION OF THE TIME-SHARING PROGRAMME "FUNFIT" FOR GENERAL NONLINEAR REGRESSION AND CURVE FITTING[a]




(9) How many parameters (i.e. parameters + constants) does the function(s) have (Note: a constant is a parameter with lower limit = upper limit) ?
(10) Enter * lines, each containing these 3 numbers in the following order
(1) the index of the parameter (i.e. 1,2,3··)
(2) its lower limit
(3) its upper limit
(11) Do you want to give an initial estimate of one more of your parameters (Default = No) ?
YES
For how many parameters ?
Enter for each of these *parameters one line containing its index followed by its value
NO
(12) Do you want to give a starting step size for one or more of your parameters (Default = No)?
YES
For how many parameters ?
Enter for each of these *parameters one line containing its index followed by its step-size
NO
(13) Enter the stopping criterion for the sum-of-squares function minimization process (Default = *)
(14) How many INDEPENDENT variables does your function(s) have ?
(15) Do you want a plot of the fitted function(s) including your data points ?
(16) Do you want to weight your data ?
YES
Enter one of the following 4 digits according to the weighting scheme you prefer (Y = dependent variable, R can be specified in next request)
1 the weight to be given individually for each observation
2 weight = Y^R (R for example = 1,-2,1,½ etc)
3 weight = LOG(RY)
4 weight = EXP(RY)
NO
1
2,3,4
Enter the value for R in the preferred weighting scheme above


(17) ENTER A LINE FOR EACH OF YOUR *OBSERVATIONS. EACH LINE MUST CONTAIN * NUMBERS IN THE FOLLOWING ORDER
(1) THE OBSERVATION NUMBER
(2) THE VALUE OF THE DEPENDENT VARIABLE(b)
(*) THE WEIGHT OF THE OBSERVATION
(*) THE VALUE(S) OF THE * INDEPENDENT VARIABLES(S)
(18) ENTER BACK FOR CORRECTION OR OK FOR A FINAL CHECK ON ALL INPUTS
BACK
OK
DO YOU WANT TO MAKE MORE CORRECTIONS ?
CORRECTION(S)
B
YES
NO
A
ALL INPUTS ARE CHECKED
ERROR FOUND
THE NATURE AND ORIGIN OF THE INPUT ERROR(S) IS PRINTED
NO ERROR FOUND
YOUR INPUTS HAVE BEEN CHECKED. DO YOU WANT A SUMMARY OF THEM ?
YES
DO YOU WANT A PLOT OF YOUR DATA POINTS INCLUDED IN THE SUMMARY ?
YES
NO
NO
INPUT SUMMARY PRINTED
ENTER BACK FOR CORRECTION OR OK FOR ACTIVATION OF THE COMPUTATIONS
BACK
OK
OUTPUT(c)
1. SUM-OF-SQUARES FUNCTION MINIMIZATION AND QUADRATIC SURFACE FITTING REPORTS
2. RESULTS
3. ADDITIONAL OUTPUT SPECIFIED BY THE USER IN THE SUBROUTINE "MODEL" THAT DEFINES THE FUNCTION(S) TO BE FITTED.


(19) IF THE FUNCTION(S) FITTED TO THE DATA POINTS IS NONLINEAR IN THE PARAMETERS A SMALLER MINIMUM THAN THE ONE FOUND BY THE MINIMIZATION PROCESS MAY EXIST. DO YOU WANT A * DIMENSIONAL SEARCH FOR THIS ?
NO
YES
HOW MANY SUM-OF-SQUARES FUNCTION EVALUATIONS DO YOU ALLOW IN THIS SEARCH (DEFAULT = *)?
LATTICE SEARCH
SMALLER MINIMUM FOUND
SMALLER MINIMUM NOT FOUND
THE MINIMIZATION PROCESS WILL BE RESTARTED IN A NEW MINIMUM REGION.
DO YOU WANT TO RESTART THE MINIMIZATION PROCESS IN THE MINIMUM REGION FOUND IN THE LATTICE SEARCH ?
YES
NO
DO YOU WANT TO MAKE ANY ADDITIONAL ALTERATIONS IN THE INPUTS BEFORE THE RESTART ?
NO
YES
A
B
(20) DO YOU WANT TO REPEAT THE LAST RUN BUT WITH SOME CHANGES IN THE INPUTS ?
YES
NO
(21) DO YOU WANT TO USE THE FINAL PARAMETER ESTIMATES FROM THE RUN ABOVE AS STARTING PARAMETERS ?
DO YOU WANT ANOTHER RUN WITH A NEW SET OF OBSERVATIONS ?
YES
NO
YES
NO
A
REQUEST NO. 2
STOP

(a) THE INPUT/OUTPUT CAN BE HANDLED BY ALL TERMINALS WITH NOT LESS THAN 80 PRINT POSITIONS PER LINE.
THE ASTERIX IN THE SCHEME DENOTES PLACES WHERE NUMBERS ARE PRINTED IN AGREEMENT WITH INFORMATION OBTAINED FROM EARLIER INPUT REQUESTS OR AS DEFAULT VALUES THAT CAN BE ENTERED, IF THERE IS DOUBT ABOUT WHICH VALUE TO ENTER.
EACH INDIVIDUAL INPUT STEP IS CHECKED FOR TYPING ERRORS, BACK-COMMAND AND NUMERICAL ERRORS ACCORDING TO SCHEME 8.2

(b) THIS PART IS ONLY INCLUDED IF WEIGHTING SCHEME 8.1 IS CHOSEN IN REQUEST NO. 16.

(c) SEE SCHEME 8.4 FOR DETAILS OF THE OUTPUT.

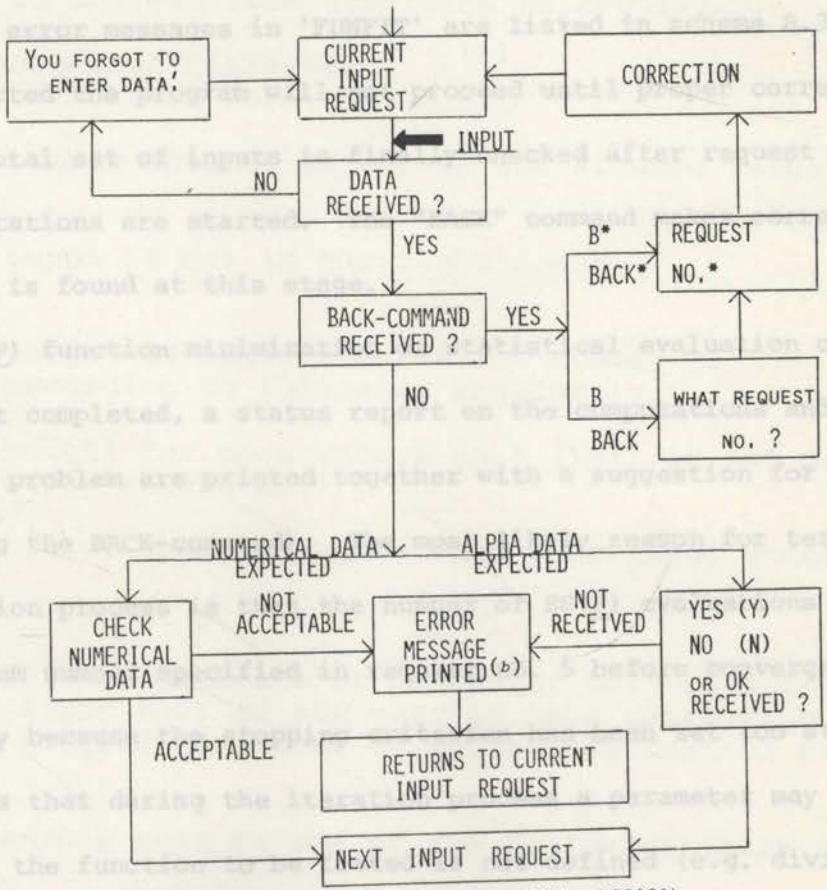
to the particular treatment chosen. The answers can be YES or NO or, in short form, Y or N. Where numerical data are requested, these can be typed anywhere on a line in integer, decimal or exponential form separated by one or more blanks or a comma. For example 17.0 can be typed as 17 or 17. and -0.00153 can be typed as such or as -.00153 or -1.53E-3, -1.53E-03, -153E-5, etc.

The computer uses previously entered information to give exact directions how to enter data. For example, in request no. 17 (scheme 8.1), the user is told how many lines to type, which quantities each line must contain and the order in which these must be typed. The inputs resulting from requests no. 8, 14 and 16 are stored and used for this instruction. The user is also helped by default values in requests no.3,5,6,8,12,13 and 19 where some understanding of the minimization process is required.

The program has a unique feature that makes it possible to go back at any time and correct or alter the input under any previous input request. If, for example, BACK4 or B4 is typed anywhere on a line, alone or along with other data, request no. 4 will be repeated and after the appropriate input, will be followed by the current request without repetition of the requests in between.

Possibly the most inconvenient feature of most computer programs is the very strict form in which data have to be entered. Execution will terminate in most cases if an input does not make sense because of a typing error. This can be of great inconvenience for the user, particularly if many data have to be entered. 'FUNFIT' does not have this disadvantage. *It will not terminate for any kind of typing error.* Instead every input is checked in several ways (scheme 8.2). Typing errors, logical errors and numerical errors are all detected. It is, for example, not uncommon to

SCHEME 8.2 STRUCTURE OF THE DATA CHECK PERFORMED AT EACH INDIVIDUAL INPUT STEP IN SCHEME 8.1 (a)



(a) THE INPUT DATA ARE CHECKED FOR NUMERICAL, LOGICAL AND TYPING ERRORS. IF AN ERROR IS MADE IT IS ESSENTIALLY IMPOSSIBLE TO PROCEED BEFORE IT HAS BEEN PROPERLY CORRECTED. ALL INPUT DATA ARE FURTHERMORE CHECKED COLLECTIVELY AFTER REQUEST No.18 IN SCHEME 8.1 SUCCESSFUL RUNS SHOULD IN THIS WAY BE GUARANTEED IN MOST CASES.

(b) SCHEME 8.3 GIVES EXAMPLES OF SOME OF THE MANY INPUT ERROR MESSAGES IN FUNFIT.

type a comma in place of a decimal point or an "0" instead of a zero; such errors are clearly indicated by the program and the last input line is repeated for immediate correction. An error message is also printed so corrections can be made if an input does not agree with an earlier input. Some of the many error messages in 'FUNFIT' are listed in scheme 8.3. If an error is detected the program will not proceed until proper corrections are made. The total set of inputs is finally checked after request no. 18 before the computations are started. The "BACK" command makes corrections easy if an error is found at this stage.

If SS($\underline{P}$) function minimization or statistical evaluation of the parameters is not completed, a status report on the computations and a diagnosis of the problem are printed together with a suggestion for a suitable correction (using the BACK-command). The most likely reason for termination of the minimization process is that the number of SS($\underline{P}$) evaluations may exceed the maximum number specified in request no. 5 before convergence is reached, probably because the stopping criterion has been set too strictly. Another reason is that during the iteration process a parameter may reach a value at which the function to be fitted is not defined (e.g. division by zero, taking the logarithm of a non-positive number, etc.). This can be avoided by a proper formulation of the user-supplied subroutine 'MODEL' and by the right choice of parameter limits.

At convergence, the point in n-dimensional parameter space defined by $\hat{\underline{P}}$ is surrounded by other points which form a polygon with $\hat{\underline{P}}$ as centroid. The size of this polygon depends on the accuracy with which the SS($\underline{P}$) minimum is determined and diminishes for small values of the stopping criterion. The SS-values at the vertices, the centroid and mid-points of the sides of the polygon provide the surface points used in the quadratic surface fitting for estimating the variability of the parameters (135). Because of the small

SCHEME 8.3 SOME INPUT ERROR MESSAGES IN FUNFIT[a]

Input error, too many numbers on line above.* are expected.
Reenter last line in correct form according to request.

Input error, too few numbers on line above.* are expected.
Reenter last line in correct form according to request.

Input error. Unrecognized character ( )[b].
Reenter last line in correct form.

Input error. The number entered is not in the allowed range.
- Try again -

Input error. The input is not in the right order.
- Try again -

Input error. You forgot to enter observation number *.
Reenter your data according to following request:

Input error. The weight for observation no.* (Y=*) is not defined by the chosen weighting scheme no.*

Input error. Your input does not agree with your input under request no.*
- Try again -

Input error. The initial estimate of parameter no.* is not within the limits given. Enter BACK if you want to make correction or OK if you want the program to choose an acceptable initial parameter estimate.

Input error, your answer must be one of the following three only

YES NO BACK (Y,N,B)
- Try again -

Input error. Your input under request no.* does not agree with the input under request no.*. What request number do you want to go back to for making corrections?

Input error. You forgot to enter data.
- Try again -

Input error. The weight of the dependent variable is negative.
Reenter last line in correct form.

(a) The appropriate numbers are printed where * appears in the text.

(b) The mistyped symbol is printed in the bracket. This feature prevents termination of the program when for example common errors such as typing ',' instead of '.' or the letter O instead of zero are committed.

size of the polygon at convergence these points will be so close together that numerical rounding-off errors can be significant in the fitting procedure. To prevent this, the polygon is expanded around the centroid before the quadratic surface is fitted until the SS-values at the ventices exceed that at the centroid by more than a given value of the expansion criterion (request no. 3). Two tests are then done to ensure that the statistical evaluation of the parameters is not significantly subject to numerical errors: The minimum of the fitted surface is compared with the minimum $SS(\hat{\underline{P}})$ found by the minimization process to check for round-off errors; also, the surface points used in the fitting procedure must lie in a convex region of the $SS(\underline{P})$ surface in order to be close to the true minimum and this is checked by evaluating whether the Hessian matrix (Eq. 8.13) is positive definite.

It is however most unlikely that there will be any computational difficulties either in the minimization process or in the statistical evaluation of the parameters if the recommended (default) values are chosen in requests no. 3,5 and 13.

Lower and upper parameter limits are specified in request no. 10 to prevent convergence on an $SS(\underline{P})$ minimum that may give unrealistic parameter estimates. If an initial estimate is not given for a parameter in request no. 11 the mid-point of the range defined by its lower and upper limits is chosen. Default values for the initial step sizes (request no. 12) are chosen in relation to the initial parameter estimates and their lower and upper limits so that a large part of the parameter space is searched in the $SS(\underline{P})$ minimization process and the chance of overlooking a smaller minimum is reduced.

Requests no. 11 and 12 also provide the option of choosing one or more initial parameter estimates and their respective step sizes independently

of their lower and upper limits. This option can be useful if it is desirable to increase the chance of finding a minimum that gives final estimates close to the initial estimates instead of finding another even smaller minimum that may exist but would give less realistic parameter estimates.

By entering 1 in request no. 5 a lattice search can be made *before* the start of a run. The coordinates of the point in the lattice which has the smallest SS($\underline{P}$) value can then be chosen as the initial estimates of the parameters, and the run can be started after input no. 5 has been modified (using a B5 command). This 'pre-search' for the best starting values increases the chance of finding the global minimum.

APPLICATIONS OF THE 'BACK' COMMAND

The B** command is not only useful for correcting input errors but has wider application. For example, if the analysis of the residuals suggests a particular weighting scheme, a new run with weighted data can easily be made. By typing B16, request no. 16 is repeated and the desired weighting scheme can be chosen (e.g. R = -2 will give weights proportional to y^{-2}). The 'communication' then continues from the current request and a new run, this time with weighted data, will be started by typing B18 followed by OK and N in succession. A new run in which one or more of the parameters is kept constant can, in a similar way, be made by repeating request no. 10 and fixing the parameters in question by setting upper limit=lower limit=the fixed value.

The lattice search that can be chosen in request no. 19 includes, in the case of two parameters (2-dimensional search), a contour map of the SS($\underline{P}$) surface, together with matrices of the SS($\underline{P}$) values and the sign of

the partial derivatives of SS(P) with respect to each of the two parameters at the points in the search grid. In the case of only one parameter (all others fixed) a plot will be printed of SS(P) *versus* the parameter.

The B** command is particularly useful in connection with a lattice search involving more than 2 parameters if a composite picture of the SS(P) surface is desired. This is done as follows: The command B10 is typed in response to request no. 19 (or after a lattice search) and all but two parameters are fixed. Then after the following 2-dimensional search, B19 is typed, followed again by B10. This procedure can be repeated as many times as desired, each time 'slicing' the parameter space in a different way.

If two or more response systems are fitted simultaneously the data for these are stacked and numbered in succession (in request no. 17). Therefore, a new run where only the first response system is fitted individually can easily be made by typing B8 followed by 1, B18 and N in succession. If the first run was a simultaneous fitting of a pharmacokinetic model using both blood and urine data the effect of including or excluding the urine response system can quickly be evaluated.

Every time the B** command is used the computer stores the current request position and returns to this immediately after correction of a previous input. It is however, possible to erase this memory and continue without returning from a previous request by typing the same B** command twice in succession. This procedure is useful if, for example, the first 7 inputs for a new run with different observations are identical to those of a previous run. These inputs are then taken to be the same and need not be repeated for the new run if B8 is typed twice in succession after completion of the previous run.

OUTPUT

Scheme 8.4 summarizes the maximum output possible from 'FUNFIT' for a single run. *The sections printed in italics denote the standard (minimum) output that is always printed* (or displayed on a terminal screen). The degree of output will range from this to the maximum output according to the user's specifications.

The optional input summary is useful as a check for numerical errors before activation of the computations or may serve as an extra copy of the experimental data. A minimization report should be chosen if the user-supplied subroutine 'MODEL' specifies parameter constraints that may cause convergence problems. The report is also of value because it provides a table of SS(P) values and parameter values that gives some information about the sum-of-squares surface and how large a section of the parameter space has been searched on the way to the minimum.

The quadratic surface fitting report is only printed in conjunction with the minimization report when statistical evaluation of the parameters has been requested (no. 3). The variance-covariance and correlation matrices of the parameters printed in this section will also appear in the general result section. The plots printed in the result section are line plots and are therefore of low accuracy but provide the essential information. The exact coordinates of the points in the plots are tabulated so precise plots can be produced manually if desired.

In fitting pharmacokinetic models it is often necessary to calculate from the final parameter estimates, quantities such as half-lives, clearances and volumes of distribution or to make a plot of the amount of drug in a peripheral compartment *versus* time. It will be demonstrated in the following section how the subroutine 'MODEL' which describes the equation(s) to be

SCHEME 8.4 SUMMARY OF OUTPUT FROM 'FUNFIT'[a]

INPUT SUMMARY[b]

- Heading for problem. - Number of: variable parameters, constant parameters, observations and independent variables for each response system. Table of parameters with lower and upper limits, initial estimates and step sizes. - Table of observations with dependent variable(s), independent variable(s), weight (if any) and normalized weight. - Plot of dependent variable(s) versus independent variable(s). - Weighting scheme used, expansion and stopping criterion. -

SUM-OF-SQUARES FUNCTION MINIMIZATION REPORT[c]

- Table of evaluation no., sum-of-squares values and parameters.- sum-of-square value and parameter estimates at convergence and number of evaluations used to reach convergence.-

FITTING OF QUADRATIC SURFACE IN REGION OF MINIMUM[c]

- Minimum of quadratic surface and parameter values at minimum.- Generalised inverse of information matrix (H^{-1}). - Information matrix (H). - Correlation matrix (Eq.8.17).- Number of evaluations used in the fitting.-

RESULTS

- Heading for problem.- *Table of parameters with their lower and upper limits, initial and final estimates.*- Table of standard deviation, coefficient of variation and 95% confidence limits of the parameters.- *Graphical illustration of the relative position of*

the calculated parameters in their specified range.- Residual sum of squares, regression sum of squares, sum of squared response, mean of response, residual mean square, regression mean square, mean of residuals and correlation coefficient.- Weighted residual sum of squares, weighted residual mean squares, mean of weighted residuals.- *Table of dependent variable(s), independent variable(s), observed and calculated responses, difference in response, differences expressed as percentages and as normal deviates.*- Table of weights, normalized weights, weighted residuals and normal deviate form of weighted residuals.- Plot of calculated (fitted) curve including experimental data points. Table of coordinates in plot.- Analysis of variance table.- Durbin-Watson statistics for serial correlation of residuals. 'Run test' and 'number test' for randomness of residuals.- Residual plots: residuals versus dependent and independent variables. Weighted residuals versus dependent and independent variables.- Variance-covariance matrix.- Correlation matrix.- OUTPUT SPECIFIED IN THE USER-SUPPLIED SUBROUTINE 'MODEL'.-

LATTICE SEARCH FOR GLOBAL MINIMUM[(d)]

The output in this section depends on the number of variable parameters as follows: (1 PARAMETER) - Plot of sum-of-squares function values versus the parameter in its specified range.- Table of coordinates of points in the plot.- Sum-of-squares and parameter values at minimum found in the unidimensional search.- (2 PARAMETERS)- Residual sum-of-squares matrix.

- Matrices of the sign of the partial derivatives of the sum-of-squares function with respect to each parameter.
- Sum-of-squares contour map.- Parameter values at the grid point

that has the smallest sum-of-squares value.- (> 2 PARAMETERS).- Total number of lattice points evaluated. Number of divisions of each parameter interval.- Parameter values of the grid point with the smallest sum-of-squares value.

(*a*) The scheme summarizes the maximum degree of output possible from FUNFIT for a single run. The sections printed in italics denote the standard, minimum output that is always printed. The degree of output will vary in this range according to the user's specifications.

(*b*) The input summary can be chosen to check all the inputs before the activation of the computations or may serve as an extra copy of the experimental data.

(*c*) It is useful to choose a minimization report if special parameter constraints have been specified in the user supplied subroutine 'MODEL' that specifies the function(s) to be fitted.

(*d*) The lattice search can be used to evaluate the contour of the sum-of-squares surface when there are only two parameters. A composite picture of the surface can be built up if there are more than two by fixing all but two parameters at a time (see text).

fitted, can include a special section containing such additional calculations and plots. This 'user-supplied output' will then be printed as the last part of the results section. The program includes an easily applicable subroutine, 'PLOT', which enables the user to make special plots in this section.

DEFINING THE EQUATION(S) TO BE FITTED

In order to use FUNFIT the function(s) to be fitted must first be defined by the user in a special subroutine called MODEL. The structure of this subroutine is:

```
SUBROUTINE MODEL (Y,X,P,IPRINT)
```

where Y denotes the dependent variable(s); X the independent variable(s); P the parameter vector and IPRINT is an integer variable which is controlled by FUNFIT and is used if additional output is desired. The subroutine must define Y as a function of X and P.

The structure can best be illustrated by some examples,

1. A single Equation, one Dependent and Independent Variable:

To fit the two-exponential equation:

$$c = p_1 e^{-p_2 t} + p_3 e^{-p_4 t} \qquad (8.52)$$

the subroutine can be written simply as:

```
SUBROUTINE MODEL (C,T,P,IPRINT)
DIMENSION P(4)
C = P(1)*EXP(-P(2)*T) + P(3)*EXP(-P(4)*T)
RETURN
END
```

Special Output

If it is desired to calculate a quantity, say:

$$A = P_1/P_2 + P_3/P_4 \tag{8.53}$$

from the *final least squares parameter estimates* this can be done by the computer by including the following *special output section* in the subroutine:

```
IF(IPRINT.EQ.Ø)RETURN
A = P(1)/P(2) + P(3)/P(4)
WRITE(6,1)A
1 FORMAT(" A=", E12.6)
```

just before the RETURN statement. It is seen that the IPRINT parameter controls the execution of the special output section. This section is printed *once* just after the general output section described in scheme 8.4.

An easily applicable plotting routine is available for special plots the user may wish to make in the special output section. Its structure is

```
SUBROUTINE PLOT (XARRY,YARRY,NPOINS,NLINES)
```

where XARRAY and YARRAY are two arrays of dimension NPOINTS which contain the coordinates to the points to be plotted using NLINES of the output device.[3]

In calculating the points for such plots it is frequently convenient to know the interval over which the observations are taken, i.e. the smallest and largest X value. This information can be made available by including the common statement:

```
COMMON XMIN, XMAX
```

---

3. It is recommended to use NPOINTS=75 and NLINES=50.

The following subroutine illustrates the use of PLOT in terms of a plot (75 points) of the function:

$$f_2 = \left| e^{-P_4 t} - e^{-P_2 t} \right| \qquad (8.54)$$

over the range of the t values, i.e. from XMIN to XMAX:

```
      SUBROUTINE MODEL (C,T,P,IPRINT)
      DIMENSION P(4),F2(75),TT(75)
      COMMON XMIN, XMAX
      C=P(1)*EXP(-P(2)*T) + P(3)*EXP(-P(4)*T)
      IF (IPRINT.EQ.Ø)RETURN
      DO 1 I = 1,75
      TT(I) = XMIN + (I-1)*XMAX/74.
    1 F2(I) = ABS(EXP(-P(4)**TT(I)) - EXP(-P(2)*TT(I)))
      CALL PLOT (TT,F2,75,50)
      RETURN
      END
```

Fitting Several Functions Simultaneously

The general structure of the user supplied subroutine MODEL when N response systems or equations are to be fitted simultaneously can be illustrated schematically as follows:

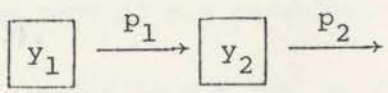
```
      SUBROUTINE MODEL (Y,X,P,IPRINT)
      DIMENSION P( )
      COMMON/FUNNUM/ITHFUN
      GO TO (1,2,...,N)ITHFUN
    1 Y = (1st function)
      RETURN
```

```
2     Y = (2nd function)
      RETURN
      .
      .
N     Y = (Nth function)
      RETURN
      END
```

The basis of this structure can be explained in the following way: During the input procedure each observation point is numbered consecutively and automatically assigned a label ITHFUN (i-th function) indicating which response system it belongs to. The user supplied subroutine MODEL is called by FUNFIT for each observation point. The parameter ITHFUN that is introduced by the COMMON block named FUNNUM (function number) transfer control to the i-th function which is then calculated. Control then returns to FUNFIT where the squared residual at that particular point is calculated, and added to the previous squared residual in the process of calculating the sum of squared residuals.

Example

Consider the following linear compartmental system:



where y_1 and y_2, which are measured at various times t, are given by:

$$dy_1/dt = -p_1 y_1 \qquad (8.55)$$

$$dy_2/dt = p_1 y_1 - p_2 y_2 \qquad (8.56)$$

which (for $y_1=y_0$ and $y_2=0$ at t=0) integrates to:

$$y_1 = y_0 \, EXP(-p_1 t) \tag{8.57}$$

$$y_2 = \frac{y_0 p_1}{p_2 - p_1} \, (EXP(-p_1 t) - EXP(-p_2 t)) \tag{8.58}$$

It is desired to estimate p_1 and p_2 from the set of observations of (y_1,t) and (y_2,t). This can be done by fitting simultaneously 8.57 and 8.58 to the observations.

These equations can conveniently be defined as follows:

```
      SUBROUTINE MODEL (Y,T,P,IPRINT)
      DIMENSION P(3)
      COMMON/FUNNUM/J
      GO TO (1,2) J
1     Y = P(3)*EXP(-P(1)*T)
      RETURN
2     Y = P(3)*P(1)/(P(2)-P(1))*(EXP(-P(1)*T)-EXP(-P(2)*T))
      RETURN
      END
```

The quantity y_0 can either be defined as a constant by setting its upper and lower limits equal to the same (constant) value in the input, or it can be defined as a parameter that is to be estimated by assigning appropriate bounds for P(3).

Fitting an Equation with 2 Independent Variables, a Simple Example

Consider a first-order reaction:

$$A \rightarrow B$$

where the fraction remaining, f_A, of A is measured at various times, t, at various temperatures T (^{0}K). A simplified model for this reaction may be written as:

$$f_A = \text{EXP}\ (-p_1 t\text{EXP}(-p_2/T)) \tag{8.59}$$

where p_1 and p_2 are parameters to be determined from a set of (f_A,t,T) data points. This model can be defined in the following way for FUNFIT:

```
SUBROUTINE MODEL (FA,X,P,IPRINT)
DIMENSION P(2),X(2)
FA = EXP(-P(1)*X(1)*EXP(-P(2)/X(2)))
RETURN
END
```

where X(1) acts as the time, t, and X(2) as the absolute temperature (^{0}K).

Fitting of Implicit Equations, A Simple Example

It is, in the following equation:

$$p_1 x = \ln y + p_2 y \qquad x,y,p_1,p_2 > 0 \tag{8.60}$$

not possible to express y explicitly as a function of the independent variable x so a special technique must be used to express and fit the function:

$$y = f(x,p_1,p_2) \tag{8.61}$$

The numerical problem of defining 8.60 in the form of 8.61 is the same as finding the root of the equation:

$$p_1 x - \ln y - p_2 y = 0 \tag{8.62}$$

for given values of x, p_1 and p_2; where $p_{1,2}$ are the parameters to be determined from a set of (x,y) observations. Let the left-hand side of 8.62 be denoted g(y) then y can be found by the Newton-Raphson iteration:

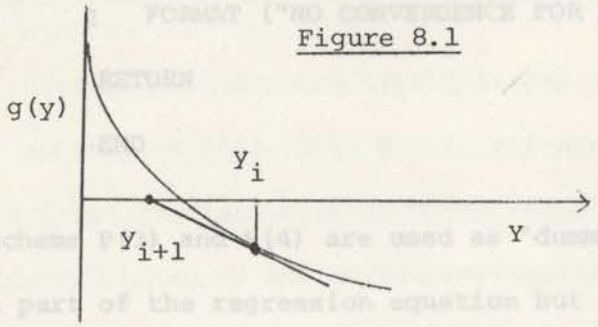
$$y_{i+1} = y_i - \frac{g(y_i)}{g'(y_i)} \tag{8.63}$$

where $$g'(y) = \frac{\partial g(y)}{\partial y} = -\frac{1}{y} - p_2 \tag{8.64}$$

i.e. 8.63 becomes: $$y_{i+1} = y_i + \frac{p_1 x - \ln y_i - p_2 y_i}{\frac{1}{y_i} + p_2} \qquad (8.65)$$

Since y and $p_2 > 0$ then $g'(y) < 0$ so the function g(y) will be monotonously decreasing:

Figure 8.1



In the iteration 8.65 y_{i+1} may become non-positive for which g(y) will not be defined in the next iteration. To prevent this to occur it is convenient to define:

$$y_{i+1} = y_i/2 \qquad \text{if} \qquad y_{i+1} \leq 0 \qquad (8.66)$$

The iteration 8.65 may be considered to have converged when the relative change in y between iterations is less than 0.0001% i.e. when $|(y_{i+1}-y_i)/y_i| < 10^{-6}$.

The subroutine to define 8.61 (i.e. 8.60) may thus be defined in the following way:

```
      SUBROUTINE MODEL(Y,X,P,IPRINT)
      DIMENSION P(4)
C     P(3) = CONVERGENCE CRITERION
C     P(4) = MAX.NO.OF ITERATIONS
      MAX  = IFIX (P(4))
      DO 1  I = 1, MAX
      YSAVE = Y
      Y = Y + (P(1)*X-ALOG(Y)-P(2)*Y)/(1/Y+P(2))
```

```
      IF(Y.LE.Ø.)Y = YSAVE/2.

      IF(ABS((Y-YSAVE)/YSAVE).LT.P(3))RETURN
    1 CONTINUE

      WRITE(6,2)X,Y,P
    2 FORMAT ("NO CONVERGENCE FOR X,Y,P =", 6E12.6)

      RETURN

      END
```

In this scheme P(3) and P(4) are used as "dummy" constant parameters which are *not* a part of the regression equation but included of computational reasons.

CHAPTER 9

CURVE FITTING AND MODELLING IN PHARMACOKINETICS

A Comparison of FUNFIT and NONLIN computer programs.

Several programs are available for nonlinear least squares parameter estimation (123-128). Nearly all are based on the Gauss-Newton or other related gradient methods since these are usually rapidly convergent and provide estimates of the variance-covariance matrix. However, such gradient methods may fail when the residuals are large (145, 146), as is often the case in fitting equations to biological data, and they may converge on a non-stationary point (147) if great care is not taken by the user in choosing a suitable value for the step size used in the finite difference approximation of derivatives. The default value specified for this step size in a program may unfortunately apply successfully only in a limited number of cases. Such practical experiences are illustrated in this paper in the application of the program NONLIN (123) which is based on Hartley's modification of the Gauss-Newton algorithm (148).[1]

Such problems are eliminated in FUNFIT. This program has implemented the adaptive simplex method of Nelder and Mead (135,150) which is a nongradient method that does not require evaluation of derivatives. This method is less efficient than Gauss-Newton based methods but considerably more robust and reliable. It will never fail even under extreme conditions where the gradient methods may be unstable due to near singularity and ill-conditioning of the matrices used in the iteration procedure.

---

1. Numerical techniques in nonlinear parameter estimation have been reviewed by Chambers (149) and an excellent discussion has been given by Dennis (146).

The possibility of multiple solutions (multiple sums of squares minima) undoubtedly represents the greatest problem in nonlinear estimation. The problem is expected to be particularly pronounced in pharmacokinetic studies, because these often involve the fitting of multiexponential equations to rather variable biological data, and because the ratio of number of data points to number of parameters is often quite small. Parameter estimation under such conditions may produce spurious results and discrimination between pharmacokinetic models may be very difficult and unreliable. The problem can be reduced, but usually not eliminated, if a graphical or numerical method is available which provides good initial parameter estimates for the iteration procedure. This is seldom the case for models describing nonlinear pharmacokinetics. The best approach should therefore be to use an algorithm which is effective in finding the (statistically) best solution in terms of the smallest residual sum-of-squares value.

It is generally accepted that the nongradient search methods perform better than the gradient methods in this respect. In particular, the adaptive simplex method used in FUNFIT appears to be very suitable because of its unique minimization method.

Regardless of the choice of algorithm the question of which starting values the parameters should be given still seems to be the greatest practical problem the user faces in nonlinear estimation. Frequently, when no preliminary estimation technique is available the initial values are simply guesses which all too often produce unacceptable results in the first run. However, by studying these results, corrections can often be made so that acceptable results can be obtained in subsequent runs.

Interactive programs are most convenient in such cases. In particular, FUNFIT has been designed so that it allows a highly interactive and flexible editing of input at any stage.

Pharmacokinetic applications of FUNFIT

1. The classical linear, compartmental models are still the most often used models in pharmacokinetic studies. The evaluation of such models is well documented and has become a routine procedure in many investigations.
2. There has, however, been an increasing awareness that linear models cannot adequately describe certain drug disposition phenomena (151,152) and various nonlinear models have been postulated. These mathematical models are often of a form which requires a special technique for least squares fitting.
3. Often several possible models are investigated to explain a pharmacokinetic phenomenon. There has been increasing interestin in discriminating between such models (152,153).

It is appropriate to discuss points 2. and 3. above:

Fitting of Implicit Functions: a simple example

Several of the models describing nonlinear pharmacokinetic phenomena can be expressed in an implicit form which can be fitted by defining the functional relationship between the variables explicitly by an iterative procedure.

Consider, for example, a simple one-compartmental model with intravenous injection in which the drug is eliminated partly by conversion to a single metabolite according to Michaelis-Menten kinetics and partly by excretion unchanged in the urine (154). The concentration of drug in plasma, c, is given *implicitly* by:

$$\ln \frac{c}{c_o} = \frac{V_m}{k_{lu} K_m} \ln \left[\frac{k_{lu} K_m + V_m + k_{lu} c_o}{k_{lu} K_m + V_m + k_{lu} c} \right] - \left[\frac{k_{lu} M_m + V_m}{K_m} \right] t \qquad (9.1)$$

where V_m and K_m are the Michaelis-Menten parameters, C_o = Dose/V_1 and k_{lu} is the urinary elimination constant. This equation can be written more simply:

$$\ln c + A_1 \ln (A_2 + A_3 c) + A_4 = 0 \qquad (9.2)$$

where
$$A_1 = \frac{V_m}{k_{lu} K_m} \qquad (9.3)$$

$$A_2 = \frac{k_{lu} K_m + V_m}{k_{lu} K_m + V_m + k_{lu} c_o} \qquad (9.4)$$

$$A_3 = \frac{k_{lu}}{k_{lu} K_m + V_m + k_{lu} c_o} \qquad (9.5)$$

$$A_4 = \frac{k_{lu} K_m + V_m}{K_m} t - \ln c_o \qquad (9.6)$$

The dependent variable, c, cannot be isolated from 9.1 but must be found by an iterative procedure by solving 9.2 for c.[2] The Newton-Raphson algorithm provides a simple and rapidly convergent method. If the expression in 9.2 is denoted g(c) then c can be determined by the following iteration:

$$c_{i+1} = c_i - \frac{g(c_i)}{g'(c_i)} \qquad (9.7)$$

where
$$g'(c) = \frac{\partial g}{\partial c} = \frac{1}{c} + \frac{A_1 A_2}{A_2 + A_3 c} \qquad (9.8)$$

---

2. It would be incorrect, as is sometimes done for equations of similar type, to fit directly the equation where t is expressed as a function of the dependent variable. The result would be unreliable because the dependent variable which accounts for nearly all of the errors in the data, is treated as an independent variable without error.

It is important that c_{i+1} in this iteration does not take a nonpositive value since this will terminate the execution of the program because g(c) (ln c) is not defined for $c \leq 0$. To prevent this it is most convenient to define

$$c_{i+1} = c_i/2 \qquad \text{if} \qquad c_{i+1} \leq 0 \tag{9.9}$$

This is an acceptable approach because g(c) is strictly increasing for $c>0$ since $A_1, A_2, A_3, A_4 > 0$ and therefore, $g'(c) > 0$. The term $\ln (A_2+A_3c)$ in g(c) will also be defined under these conditions.

The above procedure can be used for most implicit mathematical models in nonlinear pharmacokinetics. However special care must be taken to prevent the parameters wandering into a parameter space where the function(s) is not defined (e.g. logarithm of a nonpositive number, division by zero etc.).

Discrimination between models

The best criterion to use in discriminating between alternate pharmacokinetic models depends on the aim of the investigation and the application of the results (138,155-/157). If the main aim is to discriminate between models, the experiment should be designed so that the hypothesized models are placed in as much jeopardy as possible.

The problem is nevertheless considerably complicated by the substantial variation and low reproducibility of measurements in a biological system and the limited number of sample points available. Discrimination on a statistical basis requires information about the variability of the observations which can only be estimated by repeated experiments. A likely outcome of such experiments would often be that the system is "ill-conditioned" i.e. the variability of the data is too

large to allow a discrimination on a significant probability level.

Since the macroparametric representations of linear, compartmental models are all of the multi-exponential form:

$$c = \sum_{i=1}^{n} A_i e^{-\alpha_i t} \qquad (A_i, \alpha_i > 0) \qquad (9.10)$$

it appears appealing in routine investigations of raw pharmacokinetic data to apply a "multiple regression approach", similar to that used for linear systems, to determine the order, n, of the system.

This seems to be unreliable, however, for several reasons.

1. There exists no computer program which will inevitably find the "best" solution (smallest residual sum of squares) in a nonlinear least squares estimation which may have several minima.
2. Measurements in biological systems often produce substantial residual values which may give rise to multiple minima.
3. The number of minima will increase very rapidly as the number of exponential terms, n, to be fitted increases.

The problem of multiple minima can be reduced but *not* eliminated by a suitable procedure which gives good initial estimates, or by multiple runs with initial parameter values randomly taken from the parameter space, or by performing a lattice search. The interactive structure of FUNFIT makes it particularly suitable for performing multiple runs and lattice searches.

So-called "back-projection" or "stripping" is the technique most frequently used to obtain initial parameter estimates for models of the form described by 9.10. The stripping is either done graphically or automatically by the computer, in some cases employing a spline function representation of the data (152).

However, it is important to realise that this particular technique assumes that one or more exponential terms vanish in certain regions of the total drug level-time curve. In other words the method tends to disregard cases where two or more exponential terms dominate fairly equally throughout the whole time space investigated. Hence, the method may produce biased results. Discrimination between alternative models (9.10) on this basis must therefore be considered unreliable.

It would be appropriate in this connection to refer to a different method which does not introduce such a bias (158). This method is based on a linear shift operator technique which appears not to have been used previously in pharmacokinetic studies. Currently it seems to be the most suitable to use in obtaining initial parameter estimates in linear compartmental models.

In evaluating how well a model describes some data, three points must be considered:

1. How well do the calculated values agree with those observed, i.e. what is the sum of squared residuals or the correlation coefficient? 2. Does the fit agree with the basic assumptions made about the errors? 3. How predictive is the model?

A comparison of fits entirely in terms of sum of squared residuals (such as an F-test) must be considered insufficient.

Analysis of Residuals

The importance of an analysis of residuals (137) seems to have been completely ignored in most computer programs. The basic assumptions in nonlinear least squares are: 1. The independent variable(s) is without error. 2. The errors, ε_i in the dependent variable are independent

($\underset{i\neq j}{COV}(\varepsilon_i,\varepsilon_j)=0$) and normally distributed with zero mean and the same variance ($\varepsilon_i \sim N(0,\sigma^2)$).

The theory of least squares predicts that *if* the model is correct then the *residuals* should reflect the above properties of the *errors*, in an unbiased way, provided the errors possess these properties[3] (140). The assessment of a model on the basis of a residual analysis must rely on the converse principle: If the *residuals* appear to be from the same normal distribution then, under assumption 2, the model should not be rejected.

To investigate whether the residuals are in fact equally normally distributed requires repeated experiments. When only one experiment is available a test must rely on the fact that if the residuals are deviates from the same normal distribution, then collectively they will be normally distributed. The converse, however, is not true in general (140). Therefore, in the absence of repeated experiments it is necessary to make the additional assumption that if the residuals are collectively normally distributed then they are individually normally distributed also.

Possibly the best way to examine the residuals is to plot them against the independent and dependent variables (137,138). Significant systematic deviations can be visualized in this way. FUNFIT includes such plots and the following statistics which may be helpful in the assessment and comparison of models.

The Kolmogorov-Smirnov statistic (159) is used to test for normality of the residuals. THe procedure is as follows: Given N residuals, the program calculates

---

3. This is only strictly true in the linear case but approximately true for the nonlinear case.

$$D = \max_X |F^*(X) - S_N(X)| \qquad (9.11)$$

where $S_N(X)$ is the cumulative distribution of the residuals and $F^*(X)$ is the cumulative normal distribution function with the same mean and variance as the residual sample. The calculated value of D is compared with the critical value obtained from a Monte Carlo calculation at a given significance level.

The fundamental assumption of random errors is also tested in FUNFIT using the "run test" and the "number test" as discussed in Chapter 8. The Durbin-Watson statistic to test for serial correlation among the residuals was also presented in that chapter together with the test for "outliers".

If the residual analysis reveals that the residuals do not appear to be significantly random or normally distributed then *this does not necessarily mean that the model is incorrect*. More exactly it means one is faced with the problem of either rejecting the hypothesis that the model is "correct", rejecting the assumption made about the errors, or rejecting the assumption that the computer program has found the "best" solution in the case of multiple sum of squares minima. In the last two cases the model cannot be verified. This clearly emphasizes the need for a computer program which is efficient in finding a global minimum, the need for accurate data to reduce or eliminate multiple minima, and the need for carefully designed experiments which do not introduce systematic or cumulative errors.

The predictive power of the model

The ultimate goal in mathematical modelling in pharmacokinetics is to establish models with significant predictive power. A similar goal

exists in modelling of economic systems. The voluminous literature in this area can undoubtedly give inspiration to future approaches in pharmacokinetics.

A very useful test of the predictive capabilities of a model is to test the hypothesis that the parameters do not depend on the model variables. This can be done readily if sufficient data are available. The total set of data is first partitioned into 2 or more subsets. The parameters are then estimated separately for each subset and the parameter subsets are tested for any trend or for a functional relationship with the independent variable(s) by suitable correlation analysis. A test to establish whether the parameter subsets are significantly different from the parameter set obtained for the whole sample can also be employed [160,161]. The highly interactive structure of FUNFIT readily facilitates such partitioning of the data enabling the above tests to be made.

A Comparison of FUNFIT and NONLIN

FUNFIT was applied to obtain parameter estimates of the following simplified 2- and 3-compartment models:

$$c = p_1 e^{-p_2 t} + p_3 e^{-p_4 t} \qquad p_i > 0 \tag{9.12}$$

$$c = p_1 e^{-p_2 t} + p_3 e^{-p_4 t} + p_5 e^{-p_6 t} \qquad p_i > 0 \tag{9.13}$$

which were used to describe the plasma profile of pancuronium after I.V. bolus injection in 4 human subjects. The data to which 9.12 and 9.13 were fitted are shown in table 9.1.

Identical parameter limits and initial estimates were chosen to those used in applying the 1969 version of NONLIN which appears to be the

Table 9.1 Pancuronium Bloodlevel Data to which Eqs. 9.12 and 9.13 are fitted using NONLIN and FUNFIT.

	I.A.		B.A.		M.C.		J.C.	
	TIME	CONC.	TIME	CONC.	TIME	CONC.	TIME	CONC.
No.	min	µg/ml	min	µg/ml	min	µg/ml	min	µg/ml
1	7.5	1.120	5.0	.600	5.0	1.033	5.0	1.440
2	10.0	.775	10.0	.556	10.0	.830	10.0	1.000
3	20.0	.545	15.0	.550	15.0	.800	15.0	.945
4	30.0	.510	20.0	.480	20.0	.680	20.0	.805
5	60.0	.395	30.0	.370	30.0	.555	30.0	.620
6	95.0	.416	60.0	.200	60.0	.255	60.0	.463
7	120.0	.166	90.0	.160	91.0	.235	90.0	.365
8	152.0	.200	120.0	.150	120.0	.220	120.0	.355
9	180.0	.200	255.0	.090	240.0	.143	143.0	.270
10	190.0	.168	361.0	.100	413.0	.095	257.0	.160
11	250.0	.130					408.0	.083
12	400.0	.067						

most commonly used nonlinear regression program in pharmacokinetic investigations. The stopping criterion for FUNFIT was 0.1 (per cent) which gives approximately the same relative change in the SS value at convergence as NONLIN with its differently defined stopping criterion set at TEST=0.0001. The step size used in NONLIN to approximate derivatives was DEL=0.001; the same value as that used in the test problems given in the NONLIN user's manual. The experimental plasma levels were recorded to 3 significant digits after the decimal point so the precision factor in NONLIN was chosen as IDIG=-9 to avoid significant truncation of calculated values. This NONLIN and FUNFIT should give identical SS values for identical parameter values. This was verified in all runs.

FUNFIT found a different solution than NONLIN in every case where a 3-compartment model (9.13) was fitted, and in half of the cases where a 2-compartment model (9.12) was fitted (Table 9.2). The residual sum of squares values obtained using FUNFIT were substantially lower than those obtained using NONLIN in all cases where there was a difference. The average percentage difference was -55% and -29% when fitting Eqs. 9.13 and 9.12 respectively. The differences were also reflected in the parameter values. Furthermore, the run test indicates that the residuals are more randomly distributed overall in the FUNFIT results.

To test NONLIN's results in the cases where results differed for the two programmes FUNFIT was started with initial parameter values identical to NONLIN's *final* estimates and with initial parameter step sizes = 0.1% of these parameter values. At the first iteration FUNFIT gave exactly the same SS-value as NONLIN's value at convergence but it did not accept this solution as a stationary point and converged to a significantly different solution (Table 9.2).

Table 9.2: Least squares fitting of Eq's 9.12 and 9.13 using NONLIN (N) and FUNFIT (F).

PATIENT /POINTS		SS	ΔSS%[a]	RUNS[b]	P_1	P_2	P_3	P_4	P_5	P_6
IA/12	N	.7131E-1		6	.1452E+1	.2871	.4553	.3022E-1	.3986	.4165E-
	F	.2410E-1	-66.2	9	.4924E+1	.3026	.4081E-1	.1686	.6013	.6632E-
	F[c]	.2869E-1	-59.8	9	.2940E+1	.2370	.4855E-3	.3568E-1	.6025	.6712E-
JC/11	N	.1825E-1		8	.1038E+2	.9543	.1139E+1	.7307E-1	.6039	.5048E-
	F	.6631E-2	-63.7	9	.4649E+1	.5826	.8223	.5242E-1	.5597	.4721E-
	F[c]	.6074E-2	-66.7	9	.1303E+2	.7920	.8369	.5411E-1	.5647	.4744E-
BA/10	N	.8973E-2		6	.2614	.4341E-1	.3327	.3473E-1	.1697	.1873E-
	F	.4653E-2	-48.1	6	.5603E-1	.5668	.5911	.2526E-1	.1041	.2758E-
	F[c]	.4447E-2	-50.4	6	.7253E-1	.2467E-1	.5253	.2478E-1	.9702E-1	.3492E-
MC/10	N	.1382E-1		4	.8224	.4488E-1	.7388E-1	.6326E-1	.3148	.2521E-
	F	.7844E-2	-43.2	6	.1658E+1	.9200	.9374	.3623E-1	.2283	.1936E-
	F[c]	.8346E-2	-39.6	6	.8740	.3600E-1	.818E-1	.6357E-1	.2336	.2035E-
IA/12	N	.5102E-1		5	.1983E+1	.2367	.6586	.7449E-2		
	F	.2199E-1	-55.9	8	.1066E+2	.3999	.6202	.6858E-2		
	F[c]	.2199E-1	-55.9	9	.1049E+2	.3978	.6198	.6853E-2		
JC/11	N	.1621E-1		7	.1240E+1	.9424E-1	.6496	.5616E-2		
	F	.1621E-1	0	7	.1240E+1	.9425E-1	.6497	.5616E-2		
BA/10	N	.4528E-2		6	.5998	.2569E-1	.1012	.2100E-3		
	F	.4446E-2	- 1.8	6	.5975	.2470E-1	.9677E-1	.5866E-5		
	F[c]	.4449E-2	- 1.8	6	.5962	.2484	.9835E-1	.5402E-4		
MC/10	N	.8194E-2		6	.9464	.3760E-1	.2373	.2093E-2		
	F	.8194E-2	0	6	.9464	.3761E-1	.2374	.2095E-2		

(a) $\Delta SS\% = 100(SS_{FUNFIT} - SS_{NONLIN})/SS_{NONLIN}$

(b) See text for definition of runs.

(c) FUNFIT was in these cases started using NONLINS final parameter estimate as initial estimates and with initial parameter step sizes = 0.1% of these parameter values.

The detailed minimization report chosen in the investigation of this phenomenon, in fact, showed that the SS-function in NONLIN's convergence region had a significant gradient value indicating that NONLIN's solutions in these cases were not sufficiently close to the true sum of squares minimum.

The most likely reason for the failure of NONLIN to find a satisfactory solution in these cases appears to result from substantial errors in the approximation of derivatives.

The derivative with respect to the i-th parameter, p_i, of the function $f(\underline{X},\underline{P})$ to be fitted in NONLIN is approximated by a one-sided difference formula:

$$\frac{\partial f}{\partial p_i} \simeq \frac{f(\underline{X},p_1,p_2..,p_i+\delta p_i,..p_n)-f(\underline{X},\underline{P})}{\delta p_i} \qquad (9.14)$$

where the step size δ, (DEL) is chosen by the user. The value of this quantity is critical to the accuracy of the derivative. In choosing a proper value for δ, one has to steer between two hazards: 1. If the value chosen for δ is too *small* the derivative will be substantially inaccurate because of the *rounding error* which arises when the two f-values in (9.14) are too close. 2. If δ is set too *large* the derivative approximation will be too inaccurate because of the *truncation error* (9.14 is only accurate in the limit as $\delta \to 0$). This indicates that there must be an optimal value for δ. It can be shown that this value, for the i-th parameter, is approximately given by:

$$|\delta_i| \simeq \left|\frac{4\,f}{p_i H_{ii}}\right| \qquad (9.15)$$

where ϵ is the relative error in the computed f-value and H_{ii} is the i-th diagonal element of the Hessian matrix. This formula shows that the optimal

δ-value differs from parameter to parameter. It is not uncommon to find a very large value for the ratio, $\max(p_i H_{ii})/\min(p_i H_{ii})$, indicating that the choice of a *single* common δ-value as is done in NONLIN may not be adequate for all derivative evaluations and may cause convergence to a non-stationary point (147).

The above problem can be overcome in several ways:

1. By abolishing difference approximations and using exact analytical derivatives. This, however, may be of considerable inconvenience for the user who must define the analytical derivatives. It also limits the use of the program to equations for which analytical derivatives can be obtained.
2. By modifying the initial choice of δ according to Eq. 19 or by other means (162). However, even if alterations are made according to 1. or 2. the Gauss-Newton methods may still converge in some cases to a point at which the gradient does not vanish (145).
3. By abolishing the linearization approach in the Gauss-Newton methods and using a general function minimization approach (163-165).
4. By the use of an algorithm which is not based on derivatives or derivative approximations as is done in FUNFIT. The disadvantage of the last approach is that more function evaluations are required to reach convergence. For most pharmacokinetic applications this disadvantage is not significant. However, in cases where many parameters (> about 12) are to be estimated or where the equation(s) to be fitted is very time-consuming to evaluate (for example in the fitting of a functional relationship described by a system of differential equations) the disadvantage may become significant.

The presence of multiple SS-minima in fitting the 3-compartment model is evident from the fact that in the cases where FUNFIT was started

with NONLIN's final estimate it converged on a different solution (Table 9.2). The difference between SS-values found by FUNFIT in consecutive runs was much smaller than the difference between NONLIN and FUNFIT's values.

To test for multiple minima in fitting the 2-compartment model to IA's data, FUNFIT was started randomly 10 times in the chosen parameter space. In 9 of these cases it found the same solution (Table 9.2) but in one case it converged to:

$$SS = .2209E\text{-}1 \text{ and } p_{1-4} = .8780E{+}1,\ .3744,\ .6162,\ .6797E\text{-}2.$$

It is encouraging that this minimum is larger than that found in the 9+2 other cases. The more frequent occurrence of different solutions in the 3-compartment fitting confirms that the problem of multiple minima increases with an increasing number of parameters.

In only about half the cases investigated did FUNFIT and NONLIN find a smaller SS value for the 3-compartment model than for the 2-compartment model. This clearly emphasises the problems in discriminating between nonlinear mathematical models as discussed.

If, in fitting linear compartmental models, 9.10, the lower limits for the coefficient parameters A_i are set to zero then, in theory, the fit in terms of SS of a higher order model (e.g. n=3 vs n=2) should always be better or at least as good as the fit of a lower order model.[4] Therefore, if under such conditions it is found that the higher order model does not improve the fit (SS) then there is reason to believe a better minimum exists. In such cases, the higher order model should be refitted with initial estimates of the common parameters equal to the final parameter estimates of the lower order models. If this procedure also fails to give a lower SS-value, then a third run should be made where the common parameters

---

4. Provided of course that the parameter space for the lower order model is a subset of that of the higher order model.

are restricted within narrow limits around the optimal values for the lower order model, and where the parameters in the higher order term (e.g. p_5 and p_6 above) are much less restricted. The above approach will be successful in most cases and its use is recommended not only for fitting to sums of exponentials but also for any other "order system" where the terms are allowed to vanish. It should reduce significantly the problems associated with multiple minima and the problems of finding suitable initial estimates.

Truncation

In the NONLIN program it is possible by using the IDIG parameter to specify various degrees of truncation of the calculated values for the dependent variable. For example, if IDIG is set at -3 then all calculated values of the dependent variable will be truncated to 3 significant digits after the decimal point. The philosophy behind the use of this parameter is that there is no reason to calculate the predicted values to any higher precision than the observed values. In adopting such a philosophy it must be realised that the results so obtained will be specific to the NONLIN program and in general cannot be compared with results obtained using other nonlinear regression programs. The difference between results obtained specifying virtually no truncation (IDIG= -9) and specifying truncation to the precision of the observations (IDIG= -3) was found to be very pronounced (Table 9.3). The substantial difference was reflected not only in the SS-values and parameter values but also in the randomness of the residuals.

Truncation (IDIG= -3), furthermore, strongly affects the errors in the derivative approximations since the value of ε and hence, δ in Eq. 9.15 will be affected. This may explain why NONLIN in one case (Table 9.3)

Table 9.3: Least squares fitting of Eq's 9.12 and 9.13 using NONLIN with precision factor -3 and -9.

PATIENT /POINTS	PRECISION-FACTOR	SS	$\lvert\Delta SS\%\rvert$ [a]	RUNS [b]	P_1	P_2	P_3	P_4	P_5	P_6
IA/12	-9	.7131E-1	48.7	6	.1452E+1	.2871	.4553	.3022E-1	.3986	.4165E-2
	-3	.3658E-1		8	.1793E+1	.1807	.3155	.1097E-1	.2949	.3697E-2
JC/11	-9	.1825E-1	35.2	8	.1028E+2	.9543	.1139E+1	.7307E-1	.6039	.5048E-2
	-3	.1183E-1		6	.112E+1	.1813	.5571	.3310E-1	.5060	.4237E-2
BA/10	-9	.8973E-2	68.8	6	.2614	.4341E-1	.3327	.3473E-1	.1697	.1873E-2
	-3	.1515E-1		4	.1396	.5172E-1	.4649	.4265E-1	.1734	.1953E-2
MC/10	-9	.1382E-1	582	4	.8224	.4488E-1	.7388E-1	.6326E-1	.3148	.2521E-2
	-3	.9427E-1	(c)	3	.8000	.6000	.3000	.1000	.4000E-1	.2000E-2
IA/12	-9	.5102E-1	74.9	5	.1983E+1	.2367	.6586	.7449E-2		
	-3	.8924E-1		4	.8795	.4713E-1	.3312	.2706E-2		
JC/11	-9	.1621E-1	165	7	.1240E+1	.9424E-1	.6496	.5616E-2		
	-3	.4306E-1		4	.1114E+1	.4835E-1	.4251	.2649E-2		
BH/10	-9	.4528E-2	17.1	6	.5998	.2569E-1	.1012	.2100E-3		
	-3	.5303E-2		6	.5547	.3019E-1	.1610	.2121E-2		
MC/10	-9	.8194E-2	2.78	6	.9464	.3760E-1	.2373	.2093E-2		
	-3	.8422		5	.9226	.4107E-1	.2760	.2779E-2		

(a) $\lvert\Delta SS\%\rvert = 100 \times \lvert SS_{(-9)} - SS_{(-3)}\rvert / SS_{(-9)}$

(b) See text for definition of 'runs'.

(c) NONLIN failed to find a proper solution. At convergence after 5 iterations the final parameter estimates were identical to the initial estimates.

failed to converge properly.

The relative precision with which the parameter can be calculated will also be affected by truncation since it will not be possible to improve their estimates further when the computer has reduced the residual sum of squares to a value, SS, for which

$$\frac{SS}{SS(true)} < 1 + \varepsilon \qquad (9.16)$$

where the error, ε, depends on the degree of truncation chosen by IDIG (149). Furthermore the truncation procedure cannot be justified on a statistical basis since the theory of least squares assumes no errors in the calculated values. In fact great effort is often made to program the function to be fitted so that it can be evaluated with minimum errors to avoid biased results.

The user of the program NONLIN is, therefore, advised to use a value for the precision factor IDIG such that minimum truncation takes place.

The many nonlinear regression programs available have provided the scientist with a powerful tool useful for a great variety of problems. However, the results obtained have too often been accepted and used without an awareness of the limitations and possible unreliability of the program used, ignoring the numerical problems involved.

The complex structure of the program used has often resulted in an authoritative attitude which may go so far as using the program as a substitute for rational thought. There is a definite need for programs which are more reliable and which allow greater interaction between the user and the program, with the user in a more dominant role.

CHAPTER 10

A NEW METHOD FOR CHARACTERISING THE DISSOLUTION PROPERTIES OF DRUG POWDERS

Chapter 7 demonstrated good agreement between the theoretical dissolution profile and experimental data for tolbutamide. The intrinsic dissolution profile was calculated, using the rigorous mathematical approach present in Chapter 5 and 6, from optical analysis of the size distribution of the 60/85 mesh fraction powder. Such an analysis is complicated to perform for micronized powders because of the highly irregular particle shapes and the degree of aggregation often found in such powders.

However it should be possible to determine by nonlinear regression analysis whether the dissolution of such a powder can be adequately described by one of the multiparticulate dissolution models presented earlier.

In this chapter various such models, based on a log-normal particle size distribution, are fitted by nonlinear least squares regression to data from the dissolution of micronized glibenclamide using the FUNFIT program. Estimates of parameters describing the *effective* initial particle size distribution are obtained together with estimates of a quantity defined as *the specific dissolution rate parameter*. A dissolution equation based on an ideal, untruncated log-normal distribution with the single particles dissolving according to the cube root law best describes the dissolution kinetics. Dissolution behaviour of glibenclamide can be well described by this model in terms of *the specific dissolution rate parameter* and one other parameter which accounts for the distribution effect termed *the dispersion parameter*.

Estimation of these two parameters provides a means of describing dissolution characteristics of drug powders apparently more correctly than previous approaches. The method should be of interest in the quality control of drugs likely to cause bioavailability problems because of dissolution rate limited absorption.

Theoretical

The dissolution equation for a log-normal powder, considering the cube root and the square root model, was given earlier by Eq. 6.21. This equation appears to contain 5 parameters, namely σ, i, j, μ and K_m which define the dissolution profile, W/W_o *versus* time. However, an attempt to obtain least squares estimates of *all* these 5 parameters from W/W_o *vs.* time data may fail because μ and K_m can be fused into a single parameter:

$$K_m^* = e^{\frac{-3}{m}\mu} K_m \tag{10.1}$$

which will be called the *specific dissolution rate parameter*.

The uniqueness of these 4 parameters, σ, i, j and K_m^*, in defining the dissolution profile can be seen by substituting $K_m = e^{\frac{3}{m}\mu} K_m^*$ into Eq. 10.1 resulting in total cancellation of μ, in full agreement with the theory discussed previously:

$$\frac{W}{W_o} = \sum_{n=o}^{m} \binom{m}{n} (-K_m^* t)^{(m-n)} \frac{F(j-\frac{3n\sigma}{m})-A}{F(j-3\sigma)-F(-i-3\sigma)} e^{\frac{9}{2}\left[\left(\frac{n}{m}\right)^2-1\right]\sigma^2} \tag{10.2}$$

where $A = F(-i-\frac{3n\sigma}{m})$ for $t < \frac{1}{K_m^*} e^{\frac{-3i\sigma}{m}}$ (10.3)

and $A = F(\frac{m}{3\sigma} \ln(K_m^* t)-\frac{3n\sigma}{m})$ for $\frac{1}{K_m^*} e^{\frac{3j\sigma}{m}} > t > \frac{1}{K_m^*} e^{\frac{-3i\sigma}{m}}$ (10.4)

and $$\frac{W}{W_o} = 0 \quad \text{for} \quad t \geq \frac{1}{K_m^*} e^{\frac{3j\sigma}{m}} \tag{10.5}$$

The continuous flow, recording apparatus used provides dissolution *rate* data. The fraction undissolved, W/W_o, *versus* time can be obtained by integrating. The parameters σ, i, j and K_m^* can then be estimated by nonlinear least squares regression analysis using Eq. 10.2. However the integrated data will contain integration errors. The integration also tends to "smooth" the original data so estimates of the variability of the parameters will be less reliable than if the original *rate* data were used. It is therefore useful to derive an expression for the release rate in order to estimate the 4 parameters directly from the original (rate) data.

By applying Eq. 5.33 of Chapter 5 the following equation is obtained:

$$\frac{W}{W_o} = \frac{\int_{R_1}^{R_2} \left[X^{\frac{3}{m}} - K_m t\right]^m X^{-1} N(\ln x, \mu, \sigma)\, dx}{\int_{d_o}^{D_o} x^2 N(\ln X, \mu, \sigma)\, dx} \tag{10.6}$$

where $$d_o = e^{\mu - i\sigma} \tag{10.7}$$

$$D_o = e^{\mu + j\sigma} \tag{10.8}$$

$$R_1 = \left(\max\left[K_m t,\ d_o^{\frac{3}{m}}\right]\right)^{\frac{m}{3}} \tag{10.9}$$

$$R_2 = \left(\max\left[K_m t,\ D_o^{\frac{3}{m}}\right]\right)^{\frac{m}{3}} \tag{10.10}$$

An expression for the dissolution rate of drug, $Q = -dW/dt$ can now be found by differentiating Eq. 10.6 with respect to time (using Leibnitz's rule). For abbreviation, let B denote the (constant) denominator of Eq. 10.6; differentiation then gives:

$$\frac{B}{W_o} Q = mK_m \int_{R_1}^{R_2} (x^{\frac{3}{m}} - Kt)^{(m-1)} x^{-1} N(\ln x, \mu, \sigma) dx \qquad (10.11)$$

$$- (R_2^{\frac{3}{m}} - K_m t)^m R_2^{-1} N(\ln R_2, \mu, \sigma) \frac{dR_2}{dt}$$

$$+ (R_1^{\frac{3}{m}} - K_m t)^m R_1^{-1} N(\ln R_1, \mu, \sigma) \frac{dR_1}{dt}$$

Before critical time[1] $R_1 = d_o$, $R_2 = D_o$ so $dR_1/dt = dR_2/dt = 0$, and the last two terms of the right-hand side of Eq. 10.11 vanish. *After* the critical time, dR_2/dt is still zero but $R_1 = (K_m t)^{\frac{m}{3}}$ (according to 10.9), thus the last term also vanishes because then:

$$R_1^{\frac{3}{m}} - K_m t = ((K_m t)^{\frac{m}{3}})^{\frac{3}{m}} - K_m t = 0$$

Equation 10.11 can therefore be simplified to:

$$Q = mK_m W_o \frac{\int_{R_1}^{R_2} (x^{\frac{3}{m}} - K_m t)^{(m-1)} x^{-1} N(\ln x, \mu, \sigma) dx}{\int_{d_o}^{D_o} x^2 N(\ln x, \mu, \sigma) dx} \qquad (10.12)$$

The term $(x^{\frac{3}{m}} - K_m t)^{(m-1)} x^{-1}$ under the integral sign can be expanded to:

$$x - 2K_3 t + (K_3 t)^2 x^{-1} \qquad \text{(for m=3)}$$

$$x^{\frac{1}{2}} - K_2 t x^{-1} \qquad \text{(for m=2)} \qquad (10.13)$$

and the formula given previously (5.34) can be applied to express the integrals in 10.11 in terms of the function F. This leads to the following

---

1. The critical time is the time when the first particles begin to disappear in the dissolution process, i.e.

$$t = \frac{1}{K_m} e^{\frac{3}{m}(\mu - i\sigma)}$$

expression for Q that can be evaluated more readily and exactly than 10.12:

$$Q = m K_m W_o \sum_{n=o}^{(m-1)} \binom{m-1}{n} (-K_m t)^{(m-n-1)} \frac{F\left(\frac{T_2-\mu}{\sigma} - \frac{3n\sigma}{m}\right) - F\left(\frac{T_1-\mu}{\sigma} - \frac{3n\sigma}{m}\right)}{F(j-3\sigma)-F(-i-3\sigma)} \; X$$

$$e^{\frac{3}{m}(n-m)\left(\mu+\frac{3}{2m}(n+m)\sigma^2\right)} \qquad (10.14)$$

where T_1 and T_2 are defined:

$$T_1 = \max\left[\frac{m}{3}\ln(Kt),\ \mu-i\sigma\right] \qquad (10.15)$$

$$T_2 = \max\left[\frac{m}{3}\ln(Kt),\ \mu+j\sigma\right] \qquad (10.16)$$

This expression for the dissolution rate Q contains 5 parameters (σ,i,j,μ and K_m). However, as before only 4 parameters (σ,i,j and K_m^*) are needed to define uniquely the dissolution rate profile:

$$Q = m K_m^* W_o \sum_{n=o}^{(m-1)} \binom{m-1}{n} (-K_m^* t)^{(m-n-1)} \frac{F\left(j-\frac{3n\sigma}{m}\right)-A}{F(j-3\sigma)-F(-i-3\sigma)} \; X \qquad (10.17)$$

$$e^{\frac{9}{2}\left[\left(\frac{n}{m}\right)^2-1\right]\sigma^2}$$

This equation is the differential form of Eq. 10.2. The quantity A is defined as previously (10.3 and 10.4) and

$$Q = 0 \text{ for } t \geq \frac{1}{K_m^*} e^{\frac{3j\sigma}{m}}.$$

The above expression is of considerable value since it allows the *effective* initial particle size distribution parameters σ,i and j to be determined together with the specific dissolution rate parameter, K_m^*, by nonlinear regression analysis of the dissolution rate profile (dW/dt *versus* time).

It is also of interest to investigate how well the dissolution behaviour can be described if the particle size distribution is considered

ideal, i.e. if i=j=∞.

Noting that F(∞) = 1 and F(-∞) = 0, Eq. 10.17 becomes, for an ideal distribution:

$$Q = m\,K^*_m W_o \sum_{n=o}^{(m-1)} \binom{m-1}{n} (-K^*_m t)^{(m-n-1)} \left[1-F\left(\frac{m}{3\sigma}\ln(K^*_m t)-\frac{3n\sigma}{m}\right)\right] X$$

$$e^{\frac{9}{2}\left[\left(\frac{n}{m}\right)^2-1\right]\sigma^2} \qquad (10.18)$$

If the distribution is considered ideal at the lower end (i=∞) but truncated at the higher end, the expression becomes:

$$Q = m\,K^*_m W_o \sum_{n=o}^{(m-1)} \binom{m-1}{n} (-K^*_m t)^{(m-n-1)} \frac{F\left(j-\frac{3n\sigma}{m}\right)-F\left(\frac{m}{3\sigma}\ln(K^*_m t)-\frac{3n\sigma}{m}\right)}{F(j-3\sigma)} X$$

$$e^{\frac{9}{2}\left[\left(\frac{n}{m}\right)^2-1\right]\sigma^2} \qquad (10.19)$$

and if it is considered truncated at the lower end but not at the higher end (j=∞):

$$Q = m\,K^*_m W_o \sum_{n=o}^{(m-1)} \binom{m-1}{n} (-K^*_m t)^{(m-n-1)} \frac{1-A}{1-F(-i-3\sigma)} e^{\frac{9}{2}\left[\left(\frac{n}{m}\right)^2-1\right]\sigma^2} \qquad (10.20)$$

Monodisperse powder

It is of interest to look at the limiting case where the particle size distribution is infinitely narrow, i.e. σ=0, since this case allows a better understanding of the specific dissolution rate parameter, K^*_m. It also provides a method of obtaining a suitable initial estimate of this parameter to use in nonlinear curve fitting.

When σ=i=j=o, Eq. 10.17 becomes:

$$Q = m K^*_m W_o \sum_{n=o}^{(m-1)} \binom{m-1}{n} (-K^*_m t)^{(m-n-1)} \qquad (10.21)$$

which can be written more simply as:

$$Q = -\frac{dW}{dt} = m K^*_m W_o (1-K^*_m t)^{(m-1)} \qquad (10.22)$$

which, after integration, can be written:

$$\left(\frac{W}{W_o}\right)^{\frac{1}{m}} = 1 - K^*_m t \qquad (10.23)$$

As expected, since there is no size distribution effect this equation predicts that the powder will dissolve strictly according to the cube root or square root model.

When the powder is monodisperse $\mu = \ln a_o$ and Eq. 10.1 becomes:

$$K^*_m = e^{-\frac{3}{m} \ln a_o} K_m = a_o^{-\frac{3}{m}} K_m \qquad (10.24)$$

For spherical particles $K_m = (6\rho\pi)^{\frac{1}{m}} k_m$ where k_m is the rate parameter in the single particle dissolution model:

$$w^{\frac{1}{m}} = w_o^{\frac{1}{m}} - k_m t \qquad (10.25)$$

thus 10.24 becomes:

$$K^*_m = a_o^{-\frac{3}{m}} (6/\rho\pi)^{\frac{1}{m}} k_m = w_o^{-\frac{1}{m}} k_m \qquad (10.26)$$

When $k_m = w_o^{\frac{1}{m}} K^*_m$, from this expression is inserted in 10.25 the single particle dissolution model can be written:

$$\left(\frac{w}{w_o}\right)^{\frac{1}{m}} = 1 - K^*_m t \qquad (10.27)$$

A comparison of 10.27 and 10.23 shows that *for a monodisperse powder the specific dissolution rate parameter, K^*_m, is common to both multiparticulate*

(10.23) and single particle dissolution (10.27). For both m=2 and 3 it has the dimension of time^{-1}.

Equation 10.22 can be written:

$$Q^{\frac{1}{m-1}} = (mW_o K_m^*)^{\frac{1}{m-1}} - (mW_o)^{\frac{1}{m-1}} (K_m^*)^{\frac{m}{m-1}} t \qquad (10.28)$$

This equation can be used to obtain an initial estimate of K_m^* from the linear regression of $Q^{\frac{1}{m-1}}$ on t, using dissolution rate data. The intercept value, $(mW_o K_m^*)^{\frac{1}{m-1}}$, divided by $(mW_o)^{\frac{1}{m-1}}$ gives $K_m^{*\frac{1}{m-1}}$. (Fig. 10.1)

For a monodisperse powder 10.28 predicts a linear relationship between $\sqrt{Q}$ and t when the single particles dissolve according to the cube root model (m=3) and a linear relationship between Q and t when they dissolve according to the square root model (m=2).

Significant deviations from linearity were observed when dissolution rate data for micronized glibenclamide were plotted in either of these ways (Figs. 10.1-10.5). Such deviations can arise if the powder is not monodisperse or if the single particles do not dissolve according to the single particle model given by 10.25.

Under an electron microscope the micronized glibenclamide used appears to be quite polydisperse. The observed deviation from linearity in the rate plots can thus be explained as a particle size distribution effect, assuming 10.25 to be valid. However, the rate data can also be explained by other single particle dissolution models in combination with a size distribution effect. Conclusions about the validity of a single particle model can only be made when the size distribution effect can be taken into account.

Chapter 7 dealing with the dissolution of 60/85, mesh fraction tolbutamide in relation to its particle size distribution indicated that

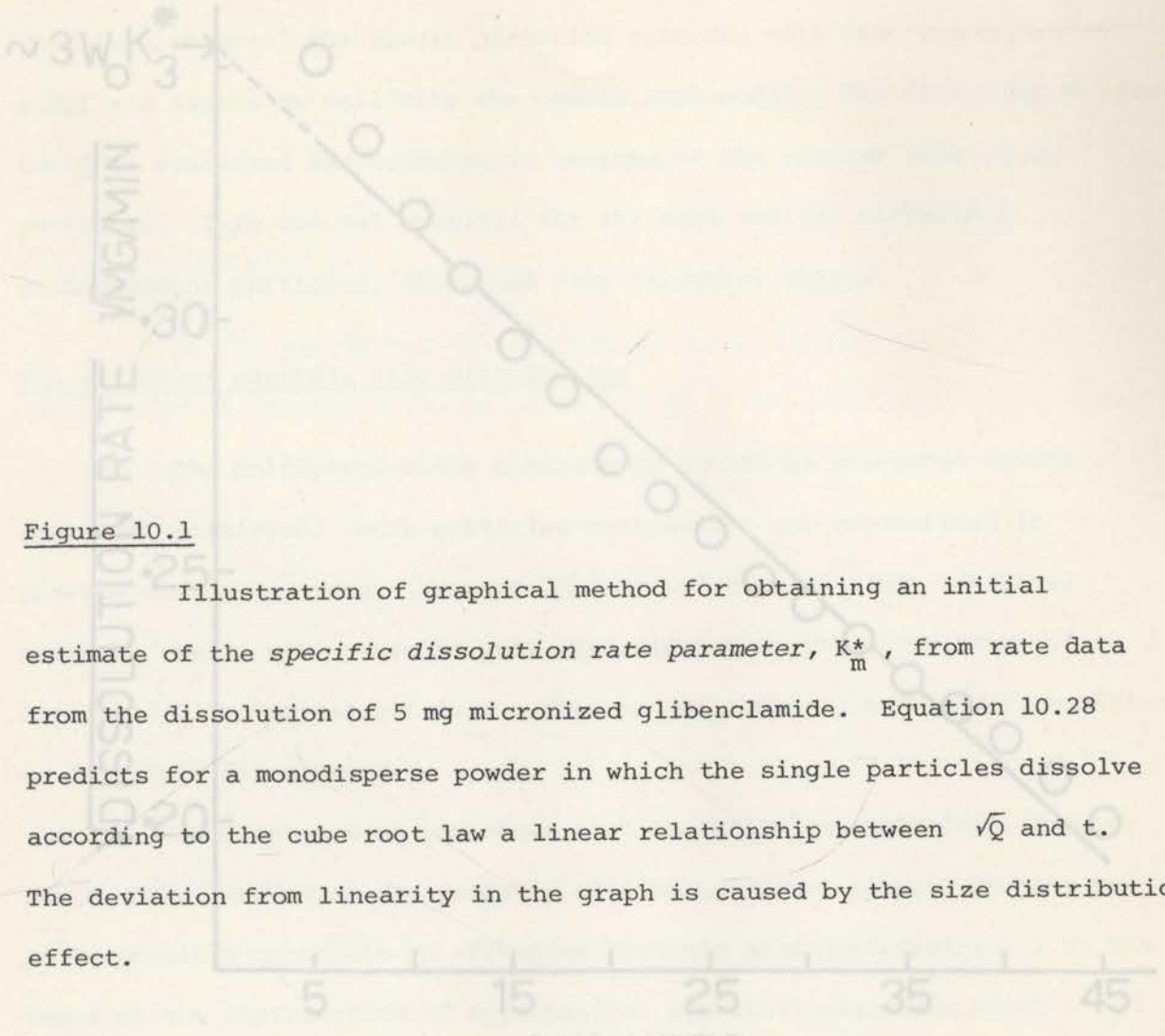
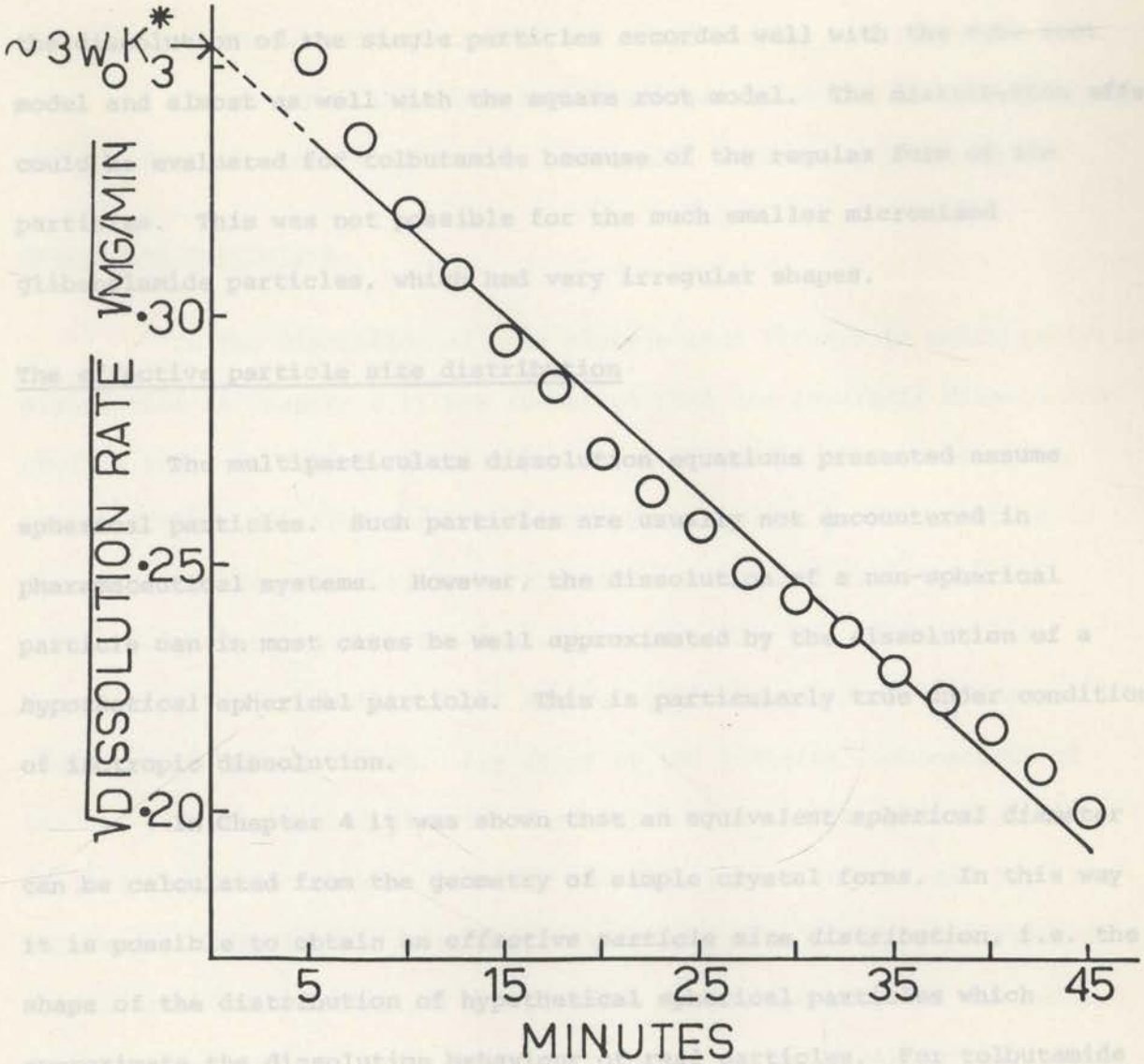


Figure 10.1

Illustration of graphical method for obtaining an initial estimate of the *specific dissolution rate parameter*, K_m^* , from rate data from the dissolution of 5 mg micronized glibenclamide. Equation 10.28 predicts for a monodisperse powder in which the single particles dissolve according to the cube root law a linear relationship between $\sqrt{Q}$ and t. The deviation from linearity in the graph is caused by the size distribution effect.


~3WK*
√DISSOLUTION RATE √MG/MIN
30
25
20
5
15
25
35
45
MINUTES

the dissolution of the single particles accorded well with the cube root model and almost as well with the square root model. The distribution effect could be evaluated for tolbutamide because of the regular form of its particles. This was not possible for the much smaller micronized glibenclamide particles, which had very irregular shapes.

The effective particle size distribution

The multiparticulate dissolution equations presented assume spherical particles. Such particles are usually not encountered in pharamaceutical systems. However, the dissolution of a non-spherical particle can in most cases be well approximated by the dissolution of a *hypothetical* spherical particle. This is particularly true under conditions of isotropic dissolution.

In Chapter 4 it was shown that an *equivalent spherical diameter* can be calculated from the geometry of simple crystal forms. In this way it is possible to obtain an *effective particle size distribution*, i.e. the shape of the distribution of hypothetical spherical particles which approximate the dissolution behaviour of *real* particles. For tolbutamide it was found that the effective distribution was approximately log-normal. Because of the irregular particle shapes it was not possible to make *a priori* conclusions about the effective particle size distribution for glibenclamide. In these investigations the distribution is assumed to be log-normal, consistent with the results in Chapter 7 and the fact that powders are often found to have a log-normal particle size distribution (122,166). *The distribution in Fig.* 5.1 *therefore illustrates a log-normal approximation to the effective particle size distribution. The initial particle diameters are the diameters of the hypothetical spherical particles*

that approximate the dissolution of the real nonspherical particles.

If the lower and upper truncation parameters, i or j, are finite the log-distribution is said to be truncated; otherwise it is ideal.

Regression parameters

In the discussion of size distribution effects in multiparticulate dissolution in Chapter 6 it was indicated that the *intrinsic dissolution profile* does not depend on the actual size of the particles but on the shape of their distribution. For the same reason it is not possible to determine by regression analysis the scale parameter, μ, as might be erroneously expected from the appearance of 6.21. If, in fact, this equation is used to obtain least squares estimates of μ and K_m, these would not be unique values. Any other of the infinite combinations of the two parameters that give the same value of K_m^* (10.1) will according to 10.2 result in the same fit (for the same values of σ, i and j). The regression analysis can thus only provide estimates of the *dimensionless* distribution parameters σ, i and j, which define the *shape* of the initial distribution, and K_m^*. *The scale or position of the distribution, given by μ, is hidden in the specific dissolution rate parameter, K_m^*.*

Curvefitting

Equation 10.17 was fitted by least squares to dissolution rate data from dissolution of 5 and 10 mg micronized glibenclamide using FUNFIT. The kinetic models given by 10.17-10.19 are defined in schemes 10.1-10.3

respectively.[2]

The estimates obtained for the truncation parameters, i and j (Table 10.1, Eq. 10.17), were all larger than 2 and in most cases exceeded 4, indicating that the effective initial particle size distribution was close to ideal. It was shown in Chapter 5 that the effect on the dissolution profile of the lower truncation parameter, i, is negligible. Simulation studies also show that the influence of an increase in the upper truncation parameter, j, becomes insignificant when j is larger than 2. It is therefore expected from the values of i and j obtained using 10.17 that the simpler model, 10.18, which assumes an ideal distribution ($i=j=\infty$) should fit the same data nearly as well. The values in table 10.1 and the curves fitted in Figs. 10.2-10.5 confirm this expectation. There does not seem to be any significant difference in either the K_m^*, the σ or the r-values for the two models. The residual plots in Figs. 10.2, 10.3 and Figs. 10.4, 10.5 also seem to be very similar. Equation 10.19 (upper truncation) and 10.20 (lower truncation) also gave similar results (and therefore, these have not been included).

Choice of the mathematical model

In agreement with the general principles of mathematical modelling, Eq. 10.17, should be considered as the model that best describes the dissolution of the micronized glibenclamide. This is because it is the

---

2. The user of FUNFIT does not need to specify the binomial coefficients or the cumulative standard normal distribution function, F, since these are present in FUNFIT as function routines named NBC and SDF respectively. The "model parameter", m, can be set at 3 or 2, depending on whether a model based on the cube- or square root model is to be fitted. The use of m is of great convenience since a fitting to both models can quickly be done with a single input of dissolution data.

TABLE 10.1 Least squares estimates of rate- and distribution parameters obtained from nonlinear regression analysis of data from the dissolution of micronized glibenclamide, considering various models for multiparticulate dissolution kinetics.[a]

W_o (mg)		SQUARE ROOT MODEL (m=2)			CUBE ROOT MODEL (m=3)		
		Eq.10.18[b] Scheme 10.2	Eq.10.19[c] Scheme 10.3	Eq.10.17[d] Scheme 10.1	Eq.10.18[b] Scheme 10.2	Eq.10.19[c] Scheme 10.3	Eq.10.17[d] Scheme 10.1
5	K_m^* (min^{-1})	0.02858 (6.39)	0.02858 (5.12)	0.02859 (6.51)	0.02367 (4.55)	0.02367 (4.44)	0.02449 (3.96)
	σ	0.4759 (2.93)	0.4760 (2.70)	0.4760 (3.28)	0.6184 (2.47)	0.6184 (2.11)	0.6315 (1.71)
	i	∞	∞	5.136	∞	∞	2.364
	j	∞	5.299	5.655	∞	9.807	4.172
	r[e]	0.9935	0.9935	0.9935	0.9952	0.9952	0.9954
10	K_m^* (min^{-1})	0.02633 (7.18)	0.02633(10.6)	0.02633 (9.35)	0.02147 (9.45)	0.02147 (4.74)	0.02176 (4.03)
	σ	0.4955 (3.20)	0.4956 (7.24)	0.4955 (5.14)	0.6374 (3.29)	0.6374 (2.03)	0.6446 (1.89)
	i	∞	∞	8.912	∞	∞	4.577
	j	∞	5.299	6.732	∞	7.060	4.200
	r[e]	0.9938	0.9938	0.9938	0.9957	0.9957	0.9958

(a) The values in brackets are relative standard deviations (percent).
(b) The initial size distribution is considered ideal (i=j=∞,Fig.1).
(c) The initial size distribution is truncated at the upper end (i=∞, j<∞,Fig.1).
(d) The initial size distribution is truncated at both ends (Fig.1).
(e) Correlation coefficient.

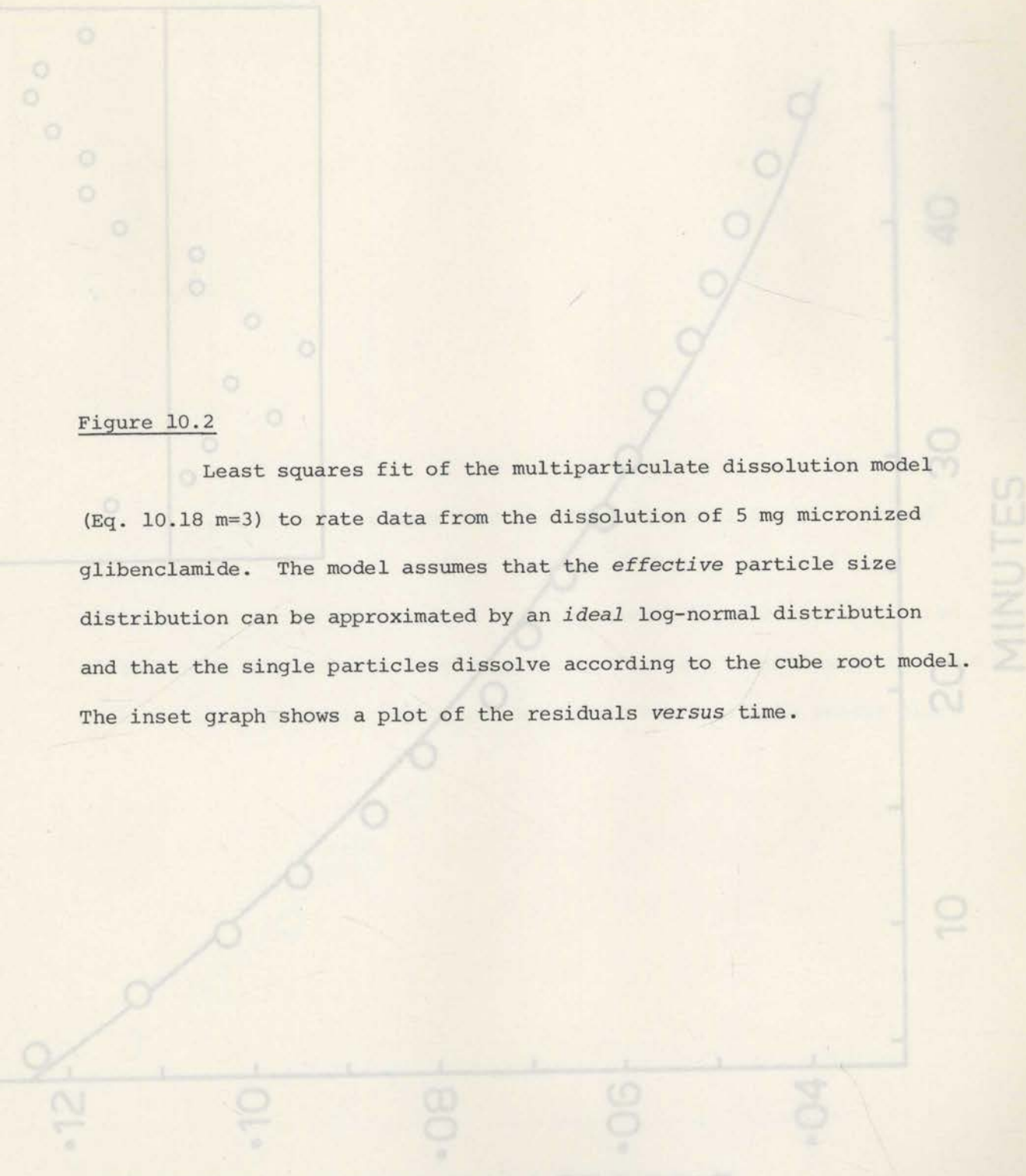


Figure 10.2

Least squares fit of the multiparticulate dissolution model (Eq. 10.18 m=3) to rate data from the dissolution of 5 mg micronized glibenclamide. The model assumes that the *effective* particle size distribution can be approximated by an *ideal* log-normal distribution and that the single particles dissolve according to the cube root model. The inset graph shows a plot of the residuals *versus* time.

DISSOLUTION RATE MG/MIN
.04
.06
.08
.10
.12
10
20
30
40
MINUTES

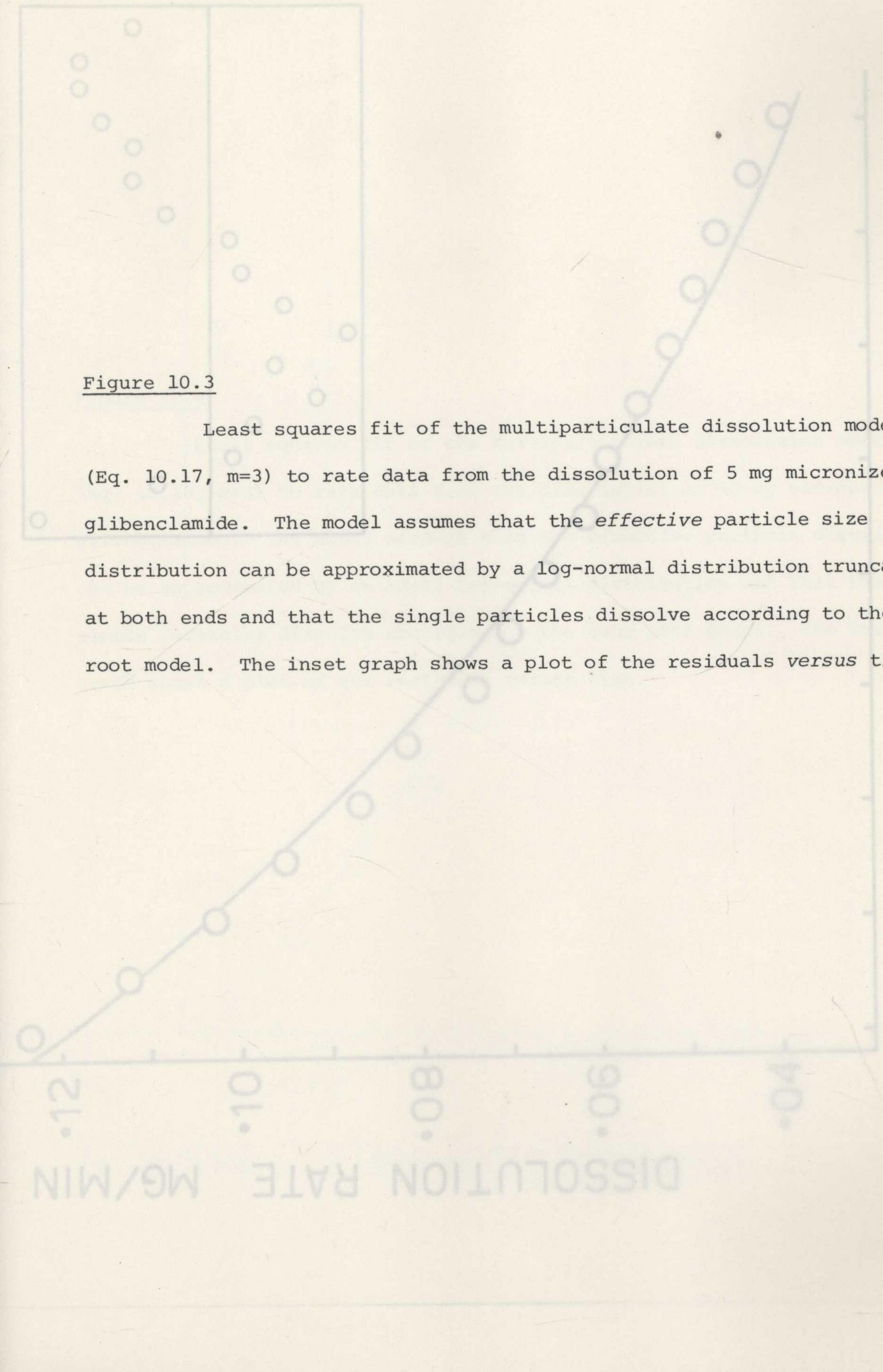


Figure 10.3

Least squares fit of the multiparticulate dissolution model (Eq. 10.17, m=3) to rate data from the dissolution of 5 mg micronized glibenclamide. The model assumes that the *effective* particle size distribution can be approximated by a log-normal distribution truncated at both ends and that the single particles dissolve according to the cube root model. The inset graph shows a plot of the residuals *versus* time.

DISSOLUTION RATE MG/MIN
.04
.06
.08
.10
.12
10
20
30
40
MINUTES

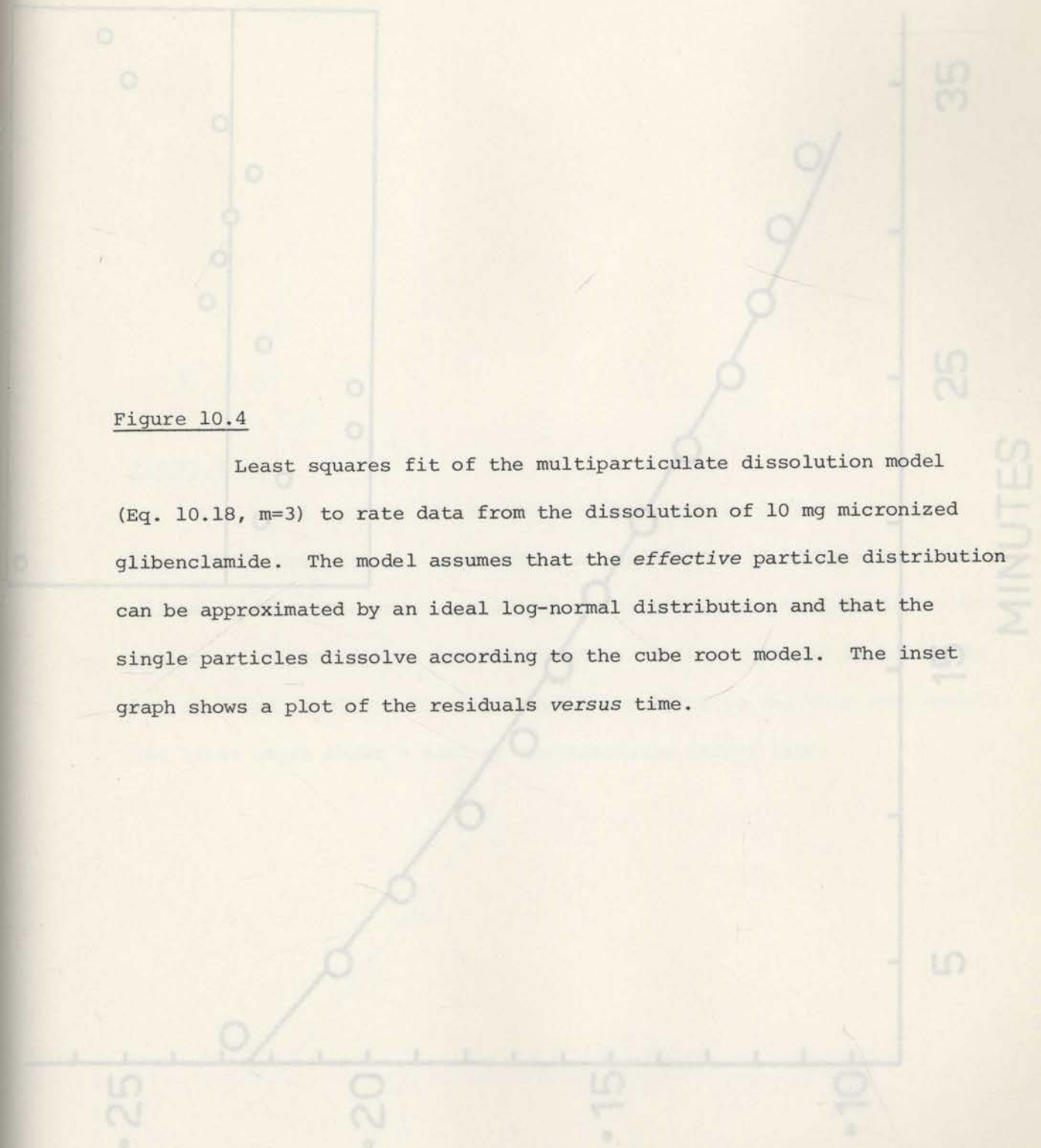
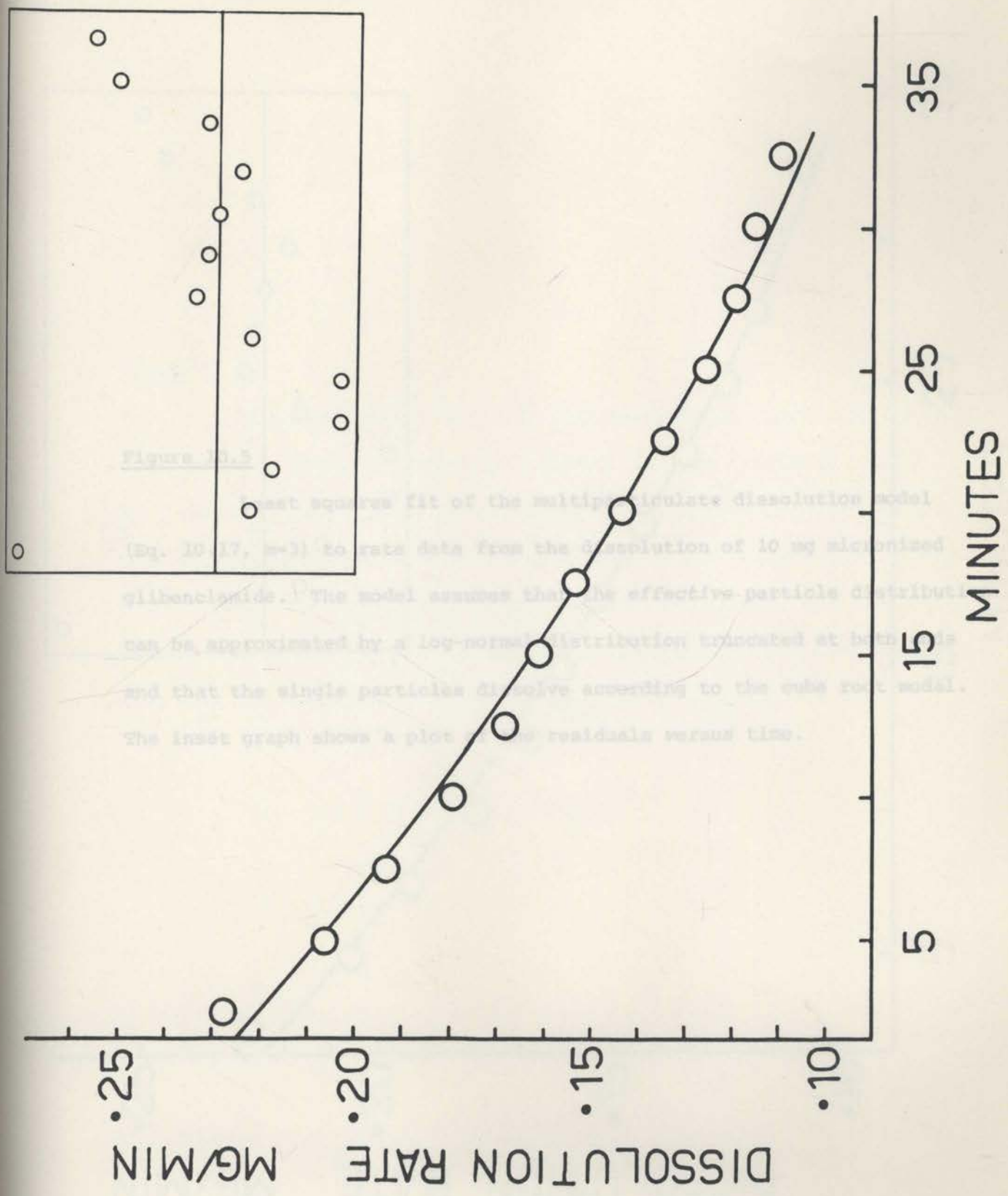


Figure 10.4

Least squares fit of the multiparticulate dissolution model (Eq. 10.18, m=3) to rate data from the dissolution of 10 mg micronized glibenclamide. The model assumes that the *effective* particle distribution can be approximated by an ideal log-normal distribution and that the single particles dissolve according to the cube root model. The inset graph shows a plot of the residuals *versus* time.


DISSOLUTION RATE MG/MIN
.25
.20
.15
.10
5
15
25
35
MINUTES

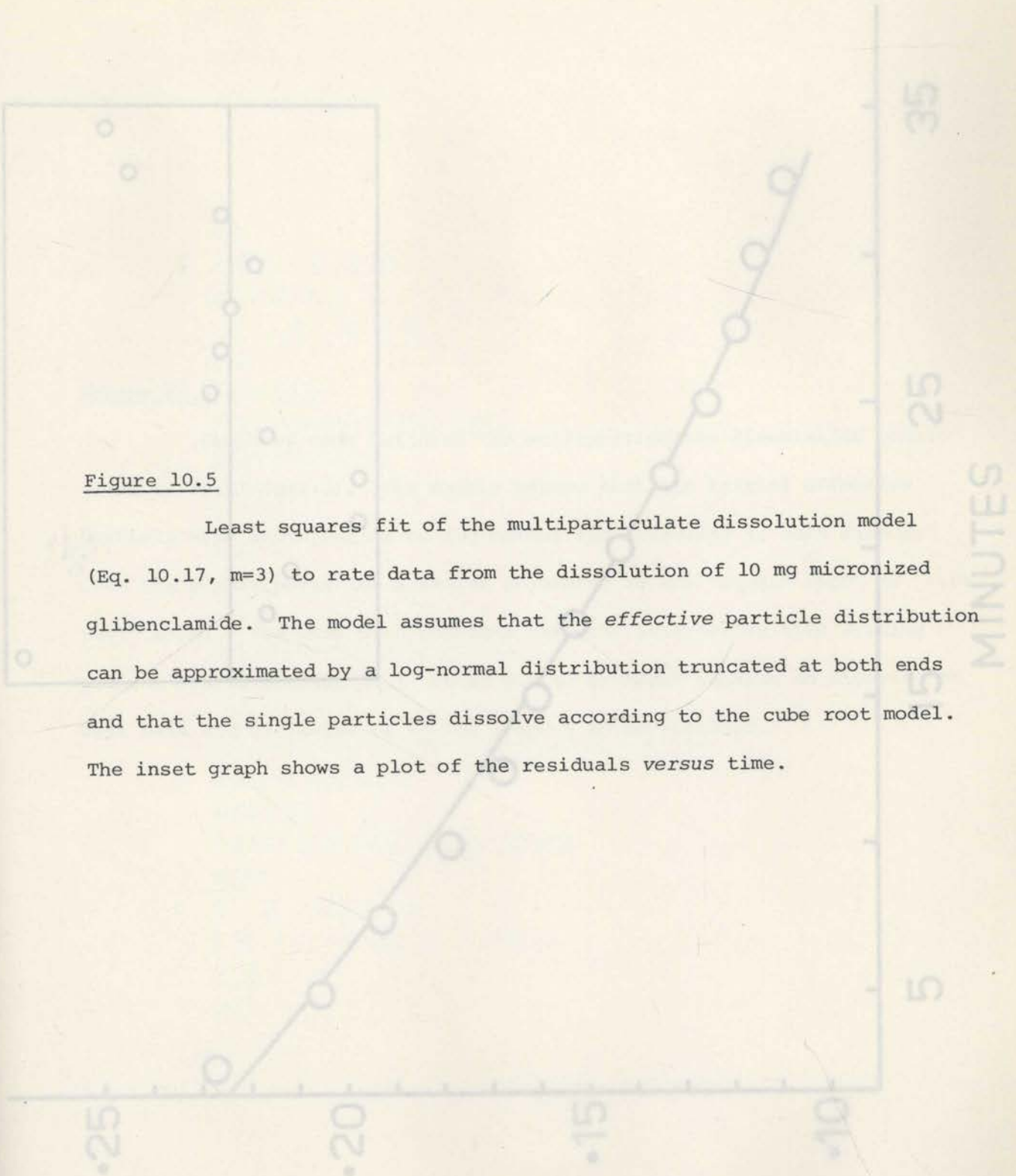


Figure 10.5

Least squares fit of the multiparticulate dissolution model (Eq. 10.17, m=3) to rate data from the dissolution of 10 mg micronized glibenclamide. The model assumes that the *effective* particle distribution can be approximated by a log-normal distribution truncated at both ends and that the single particles dissolve according to the cube root model. The inset graph shows a plot of the residuals *versus* time.

DISSOLUTION RATE MG/MIN
·10
·15
·20
·25
5
15
25
35
MINUTES

Scheme 10.1

Computer code defining the multiparticulate dissolution models given by 10.17 (m=2,3). The models assume that the initial *effective* particle size distribution is log-normal and *truncated at both ends* and that the single particles dissolve according to the "square root" or "cube root" model. The code used together with the interactive time sharing program FUNFIT enables 10.17 to be fitted by least squares to dissolution rate data and estimates of K_m^*, σ, i and j to be obtained.

```
      SUBROUTINE MODEL(Q,T,P,IPRINT)
      DIMENSION P(6)

       P(I) = SIGMA,KSTAR,I,J,W0,M        Q = DISSOLUTION RATE
              1    2   3 4 5  6          T = TIME


      Q=0.
      IF(P(2).LE.0.)RETURN
      M=IFIX(P(6)+.1)
      IF(P(1).EQ.0.)GO TO 4
      AM=M
      B=3.*P(1)/AM
      IF(T.GE.EXP(B*P(4))/P(2))RETURN
      C=EXP(-B*P(3))/P(2)
      DO 1 L=1,M
      N=L-1
      AN=N
      D=B*AN
      IF(T.LT.C)A=SDF(-P(3)-D)
      IF(T.GE.C)A=SDF(ALOG(P(2)*T)/B-D)
    1 Q=Q+FLOAT(NBC(M-1,N))*((-P(2)*T)**(M-N-1))*(SDF(P(4)-D)-A)*
     + EXP(4.5*(AN*AN/AM/AM-1.)*P(1)*P(1))
      Q=P(6)*P(2)*P(5)*Q/(SDF(P(4)-3.*P(1))-SDF(-P(3)-3.*P(1)))
    2 IF(IPRINT.EQ.0)RETURN
      WRITE(6,3)P
    3 FORMAT(' SIGMA,KSTAR,I,J,W0,M='/6E12.4)
      RETURN
    4 IF(T.GE.1./P(2))RETURN
      Q=P(6)*P(2)*P(5)*((1.-P(2)*T)**(M-1))
      GO TO 2
      END
```

Scheme 10.2

Computer code defining the multiparticulate dissolution models given by 10.18 (m=2,3). The models assume that the initial *effective* particle size distribution is *ideal* log-normal and that the single particles dissolve according to the "square root" or "cube root" model. This code used together with the interactive time sharing program FUNFIT enables 10.18 to be fitted by least squares to dissolution rate data and estimates of K_m^* and σ to be obtained.

```
      SUBROUTINE MODEL(Q,T,P,IPRINT)
      DIMENSION P(4)
C
C       P(I) = SIGMA,KSTAR,WØ,M      Q = DISSOLUTION RATE
C                1     2   3  4      T = TIME
C
      Q=Ø.
      IF(P(2).LE.Ø.) RETURN
      M=IFIX(P(4)+.1)
      IF(P(1).EQ.Ø.)GO TO 4
      AM=M
      A=Ø.
      DO 1 L=1,M
      N=L-1
      AN=N
      IF(T.EQ.Ø.)GO TO 1
      A=AM/(3.*P(1))
      A=SDF(A*ALOG(P(2)*T)-AN/A)
1     Q=Q+FLOAT(NBC(M-1,N))*((-P(2)*T)**(M-N-1))*(1.-A)*
     +EXP(4.5*(AN*AN/AM/AM-1.)*P(1)*P(1))
      Q=P(4)*P(2)*P(3)*Q
2     IF(IPRINT.EQ.Ø)RETURN
      WRITE(6,3)P
3     FORMAT(' SIGMA,KSTAR,WØ,M='4E11.4)
      RETURN
4     IF(T.GE.1./P(2))RETURN
      Q=P(4)*P(2)*P(3)*((1.-P(2)*T)**(M-1))
      GO TO 2
      END
```

Scheme 10.3

Computer code defining the multiparticulate dissolution models given by 10.19 (m=2,3). The models assume that the initial *effective* particle size distribution is log-normal and *truncated at the upper end* ($i=\infty, j<\infty$) and that the single particles dissolve according to the "square root" or "cube root" model. The code used together with the interactive time sharing program FUNFIT enables 10.19 to be fitted by least squares to dissolution rate data and estimates of K_m^*, σ and j to be obtained.

```
      SUBROUTINE MODEL(Q,T,P,IPRINT)
      DIMENSION P(5)
C
C           P(I) = SIGMA,KSTAR,J,WØ,M   Q = DISSOLUTION RATE
C                  1     2   3 4  5    T = TIME
C
      Q=Ø.
      IF(P(2).LE.Ø.)RETURN
      M=IFIX(P(5)+.1)
      IF(P(1).EQ.Ø.)GO TO 4
      AM=M
      B=3.*P(1)/AM
      IF(T.GE.EXP(B*P(3))/P(2))RETURN
      A=Ø.
      DO 1 L=1,M
      N=L-1
      AN=N
      IF(T.EQ.Ø.)GO TO 1
      A=SDF(ALOG(P(2)*T)/B-B*AN)
    1 Q=Q+FLOAT(NBC(M-1,N))*((-P(2)*T)**(M-N-1))*(SDF(P(3)-B*AN)-A)*
     +EXP(4.5*(AN*AN/AM/AM-1.)*P(1)*P(1))
      Q=P(5)*P(2)*P(4)*Q/SDF(P(3)-3.*P(1))
    2 IF(IPRINT.EQ.Ø)RETURN
      WRITE(6,3)P
    3 FORMAT(' SIGMA,KSTAR,J,WØ,M= '5E1Ø.4)
      RETURN
    4 IF(T.GE.1./P(2))RETURN
      Q=P(5)*P(2)*P(4)*((1.-P(2)*T)**(M-1))
      GO TO 2
      END
```

simplest of the models, containing only the two parameters K_m^* and σ, and fits the dissolution data just as well as the other models that contain more parameters (i and j). Furthermore, it is clear (Table 10.1) that the multiparticulate dissolution model based on the cube root model (m=3) agrees best with the dissolution data. This is true for either a truncated or ideal distribution using either 5 or 10 mg of powder (Table 10.1). Of the 8 multiparticulate models investigated it can therefore be concluded that the following equation:

$$\frac{dW}{dt} = -3K^*_3 W_o \sum_{n=o}^{2} \binom{2}{n} (-K^*_3 t)^{(2-n)} \left[1-F\left(\frac{1}{\sigma} \ln(K^*_3 t) - n\sigma\right)\right] e^{\frac{\sigma^2}{2}\left[n^2-9\right]} \qquad (10.29)$$

which can be written in integrated form as:

$$\frac{W}{W_o} = \sum_{n=o}^{3} \binom{3}{n} (-K^*_3 t)^{(3-n)} \left[1-F\left(\frac{1}{\sigma} \ln(K^*_3 t) - n\sigma\right)\right] e^{\frac{\sigma^2}{2}\left[n^2-9\right]} \qquad (10.30)$$

best describes the dissolution kinetics of the micronized glibenclamide. These two equations uniquely characterise the dissolution behaviour in terms of the rate parameter, K_3^*, and the distribution parameter, σ.

It appears from the residual plots in Figs. 10.2-10.5 that there is a serial correlation between the residuals. The Durbin-Watson statistic indicates ($\alpha < 0.05$) that this correlation is significant (142). Systematic deviation can be caused by non-random experimental errors. It can also be caused by a significant departure from the assumed single particle dissolution model 10.25 or by a deviation from log-normality. The residual values are however so small in relation to the accuracy of the experimental technique that the correlation seems of little importance.

The specific dissolution rate parameter, K_3^*, should theoretically

be independent of the initial amount used, W_o. This is only found to be approximately true. Values of the rate parameter, K_m^*, obtained for W_o=10 mg are consistently lower than for W_o=5 mg (Table 10.1). The dissolution models consider dissolution under *complete sink conditions* i.e. conditions where there is no interaction between the dissolving particles. Using 10 mg of the very fine powder it has not been possible to load the dissolution cell with a single "layer" of particles in such a way that dissolved drug from any particle does not pass over other particles. The slightly larger K_m^* values observed when less powder is used in the cell agree with an expected smaller particle interaction.

Characterisation and quality control of drug powders

Previous approaches to characterizing the dissolution properties of drug powders have been based on equations describing monodisperse systems. In many cases, the so-called "dissolution rate constant" evaluated using such equations will not be sufficient to characterize the dissolution behaviour because the size distribution effect is not accounted for. This is particularly true for pharmaceutical systems which frequently involve highly polydisperse fine powders.

The use of nonlinear regression analysis to evaluate the specific dissolution rate parameter, K_m^*, and the distribution parameter, σ, represents a more exact and meaningful approach.

The properties of K_m^* make its interpretation particularly meaningful. These properties are best understood in relation to the concepts of *time scaling* and *the intrinsic dissolution profile* from which the following conclusion can be made:[3]

---

3. For example if it takes x min for a powder to dissolve, say 30% for a given K_m^* value, then it will take the powder x/2 minutes to dissolve to the same extent if the value of K_m^* is doubled (for illustration see Fig. 6.1).

If K_m^* is changed by a factor α then the time for complete dissolution, or the time for any particular fraction to dissolve, is changed by a factor of $1/\alpha$. This simple property makes K_m^* a particularly useful parameter.

The distribution parameter, σ, which appropriately should be called *the dispersion parameter,* is a single measure of how polydisperse a powder is, or more exactly, how much the dissolution behaviour deviates from that expected if the powder were completely monodisperse. A value close to zero characterizes a nearly monodisperse powder, while higher values indicate increasing "degrees of dispersion". Probably the most important property of σ is that it is a measure of how long it takes the last fraction of a polydisperse powder to dissolve. For example, it is seen from 10.5 that the time for complete dissolution increases exponentially with σ. For this reason it is most likely that very slightly soluble drugs, which exhibit dissolution rate limited absorption, will show a significant correlation between σ and systemic availability. Research in this area should be of considerable pharmaceutical interest.

Although the multiparticulate dissolution model defining K_m^* and σ may seem complex, the interpretation of these parameters is simple and they can be readily obtained. The experimental technique used requires a high precision, flow-through dissolution apparatus which is easy to standardize in combination with a nonlinear regression program.

The method could well become established as a routine procedure in quality control and further investigation could eventually result in improved standards for drug dissolution.

CHAPTER 11

A METHOD OF OBTAINING DRUG-MACROMOLECULE BINDING PARAMETERS DIRECTLY FROM DYNAMIC DIALYSIS DATA

The dynamic dialysis method for characterizing interactions of small molecules with macromolecules is well established (167-181). It has a number of advantages compared to equilibrium dialysis and ultrafiltration. A complete binding profile can be obtained rapidly in one experiment and the method utilizes only a small sample of macromolecule. As the method is based on a dynamic process, an equilibrium state does not need to be defined and compared to ultrafiltration there is no change in concentration of macromolecule.

Meyer and Guttman (169) designed a dynamic dialysis method to characterize drug-protein interactions. However this method has a number of limitations. The experimental data must be differentiated to evaluate binding parameters. It is recognised in numerical analysis that differentiation of discrete data may introduce substantial errors particularly if the number of data points is limited. An empirical equation was used to fit dialysis data to obtain instantaneous rates. A recent publication (179) has shown that the various empirical equations used can yield substantially different binding parameters.

The technique of Meyer and Guttman (169) requires that the rate constant for dialysis be determined in a separate experiment in the absence of macromolecules. It is assumed that the same rate constant will apply in the presence of macromolecules. This may be an unreasonable assumption as the rate constant depends on several factors such as the physico-chemical

state of the dialysis membrane (182,[183]) which may change between runs.

A number of compounds are significantly bound to the membrane material (171,177). Using previous methods it has not been possible to determine the dialysis rate constant and account for the membrane binding in the determination of macromolecule binding parameters of such compounds.

In this chapter a new approach is presented which rigorously describes the total kinetics of the system in a form that enables binding parameters to be estimated accurately, directly from dialysis data. It eliminates the need to determine an accurate dialysis rate constant in a separate experiment. The method does not rely on differentiation of experimental data and should be applicable to compounds that are membrane bound.

THEORY

Consider an interaction between small molecules and macromolecules which can be described by the general binding expression:

$$\bar{\nu} = \sum_{i=1}^{j} \frac{n_i K_i D_f}{1+K_i D_f} \qquad (11.1)$$

where $\bar{\nu}$ is the number of moles of small molecules bound per mole macro molecule, n_i is the number of binding sites in the i-th class of sites, K_i is the association constant for the interaction and D_f is the molar concentration of unbound small molecules. If a model with two classes (j=2) is assumed then the total concentration of drug, D_t, in the protein compartment is given by (171):

$$D_t = D_f + P_t D_f \left[\frac{n_1 K_1}{1+K_1 D_f} + \frac{n_2 K_2}{1+K_2 D_f} \right] \qquad (11.2)$$

If sink conditions prevail the small molecules will leave the

protein compartment by a first order process:

$$\frac{dD_t}{dt} = -K_e D_f \tag{11.3}$$

where K_e is the dialysis rate constant.

It is convenient to introduce a variable, s, defined as:

$$s = -\frac{dD_t}{dt} \tag{11.4}$$

so that $D_f = s/K_e$ and Eq. 11.6 can be written:

$$D_t = \frac{s}{K_e} + P_t s \left[\frac{n_1 K_1}{K_e + K_1 s} + \frac{n_2 K_2}{K_e + K_2 s}\right] \tag{11.5}$$

Taking the differential of this equation, noting $dD_t = -s\,dt$, it becomes

$$dt = -\left[\frac{1}{K_e s} + P_t K_e \left(\frac{n_1 K_1}{s(K_e + K_1 s)^2} + \frac{n_2 K_2}{s(K_e + K_2 s)^2}\right)\right] ds \tag{11.6}$$

which integrated from t=0 to t corresponding to $s = s_o$ to s yields:

$$t = \left|\frac{1}{K_e} \ln s + P_t n_1 K_1 \left[\frac{1}{K_e + K_1 s} + \frac{1}{K_e} \ln\left(\frac{s}{K_e + K_1 s}\right)\right]\right. \tag{11.7}$$

$$\left. + P_t n_2 K_2 \left[\frac{1}{K_e + K_2 s} + \frac{1}{K_e} \ln\left(\frac{s}{K_e + K_2 s}\right)\right]\right|_{s}^{s_o}$$

If the following function is defined:

$$f(x) = \frac{1}{K_e} \ln x + P_t n_1 K_1 \left[\frac{1}{K_e + K_1 x} + \frac{1}{K_e} \ln\left(\frac{x}{K_e + K_1 x}\right)\right. \tag{11.8}$$

$$\left. + P_t n_2 K_2 \left[\frac{1}{K_e + K_2 x} + \frac{1}{K_e} \ln\left(\frac{x}{K_e + K_2 x}\right)\right]\right.$$

then Eq. 11.7 can be written more simply as:

$$t = f(s_o) - f(s) \tag{11.9}$$

The exact functional relationship describing the change of D_t with t is now described by Eq's 11.5 and 11.9 in *parametric form* where the variable s is the parameter. Each value of s defines by these equations a *unique* pair of D_t and t values.

The quantity s_o is the initial (t=0) value of $-dD_t/dt$ (Eq. 11.4), which would normally be determined by extrapolation. To avoid the errors and problems of such an extrapolation *it is convenient to define t=0 at the first sampling time. In this way* s_o *is* $-dD_t/dt$ *at first sampling.*

In order to determine the binding parameters by nonlinear regression it is necessary to define the exact functional relationship between D_t and t, for any values of n_1, K_1, n_2, K_2, s_o and K_e which are changing during the nonlinear fitting procedure. This can be done by determining the particular values of s which satisfy Eq. 11.9. These values are then used to determine the corresponding values of D_t by Eq. 11.5.

However Eq. 11.9 cannot be expressed explicitly in terms of s so some iterative procedure is needed to solve for s. The Newton-Raphson algorithm is particularly suitable because it is computationally compact and exhibits quadratic convergence.

If
$$\phi(s) = t - f(s_o) + f(s) \tag{11.10}$$
then Eq. 11.9 can be solved by the Newton-Raphson method using the following iteration[1]:
$$s_{i+1} = s_i - \frac{\phi(s_i)}{\phi'(s_i)} \tag{11.11}$$
where the functions ϕ and ϕ' are defined by:

---

1. The function ϕ is not defined for $s \leq 0$ so it is necessary during the iteration procedure to prevent s from taking a nonpositive value. This is conveniently done by defining $s_{i+1} = s_i/2$ if $s_{i+1} \leq 0$ since ϕ is a monotone increasing function of s because $\phi' = f' > 0$ $(s>0)$.

$$\phi'(s) = f'(s) = \frac{1}{K_e s} + \frac{P_t K_e}{s}\left[\frac{n_1 K_1}{(K_e+K_1 s)^2} + \frac{n_2 K_2}{(K_e+K_2 s)^2}\right] \quad (11.12)$$

$$\phi(s) = t + \frac{1}{K_e}\ln\frac{s}{s_o} + P_t n_1 K_1\left[\frac{K_1(s_o-s)}{(K_e+K_1 s)(K_e+K_1 s_o)} + \frac{1}{K_e}\ln\frac{s}{s_o}\frac{(K_e+K_1 s_o)}{(K_e+K_1 s)}\right] \quad (11.13)$$

$$+ P_t n_2 K_2\left[\frac{K_2(s_o-s)}{(K_e+K_2 s)(K_e+K_2 s_o)} + \frac{1}{K_e}\ln\frac{s}{s_o}\frac{(K_e+K_2 s_o)}{(K_e+K_2 s)}\right]$$

In order for the algorithm (11.11) to converge within a given number of iterations it is necessary that the initial estimates of s are not too far from the s values for which 11.9 is satisfied. These initial estimates can, according to 11.4, be obtained as the (positive) values of $-dD_t/dt$ estimated from the observed values of D_t *vs.* t. Any simple technique for slope estimation can be used since the actual accuracy of the estimates is of no importance for the final result.

The above derivations for 2 binding classes can easily be extended to any number (j) of classes for which then:

$$D_t = \frac{s}{K_e} + P_t s \sum_{i=1}^{j} \frac{n_i K_i}{K_e+K_i s} \quad (11.14)$$

$$t = \left| \frac{1}{K_e}\ln s + P_t \sum_{i=1}^{j} n_i K_i\left[\frac{1}{K_e+K_i s} + \frac{1}{K_e}\ln\left(\frac{s}{K_e+K_i s}\right)\right]\right|_{s}^{s_o} \quad (11.15)$$

and the iteration, 11.11, can still be used with

$$\phi'(s) = \frac{1}{K_e s} + \frac{P_t K_e}{s}\sum_{i=1}^{j}\frac{n_i K_i}{(K_e+K_i s)^2} \quad (11.16)$$

and

$$\phi(s) = t + \frac{1}{K_e} \ln \frac{s}{s_o} \qquad (11.17)$$

$$+ P_t \sum_{i=1}^{j} n_i K_i \left[\frac{K_i (s_o - s)}{(K_e + K_i s)(K_e + K_i s_o)} + \frac{1}{K_e} \ln \frac{s(K_e + K_i s_o)}{s_o (K_e + K_i s)} \right]$$

Binding of Small Molecules by the Dialysis Membrane

The treatment outlined above is based on an assumption that binding occurs only to protein. However some drugs, particularly those strongly protein bound can become significantly bound to the dialysis membrane. This means that K_e cannot be estimated from plots of ln D_t *vs.* t in the absence of protein using previous techniques because of curvature (Fig. 11.1).

A special technique is therefore required to estimate the dialysis rate constant and the membrane binding must be taken into account in the treatment of the small molecule-macromolecule system.

Determination of K_e and membrane binding parameters in absence of macromolecule.

The binding of small molecules to the membrane can often be considered as a Langmuir type absorption phenomenon (171), which is mathematically analogous to binding to a single class of sites, and can be described by the equation:

$$\bar{\nu}^* = \frac{n^* K^* D_f}{1 + K^* D_f} \qquad (11.18)$$

where $\bar{\nu}^*$ is the *amount* of small molecules bound *per amount* of available membrane material and K* is the association constant for membrane binding.

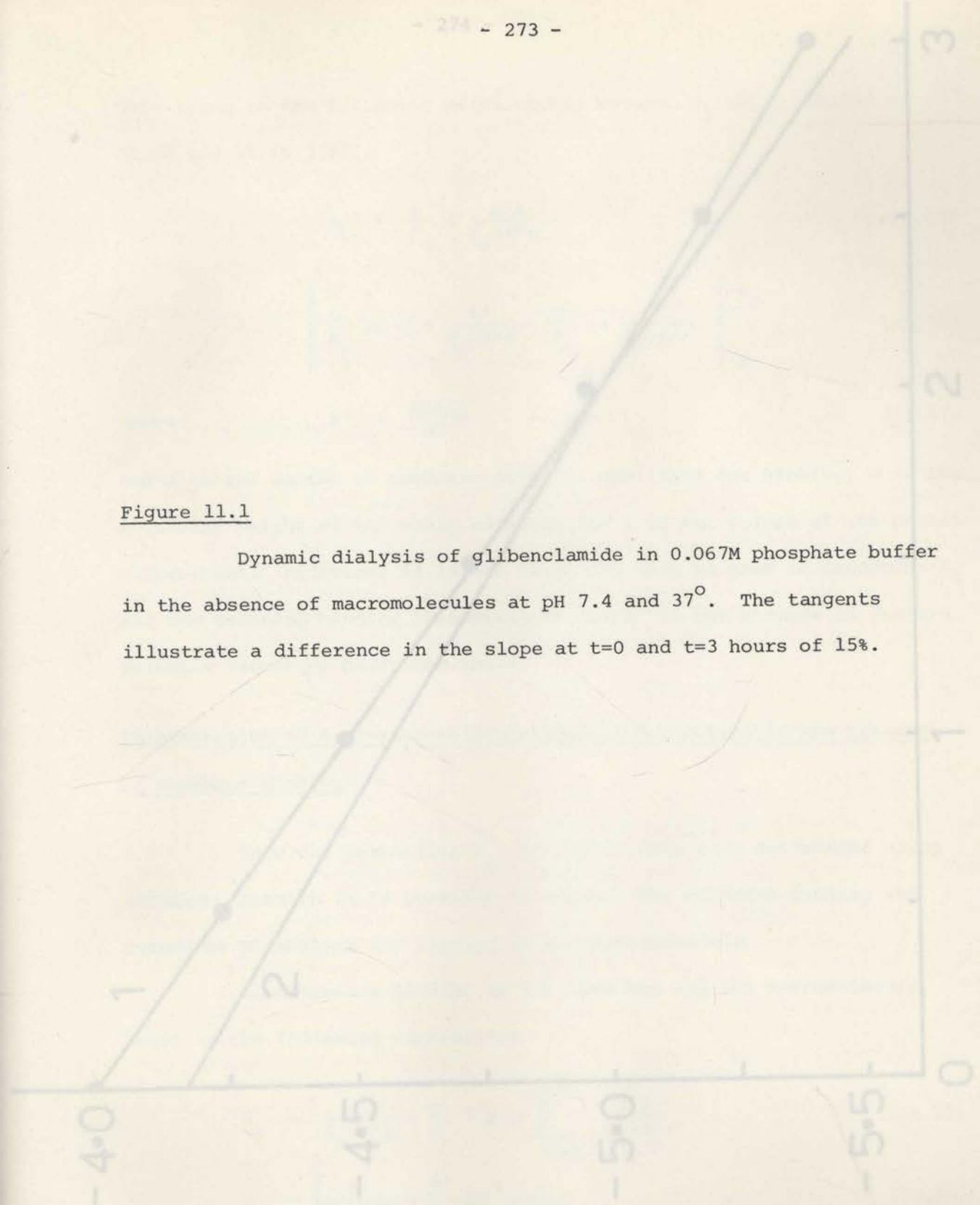
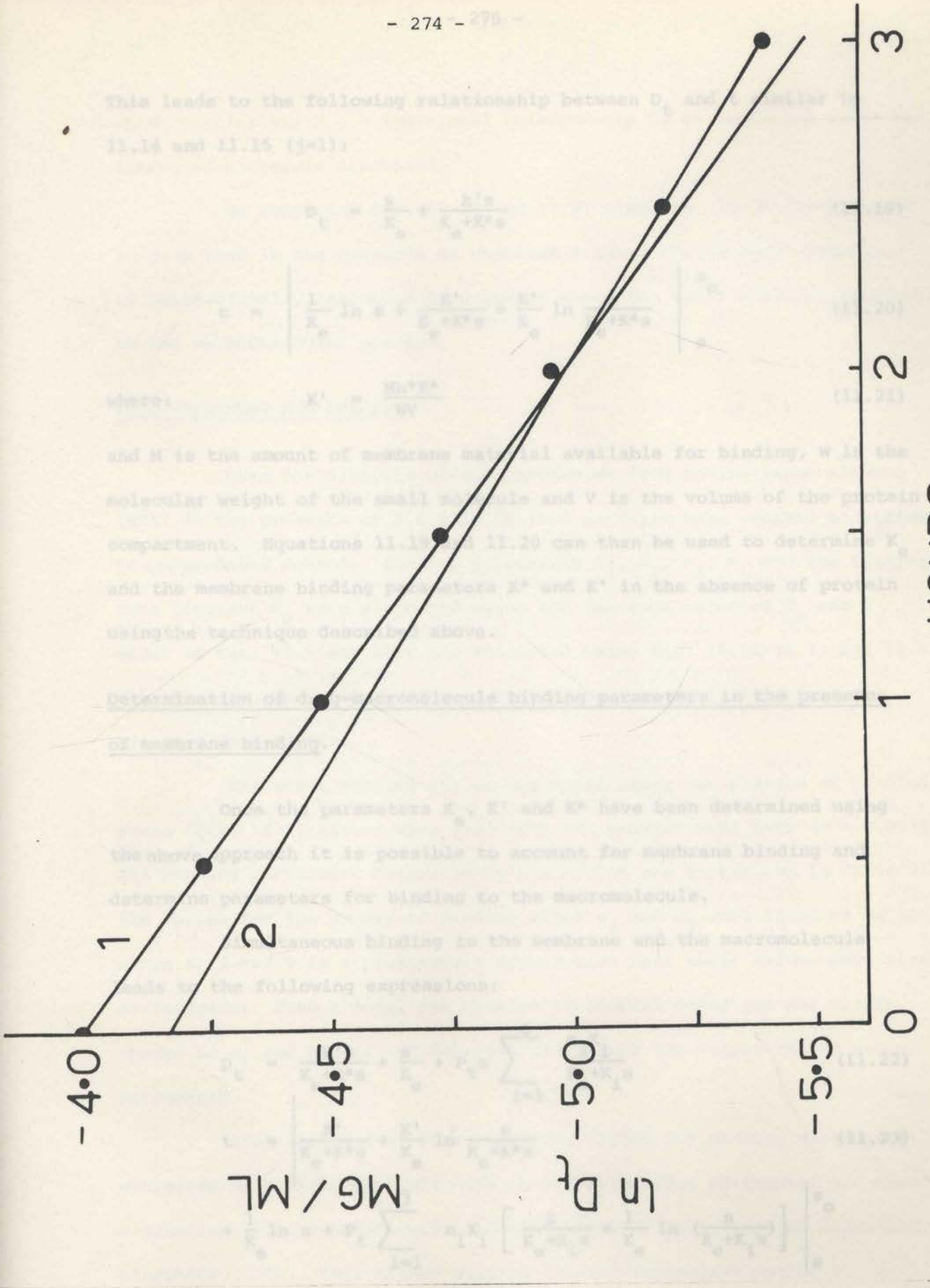


Figure 11.1

Dynamic dialysis of glibenclamide in 0.067M phosphate buffer in the absence of macromolecules at pH 7.4 and 37°. The tangents illustrate a difference in the slope at t=0 and t=3 hours of 15%.


1
2
-4·0
-4·5
-5·0
-5·5
MG / ML
ln D_t
0
1
2
3
HOURS

This leads to the following relationship between D_t and t similar to 11.14 and 11.15 (j=1):

$$D_t = \frac{s}{K_e} + \frac{k's}{K_e+K^*s} \tag{11.19}$$

$$t = \left| \frac{1}{K_e} \ln s + \frac{K'}{K_e+K^*s} + \frac{K'}{K_e} \ln \frac{s}{K_e+K^*s} \right|_{s}^{s_o} \tag{11.20}$$

where: $$K' = \frac{Mn^*K^*}{WV} \tag{11.21}$$

and M is the amount of membrane material available for binding, W is the molecular weight of the small molecule and V is the volume of the protein compartment. Equations 11.19 and 11.20 can then be used to determine K_e and the membrane binding parameters K* and K' in the absence of protein usingthe technique described above.

Determination of drug-macromolecule binding parameters in the presence of membrane binding.

Once the parameters K_e, K' and K* have been determined using the above approach it is possible to account for membrane binding and determine parameters for binding to the macromolecule.

Simultaneous binding to the membrane and the macromolecule leads to the following expressions:

$$D_t = \frac{K's}{K_e+K^*s} + \frac{s}{K_e} + P_t s \sum_{i=1}^{i} \frac{n_i K_i}{K_e+K_i s} \tag{11.22}$$

$$t = \left| \frac{K'}{K_e+K^*s} + \frac{K'}{K_e} \ln \frac{s}{K_e+K^*s} + \frac{1}{K_e} \ln s + P_t \sum_{i=1}^{j} n_i K_i \left[\frac{1}{K_e+K_i s} + \frac{1}{K_e} \ln \left(\frac{s}{K_e+K_i s} \right) \right] \right|_{s}^{s_o} \tag{11.23}$$

which enables the D_t, t functional relationship to be evaluated using the iterative procedure discussed.

By comparing Eqs. 11.22 and 11.23 with Eqs. 11.14 and 11.15 it is seen that in the presence of membrane binding the dialysis behaviour is mathematically analogous to a system where the small molecule is binding to two macromolecular species.

DATA TREATMENT AND RESULTS

Data for dialysis of chlorpropamide from bovine serum albumin (BSA) in the presence of 1.6×10^{-5} M free warfarin were treated according to the proposed method. Binding parameters n_1, K_1, n_2, K_2 and the dialysis rate constant K_e were estimated using the function relating D_t and t given in Eqs. 11.5 and 11.9 and evaluated using Eqs. 11.11-11.13 and 11.5. This function was programmed in a subroutine (scheme 11.1) which was executed with FUNFIT.

The least squares fit of the model using two classes of binding sites (Fig. 11.2) agreed very well with the experimental data (r = 0.99986). The binding parameters estimated by the method are summarised in Table 11.1. The values for the number of binding sites n_1 and n_2 were found to be so close to 2 and 9 in a preliminary computation that their values were fixed as integers. Such a model has greater conceptual value and the slight change in n_1 and n_2 did not substantially alter the values of the other parameters.

Also included in Table 11.1 are values for binding parameters estimated by fitting dialysis data to a fourth order polynomial and then evaluation using a modification of the method of Hart (184) as described elsewhere (175). This method will be denoted derivative method I. A

Scheme 11.1

The subroutine MODEL defines the functional relationship between D_t and t in a dynamic dialysis process. This function, given in parametric form by Eqs. 11.5 and 11.9, is calculated according to Eqs. 11.11-11.13 and 11.5. The functions PHI and PHIDER used in MODEL defines $\phi(s)$ and $\phi'(s)$ as given by Eqs. 11.13 and 11.12. The subroutine which is written specifically for FUNFIT enables least squares estimates of the binding parameters n_1, K_1, n_2, K_2 (Eq. 11.1) and the dialysis rate constant K_e (Eq. 11.3) to be determined directly from dynamic dialysis data (Fig. 11.2). The subroutine specifies s as a second "independent variable", x(2), which is used as an initial value in the iteration, Eq. 11.11, that in less than MAX cycles should determine s to an accuracy corresponding to an accuracy of t better than 10^{-6}%. The last two parameters (P(7) = total macromolecule concentration and P(8) = MAX) of the eight formal parameters are fixed during the least squares fitting procedure. A value of 50 for MAX should be sufficient to reach convergence (Eq. 11.11) even for pure starting values of P(1) to P(6). However if convergence is not reached IPRINT is made equal to 1, the current values of the parameters are printed and the control is returned to the main program so that appropriate action can be taken.

```
      SUBROUTINE MODEL (DT,X,P,IPRINT)
      DIMENSION X(2),P(8)

C
C  P=N1,K1,N2,K2,SØ,KE,PT,MAX   DT = TOTAL CONC. OF SMALL MOLECULES
C    1  2  3  4  5  6  7  8     X(1) = TIME
C                                X(2) = SLOPE ESTIMATE
C
C

      S=X(2)
      MAX=IFIX(P(8))
      DO 1 I=1,MAX
      SSAVE=S
      PHIVAL=PHI(S,P,X(1))
      S=S-PHIVAL/PHIDER(S,P)
      IF(S.LE.Ø.)S=SSAVE/2.
      TIME=X(1)
      IF(TIME.EQ.Ø.)TIME=1.
      IF(ABS(PHIVAL)/TIME.LT.1E-8) GO TO 2
    1 CONTINUE
      IPRINT=1
    2 DT=S/P(6) + P(7)*S*(P(1)*P(2)/(P(6)+P(2)*S)+
     +                P(3)*P(4)/(P(6)+P(4)*S))
      IF(IPRINT.EQ.Ø)RETURN
      WRITE(6,3)P
    3 FORMAT(' N1,K1,N2,K2,SØ,KE,PT,MAX =',(4E11.5))
      RETURN
      END

C
C
      FUNCTION PHIDER(S,P)
      DIMENSION P(8)
      PHIDER=1./(P(6)*S) + (P(7)*P(6)/S)*(P(1)*P(2)/((P(6)+P(2)*S)**2)+
     +                                    P(3)*P(4)/((P(6)+P(4)*S)**2))
      RETURN
      END
C
C
      FUNCTION PHI(S,P,T)
      DIMENSION P(8)
      A=ALOG(S/P(5))/P(6)
      B=P(5)-S
      C=P(6)+P(2)*P(5)
      D=P(6)+P(2)*S
      E=P(6)+P(4)*P(5)
      F=P(6)+P(4)*S
      PHI=T+A+P(7)*P(1)*P(2)*(P(2)*B/(C*D)+A+ALOG(C/D)/P(6))+
     +       P(7)*P(3)*P(4)*(P(4)*B/(E*F)+A+ALOG(E/F)/P(6))
      RETURN
      END
```

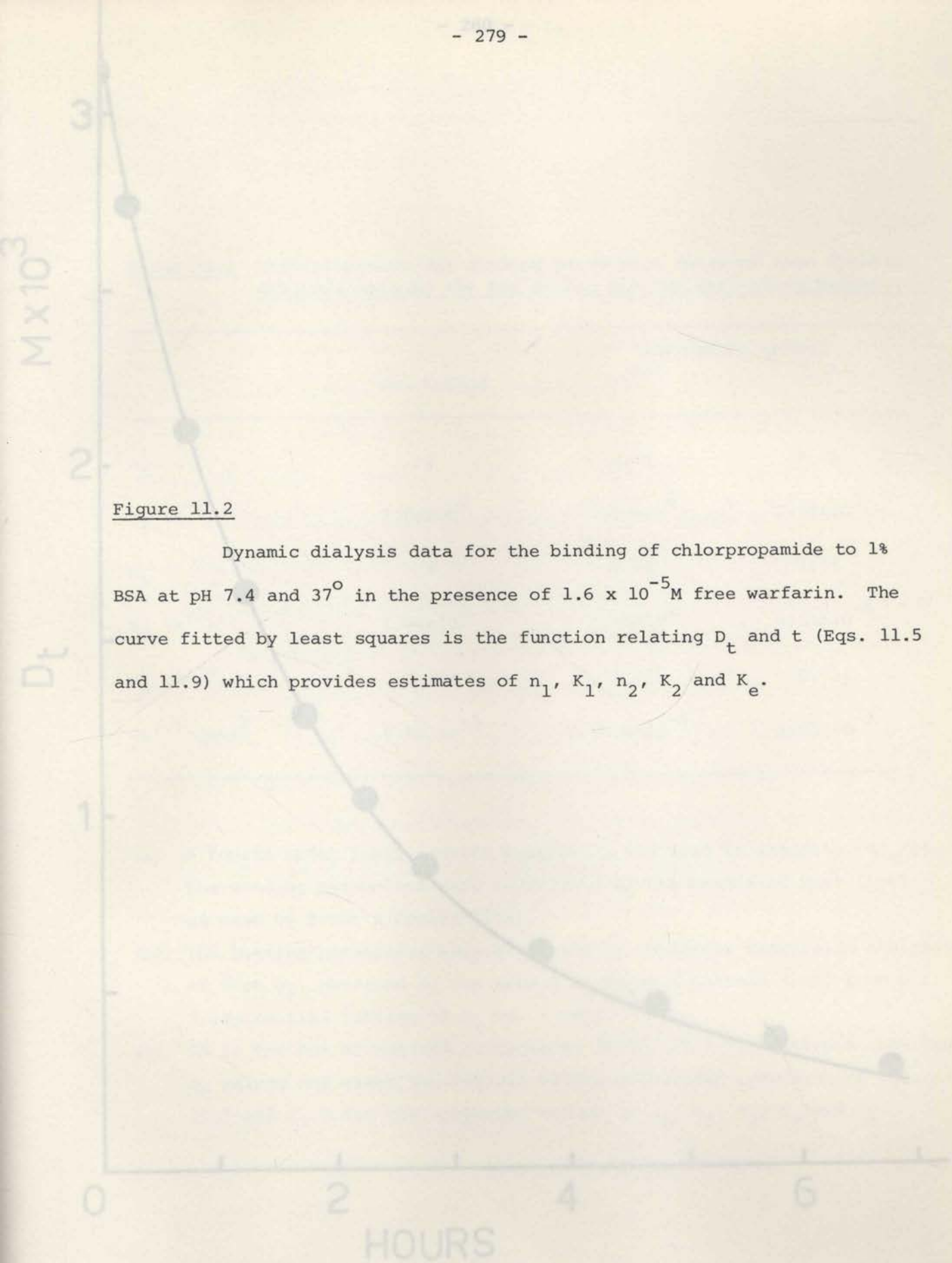


Figure 11.2

Dynamic dialysis data for the binding of chlorpropamide to 1% BSA at pH 7.4 and 37^{o} in the presence of 1.6 x 10^{-5}M free warfarin. The curve fitted by least squares is the function relating D_t and t (Eqs. 11.5 and 11.9) which provides estimates of n_1, K_1, n_2, K_2 and K_e.

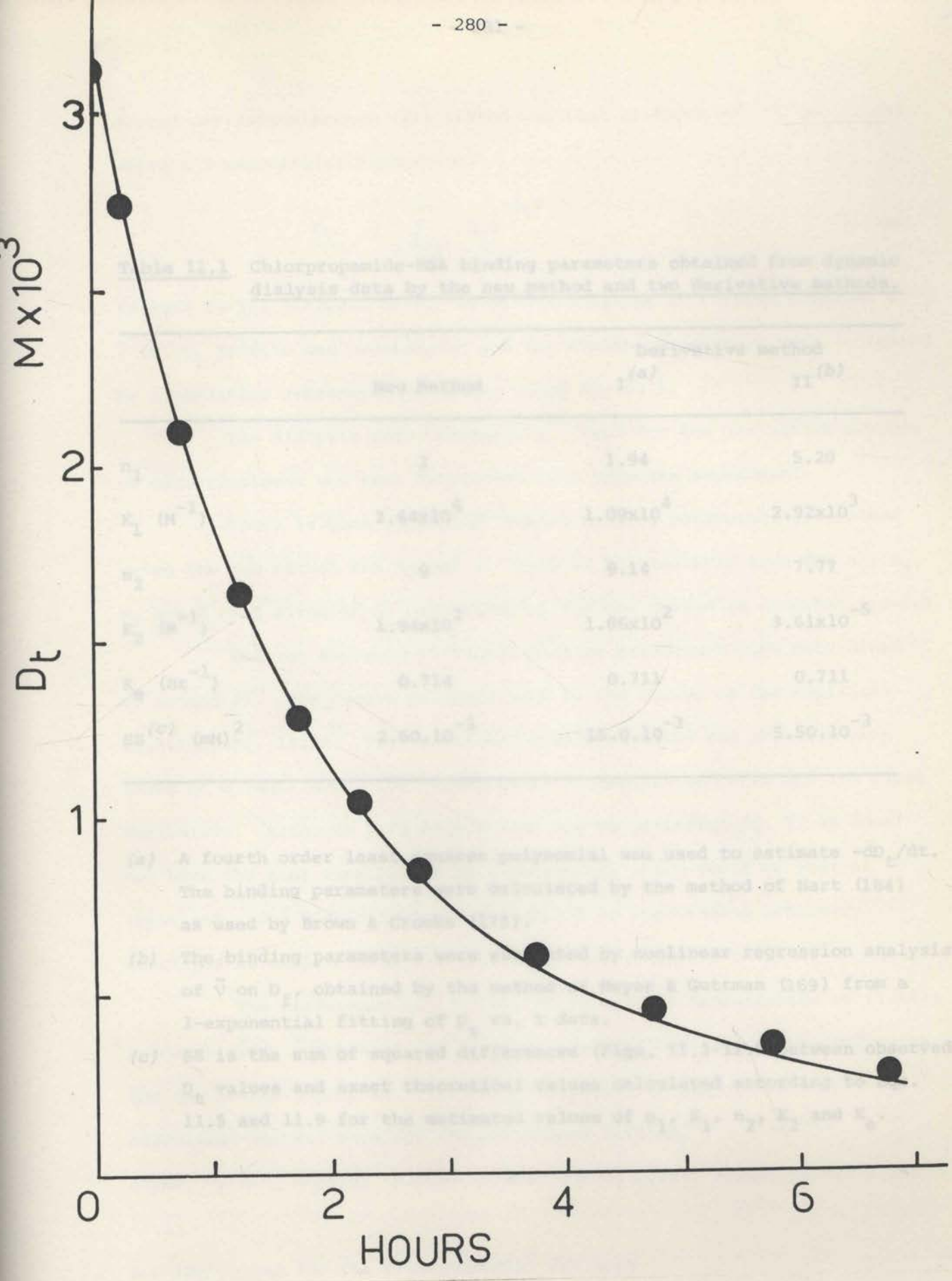

M x 10^3
3
2
1
D_t
0
2
4
6
HOURS

Table 11.1 Chlorpropamide-BSA binding parameters obtained from dynamic dialysis data by the new method and two derivative methods.

	New Method	Derivative method I[(a)]	Derivative method II[(b)]
n_1	2	1.94	5.28
K_1 (M^{-1})	2.64×10^4	1.09×10^4	2.92×10^3
n_2	9	9.14	7.77
K_2 (M^{-1})	1.94×10^2	1.86×10^2	3.61×10^{-5}
K_e (Hr^{-1})	0.714	0.711	0.711
SS[(c)] $(mM)^2$	$2.60.10^{-3}$	$15.0.10^{-3}$	$5.50.10^{-3}$

(a) A fourth order least squares polynomial was used to estimate $-dD_t/dt$. The binding parameters were calculated by the method of Hart (184) as used by Brown & Crooks (175).

(b) The binding parameters were estimated by nonlinear regression analysis of $\bar{\nu}$ on D_f, obtained by the method of Meyer & Guttman (169) from a 3-exponential fitting of D_t vs. t data.

(c) SS is the sum of squared differences (Figs. 11.3-11.5) between observed D_t values and exact theoretical values calculated according to Eqs. 11.5 and 11.9 for the estimated values of n_1, K_1, n_2, K_2 and K_e.

second derivative method (II) tested was that of Meyer and Guttman (169) where a 3-exponential expression:

$$D_t = \sum_{i=1}^{3} A_i e^{-\alpha_i t} \qquad (11.24)$$

is used to fit dialysis data. By differentiation at various t values a $\bar{\nu}$ vs. D_f profile was constructed and the binding parameters were estimated by a nonlinear regression technique using Eq. 11.1.

The dialysis rate constant, K_e, used for the derivative methods of data treatment was that determined in a separate experiment.

There is good agreement between binding parameters determined using the new method and method I. This is particularly true for n_1, n_2, K_2 and K_e and although K_1 determined by the new method is somewhat greater.

However entirely different binding parameters were determined by method II. The reason probably lies in the choice of the empirical equation, Eq. 11.24. The selection of this equation has probably been based on a requirement for "smoothness" of the fitted curve and its first derivative. Although this requirement may be satisfied, Eq. 11.24 does not have the same flexibility as a polynomial which could be just as important. The flexibility of polynomials to approximate arbitrary functions is explained by the well-known Taylor series theorem.

The difference in flexibility is clearly demonstrated by fitting a fourth order polynomial and 11.24 to *exact* dialysis data obtained using the new method (Table 11.1). The D_t residuals (expressed as % of calculated values) were for the polynomial fitting:

0.286, -0.217, -0.519, -0.0609, 0.466, 0.779, 0.381, -1.49, -1.42, 3.14, -1.12

and the values for the triexponential fit were:

1.73, 1.91. 1.46, 3.38, -1.19, -2.65, -3.91, -4.55, -2.19, 3.10, 10.7.

This shows that the polynomial is considerably more flexible. The triexponential fitting resulted in significant systematic deviation in residuals leading to bias in the slope values and D_t values and therefore a bias in the final results. The residual sum of squares (mM^2) were $5.85.10^{-4}$ and $1.04.10^{-2}$ for the polynomial and triexponential fitting respectively.

A further disadvantage of using 11.24 is that multiple solutions are possible because this equation is nonlinear (in α) and therefore may result in several sum-of-squares minima. This is not the case with a polynomial which has a *unique* least squares solution.

The fitting of 11.24 to the generated, exact D_t, t data was repeated several times with different initial estimates for A_i and α_i but the same solution was obtained each time suggesting that the above fit is the best possible using the triexponential equation.

The use of a lease squares polynomial to represent dialysis data, is on the other hand, expected to be less suitable than the triexponential when the experimental errors are large or where there are significant 'gaps' between observation points. This is due to the fact that the ordinary least squares polynomial fitting completely disregards derivative values. The derivative values are commonly found to be in large error at first and last observation points and just before or after 'gaps' in the data. This disadvantage of polynomials can be reduced considerably by imposing constraints on the derivative values by using least squares spline polynomials.

Such polynomials will compete favourably with 11.24 on data with large errors and 'gaps' particularly considering the fact that the problem of multiple minima, using 11.24, is much larger for large residual problems.

However, no matter which empirical equation is used the results obtained will theoretically never be as exact as obtained using the *true* equation as in fact done in the proposed method.

To investigate the bias introduced by using method I and II the exact D_t, t profile was calculated using the parameter values (Table 11.1) obtained using the two methods. The differences between the observed D_t values and the calculated values (Figs. 11.4 and 11.5) shows that the residuals are significantly larger in those methods than in the new method (Fig. 11.3). The residuals from method I are particularly biased in a positive direction although their pattern resembles that from the new method (Fig. 11.3).

On the other hand the residuals from method II (Fig. 11.5) show an entirely different pattern consistent with the fact that the binding parameters obtained using method II represent an entirely different solution.

A final check on the bias introduced by method I and II was made by applying them to *exact* D_t, t data generated from the parameter values (Table 11.1) obtained using the new method.

Method I found binding parameters relatively close to the true values although K_1 seems to be somewhat different (Table 11.2). Method II found however an entirely different solution that is similar to the solution obtained using the real experimental data.

The value for the dialysis rate constant, K_e, estimated by the new method agrees very well with the value determined experimentally (Table 11.1). As stated previously this may not always be true as the permeability of the membrane may change between experiments.

Chlorpropamide does not appear to be significantly membrane bound.

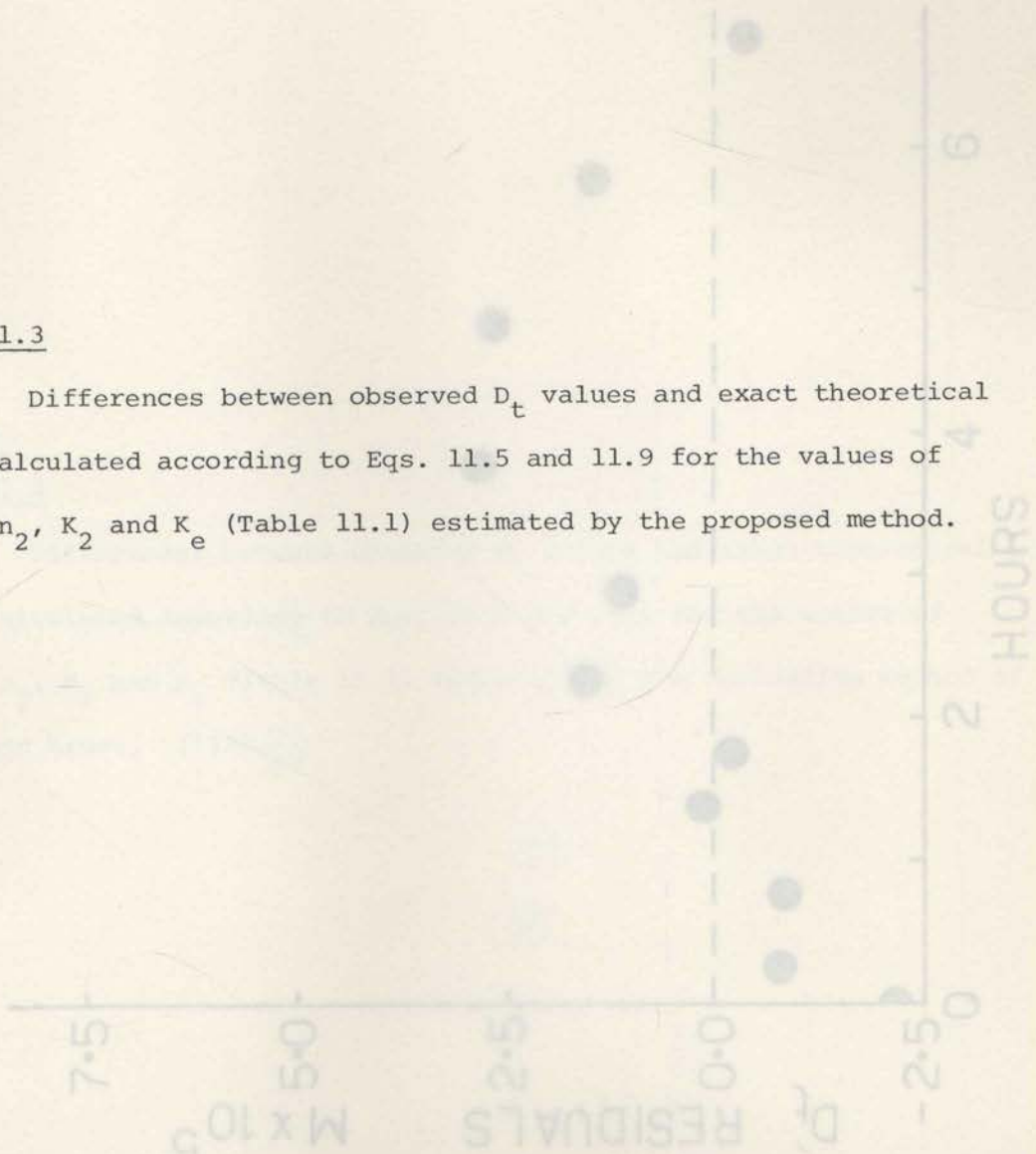
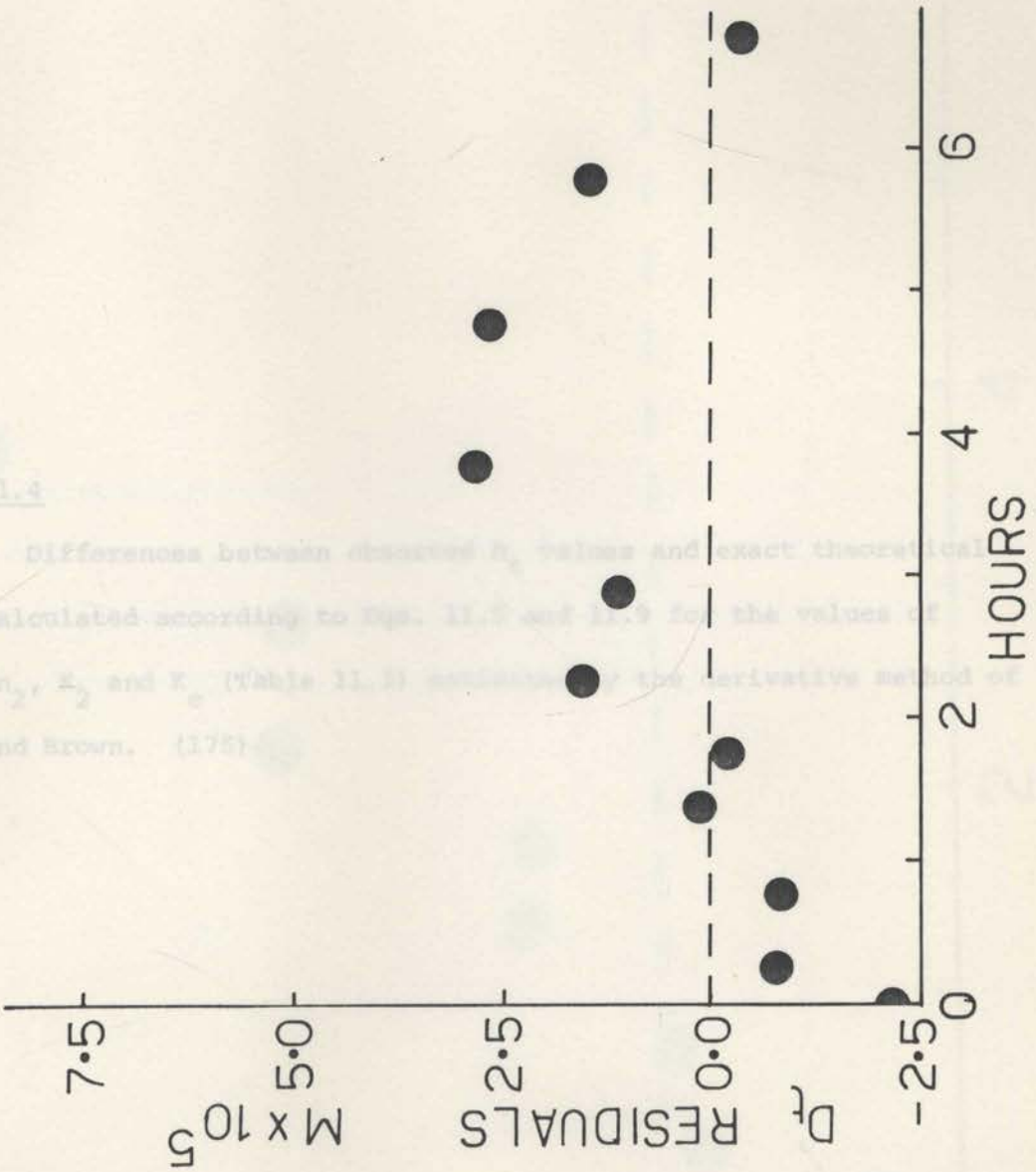


Figure 11.3

Differences between observed D_t values and exact theoretical values calculated according to Eqs. 11.5 and 11.9 for the values of n_1, K_1, n_2, K_2 and K_e (Table 11.1) estimated by the proposed method.


7·5
5·0
2·5
0·0
-2·5
M x 10^5
D_t RESIDUALS
0
2
4
6
HOURS

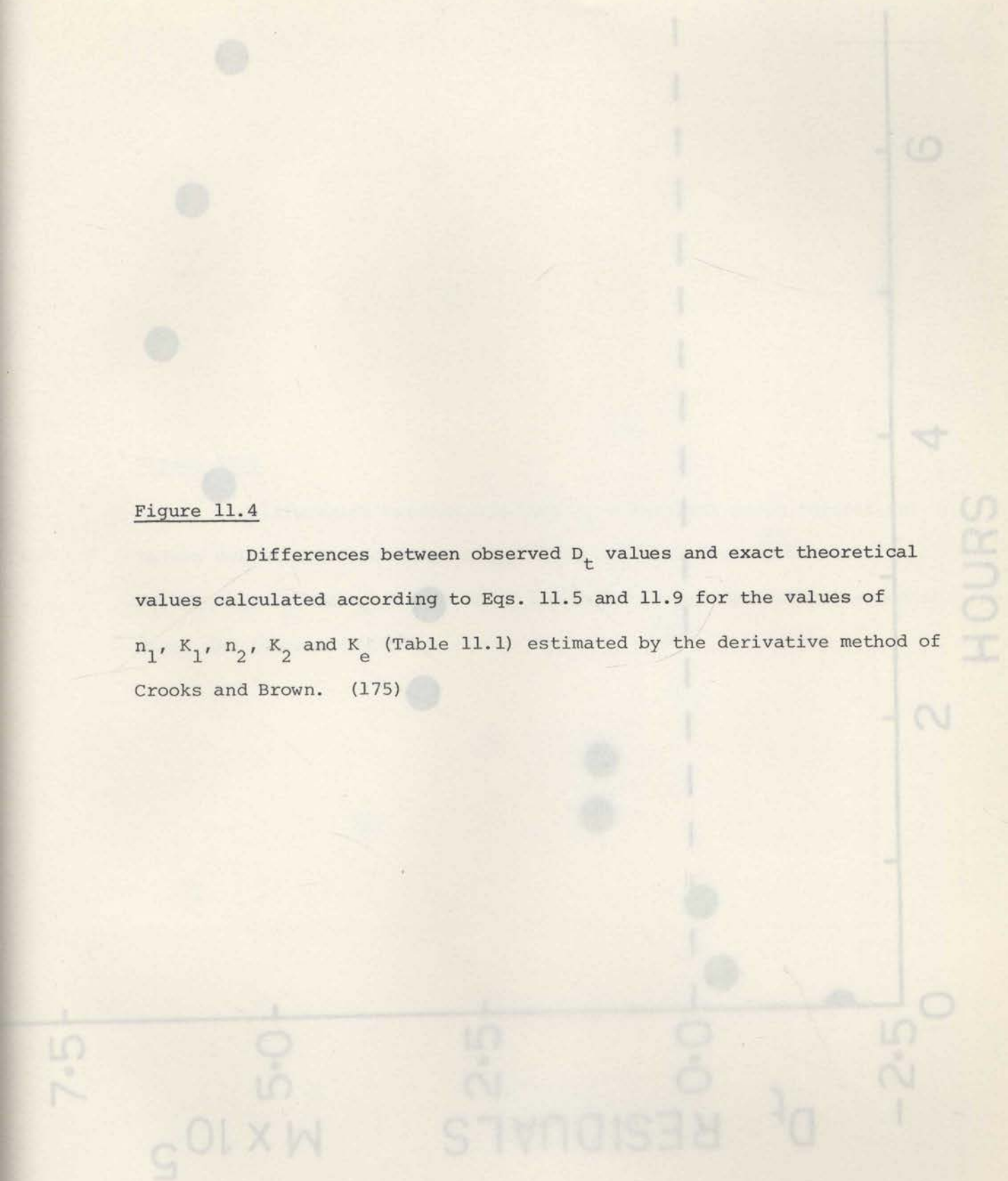


Figure 11.4

Differences between observed D_t values and exact theoretical values calculated according to Eqs. 11.5 and 11.9 for the values of n_1, K_1, n_2, K_2 and K_e (Table 11.1) estimated by the derivative method of Crooks and Brown. (175)

7·5
5·0
2·5
0·0
-2·5
M x 10^5
D_t RESIDUALS
0
2
4
6
HOURS

Figure 11.5

Differences between observed D_t values and exact theoretical values calculated according to Eqs. 11.5 and 11.9 for the values of n_1, K_1, n_2, K_2 and K_e (Table 11.1) estimated by the derivative method of Meyer and Guttman. (169)

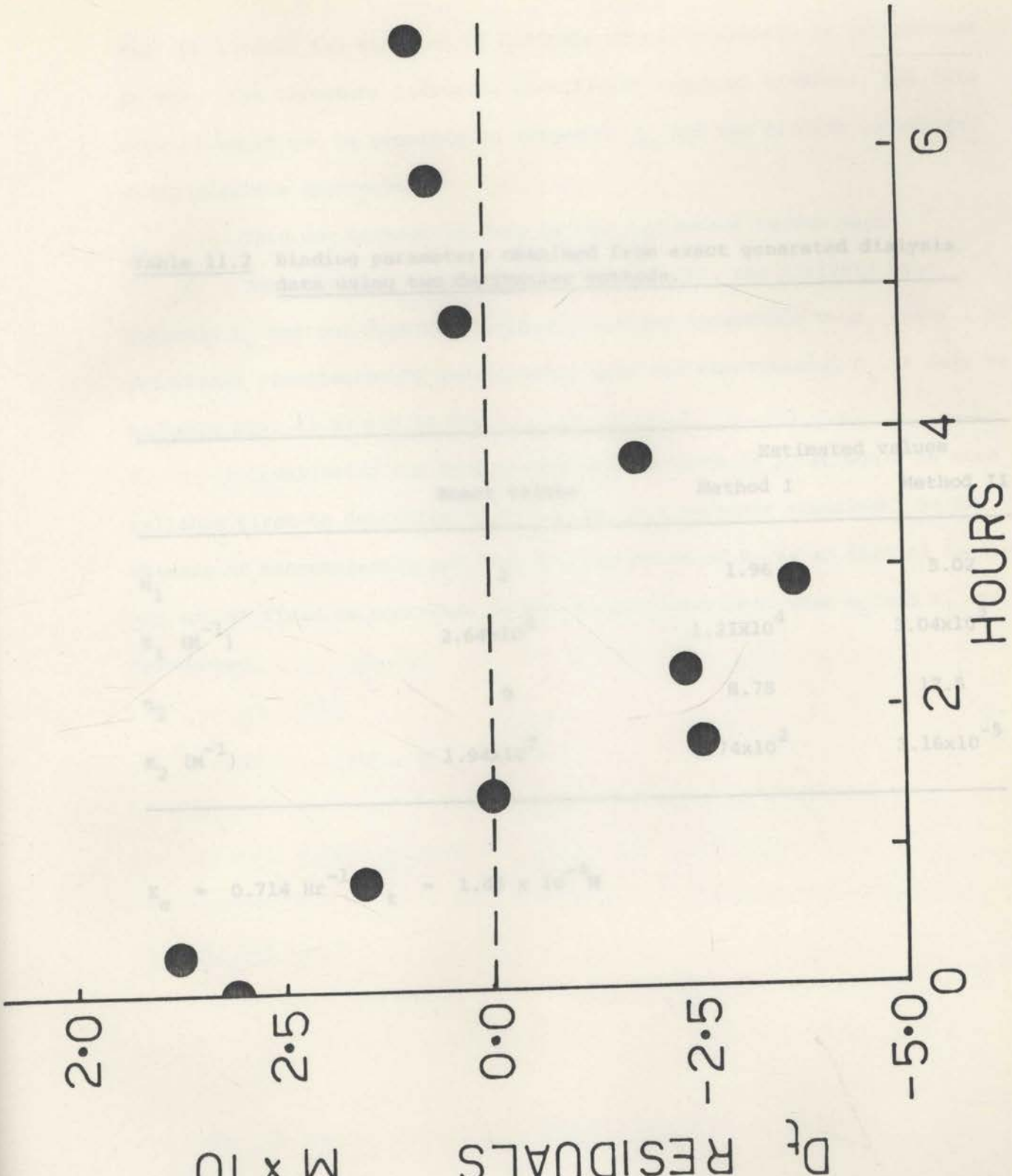

2·0
2·5
0·0
-2·5
-5·0
0
2
4
6
HOURS
D_t RESIDUALS
M x 10^5

Table 11.2 Binding parameters obtained from exact generated dialysis data using two derivative methods.

	Exact values	Estimated values	
		Method I	Method II
n_1	2	1.96	5.02
K_1 (M^{-1})	2.64×10^4	1.21×10^4	3.04×10^3
n_2	9	8.78	12.5
K_2 (M^{-1})	1.94×10^2	1.74×10^2	3.16×10^{-5}

$K_e = 0.714\ Hr^{-1}$, $P_t = 1.45 \times 10^{-4} M$

Fig. 11.1 shows the kinetics of dialysis of glibenclamide in the absence of BSA. The curvature indicates significant membrane binding. For this drug it would not be possible to determine K_e and the binding parameters using previous approaches.

This can however be done by the new method in two ways:

1. The membrane binding parameters, K', K*, the dialysis rate constant K_e and the drug-macromolecule binding parameters n_i, K_i can all be determined simultaneously and directly from the experimental D_t, t data by applying Eqs. 11.22 and 11.23.

2. Considering the many parameters involved in 1. it would be more reliable first to determine K_e, K' and K* in a separate experiment in the absence of macromolecule and then use the value of K_e as an initial estimate and K', K* fixed as constants in the second experiment when n_i and K_i are determined.

BIBLIOGRAPHY

1. Wagner, J.G., J. Pharm. Sci., 50, 359 (1961)
2. Fincher, J.H., ibid., 57, 1825 (1968)
3. Wurster, D.E. and Taylor, P.W., ibid., 54, 169 (1965).
4. Higuchi, W.I., ibid., 56, 315 (1967)
5. Wagner, J.G., Drug Intelligence Clin. Pharm., 4, 17 (1970).
6. Hersey, J.A., Mfg. Chemist Aerosol News, 40, 32 (1969).
7. Baun, D.C. and Walker, G.C., J. Pharm. Sci., 58, 611 (1969).
8. Hersey, J.A. and Marty, J.J., Manuf. Chemist, 46, 43 (1975).
9. Groves, M.J., ibid., 46, 37 (1975).
10. Dare, J.G., Australian J. Pharm., 45, 5 (1967).
11. Levy, G. and Hayes, B.A., New Engl. J. Med., 262, 1053 (1960).
12. Parrot, E.L., Wurster, D.E. and Higuchi, T., J. Pharm. Sci., 44, 269-272 (1955).
13. Nelson, E., ibid., 46, 607 (1957).
14. Mitchell, A.G. and Saville, D.J., ibid., 19, 729 (1967)
15. Shelfer, E. and Higuchi, T., ibid., 52, 781 (1963).
16. Brudney, N., Stewart, D.J. and Eustace, B.T., Can. Med. Assoc. J., 90, 980 (1976).
17. Levy, G. and Procknall, J.A., J. Pharm. Sci., 53, 656 (1964).
18. Lu, F.C., Rice, W.B. and Mainville, C.W., Can. Med. Assoc. J., 92, 1166 (1965).
19. Levy, G. and Sahli, B.A., J. Pharm. Sci., 51, 58 (1962).
20. Levy, G. and Tauski, W., ibid., 53, 679 (1964).
21. Wood, J.H., Syarto, J.E. and Leherman, H., ibid., 54, 1068 (1965).
22. Paikoff, M. and Drum, G., ibid., 54, 1693 (1965).

23. Cleverly, B. and Williams, P.P., Chem. & Ind. London, Jan. 10, p. 49 (through Wagner J.G., J. Pharm. Sci., 50, 359 (1961)).

24. Klein, L., Bull. Biol. Pharm., 273 (1932).

25. Elliott, G.H., Pharm. Z., 131, 514 (1933).

26. Gibaldi, M. and Feldman, S., J. Pharm. Sci., 56, 1238 (1967).

27. Hamlin, W.E., Nelson, E., Ballard, B.E. and Wagner, J.G., ibid., 51, 432 (1962).

28. Schroeter, L.C. & Wagner, J.G., ibid., 51, 957 (1962).

29. Langenbucher, F., ibid., 58, 1265 (1969).

30. Tingstad, J.E. and Riegelman, S., ibid., 59, 692 (1970).

31. Tingstad, J.E., Gropper, E., Lachman, L. and Shami, E., ibid., 61, 1985 (1972).

32. Needham, T.E., Luzzi, L.A. and Mason, W.D., ibid., 62, 1860 (1973).

33. Lapidus, H. and Lordi, N.G., ibid., 55, 840 (1966).

34. Fatton, R., Fielding, D., Khanna, S.C. and Richler, A., Pharm. Acta. Helv., 47, 487 (1971).

35. Crooks, M.J., Ph.D. thesis, Sydney University (1973).

36. Meyer, M.C. and Guttman, D.E., J. Pharm. Sci., 57, 1627 (1968).

37. Brown, K.F. and Crooks, M.J., Biochem. Pharmacol., 25, 1175,(1976).

38. Veng Pedersen, P., M.Sc. thesis, Sydney University (1975).

39. Barr, W.H., Drug Inf. Bull., 3, 27-45 (1969).

40. Castillo, R.A. & Grim, W.M., Paper presented to the Industrial Pharmaceutical Technology Section, AphA Academy of Pharmaceutical Science, Las Vegas Meeting, April, 1967.

41. Dare, J.G., Australian J. Pharm., 45, 5 (1967).

42. Hunt, J.N. & McDonald, I., J. Physiol. 126, 459 (1954).

43. Hunt, J.N. & Spurrel, W.R., ibid., 113, 159 (1951).

44. Levy, G. "Physio-chemical Aspects of Drug Actions" Pergamon Press, Oxford, New York (1968).

45. Wagner, J.G., "Biopharmaceutics and Relevant Pharmacokinetics" Drug Intelligence publications, Illinois (1970).

46. Hixson, A. & Baum, S., Ind. Eng. Chem., 33, 478 (1941).

47. ibid., 34, 120 (1942).

48. Abramson, M. & King, C.V., J. Am. Chem. Soc. 61, 2290 (1939).

49. Levy, G. & Jusko, W., J. Pharm. Sci., 54, 219 (1965).

50. Card, W.I., Am. J. Digest. Dis., 8, 47 (1941).

51. Noyes, A. & Whitney, W., Z. Physik. Chem., 23, 689 (1897).

52. Levy, G., J. Pharm. Sci., 52, 1039 (1963).

53. Wagner, C., J. Phys. Chem., 53, 1030 (1949).

54. Roehl, E., King, C.V. & Kipness, S., J. Am. Chem. Soc., 61, 2290 (1939).

55. Roller, O., J. Phys. Chem., 39, 221 (1935).

56. Gibaldi, M., "Introduction to Biopharmaceutics" Lea & Febiger, Philadelphia (1971).

57. Finholt, P. & Solvang, S., J. Pharm. Sci., 57, 1322 (1968).

58. Elworthy, P.H. & Lipscombe, F.J., J. Pharm. Pharmacol., 20, 923 (1968).

59. Walter, V., ibid., 20, suppl. 228s (1968).

60. Higuchi, W.I., Prakonpau, S., Surpuriya, V. & Young, F. Nature, 178, 633 (1972).

61. Krondi, A., J. Can. Med. Assoc., 29, 360 (1970).

62. Bates, T.R., Gibaldi, M. & Kanig, J.L., J. Pharm. Sci., 55, 191 (1966).

63. Bates, T.R., Lin, S.L. & Gibaldi, M., ibid., 56, 1492 (1967).

64. Thakker, A.Z. & Hall, N.A., ibid., 58, 68 (1969).

65. Higuchi, W.I., Lau, P.K., Higuchi, T. & Shell, J.W., ibid., 150 (1963).

66. Leonards, J.R., Clin. Pharmacol. Therap., 4, 476 (1963).

67. Higuchi, W.I., Nelson, E. & Wagner, J.G., J. Pharm. Sci., 53, 333 (1964).

66. Smolen, V.F. & Kildsig, D.O., ibid., 60, 130 (1971).

67. Mullins, J.D. & Macek, T.J., J. Am. Pharm. Assoc., 49, 245 (1960).

68. Alimirante, L., De Carneri, I., & Coppi, G., Ill. Pharmaco. Ed. Prat., 15, 471 (1960).

69. Taylor, P.W. & Wurster, D.E., J. Pharm. Sci., 54, 1654 (1965).

70. Higuchi, W.I., Bernado, P.D. & Mehta, S.C., ibid., 56, 200 (1967).

71. Tawaski, L., Science, 160, 76 (1968).

72. Shefter, E., & Higuchi, T., J. Pharm. Sci., 52, 781 (1963).

73. Nelson, E., Knoechel, E.L., Hamlin, W.E., & Wagner, J.G., ibid., 51, 509 (1962).

74. Unger, R.H. & Madison, L.L., J. Clin. Invest., 37, 627 (1958).

75. Middleton, E.J., Shang, H.S., & Cook, D., Can. J. Pharm. Sci., 4, 97 (1969).

76. Mullins, J.D. & Macek., T.J., J. Am. Pharm. Ass. Sci. Ed., 49, 249 (1960).

77. Feinberg, M., ibid., 9, 113 (1969).

78. Reinhold, J.G., Phillips, F.J. & Flipper, H.J., Am. J. Med. Sci., 210, 171 (1945).

79. Atkinson, R.M., Bedford, C.B., Child, K.J. & Tamic, E.G., Nature, 193, 588 (1962).

80. Levy, G., Lancet, 2, 723 (1962).

81. Bauer, G., Rieckman, P. & Shanmann, W., Arzneim. Forsch. 12, 487 (1962).

82. Boyed, E.M. & Dingwall, R.W., Am. J. Med. Sci., 213, 549 (1947).

83. Kakani, K., Arita, T. & Okashi, S., "Absorption and exretion of chloramphenical" Paper presented to the Am. Pharm. Assoc. National Mtg., Las Vegas, Nev. (1962) from ref. 56.

84. Lees, K.A., Pharm. J., 289 (1963).

85. Dare, J.G., Australian J. Pharm., 45, 5-58 (1964).

86. Kornblum, S.S., & Hirschorn, J. Pharm. Sci., 59, 606, (1970).

87. Lin, S.L. & Lachman, L., ibid., 57, 2143 (1968).

88. Chiou, W.L. & Riegelman, S., ibid., 58, 1505 (1969).

89. ibid., 60, 1281 (1971).

90. Goldberg, A.H., Gibaldi, M. & Kanig, J.L., ibid., 55, 487 (1966).

91. Gibaldi, M., Feldman, S., & Bates, T.R., ibid., 57, 708 (1968).

92. Carstensen, J.T. & Musa, M.N., ibid., 61, 223 (1972).

93. Brooke, D., ibid., 62, 795 (1973).

94. ibid., 63, 344 (1974).

95. Veng Pedersen, P. & Brown, K.F., ibid., 64, 1192 (1975).

96. ibid., 65, 1981 (1976).

97. Hixson, A.W. & Crowell, J.H., Ind. Eng. Chem., 23, 923 (1931).

98. Nernst, W., Z. Phys. Chem., 47, 52 (1904).

99. Dankwerts, P.V., Ind. Eng. Chem. 43, 1460 (1951).

100. Bertroud, A., J. Phys. Chem., 10, 624 (1912).

101. Nedich, R.L. and Kildsig, D.O., J. Pharm. Sci., 61, 214 (1972).

102. Nernst, W. and Brunner, E., Zeit. Phys. Chem., 47, 52 (1904).

103. Wilderman, M., ibid., 60, 445 (1909).

104. Wurster, D.E. and Taylor, P.W., J. Pharm. Sci., 54, 169 (1965).

105. Goyan, J.E., ibid., 54, 645 (1965).

106. Niebergall, P.J., Milosovich, G. & Goyan, J.E., ibid., 52, 236 (1963).

107. Higuchi, W.I. & Hiestand, E.N., ibid., 52, 67 (1963).

108. Rosner, D.E. and Epstein, M., J. Phys. Chem., 74, 4001 (1970).

109. Cable, M. and Evans, D.J., J. Appl. Phys. 37, 2377 (1966).

110. Scriven, L.E., Chem. Engng. Sci., 10, 1 (1959).

111. Duda, J.L. and Vrentas, J.S., AI Ch.E.J., 15, 351 (1969).

112. Ready, D.W. and Cooper, A.R., Chem. Engng. Sci., 21, 917 (1966).

113. Rosner, D.E., J. Phys. Chem., 73, 383 (1969).

114. Ridgeway, K. & Peacock, D.G., J. Pharm. Pharmac., 23, 1215 (1971).

115. Crank, J., "The Mathematics of Diffusion", Oxford University Press, London (1956).

116. Cleave, J.P., J. Pharm. Pharmacol., 17, 698 (1965).

117. Cobby, J., Mayersohn, M. and Walker, G.C., J. Pharm. Sci., 63, 725 (1974).

118. Edmundson, I.C. and Lees, K.A. J. Pharm. Pharmacol., 17, 193 (1965).

119. Withey, R.J., ibid., 23, 573 (1971).

120. Veng Pedersen, P. & Brown, K.F., J. Pharm. Sci., in press.

121. Walpole, R.E. and Myers, R.H. "Probability and Statistics for Engineers and Scientists", Macmillan, New York, N.Y., 1972.

122. "Symposium on Particle Size Measurement", American Society for Testing Materials Special Technical Publication No. 234, Philadelphia, Pa., (1958).

123. Metzler, C.M., 'A Users Manual for NONLIN', The Upjohn Co., Technical Report 7292/69/7292/005, Kalamazoo, Mich. (1969).

124. Booth, G.W., & Peterson, T.L., 'Nonlinear Estimation', I.B.M. share program., no. 687 WLNL7 (1958).

125. Marquardt, D.W., 'Least Squares Estimation of Nonlinear Parameters', I.B.M. share library program, Distribution no. 309401 (1966).

126. Meeter, D.A., 'Program GAUSHAUS', Numerical Analysis Laboratory, University of Wisconsin, Madison, Wisconsin (1966).

127. Pfeffer, M., 'J. Pharmacokin. Biopharm.'2, 137 (1973).

128. Berman, M. & Weiss, M.F., 'Users Manual for SAAM', National Institutes of Health, Bethesda, Md. (1968).

129. Chamber, J.M., Appl. Statist., 18, 249 (1969).

130. Fletcher, R., Report AERE, R6799, London: H.M.S.O. (1971).

131. Niebergall, P.J., Schnaare, R.L. and Sugita, E.T., J. Pharm. Sci., 60, 1393 (1971).

132. Atkins, G.L., Biochim. Biophys. Acta, 252, 405 (1971).

133. Himmelblau, D.M., 'Applied Nonlinear Programming', McGraw-Hill, New York, (a) p. 9 (b) p. 474 (1972).

134. Gauss, C.F., 'Theory of Least Squares'. English translation by H.F. Trotter of original from 1821. Princeton University, Statistical Techniques Research Group. Technical Report No. 5 (1959).

135. Nelder, J.H. and Mead, R., Comput. J., 7, 308 (1965).

136. Hogg, R.V., & Craig, A.T., 'Introduction to Mathematical Statistics'. McMillan, New York, (a) p. 383 (b) p. 138 (1970).

137. Draper, N.R., & Smith, H., 'Applied Regression Analysis'. John Wiley & Sons, New YOrk. p. 265 (1966).

138. Boxenbaum, H.G., Riegelman, S., and Elashoff, R.M., J. Pharmacokin. Biopharm., 2, 123 (1974).

139. Davis, O.L., & Goldsmith, P.L., 'Statistical Methods in Research and Production', Oliver and Boyed, Edinburgh, p. 236 (1972).

140. Kendall, M.G., 'The Advanced Theory of Statistics', Charles Griffin & Co., London, Chapter 10 (1963).

141. Daniel, C., & Wood, F.S., 'Fitting Equations to Data', Wiley & Sons, New York, p. 28 (1971).

142. Durbin, J., & Watson, G.S., Biometrica, 37, 409 (1950).

143. Sweed, F.S., & Eisenhart, C., Ann. Math. Stats., 14, 66 (1943).

144. Anscombe, F.J., Technometrics, 2, 123 (1960).

145. Powell, M.J.D. in Rabinowitz, P. (ed.) "Numerical Methods for Non-Linear Algebraic Equations", Gordon & Breach, p. 87 (1970).

146. Dennis, J.E., in Byrne, G.D. & Hall, C.H. (eds.) "Numerical Solution of System of Nonlinear Algebraic Equations", Academic Press, New York, p. 157 (1973).

147. Dennis, J.e., Proc. ACM Natl. Conf. 1970, ACM, N.Y. (1971).

148. Hartley, H.O., Technometrics, 3, 269 (1961).

149. Chambers, J.M., Biometrica, 60, 1 (1973).

150. Spendley, W. in Fletcher, R. (ed.), "Optimization", Academic Press, New York, 259 (1969).

151. Krüger-Thiemer, E. & Levine, R.R., Arzneimittel-Forschung, 18, 1575 (1968).

152. Wagner, J., "Fundamentals of Clinical Pharmacokinetics", Illinois, Drug Intelligence Publications (1975).

153. Wagner, J., J. Pharmacokin. Biopharm., 3, 457 (1975).

154. Wagner, J., "Biopharmaceutics and Relevant Pharmacokinetics", The Hamilton Press, Hamilton, Illinois (1971).

155. Berman, M., Ann. N.Y. Acad. Sci., 108, 182 (1963).

156. Nooney, G.C., J. Chron. Dis., 19, 325 (1966).

157. Berman, M., Shahn, E. & Weiss, M.F., Biophys. J., 2, 275 (1962).

158. Hiller, J. in, Andersen, R.S. & Osborne, M.R. (Eds.) "Data Representation" University of Queensland Press, Queensland, Australia, 62 (1970).

159. Killifors, H.W., J. Amer. Stat. Ass., 62, 399 (1967).

160. Hunter, W.G. & Mezaki, R., A.I. ChE.J., 10, 315 (1964).

161. Box, G.E.P. & Hunter, W.G., Technometrics, 7, 23 (1965).

162. Stewart, G.W., J. Assoc. Comput. Mach., 14, 72 (1967).

163. Fletcher, R. (ed.) "Optimization", Academic Press, New York, p. 1 (1969).

164. Brown, K.M. & Dennis, J.E., Yale University Report 71-6 (1971).

165. Powell, M.J.D., in Murray, W. (ed.) "Numerical Methods for Unconstrained Optimization", Academic Press, p. 29 (1972).

166. Herdan, G. & Smith, M.L., "Small Particle Statistics", Elsevier, New York, N.Y. (1953).

167. Stein, H.H., Analyt. Biochem., 13, 605 (1965).

168. Agren, A., Acta. Pharm. Suec., 5, 37 (1968).

169. Meyer, M.C. & Guttman, D.E., J. Pharm. Sci., 57, 1627 (1968).

170. Meyer, M.C. & Guttman, D.E., J. Pharm. Sci., 59, 33 (1970).

171. Meyer, M.C. & Guttman, D.E., J. Pharm. Sci., 59, 39 (1970).

172. Dearden, J.C. & Tomlinson, E., J. Pharm. Pharmac., 22, Suppl., 53S. (1970)

173. Goto, S., Ohki, T., Kiryu, S. & Iguchi, S., Yakuzaigaka, 31, 247 (1971).

174. Brown, K.F. & Crooks, M.J., Pharm. Acta. Helv., 48, 494 (1973).

175. Crooks, M.J. & Brown, K.F., J. Pharm. Sci., 62, 1904 (1973).

176. Crooks, M.J. & Brown, K.F., J. Pharm. Pharmac., 26, 235 (1974).

177. Crooks, M.J. & Brown, K.F., J. Pharm. Pharmac., 26, 304 (1974).

178. Brown, K.F. & Crooks, M.J., Biochem. Pharmacol., In the press (1976).

179. Kanfer, I. & Cooper, D.R., J. Pharm. Pharmac., 28, 58 (1976).

180. Kazmi, S.J.A. & Mitchell, J. Pharm. Sci., 62, 1299 (1973).

181. Bottari, F., Di Colo, G., Nannipieri, E., Saettone, M.F. & Serafini, M.F., J. Pharm. Sci., 64, 946 (1975).

182. Craig, L.C., Advan. Anal. Chem. Instr., 4, 35 (1966).

183. Matsumoto, H., Matsumura, H. & Iguchi, S., Chem. Pharm. Bull., 14, 385 (1966).

184. Hart, H.E., Bull. Math. Biophys., 27, 87 (1965).

- o O o -

A P P E N D I X

Table of Symbols

a diameter or equivalent spherical diameter.

$\bar{a}$ mean diameter.

A surface area.

α acute angle of crystal (p. 69).

b_o side length of crystal (t=0).

c_o solubility of compound in unionised form.

c_b solute bulk concentration.

c_i interfacial concentration of solute.

c^* dimensionless concentration.

$COV(\ )$ covariance.

d_o initial diameter of the smallest particle in a powder.

D_o initial diameter of the largest particle in a powder.

D diffusion coefficient.

D_t total concentration of small molecule.

D_f free concentration of small molecule.

∇^2 Laplacian operator.

e_i i-th residual value.

$E(\ )$ expected value.

$E_t(w)$ mean particle weight of time t.

$erfc(x) = 1-erf(x)$ complementary error function.

$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2} du$ error function.

$f(\ )$ particle weight density function of time t.

F shape ratio.

$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-u^2/2} du$ cumulative normal distribution function.

$F_{\alpha,n,p}$ critical point of F-distribution.

ψ stream function, time function.

g acceleration of gravity.

$g( )$ single particle dissolution function (p. 97).

$g^{-1}( )$ inverse single particle dissolution function (p. 97).

γ mean rate of surface renewal (p. 51).

h diffusion layer thickness.

H Hessian matrix.

$H+$ H_3^+0 concentration.

H_{ii} i-th diagonal element of Hessian matrix.

i lower truncation parameter (Fig. 5.1)

j upper truncation parameter (Fig. 5.1)

J interfacial mass flux (p. 66).

J_D diffusion flux.

k_i effective interfacial transport rate constant (p. 52).

k_m rate parameter for model m.

K_a dissolution constant of acid.

K_b dissociation constant of base.

K_e dialysis rate constant (Chapter 11).

K_i association constant for i-th class (Chapter 11)

K_m Michaelis Menten parameter (Chapter 11), rate parameter for model m.

K_w ion product of water.

K^* specific dissolution rate parameter, association constant for membrane binding.

l_o sidelength of crystal (Fig. 4.4).

$l_o( )$ initial particle size distribution.

$L( )$ likelihood function.

m	model parameter.
m_o	initial weight of the smallest particle.
M_o	initial weight of the largest particle.
μ	logarithmic mean (Fig. 5.1). viscosity (Chapter 4).
n_i	number of sites in i-th class (Chapter 11)
$N( )$	normal distribution function (p. 101).
N_t	number of particles at time t.
$\overline{\nu}$	number of moles of small molecules bound per mol macromalecule.
$\overline{\nu}^*$	amount of small molecule bound per amount of available membrane material.
$\underline{P}$	parameter vector.
P	operator (defined p. 99).
P_t	total concentration of macromolecule (Chapter 11).
$\pi = 3.1415\cdots$	
r_t	radius of particle at time t.
r^*	dimensionless radius.
ρ	density
$\rho_{i,j}$	correlation coefficient between i-th and j-th element.
R_e	Reynolds number.
s	distance to fixed point (Chapter 4)
	dispersion product (Chapter 6)
	variable defined p. 269.
Sc	Schmidt number.
SD_i	standard deviation of i-th element.
SS	sum of squared residuals.
$S_N( )$	cumulative distribution of residuals.
σ	standard deviation of normal distribution (Fig. 5.1.)

t	time.
t^*	dimensionless time (Chapter 4).
$t_{\alpha,n}$	critical value of t-distribution.
V_∞	vertical velocity of free falling spherical particle.
V	variance-covariance matrix.
V_m	Michaelis Menten parameter.
Var( )	variance.
w	weight of single particle.
W	weight of undissolved powder.
θ	spherical coordinate (Chapter 4).
$\binom{x}{y}$ =	$x!/(y!(x-y)!)$ binomial coefficient.

- o O o -

```
      PROGRAM FUNFIT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)                    10
C     ------------------------------------------------------------------       20
C     FUNFIT IS AN INTERACTIVE TIME-SHARING PROGRAM FOR GENERAL                30
C     NONLINEAR REGRESSION AND CURVE FITTING                                   40
C     THE PROGRAM CONSISTS OF-                                                 50
C     MAIN PROGRAM / FUNFIT                                                    60
C     SUBROUTINES  / NELDR,CHOL,SYMINV,LSQ,XYPLOT,PLOT,                        70
C                  / SEARCH,READ,PLACE,PROB1,KOLMIR                            80
C     FUNCTIONS    / SIGNOF,WEIGHT,SEQPRO,NBC,SDF                              90
C     A SUBROUTINE MODEL(Y,X,P,IPRINT) MUST BE SUPPLIED BY THE USER           100
C     TO DEFINE THE FUNCTION(S) TO BE FITTED. FOR DETAILS ABOUT THIS          110
C     SUBROUTINE CONTACT P.VENG PEDERSEN,THE PHARMACY DEPARTMENT              120
C     THE UNIVERSITY OF SYDNEY,N.S.W. 2006,AUSTRALIA.                         130
C     ------------------------------------------------------------------      140
      DIMENSION F(20),F1(20),F2(20),FSAVE(20),NF(20),FS(20),NT(20),            150
     *STPS(20),STEP(20),STEPS(20),VAR(20),TCRIT(30),X(100),Z(100),             160
     *Y(100),WY(100),WYN(100),RES(100),DIFPCT(100),YEST(100),IOUTL(10),        170
     *VARCOV(55),INDEX(100),WRES(100),FNEW(7),STPNEW(7),NRECOM(7),             180
     *NCRIT(7),VC(55),A(650),FMIN(20),AR(12),XX(9,100),G(9),LABEL(100),        190
     *ITHSET(10),LIM1(10),LIM2(10),RSS(10),WRSS(10)                            200
      DOUBLE PRECISION VC                                                      210
      COMMON XMIN,XMAX                                                         220
      COMMON /FUNNUM/ITHFUN                                                    230
      COMMON /DATA/ XX,Y,WYN,NOBS                                              240
      COMMON /PARLIM/ F1,F2                                                    250
      COMMON /CONSTR/LIMITS                                                    260
      COMMON /B1/ X                                                            270
      COMMON /B2/ Z                                                            280
      COMMON/B3/VARCOV,VC,NEVAL1,NEVAL2,AMINQ                                  290
      COMMON /B4/ NIND                                                         300
      COMMON/B5/NQ,NS,IGO                                                      310
      COMMON /B6/ NOP                                                          320
      COMMON /B7/ ITABLE                                                       330
      COMMON /B8/NFUNC                                                         340
      COMMON /B9/LIM1,LIM2                                                     350
      COMMON /B10/LABEL                                                        360
      COMMON /B11/AR                                                           370
      COMMON /B12/NPR                                                          380
      COMMON /B13/IGNORE                                                       390
      COMMON /B14/NVIOL                                                        400
      COMMON /B15/SKIP                                                         410
      LOGICAL SKIP,SKIPS                                                       420
      EQUIVALENCE (AR(1),ANS)                                                  430
      EXTERNAL LSQ                                                             440
      DATA TCRIT/12.7,4.30,3.18,2.78,2.57,2.45,2.36,2.31,2.26,2.23,            450
     *2.20,2.18,2.16,2.14,2.13,2.12,2.11,2.10,2.09,2.09,2.08,                  460
     *2.07,2.07,2.06,2.06,2.06,2.05,2.05,2.04,2.04/                            470
      DATA NRECOM /0,0,343,625,1024,729,2187 /                                 480
      DATA NCRIT /0,0,8,16,32,64,128 /                                         490
      WRITE(6,500)                                                             500
      LINES=30                                                                 510
      NRUN=0                                                                   520
      ITABLE=0                                                                 530
      LIMITS=0                                                                 540
      NCONR=0                                                                  550
      NDUMMY=0                                                                 560
      NS=1                                                                     570
      NFUNC=-1                                                                 580
      NOP=-1                                                                   590
      NOP2=-1                                                                  600
      NOBS2=-1                                                                 610
```



```
      IQUAD=1                                                                  620
      SIMP=1E-6                                                                630
      ANS4=2HNO                                                                640
      PLOTRS=2HNO                                                              650
      MAX=800                                                                  660
      NLOOP=20                                                                 670
      IPRINT=-1                                                                680
      ANS11=2HNO                                                               690
      ANS12=2HNO                                                               700
      STOPCR=0.001                                                             710
      NIND=1                                                                   720
      PLT=2HNO                                                                 730
      ANS16=2HNO                                                               740
    5 NQ=1                                                                     750
      WRITE(6,505)                                                             760
      IGNORE=0                                                                 770
      CALL READ(1,AR)                                                          780
```

```
      J=IOUTL(I)                                                        9300
  370 WRITE(6,822)J                                                     9310
      IF(NOUTL.EQ.1) WRITE(6,824)                                       9320
      IF(NOUTL.GT.1) WRITE(6,826)                                       9330
 3701 CONTINUE                                                          9340
      IF(ANS16.EQ.2HNO.OR.NOWGHT.EQ.1) GO TO 3704                       9350
      WRITE(6,8241)T                                                    9360
      NOUTL=0                                                           9370
      STAR=2H *                                                         9380
      NSAVE=1                                                           9390
      DO 3703 I=1,NOBS                                                  9400
      N=LABEL(I)                                                        9410
      IF(N.EQ.NSAVE+1) WRITE(6,8240) N                                  9420
      NSAVE=N                                                           9430
      WNRDV=WRES(I)/WSDVRS                                              9440
      IF(ABS(WNRDV).LT.T) GO TO 3702                                    9450
      NOUTL=NOUTL+1                                                     9460
      WRITE(6,8242)I,STAR,WY(I),WYN(I),WRES(I),WNRDV                    9470
      GO TO 3703                                                        9480
 3702 WRITE(6,8243)I,WY(I),WYN(I),WRES(I),WNRDV                         9490
 3703 CONTINUE                                                          9500
      WRITE(6,8244)                                                     9510
      IF(NOUTL.EQ.1) WRITE(6,8245)                                      9520
      IF(NOUTL.GT.1) WRITE(6,8246)                                      9530
 3704 CONTINUE                                                          9540
      IF(NIND.GT.1.OR.PLT.NE.3HYES) GO TO 371                           9550
      CALL XYPLOT(F)                                                    9560
      DO 3709 II=1,NFUNC                                                9570
      N1=LIM1(II)                                                       9580
      N2=N1+1                                                           9590
      N3=LIM2(II)                                                       9600
      NN=N3-1                                                           9610
      XMAX=X(N1)                                                        9620
      XMIN=X(N1)                                                        9630
      DO 3705 I=N2,N3                                                   9640
      IF(X(I).GT.XMAX) XMAX=X(I)                                        9650
 3705 IF(X(I).LT.XMIN) XMIN=X(I)                                        9660
      DELTA=(XMAX-XMIN)/50.                                             9670
      IF(II.EQ.1) WRITE(6,8261)                                         9680
      IF(II.GT.1) WRITE(6,8264)II                                       9690
      DO 3706 I=1,26                                                    9700
      J=26+I                                                            9710
      XP1=XMIN+FLOAT(I-1)*DELTA                                         9720
      XP2=XMIN+FLOAT(J-1)*DELTA                                         9730
      DO 3708 L=N1,NN                                                   9740
      IF(XP1.GT.X(L).AND.XP1.LE.X(L+1))                                 9750
     *YP1=Y(L)+(Y(L+1)-Y(L))*(XP1-X(L))/(X(L+1)-X(L))                   9760
 3708 IF(XP2.GT.X(L).AND.XP2.LE.X(L+1))                                 9770
     *YP2=Y(L)+(Y(L+1)-Y(L))*(XP2-X(L))/(X(L+1)-X(L))                   9780
      ITHFUN=II                                                         9790
      CALL MODEL(YP1,XP1,F,0)                                           9800
      CALL MODEL(YP2,XP2,F,0)                                           9810
      IF(I.EQ.26) GO TO 3707                                            9820
 3706 WRITE(6,8262) I,XP1,YP1,J,XP2,YP2                                 9830
 3707 WRITE(6,8262) I,XP1,YP1                                           9840
 3709 CONTINUE                                                          9850
      WRITE(6,8263)                                                     9860
  371 CONTINUE                                                          9870
      IF(IQUAD.NE.1) GO TO 376                                          9880
      WRITE(6,827)                                                      9890
      DO 375 K=1,2                                                      9900
      IF(K.EQ.2) WRITE(6,8270)                                          9910
```



```
      L=1                                                               9920
  372 IF(L.GT.NPR) GO TO 375                                            9930
      II=L*(L-1)/2                                                      9940
      DO 373 I=L,NPR                                                    9950
      IA=II+L                                                           9960
      II=II+I                                                           9970
      IB=MIN0(II,IA+5)                                                  9980
      IF(K.EQ.2) GO TO 3721                                             9990
      DO 3720 J=IA,IB                                                  10000
 3720 VARCOV(J)=2.*FUNC*VARCOV(J)/FLOAT(NRSDF)                         10010
      WRITE(6,829)(VARCOV(J),J=IA,IB)                                  10020
      GO TO 373                                                        10030
 3721 DO 3722 KK=IA,IB                                                 10040
 3722 VARCOV(KK)=VC(KK)                                                10050
      WRITE(6,829)(VARCOV(J),J=IA,IB)                                  10060
  373 CONTINUE                                                         10070
      L=L+6                                                            10080
```

```
      L=I*(I-1)/2+J                                                     18600
      BMAT(L)=2D0*((HSTST*A0)-(AVAL(I)+AVAL(J)))                        18610
  208 CONTINUE                                                          18620
  207 CONTINUE                                                          18630
      L=0                                                               18640
      DO 210 I=1,NAP                                                    18650
      I1=I+1                                                            18660
      L=L+I                                                             18670
      BMAT(L)=2D0*((H(I1)+A0)-2D0*AVAL(I))                              18680
  210 CONTINUE                                                          18690
      DO 237 I=1,NAP                                                    18700
      I1=I+1                                                            18710
  237 AVAL(I)=2D0*AVAL(I)-(H(I1)+3D0*A0)/2D0                            18720
      DO 219 I=1,NOP                                                    18730
  219 PMIN(I)=G(1,I)                                                    18740
      DO 211 I=1,NAP                                                    18750
      I1=I+1                                                            18760
      DO 211 J=1,NOP                                                    18770
      G(I1,J)=G(I1,J)-G(1,J)                                            18780
  211 CONTINUE                                                          18790
      DO 212 I=1,NAP                                                    18800
      I1=I+1                                                            18810
      DO 212 J=1,NOP                                                    18820
      G(I,J)=G(I1,J)                                                    18830
  212 CONTINUE                                                          18840
      CALL SYMINV(BMAT,NAP,BMAT,TEMP,NULLTY,IFAULT)                     18850
      IF(IFAULT.NE.0) GO TO 450                                         18860
      IRANK = NAP-NULLTY                                                18870
      GO TO 441                                                         18880
  450 WRITE(IW,302)                                                     18890
  302 FORMAT(49H MATRIX TO BE INVERTED NOT POSITIVE SEMI-DEFINITE)      18900
      IF(NVIOL.GT.0.AND.IPRINT.EQ.-1) WRITE(IW,8031)NVIOL               18910
      IF(NFIX.GT.0) WRITE(6,3020) NFIX                                  18920
 3020 FORMAT(" NO. OF VERTICES OF SIMPLEX CONSTRAINED BY THE",          18930
     *" PARAMETER LIMITS =",I2/  )                                      18940
      IFAULT = 2                                                        18950
      RETURN                                                            18960
  441 DO 213 I=1,NAP                                                    18970
      H(I)=0D0                                                          18980
      DO 214 J=1,NAP                                                    18990
      IF(J-I)216,216,215                                                19000
  216 IJ=I*(I-1)/2+J                                                    19010
      GO TO 217                                                         19020
  215 IJ=J*(J-1)/2+I                                                    19030
  217 H(I)=H(I)+BMAT(IJ)*AVAL(J)                                        19040
  214 CONTINUE                                                          19050
  213 CONTINUE                                                          19060
      YMIN=0D0                                                          19070
      DO 218 I=1,NAP                                                    19080
  218 YMIN=YMIN+H(I)*AVAL(I)                                            19090
      YMIN=A0-YMIN                                                      19100
      DO 220 I=1,NOP                                                    19110
      PSTST(I)=0.0                                                      19120
      DO 220 J=1,NAP                                                    19130
  220 PSTST(I)=PSTST(I)+H(J)*G(J,I)                                     19140
      DO 221 I=1,NOP                                                    19150
  221 PMIN(I)=PMIN(I)-PSTST(I)                                          19160
      AMINQ=YMIN                                                        19170
      IF(IPRINT)223,222,222                                             19180
  222 WRITE(IW,303) YMIN,(PMIN(I),I=1,NPR)                              19190
  303 FORMAT(42H MINIMUM OF FITTED QUADRATIC SURFACE IS   ,D15.8,3H AT// 19200
     *(5E14.6))                                                         19210
```



```
      WRITE(IW,304)FUNC,(F(I),I=1,NPR)                                  19220
  304 FORMAT(/42H COMPARE WITH MINIMUM FOUND BY ITERATION  ,D15.8,3H AT/ 19230
     */(5E14.6))                                                        19240
      D=SNGL(1D0-YMIN/FUNC)                                             19250
      IF(ABS(D).GT..1) WRITE(IW,305)                                    19260
  305 FORMAT(/                                                          19270
     *" IF DIFFERENCE IS LARGE,INFORMATION MATRIX IS INACCURATE AND A"/ 19280
     *" NEW RUN WITH A DIFFERENT STOPPING CRITERION, EXPANSION",        19290
     *" CRITERION"/" OR PARAMETER RANGE IS RECOMMENDED."/  )             19300
  223 CONTINUE                                                          19310
      DO 2230 I=1,NOP                                                   19320
      IF(PMIN(I).LT.F1(I))PMIN(I)=F1(I)                                 19330
 2230 IF(PMIN(I).GT.F2(I))PMIN(I)=F2(I)                                 19340
      CALL FUNCTN(PMIN,QMIN)                                            19350
      IF(QMIN.GE.FUNC) GO TO 2234                                       19360
      FUNC=QMIN                                                         19370
      DO 2231 I=1,NOP                                                   19380
 2231 F(I)=PMIN(I)                                                      19390
```

```
      DO 260 I7=1,NP                                                    27900
      PP(7,I7)=F1(7)+FLOAT(I7-1)*DEL(7)                                 27910
      FF(7)=PP(7,I7)                                                    27920
      DO 250 I6=1,NP                                                    27930
      PP(6,I6)=F1(6)+FLOAT(I6-1)*DEL(6)                                 27940
      FF(6)=PP(6,I6)                                                    27950
      DO 240 I5=1,NP                                                    27960
      PP(5,I5)=F1(5)+FLOAT(I5-1)*DEL(5)                                 27970
      FF(5)=PP(5,I5)                                                    27980
      DO 230 I4=1,NP                                                    27990
      PP(4,I4)=F1(4)+FLOAT(I4-1)*DEL(4)                                 28000
      FF(4)=PP(4,I4)                                                    28010
      DO 220 I3=1,NP                                                    28020
      PP(3,I3)=F1(3)+FLOAT(I3-1)*DEL(3)                                 28030
      FF(3)=PP(3,I3)                                                    28040
      DO 210 I2=1,NP                                                    28050
      PP(2,I2)=F1(2)+FLOAT(I2-1)*DEL(2)                                 28060
      FF(2)=PP(2,I2)                                                    28070
      DO 200 I1=1,NP                                                    28080
      PP(1,I1)=F1(1)+FLOAT(I1-1)*DEL(1)                                 28090
      FF(1)=PP(1,I1)                                                    28100
      CALL LSQ(FF,SUM)                                                  28110
      NEVAL=NEVAL+1                                                     28120
      IF(SUM.GE.SSMIN) GO TO 195                                        28130
      SSMIN=SUM                                                         28140
      IS(1)=I1                                                          28150
      IS(2)=I2                                                          28160
      IS(3)=I3                                                          28170
      IS(4)=I4                                                          28180
      IS(5)=I5                                                          28190
      IS(6)=I6                                                          28200
      IS(7)=I7                                                          28210
  195 IF(F1(1).EQ.F2(1)) GO TO 205                                      28220
  200 CONTINUE                                                          28230
  205 IF(F1(2).EQ.F2(2)) GO TO 215                                      28240
  210 CONTINUE                                                          28250
  215 IF(F1(3).EQ.F2(3)) GO TO 225                                      28260
  220 CONTINUE                                                          28270
  225 IF(NOP.EQ.3) GO TO 270                                            28280
      IF(F1(4).EQ.F2(4)) GO TO 235                                      28290
  230 CONTINUE                                                          28300
  235 IF(NOP.EQ.4) GO TO 270                                            28310
      IF(F1(5).EQ.F2(5)) GO TO 245                                      28320
  240 CONTINUE                                                          28330
  245 IF(NOP.EQ.5) GO TO 270                                            28340
      IF(F1(6).EQ.F2(6)) GO TO 255                                      28350
  250 CONTINUE                                                          28360
  255 IF(NOP.EQ.6) GO TO 270                                            28370
      IF(F1(7).EQ.F2(7)) GO TO 270                                      28380
  260 CONTINUE                                                          28390
  270 CONTINUE                                                          28400
      DO 280 I=1,7                                                      28410
      FNEW(I)=PP(I,IS(I))                                               28420
  280 STPNEW(I)=DEL(I)/2.                                               28430
      RETURN                                                            28440
 1000 FORMAT(// 11X,                                                    28450
     *" PLOT OF RESIDUAL SUM OF SQUARES VERSUS PARAMETER NO.",I2/        28460
     * 11X," RANGING FROM ",E11.4," TO ",E11.4  )                       28470
 1010 FORMAT(// 6X,                                                     28480
     *" PLOT OF WEIGHTED RESIDUAL SUM OF SQUARES VERSUS PARAMETER",      28490
     *" NO.",I2/6X," RANGING FROM ",E11.4," TO ",E11.4  )               28500
 1020 FORMAT(/ " TABLE OF PLOT ABOVE //                                 28510
```



```
     *2(" NO.  SUM OF SQUARES   PARAMETER  ")/1X,68(1H-))               28520
 1025 FORMAT(/ " TABLE OF PLOT ABOVE"//                                 28530
     *2(" NO. WGHT.SUM OF SQUARES PARAMETER  ")/1X,68(1H-))             28540
 1026 FORMAT(I4,2E14.4,3X,I4,2E14.4)                                    28550
 1027 FORMAT(1X,68(1H-))                                                28560
 1030 FORMAT(//" THE SEARCH GRID CONSISTS OF 100 POINTS GIVEN BY",      28570
     *" PAIRED"/" COMBINATION OF FOLLOWING PARAMETER VALUES"//           28580
     *" PAR.NO.",I1,"  (J=1-10)"//(5(" (",I2,")",E10.4))  )             28590
 1040 FORMAT(/" PAR.NO.",I1,"  (I=1-10)" //(5(" (",I2,")",E10.4))  )     28600
 1050 FORMAT(/                                                          28610
     *" MATRIX OF RESIDUAL SUM OF SQUARES AT POINTS IN SEARCH GRID."  )  28620
 1060 FORMAT( /" MATRIX OF WEIGHTED RESIDUAL SUM OF SQUARES AT POINTS", 28630
     *" IN SEARCH GRID."  )                                             28640
 1070 FORMAT(                                                           28650
     *" THE I,J-TH ELEMENT IN THE MATRIX CORRESPONDS TO THE J-TH"/       28660
     *" VALUE OF PAR.NO.",I1," AND THE I-TH VALUE OF PAR.NO.",I1// )     28670
 1075 FORMAT(5(2X," J=",I2),3X,5(2X," J=",I2) / )                       28680
 1080 FORMAT(5(E10.4,5X))                                               28690
```

```
      COMMON /B10/LABEL                                                 360
      COMMON /B11/AR                                                    370
      COMMON /B12/NPR                                                   380
      COMMON /B13/IGNORE                                                390
      COMMON /B14/NVIOL                                                 400
      COMMON /B15/SKIP                                                  410
      LOGICAL SKIP,SKIPS                                                420
      EQUIVALENCE (AR(1),ANS)                                           430
      EXTERNAL LSQ                                                      440
      DATA TCRIT/12.7,4.30,3.18,2.78,2.57,2.45,2.36,2.31,2.26,2.23,     450
     *2.20,2.18,2.16,2.14,2.13,2.12,2.11,2.10,2.09,2.09,2.08,           460
     *2.07,2.07,2.06,2.06,2.06,2.05,2.05,2.04,2.04/                     470
      DATA NRECOM /0,0,343,625,1024,729,2187 /                          480
      DATA NCRIT /0,0,8,16,32,64,128 /                                  490
      WRITE(6,500)                                                      500
      LINES=30                                                          510
      NRUN=0                                                            520
      ITABLE=0                                                          530
      LIMITS=0                                                          540
      NCORR=0                                                           550
      NDUMMY=0                                                          560
      NS=1                                                              570
      NFUNC=-1                                                          580
      NOP=-1                                                            590
      NOP2=-1                                                           600
      NOBS2=-1                                                          610
```

B01

```
      IQUAD=1                                                           620
      SIMP=1E-6                                                         630
      ANS4=2HNO                                                         640
      PLOTRS=2HNO                                                       650
      MAX=800                                                           660
      NLOOP=20                                                          670
      IPRINT=-1                                                         680
      ANS11=2HNO                                                        690
      ANS12=2HNO                                                        700
      STOPCR=0.001                                                      710
      NIND=1                                                            720
      PLT=2HNO                                                          730
      ANS16=2HNO                                                        740
    5 NQ=1                                                              750
      WRITE(6,505)                                                      760
      IGNORE=0                                                          770
      CALL READ(1,AR)                                                   780
      GO TO (1000,1001) IGO                                             790
    6 IF(ANS.NE.3HYES) GO TO 8                                          800
      WRITE(6,511)                                                      810
      WRITE(6,512)                                                      820
    8 IF(NS.GT.NQ) GO TO 1006                                           830
    9 CONTINUE                                                          840
   10 NQ=2                                                              850
      WRITE(6,520)                                                      860
      READ(5,525)                                                       870
      IF(EOF(5))10,11,10                                                880
   11 IF(NS.GT.NQ) GO TO 1006                                           890
   15 NQ=3                                                              900
      WRITE(6,530)                                                      910
      CALL READ(1,AR)                                                   920
      GO TO (1000,1001) IGO                                             930
   16 IF(ANS.EQ.3HYES) IQUAD=1                                          940
      IF(ANS.EQ.2HNO) IQUAD=0                                           950
      IF(ANS.EQ.2HNO) GO TO 18                                          960
      WRITE(6,535)                                                      970
      CALL READ(1,AR)                                                   980
      IF(IGO.EQ.2) GO TO 1001                                           990
      IF(ANS.EQ.4HBACK) GO TO 1004                                     1000
      SIMP=ANS                                                         1010
      IF(SIMP.LT.0..OR.SIMP.GT.10.)GO TO 1007                          1020
   18 IF(NS.GT.NQ) GO TO 1006                                          1030
   20 NQ=4                                                             1040
      WRITE(6,540)                                                     1050
      CALL READ(1,AR)                                                  1060
      GO TO (1000,1001) IGO                                            1070
   21 ANS4=ANS                                                         1080
      IF(ANS4.EQ.2HNO) GO TO 24                                        1090
      WRITE(6,541)                                                     1100
   22 CALL READ(1,AR)                                                  1110
      IF(IGO.EQ.2) GO TO 1001                                          1120
      IF(ANS.EQ.4HBACK) GO TO 1004                                     1130
      PLOTRS=ANS                                                       1140
      IF(PLOTRS.NE.2HNO.AND.PLOTRS.NE.3HYES) GO TO 23                  1150
      GO TO 24                                                         1160
   23 WRITE(6,1020)                                                    1170
      GO TO 22                                                         1180
   24 IF(NS.GT.NQ) GO TO 1006                                          1190
   25 NQ=5                                                             1200
      WRITE(6,545)                                                     1210
      CALL READ(1,AR)                                                  1220
      GO TO (1003,1001)IGO                                             1230
```

C01

```
   26 MAX=IFIX(ANS+.1)                                                 1240
      IF(MAX.LT.1.OR.MAX.GT.2000) GO TO 1007                           1250
      IF(NS.GT.NQ) GO TO 1006                                          1260
   30 NQ=6                                                             1270
      WRITE(6,550)                                                     1280
      CALL READ(1,AR)                                                  1290
      GO TO (1003,1001)IGO                                             1300
   31 NLOOP=IFIX(ANS+.1)                                               1310
      IF(NLOOP.LT.1) GO TO 1007                                        1320
      IF(NLOOP.GT.MAX.AND.MAX.NE.1) GO TO 32                           1330
      GO TO 33                                                         1340
   32 MAXR=5                                                           1350
      GO TO 1008                                                       1360
   33 IF(NS.GT.NQ) GO TO 1006                                          1370
   35 NQ=7                                                             1380
```

```
      DELTA=(XMAX-XMIN)/50.                                                  9670
      IF(II.EQ.1) WRITE(6,8261)                                              9680
      IF(II.GT.1) WRITE(6,8264)II                                            9690
      DO 3706 I=1,26                                                         9700
      J=26+I                                                                 9710
      XP1=XMIN+FLOAT(I-1)*DELTA                                              9720
      XP2=XMIN+FLOAT(J-1)*DELTA                                              9730
      DO 3708 L=N1,NN                                                        9740
      IF(XP1.GT.X(L).AND.XP1.LE.X(L+1))                                      9750
     *YP1=Y(L)+(Y(L+1)-Y(L))*(XP1-X(L))/(X(L+1)-X(L))                         9760
 3708 IF(XP2.GT.X(L).AND.XP2.LE.X(L+1))                                      9770
     *YP2=Y(L)+(Y(L+1)-Y(L))*(XP2-X(L))/(X(L+1)-X(L))                         9780
      ITHFUN=II                                                              9790
      CALL MODEL(YP1,XP1,F,0)                                                9800
      CALL MODEL(YP2,XP2,F,0)                                                9810
      IF(I.EQ.26) GO TO 3707                                                 9820
 3706 WRITE(6,8262) I,XP1,YP1,J,XP2,YP2                                      9830
 3707 WRITE(6,8262) I,XP1,YP1                                                9840
 3709 CONTINUE                                                               9850
      WRITE(6,8263)                                                          9860
  371 CONTINUE                                                               9870
      IF(IQUAD.NE.1) GO TO 376                                               9880
      WRITE(6,827)                                                           9890
      DO 375 K=1,2                                                           9900
      IF(K.EQ.2) WRITE(6,8270)                                               9910
```

B02

```
      L=1                                                                    9920
  372 IF(L.GT.NPR) GO TO 375                                                 9930
      II=L*(L-1)/2                                                           9940
      DO 373 I=L,NPR                                                         9950
      IA=II+L                                                                9960
      II=II+I                                                                9970
      IB=MIN0(II,IA+5)                                                       9980
      IF(K.EQ.2) GO TO 3721                                                  9990
      DO 3720 J=IA,IB                                                       10000
 3720 VARCOV(J)=2.*FUNC*VARCOV(J)/FLOAT(NRSDF)                              10010
      WRITE(6,829)(VARCOV(J),J=IA,IB)                                       10020
      GO TO 373                                                             10030
 3721 DO 3722 KK=IA,IB                                                      10040
 3722 VARCOV(KK)=VC(KK)                                                     10050
      WRITE(6,829)(VARCOV(J),J=IA,IB)                                       10060
  373 CONTINUE                                                              10070
      L=L+6                                                                 10080
      GO TO 372                                                             10090
  375 CONTINUE                                                              10100
  376 IF(ANS4.EQ.2HNO) GO TO 395                                            10110
      DWSTAT=0.                                                             10120
      DO 380 I=2,NOBS                                                       10130
  380 DWSTAT=DWSTAT+(RES(I-1)-RES(I))**2                                    10140
      DWSTAT=DWSTAT/SSRES                                                   10150
      WRITE(6,830)DWSTAT                                                    10160
      N=0                                                                   10170
      DO 381 I=1,NOBS                                                       10180
      IF(WYN(I).EQ.0.) GO TO 381                                            10190
      N=N+1                                                                 10200
      VARCOV(N)=RES(I)                                                      10210
  381 CONTINUE                                                              10220
      PR=SEQPRO(VARCOV,N)                                                   10230
      WRITE(6,832)PR                                                        10240
      IF(PR.LT.0.05) WRITE(6,833)                                           10250
      IF(NOBS.GT.3) CALL KOLMIR(VARCOV,N,DN,CRIT,1)                         10260
      CALL PROB1(VARCOV,N,MM,PR)                                            10270
      WRITE(6,8320)MM,N,PR                                                  10280
      IF(PR.LT.0.05) WRITE(6,833)                                           10290
      RATIO=FLOAT(NRSDF)/ FLOAT(NOBS-NDUMMY)                                10300
      IF(RATIO.LE..5) WRITE(6,834)RATIO                                     10310
      IGNORE=1                                                              10320
      IF(PLOTRS.NE.3HYES) GO TO 395                                         10330
      IF(NIND.GT.1) GO TO 390                                               10340
      WRITE(6,836)                                                          10350
      CALL PLOT(X,RES,NOBS,LINES)                                           10360
      WRITE(6,838)                                                          10370
      CALL PLOT(YEST,RES,NOBS,LINES)                                        10380
      GO TO 391                                                             10390
  390 WRITE(6,840)                                                          10400
      CALL PLOT(X,RES,NOBS,LINES)                                           10410
      WRITE(6,842)                                                          10420
      CALL PLOT(Z,RES,NOBS,LINES)                                           10430
      WRITE(6,838)                                                          10440
      CALL PLOT(YEST,RES,NOBS,LINES)                                        10450
  391 IF(ANS16.EQ.2HNO.OR.NOWGHT.EQ.1) GO TO 395                            10460
      IF(NIND.GT.1) GO TO 392                                               10470
      WRITE(6,8360)                                                         10480
      CALL PLOT(X,WRES,NOBS,LINES)                                          10490
      WRITE(6,8380)                                                         10500
      CALL PLOT(YEST,WRES,NOBS,LINES)                                       10510
      GO TO 395                                                             10520
  392 WRITE(6,8400)                                                         10530
```

C02

```
      CALL PLOT(X,WRES,NOBS,LINES)                                          10540
      WRITE(6,8420)                                                         10550
      CALL PLOT(Z,WRES,NOBS,LINES)                                          10560
      WRITE(6,8380)                                                         10570
      CALL PLOT(YEST,WRES,NOBS,LINES)                                       10580
  395 CONTINUE                                                              10590
      WRITE(6,8401)STOPCR                                                   10600
      IF(IQUAD.NE.1) GO TO 396                                              10610
      WRITE(6,8402)SIMP                                                     10620
      D=ABS(1.-AMING/FUNC)                                                  10630
      IF(IPRINT.EQ.-1.AND.D.GT.0.1 )  WRITE(6,8403)AMING,FUNC               10640
  396 NEVAL1=NEVAL1-NVIOL                                                   10650
      WRITE(6,8422)NEVAL1,NVIOL                                             10660
      IF(IQUAD.EQ.1.AND.IPRINT.EQ.-1)WRITE(6,8424)NEVAL2                    10670
      ITABLE=1                                                              10680
      ITHFUN=NFUNC                                                          10690
```

```
      H(I)=0D0                                                          18980
      DO 214 J=1,NAP                                                    18990
      IF(J-I)216,216,215                                                19000
  216 IJ=I*(I-1)/2+J                                                    19010
      GO TO 217                                                         19020
  215 IJ=J*(J-1)/2+I                                                    19030
  217 H(I)=H(I)+BMAT(IJ)*AVAL(J)                                        19040
  214 CONTINUE                                                          19050
  213 CONTINUE                                                          19060
      YMIN=0D0                                                          19070
      DO 218 I=1,NAP                                                    19080
  218 YMIN=YMIN+H(I)*AVAL(I)                                            19090
      YMIN=A0-YMIN                                                      19100
      DO 220 I=1,NOP                                                    19110
      PSTST(I)=0.0                                                      19120
      DO 220 J=1,NAP                                                    19130
  220 PSTST(I)=PSTST(I)+H(J)*G(J,I)                                     19140
      DO 221 I=1,NOP                                                    19150
  221 PMIN(I)=PMIN(I)-PSTST(I)                                          19160
      AMINQ=YMIN                                                        19170
      IF(IPRINT)223,222,222                                             19180
  222 WRITE(IW,303) YMIN,(PMIN(I),I=1,NPR)                              19190
  303 FORMAT(42H MINIMUM OF FITTED QUADRATIC SURFACE IS   ,D15.8,3H AT//19200
     *(5E14.6))                                                         19210
```

803

```
      WRITE(IW,304)FUNC,(F(I),I=1,NPR)                                  19220
  304 FORMAT(/42H COMPARE WITH MINIMUM FOUND BY ITERATION  ,D15.8,3H AT/ 19230
     */(5E14.6))                                                        19240
      D=SNGL(1D0-YMIN/FUNC)                                             19250
      IF(ABS(D).GT..1) WRITE(IW,305)                                    19260
  305 FORMAT(/                                                          19270
     *' IF DIFFERENCE IS LARGE,INFORMATION MATRIX IS INACCURATE AND A'/  19280
     *' NEW RUN WITH A DIFFERENT STOPPING CRITERION, EXPANSION',        19290
     *' CRITERION'/' OR PARAMETER RANGE IS RECOMMENDED.'/  )             19300
  223 CONTINUE                                                          19310
      DO 2230 I=1,NOP                                                   19320
      IF(PMIN(I).LT.F1(I))PMIN(I)=F1(I)                                 19330
 2230 IF(PMIN(I).GT.F2(I))PMIN(I)=F2(I)                                 19340
      CALL FUNCTN(PMIN,QMIN)                                            19350
      IF(QMIN.GE.FUNC) GO TO 2234                                       19360
      FUNC=QMIN                                                         19370
      DO 2231 I=1,NOP                                                   19380
 2231 F(I)=PMIN(I)                                                      19390
      IF(IPRINT.LT.0) GO TO 2234                                        19400
      WRITE(IW,2232)FUNC                                                19410
 2232 FORMAT(/' IMPROVED FUNCTION VALUE =',D15.8/)                      19420
      IF(NPR.LE.5) WRITE(IW,1080)(F(I),I=1,NPR)                         19430
      IF(NPR.GT.5) WRITE(IW,108)(F(I),I=1,NPR)                          19440
 2234 CONTINUE                                                          19450
      DO 224 I=1,NOP                                                    19460
      DO 225 J=1,NAP                                                    19470
      H(J)=0D0                                                          19480
      DO 226 K=1,NAP                                                    19490
      IF(K-J)227,227,228                                                19500
  227 JK=J*(J-1)/2+K                                                    19510
      GO TO 229                                                         19520
  228 JK=K*(K-1)/2+J                                                    19530
  229 H(J)=H(J)+BMAT(JK)*DBLE(G(K,I))/2D0                               19540
  226 CONTINUE                                                          19550
  225 CONTINUE                                                          19560
      DO 230 J=I,NOP                                                    19570
      IJ = J*(J-1)/2+I                                                  19580
      VC(IJ)=0D0                                                        19590
      DO 231 K=1,NAP                                                    19600
  231 VC(IJ)=VC(IJ)+H(K)*DBLE(C(K,J))                                   19610
  230 CONTINUE                                                          19620
  224 CONTINUE                                                          19630
      J = 0                                                             19640
      DO 234 I=1,NOP                                                    19650
      J = J+I                                                           19660
  234 VAR(I)=VC(J)                                                      19670
  236 IF(IPRINT.GE.0) WRITE(IW,306) IRANK                               19680
  306 FORMAT(/' INFORMATION MATRIX HAS RANK',I3//                       19690
     *' GENERALIZED INVERSE OF INFORMATION MATRIX'/  )                  19700
      IJK=1                                                             19710
      GO TO 710                                                         19720
  717 CONTINUE                                                          19730
      IF(IPRINT.GE.0) WRITE(IW,308)                                     19740
  308 FORMAT(                                                           19750
     *' THE MATRIX ABOVE MUST BE MULTIPLIED BY TWICE THE ESTIMATE OF',  19760
     *' THE RESIDUAL'/' VARIANCE TO GIVE THE VARANCE-COVARIANCE MATRIX',19770
     *' OF THE PARAMETERS.'/  )                                         19780
      CALL SYMINV(VC,NAP,BMAT,TEMP,NULLTY,IFAULT)                       19790
      IF(IPRINT.GE.0) WRITE(IW,320)                                     19800
  320 FORMAT(/' INFORMATION MATRIX (HESSIAN MATRIX)' / )                19810
      IJK=3                                                             19820
      GO TO 710                                                         19830
```

C03

```
  711 IJK=2                                                             19840
      II = 0                                                            19850
      IJ = 0                                                            19860
      DO 701 I=1,NPR                                                    19870
      II = II+I                                                         19880
      IF(VC(II).LE.0D0) GO TO 702                                       19890
      VC(II)=1D0/DSQRT(VC(II))                                          19900
      GO TO 703                                                         19910
  702 VC(II)=0D0                                                        19920
  703 IF(I.EQ.1) GO TO 701                                              19930
      IM1 = I-1                                                         19940
      JJ = 0                                                            19950
      DO 705 J=1,IM1                                                    19960
      JJ = JJ+J                                                         19970
      IJ = IJ+1                                                         19980
  705 VC(IJ) = VC(IJ)*VC(II)*VC(JJ)                                     19990
```

```
  225 IF(NOP.EQ.3) GO TO 270                                              28280
      IF(F1(4).EQ.F2(4)) GO TO 235                                        28290
  230 CONTINUE                                                            28300
  235 IF(NOP.EQ.4) GO TO 270                                              28310
      IF(F1(5).EQ.F2(5)) GO TO 245                                        28320
  240 CONTINUE                                                            28330
  245 IF(NOP.EQ.5) GO TO 270                                              28340
      IF(F1(6).EQ.F2(6)) GO TO 255                                        28350
  250 CONTINUE                                                            28360
  255 IF(NOP.EQ.6) GO TO 270                                              28370
      IF(F1(7).EQ.F2(7)) GO TO 270                                        28380
  260 CONTINUE                                                            28390
  270 CONTINUE                                                            28400
      DO 280 I=1,7                                                        28410
      FNEW(I)=PP(I,IS(I))                                                 28420
  280 STPNEW(I)=DEL(I)/2.                                                 28430
      RETURN                                                              28440
 1000 FORMAT(// 11X,                                                      28450
     *" PLOT OF RESIDUAL SUM OF SQUARES VERSUS PARAMETER NO.",I2/          28460
     * 11X," RANGING FROM ",E11.4," TO ",E11.4  )                         28470
 1010 FORMAT(// 6X,                                                       28480
     *" PLOT OF WEIGHTED RESIDUAL SUM OF SQUARES VERSUS PARAMETER",       28490
     *" NO.",I2/6X," RANGING FROM ",E11.4," TO ",E11.4  )                28500
 1020 FORMAT(/ " TABLE OF PLOT ABOVE"//                                   28510
```

804

```
     *2(" NO.  SUM OF SQUARES   PARAMETER  ")/1X,68(1H-))                 28520
 1025 FORMAT(/ " TABLE OF PLOT ABOVE"//                                   28530
     *2(" NO. WGHT.SUM OF SQUARES PARAMETER  ")/1X,68(1H-))               28540
 1026 FORMAT(I4,2E14.4,3X,I4,2E14.4)                                      28550
 1027 FORMAT(1X,68(1H-))                                                  28560
 1030 FORMAT(//" THE SEARCH GRID CONSISTS OF 100 POINTS GIVEN BY",        28570
     *" PAIRED"/" COMBINATION OF FOLLOWING PARAMETER VALUES"//            28580
     *" PAR.NO.",I1,"  (J=1-10)"//(5(  "(",I2,")",E10.4))  )              28590
 1040 FORMAT(/" PAR.NO.",I1,"  (I=1-10)" //(5(  "(",I2,")",E10.4))  )     28600
 1050 FORMAT(/                                                            28610
     *" MATRIX OF RESIDUAL SUM OF SQUARES AT POINTS IN SEARCH GRID."  )    28620
 1060 FORMAT( /" MATRIX OF WEIGHTED RESIDUAL SUM OF SQUARES AT POINTS",   28630
     *" IN SEARCH GRID."  )                                               28640
 1070 FORMAT(                                                             28650
     *" THE I,J-TH ELEMENT IN THE MATRIX CORRESPONDS TO THE J-TH"/        28660
     *" VALUE OF PAR.NO.",I1," AND THE I-TH VALUE OF PAR.NO.",I1// )      28670
 1075 FORMAT(5(2X," J=",I2),3X,5(2X," J=",I2) / )                         28680
 1080 FORMAT(5(E10.4,5X))                                                 28690
 1085 FORMAT(2X,5(5X,E10.4))                                              28700
 1090 FORMAT(//6X," SIGN OF THE PARTIAL DERIVATIVES AT THE POINTS IN",    28710
     *" THE SEARCH GRID." // 6X,                                          28720
     *" WITH RESPECT TO PAR.NO.",I1,15X," WITH RESPECT TO PAR.NO.",I1/ )  28730
 1100 FORMAT(10X,10A2,20X,10A2)                                           28740
 1110 FORMAT(//12X," CONTOUR CHART"/45X,12(1H-)," KEY ",13(1H-)  )        28750
 1120 FORMAT(5X,10A3 )                                                    28760
 1130 FORMAT(45X,I1,"  = ",E10.4,"  TO ",E10.4  )                         28770
      END                                                                 28780
C     ****************************************************************   28790
      FUNCTION SIGNOF(A)                                                  28800
      IF(A.LT.0.)SIGNOF=2H-                                               28810
      IF(A.GT.0.)SIGNOF=2H+                                               28820
      IF(ABS(A).LT.1.E-15) SIGNOF=2H0                                     28830
      RETURN                                                              28840
      END                                                                 28850
```

C04

```
      WRITE(6,555)                                                      970
      CALL READ(1,AR)                                                   980
      IF(IGO.EQ.2) GO TO 1001                                           990
      IF(ANS.EQ.4HBACK) GO TO 1004                                     1000
      SIMP=ANS                                                         1010
      IF(SIMP.LT.0..OR.SIMP.GT.10.)GO TO 1007                          1020
   18 IF(NS.GT.NQ) GO TO 1006                                          1030
   20 NQ=4                                                             1040
      WRITE(6,540)                                                     1050
      CALL READ(1,AR)                                                  1060
      GO TO (1000,1001) IGO                                            1070
   21 ANS=ANS                                                          1080
      IF(ANS4.EQ.2HNO) GO TO 24                                        1090
      WRITE(6,541)                                                     1100
   22 CALL READ(1,AR)                                                  1110
      IF(IGO.EQ.2) GO TO 1001                                          1120
      IF(ANS.EQ.4HBACK) GO TO 1004                                     1130
      PLOTRS=ANS                                                       1140
      IF(PLOTRS.NE.2HNO.AND.PLOTRS.NE.3HYES) GO TO 23                  1150
      GO TO 24                                                         1160
   23 WRITE(6,1020)                                                    1170
      GO TO 22                                                         1180
   24 IF(NS.GT.NQ) GO TO 1006                                          1190
   25 NQ=5                                                             1200
      WRITE(6,545)                                                     1210
      CALL READ(1,AR)                                                  1220
      GO TO (1003,1001)IGO                                             1230
```

C01

```
   26 MAX=IFIX(ANS+.1)                                                 1240
      IF(MAX.LT.1.OR.MAX.GT.2000) GO TO 1007                           1250
      IF(NS.GT.NQ) GO TO 1006                                          1260
   30 NQ=6                                                             1270
      WRITE(6,550)                                                     1280
      CALL READ(1,AR)                                                  1290
      GO TO (1003,1001)IGO                                             1300
   31 NLOOP=IFIX(ANS+.1)                                               1310
      IF(NLOOP.LT.1) GO TO 1007                                        1320
      IF(NLOOP.GT.MAX.AND.MAX.NE.1) GO TO 32                           1330
      GO TO 33                                                         1340
   32 NAGR=5                                                           1350
      GO TO 1008                                                       1360
   33 IF(NS.GT.NQ) GO TO 1006                                          1370
   35 NQ=7                                                             1380
      WRITE(6,555)                                                     1390
      CALL READ(1,AR)                                                  1400
      GO TO (1003,1001)IGO                                             1410
   36 IPRINT=IFIX(ANS+.1)                                              1420
      IF(ANS.LT.0..AND.ANS.GE.-1.)  IPRINT=-1                          1430
      IF(IPRINT.LT.-1.OR.IPRINT.GE.MAX.AND.MAX.NE.1) GO TO 1007        1440
      IF(NS.GT.NQ) GO TO 1006                                          1450
   40 NQ=8                                                             1460
      WRITE(6,560)                                                     1470
      CALL READ(1,AR)                                                  1480
      GO TO (1003,1001)IGO                                             1490
   41 NFUNC=IFIX(ANS+.1)                                               1500
      IF(NFUNC.LT.1.OR.NFUNC.GT.10) GO TO 1007                         1510
      IF(NFUNC.EQ.1) GO TO 4405                                        1520
      WRITE(6,561)NFUNC                                                1530
      NOBS=0                                                           1540
      DO 4400 I=1,NFUNC                                                1550
   42 CALL READ(2,AR)                                                  1560
      IF(IGO.EQ.2) GO TO 1001                                          1570
      IF(ANS.EQ.4HBACK) GO TO 1004                                     1580
      J=IFIX(ANS+.1)                                                   1590
      IF(J.LT.1.OR.J.GT.10) GO TO 43                                   1600
      GO TO 44                                                         1610
   43 WRITE(6,1024)                                                    1620
      GO TO 42                                                         1630
   44 ITHSET(J)=IFIX(AR(2)+.1)                                         1640
      IF(ITHSET(J).LT.1.OR.ITHSET(J).GT.100) GO TO 43                  1650
 4400 NOBS=NOBS+ITHSET(J)                                              1660
      GO TO 4410                                                       1670
 4405 WRITE(6,562)                                                     1680
      CALL READ(1,AR)                                                  1690
      IF(IGO.EQ.2) GO TO 1001                                          1700
      IF(ANS.EQ.4HBACK) GO TO 1004                                     1710
      NOBS=IFIX(ANS+.1)                                                1720
      ITHSET(1)=NOBS                                                   1730
 4410 IF(NOBS.LT.2.OR.NOBS.GT.200) GO TO 1007                          1740
      N=1                                                              1750
      DO 4415 I=1,NFUNC                                                1760
      LIM1(I)=N                                                        1770
      N1=N                                                             1780
      LIM2(I)=LIM1(I)+ITHSET(I)-1                                      1790
      N2=LIM2(I)                                                       1800
      N=N2+1                                                           1810
      DO 4415 J=N1,N2                                                  1820
 4415 LABEL(J)=I                                                       1830
      IF(NS.GT.NQ) GO TO 1006                                          1840
   45 NQ=9                                                             1850
```

D01

```
      WRITE(6,565)                                                     1860
      CALL READ(1,AR)                                                  1870
      GO TO (1003,1001)IGO                                             1880
   46 NOP=IFIX(ANS+.1)                                                 1890
      IF(NOP.LT.1) GO TO 1007                                          1900
      IF(NS.GT.NQ) GO TO 1006                                          1910
   50 NQ=10                                                            1920
      WRITE(6,570)NOP                                                  1930
      NOP2=NOP+NOP/2                                                   1940
      DO 54 I=1,NOP2                                                   1950
      CALL READ(3,AR)                                                  1960
      GO TO (1003,1001)IGO                                             1970
   51 J=IFIX(ANS+.1)                                                   1980
      IF(AR(2).GT.AR(3)) GO TO 1009                                    1990
```

```
      CALL PROBT(VARLOV,M,MM,PR)                                        10270
      WRITE(6,8320)MM,N,PR                                              10280
      IF(PR.LT.0.05) WRITE(6,833)                                       10290
      RATIO=FLOAT(NRSDF)/ FLOAT(NOBS-NDUMMY)                            10300
      IF(RATIO.LE..5) WRITE(6,834)RATIO                                 10310
      IGNORE=1                                                          10320
      IF(PLOTRS.NE.3HYES) GO TO 395                                     10330
      IF(NIND.GT.1) GO TO 390                                           10340
      WRITE(6,836)                                                      10350
      CALL PLOT(X,RES,NOBS,LINES)                                       10360
      WRITE(6,838)                                                      10370
      CALL PLOT(YEST,RES,NOBS,LINES)                                    10380
      GO TO 391                                                         10390
  390 WRITE(6,840)                                                      10400
      CALL PLOT(X,RES,NOBS,LINES)                                       10410
      WRITE(6,842)                                                      10420
      CALL PLOT(Z,RES,NOBS,LINES)                                       10430
      WRITE(6,838)                                                      10440
      CALL PLOT(YEST,RES,NOBS,LINES)                                    10450
  391 IF(ANS16.EQ.2HNO.OR.NOWGHT.EQ.1) GO TO 395                        10460
      IF(NIND.GT.1) GO TO 392                                           10470
      WRITE(6,8360)                                                     10480
      CALL PLOT(X,WRES,NOBS,LINES)                                      10490
      WRITE(6,8380)                                                     10500
      CALL PLOT(YEST,WRES,NOBS,LINES)                                   10510
      GO TO 395                                                         10520
  392 WRITE(6,8400)                                                     10530
```

C02

```
      CALL PLOT(X,WRES,NOBS,LINES)                                      10540
      WRITE(6,8420)                                                     10550
      CALL PLOT(Z,WRES,NOBS,LINES)                                      10560
      WRITE(6,8380)                                                     10570
      CALL PLOT(YEST,WRES,NOBS,LINES)                                   10580
  395 CONTINUE                                                          10590
      WRITE(6,8401)STOPCR                                               10600
      IF(IQUAD.NE.1) GO TO 396                                          10610
      WRITE(6,8402)SIMP                                                 10620
      D=ABS(1.-AMINQ/FUNC)                                              10630
      IF(IPRINT.EQ.-1.AND.D.GT.0.1 )  WRITE(6,8403)AMINQ,FUNC           10640
  396 NEVAL1=NEVAL1-NVIOL                                               10650
      WRITE(6,8422)NEVAL1,NVIOL                                         10660
      IF(IQUAD.EQ.1.AND.IPRINT.EQ.-1)WRITE(6,8424)NEVAL2                10670
      ITABLE=1                                                          10680
      ITHFUN=NFUNC                                                      10690
      N=LIM1(NFUNC)                                                     10700
      AA=YEST(N)                                                        10710
      XMIN=X(1)                                                         10720
      XMAX=X(1)                                                         10730
      DO 3960 I=2,NOBS                                                  10740
      IF(X(I).GT.XMAX) XMAX=X(I)                                        10750
 3960 IF(X(I).LT.XMIN) XMIN=X(I)                                        10760
      DO 397 J=1,NIND                                                   10770
  397 G(J)=XX(J,N)                                                      10780
      WRITE(6,8425)                                                     10790
      IGNORE=0                                                          10800
      IF(NIND.EQ.1) CALL MODEL(AA,G(1),F,1)                             10810
      IF(NIND.GT.1) CALL MODEL(AA,G,F,1)                                10820
      ITABLE=0                                                          10830
      WRITE(6,844)NRUN                                                  10840
      ANS11=2HNO                                                        10850
      ANS12=2HNO                                                        10860
      NS=19                                                             10870
      GO TO 160                                                         10880
  500 FORMAT(//13(1H*),                                                 10890
     *" Y O U  A R E  N O W  L I N K E D  T O  F U N F I T ",14(1H*)/    10900
     *66X," VERSION 7-76"/  )                                           10910
  505 FORMAT(" (1) DO YOU WANT SOME INFORMATION ABOUT THIS PROGRAM",     10920
     *" (TYPE YES OR NO)" //  )                                         10930
  511 FORMAT(/                                                          10940
     *" IN THE YES-OR-NO TYPE QUESTIONS THAT FOLLOW,ONLY FOUR ANSWERS"/ 10950
     *" ARE ALLOWED,NAMELY  YES  NO  BACK  REQUEST"/                    10960
     *" OR IN SHORT FORM     Y    N    B    R"/                        10970
     *" BACK (B) CAN ALSO BE ENTERED AT ANY TIME ON REQUESTS THAT ASK"/ 10980
     *" FOR NUMERICAL DATA. IT ALLOWS YOU TO GO BACK AND CORRECT ANY"/  10990
     *" OF YOUR PREVIOUS INPUTS AND THEN CONTINUE FROM CURRENT INPUT"/  11000
     *" REQUEST. SIMPLY TYPE BACK** (OR B**) ANYWHERE ON A LINE IF YOU"/ 11010
     *" WANT TO GO BACK AND CORRECT THE INPUT UNDER REQUEST NUMBER **."/ 11020
     *" THE B** FEATURE ALSO MAKES IT EASY TO REPEAT A RUN WITH ONLY"/  11030
     *" A FEW CHANGES IN THE INPUTS. BY TYPING R** YOU WILL BE ",       11040
     *"TRANFERED"/" BACKWARD OR FORWARD TO REQUEST NUMBER **"/          11050
     *" THE R-COMMAND AND THE DEFAULT VALUES ENABLE A FAST INPUT."/     11060
     *" INPUT ERRORS IN REQUEST NO.10 OR 17 CAN BE CORRECTED",          11070
     *" BY RETYPING"/" THE WRONG LINE(S) IN CORRECT FORM. THE LAST LINE" 11080
     *," STARTS THE NEXT REQUEST."/)                                    11090
  512 FORMAT(                                                           11100
     *" A PARAMETER CAN BE CHANGED TO A CONSTANT IN REQUEST NO.10 BY"/  11110
     *" ASSIGNING IT THE SAME LOWER AND UPPER LIMITS. THIS ENABLES ANY"/ 11120
     *" SUM-OF-SQUARES CONTOUR TO BE EVALUATED (REQUEST NO.19) BY "/    11130
     *" FIXING ALL EXECEPT 2 PARAMETERS."/                              11140
     *" A LATTICE SEARCH FOR OPTIMAL INITIAL PARAMETER ESTIMATES CAN"/  11150
```

D02

```
     *" BE MADE BEFORE THE START OF THE FITTING PROCEDURE BY ENTERING"/  11160
     *" 1 AS THE MAX. NO. OF SUM-OF-SQUARES FUNCTION EVALUATIONS"/       11170
     *" ALLOWED IN REQUEST NO.5."/                                       11180
     *" THE DEFAULT VALUE SPECIFIED IN SOME REQUESTS CAN BE ENTERED IF"/ 11190
     *" THERE IS DOUBT ABOUT WHICH VALUE TO ENTER."/                     11200
     *" FOR MORE DETAILED INFORMATION ABOUT THE PROGRAM CONTACT"/        11210
     *" P.VENG PEDERSEN."//)                                            11220
  520 FORMAT(" (2) ENTER A HEADING BUT NOT MORE THAN ONE LINE"// )       11230
  525 FORMAT(80H                                                        11240
     *                          )                                      11250
  530 FORMAT(" (3) DO YOU WANT STATISTICAL EVALUATIONS OF THE",         11260
     *" PARAMETERS (DEFAULT=YES)"// )                                   11270
  535 FORMAT(" ENTER THE EXPANSION CRITERION TO AVOID ROUNDING-OFF",     11280
     *" ERRORS"/" IN THE STATISTICAL EVALUATION OF THE PARAMETERS",     11290
```

```
      IJ = J*(J-1)/2+I                                                  19580
      VC(IJ)=0D0                                                        19590
      DO 231 K=1,NAP                                                    19600
  231 VC(IJ)=VC(IJ)+H(K)*DBLE(C(K,J))                                   19610
  230 CONTINUE                                                          19620
  224 CONTINUE                                                          19630
      J = 0                                                             19640
      DO 234 I=1,NOP                                                    19650
      J = J+I                                                           19660
  234 VAR(I)=VC(J)                                                      19670
  236 IF(IPRINT.GE.0) WRITE(IW,306) IRANK                               19680
  306 FORMAT(/" INFORMATION MATRIX HAS RANK",I3//                       19690
     *" GENERALIZED INVERSE OF INFORMATION MATRIX"/  )                  19700
      IJK=1                                                             19710
      GO TO 710                                                         19720
  717 CONTINUE                                                          19730
      IF(IPRINT.GE.0) WRITE(IW,308)                                     19740
  308 FORMAT(                                                           19750
     *" THE MATRIX ABOVE MUST BE MULTIPLIED BY TWICE THE ESTIMATE OF",  19760
     *" THE RESIDUAL VARIANCE TO GIVE THE VARANCE-COVARIANCE MATRIX",   19770
     *" OF THE PARAMETERS." / )                                         19780
      CALL SYMINV(VC,NAP,BMAT,TEMP,NULLTY,IFAULT)                       19790
      IF(IPRINT.GE.0) WRITE(IW,320)                                     19800
  320 FORMAT(/" INFORMATION MATRIX (HESSIAN MATRIX)" / )                19810
      IJK=3                                                             19820
      GO TO 710                                                         19830
```

C03

```
  711 IJK=2                                                             19840
      II = 0                                                            19850
      IJ = 0                                                            19860
      DO 701 I=1,NPR                                                    19870
      II = II+I                                                         19880
      IF(VC(II).LE.0D0) GO TO 702                                       19890
      VC(II)=1D0/DSQRT(VC(II))                                          19900
      GO TO 703                                                         19910
  702 VC(II)=0D0                                                        19920
  703 IF(I.EQ.1) GO TO 701                                              19930
      IM1 = I-1                                                         19940
      JJ = 0                                                            19950
      DO 705 J=1,IM1                                                    19960
      JJ = JJ+J                                                         19970
      IJ = IJ+1                                                         19980
  705 VC(IJ) = VC(IJ)*VC(II)*VC(JJ)                                     19990
  701 IJ = IJ+1                                                         20000
      IF(IPRINT.GE.0) WRITE(IW,312)                                     20010
  312 FORMAT(" CORRELATION MATRIX" / )                                  20020
      II = 0                                                            20030
      DO 706 I=1,NPR                                                    20040
      II = II+I                                                         20050
      IF(VC(II).NE.0D0) VC(II)=1D0                                      20060
  706 CONTINUE                                                          20070
      GO TO 710                                                         20080
  712 IF(IPRINT.GE.0) WRITE(IW,310)NEVAL                                20090
  310 FORMAT(    " A FUTHER",I4," FUNCTION EVALUATIONS HAVE BEEN USED"/) 20100
      NEVAL2=NEVAL                                                      20110
      IF(NFIX.GT.0.AND.IPRINT.GE.0) WRITE(6,3020)NFIX                   20120
  235 RETURN                                                            20130
  710 L=1                                                               20140
  716 IF(L.GT.NPR) GO TO (717,712,711),IJK                              20150
      II=L*(L-1)/2                                                      20160
      DO 713 I=L,NPR                                                    20170
      I1 = II+L                                                         20180
      II=II+I                                                           20190
      I2=MIN0(II,I1+5)                                                  20200
      IF(IJK.EQ.3) GO TO 718                                            20210
      IF(IPRINT.GE.0) WRITE(IW,714)(VC(J),J=I1,I2)                      20220
      IF(IJK.NE.1) GO TO 713                                            20230
      DO 500 J=I1,I2                                                    20240
  500 VARCOV(J)=VC(J)                                                   20250
      GO TO 713                                                         20260
  718 IF(IPRINT.GE.0) WRITE(IW,714)(BMAT(J),J=I1,I2)                    20270
  713 CONTINUE                                                          20280
  714 FORMAT(1X,6D13.6)                                                 20290
      WRITE(IW,715)                                                     20300
  715 FORMAT(1H ,/)                                                     20310
      L=L+6                                                             20320
      GO TO 716                                                         20330
      END                                                               20340
C     ******************************************************************20350
      SUBROUTINE CHOL  (A,N,U,NULLTY,IFAULT)                            20360
      DIMENSION A(55),U(55)                                             20370
      DOUBLE PRECISION A,U,W,ETA                                        20380
      ETA=1D-16                                                         20390
      NN=0                                                              20400
      IFAULT=1                                                          20410
      IF (N.LE.0) GO TO 100                                             20420
      IFAULT=2                                                          20430
      NULLTY=0                                                          20440
      J=1                                                               20450
```

D03

```
      K=0                                                               20460
      DO 10 ICOL = 1,N                                                  20470
      L=0                                                               20480
      DO 11 IROW = 1,ICOL                                               20490
      K=K+1                                                             20500
      W=A(K)                                                            20510
      M=J                                                               20520
      DO 12 I = 1,IROW                                                  20530
      L=L+1                                                             20540
      IF (I.EQ.IROW) GO TO 13                                           20550
      W=W-U(L)*U(M)                                                     20560
      M=M+1                                                             20570
   12 CONTINUE                                                          20580
   13 IF (IROW.EQ.ICOL) GO TO 14                                        20590
```

```
      GO TO 44                                                          1610
   43 WRITE(6,1024)                                                     1620
      GO TO 42                                                          1630
   44 ITHSET(J)=IFIX(AR(2)+.1)                                          1640
      IF(ITHSET(J).LT.1.OR.ITHSET(J).GT.100) GO TO 43                   1650
 4400 NOBS=NOBS+ITHSET(J)                                               1660
      GO TO 4410                                                        1670
 4405 WRITE(6,562)                                                      1680
      CALL READ(1,AR)                                                   1690
      IF(IGO.EQ.2) GO TO 1001                                           1700
      IF(ANS.EQ.4HBACK) GO TO 1004                                      1710
      NOBS=IFIX(ANS+.1)                                                 1720
      ITHSET(1)=NOBS                                                    1730
 4410 IF(NOBS.LT.2.OR.NOBS.GT.200) GO TO 1007                           1740
      N=1                                                               1750
      DO 4415 I=1,NFUNC                                                 1760
      LIM1(I)=N                                                         1770
      N1=N                                                              1780
      LIM2(I)=LIM1(I)+ITHSET(I)-1                                       1790
      N2=LIM2(I)                                                        1800
      N=N2+1                                                            1810
      DO 4415 J=N1,N2                                                   1820
 4415 LABEL(J)=I                                                        1830
      IF(NS.GT.NQ) GO TO 1006                                           1840
   45 NQ=9                                                              1850
```

D01

```
      WRITE(6,565)                                                      1860
      CALL READ(1,AR)                                                   1870
      GO TO (1003,1001)IGO                                              1880
   46 NOP=IFIX(ANS+.1)                                                  1890
      IF(NOP.LT.1) GO TO 1007                                           1900
      IF(NS.GT.NQ) GO TO 1006                                           1910
   50 NQ=10                                                             1920
      WRITE(6,570)NOP                                                   1930
      NOP2=NOP+NOP/2                                                    1940
      DO 54 I=1,NOP2                                                    1950
      CALL READ(3,AR)                                                   1960
      GO TO (1003,1001)IGO                                              1970
   51 J=IFIX(ANS+.1)                                                    1980
      IF(AR(2).GT.AR(3)) GO TO 1009                                     1990
      IF(J.LT.1.OR.J.GT.NOP) GO TO 52                                   2000
      GO TO 53                                                          2010
   52 NAGR=9                                                            2020
      GO TO 1008                                                        2030
   53 F1(J)=AR(2)                                                       2040
      F2(J)=AR(3)                                                       2050
      IF(J.EQ.NOP) GO TO 5400                                           2060
   54 CONTINUE                                                          2070
      WRITE(6,571)NOP                                                   2080
      GO TO 50                                                          2090
 5400 IF(NS.GT.NQ) GO TO 1006                                           2100
   55 NQ=11                                                             2110
      WRITE(6,575)                                                      2120
      CALL READ(1,AR)                                                   2130
      GO TO (1000,1001) IGO                                             2140
   56 ANS11=ANS                                                         2150
      IF(ANS.EQ.2HNO) GO TO 66                                          2160
      WRITE(6,580)                                                      2170
      CALL READ(1,AR)                                                   2180
      IF(IGO.EQ.2) GO TO 1001                                           2190
      IF(ANS.EQ.4HBACK) GO TO 1004                                      2200
      N11=IFIX(ANS+.1)                                                  2210
      IF(N11.LT.1) GO TO 1007                                           2220
      IF(N11.GT.NOP) GO TO 57                                           2230
      GO TO 58                                                          2240
   57 NAGR=9                                                            2250
      GO TO 1008                                                        2260
   58 IF(N11.GT.1) GO TO 59                                             2270
      WRITE(6,585)                                                      2280
      GO TO 60                                                          2290
   59 WRITE(6,590)N11                                                   2300
   60 CONTINUE                                                          2310
      DO 65 I=1,N11                                                     2320
      CALL READ(2,AR)                                                   2330
      IF(IGO.EQ.2) GO TO 1001                                           2340
      IF(ANS.EQ.4HBACK) GO TO 1004                                      2350
      J=IFIX(ANS+.1)                                                    2360
      IF(J.LT.1) GO TO 1007                                             2370
      IF(J.GT.NOP) GO TO 61                                             2380
      GO TO 62                                                          2390
   61 NAGR=9                                                            2400
      GO TO 1008                                                        2410
   62 IF(AR(2).LT.F1(J).OR.AR(2).GT.F2(J)) GO TO 63                     2420
      GO TO 64                                                          2430
   63 NAGR=10                                                           2440
      GO TO 1008                                                        2450
   64 NF(I)=J                                                           2460
   65 FS(J)=AR(2)                                                       2470
```

E01

```
   66 IF(NS.GT.NQ) GO TO 1006                                           2480
   70 NQ=12                                                             2490
      WRITE(6,595)                                                      2500
      CALL READ(1,AR)                                                   2510
      GO TO (1000,1001) IGO                                             2520
   71 ANS12=ANS                                                         2530
      IF(ANS.EQ.2HNO) GO TO 81                                          2540
      WRITE(6,600)                                                      2550
      CALL READ(1,AR)                                                   2560
      IF(IGO.EQ.2) GO TO 1001                                           2570
      IF(ANS.EQ.4HBACK) GO TO 1004                                      2580
      N12=IFIX(ANS+.1)                                                  2590
      IF(N12.LT.1) GO TO 1007                                           2600
      IF(N12.GT.NOP) GO TO 72                                           2610
      GO TO 73                                                          2620
```

```
   *66X," VERSION 7-76"/  )                                              10910
505 FORMAT(" (1) DO YOU WANT SOME INFORMATION ABOUT THIS PROGRAM",        10920
   *" (TYPE YES OR NO)" //  )                                            10930
511 FORMAT(/                                                             10940
   *" IN THE YES-OR-NO TYPE QUESTIONS THAT FOLLOW,ONLY FOUR ANSWERS"/    10950
   *" ARE ALLOWED,NAMELY  YES  NO  BACK  REQUEST"/                       10960
   *" OR IN SHORT FORM     Y    N    B    R"/                           10970
   *" BACK (B) CAN ALSO BE ENTERED AT ANY TIME ON REQUESTS THAT ASK"/     10980
   *" FOR NUMERICAL DATA. IT ALLOWS YOU TO GO BACK AND CORRECT ANY"/      10990
   *" OF YOUR PREVIOUS INPUTS AND THEN CONTINUE FROM CURRENT INPUT"/      11000
   *" REQUEST. SIMPLY TYPE BACK** (OR B**) ANYWHERE ON A LINE IF YOU"/    11010
   *" WANT TO GO BACK AND CORRECT THE INPUT UNDER REQUEST NUMBER **."/    11020
   *" THE B** FEATURE ALSO MAKES IT EASY TO REPEAT A RUN WITH ONLY"/      11030
   *" A FEW CHANGES IN THE INPUTS. BY TYPING R** YOU WILL BE ",           11040
   *"TRANFERED"/" BACKWARD OR FORWARD TO REQUEST NUMBER **"/              11050
   *" THE R-COMMAND AND THE DEFAULT VALUES ENABLE A FAST INPUT."/        11060
   *" INPUT ERRORS IN REQUEST NO.10 OR 17 CAN BE CORRECTED",             11070
   *" BY RETYPING"/" THE WRONG LINE(S) IN CORRECT FORM. THE LAST LINE"   11080
   *," STARTS THE NEXT REQUEST."/)                                       11090
512 FORMAT(                                                              11100
   *" A PARAMETER CAN BE CHANGED TO A CONSTANT IN REQUEST NO.10 BY"/     11110
   *" ASSIGNING IT THE SAME LOWER AND UPPER LIMITS. THIS ENABLES ANY"/   11120
   *" SUM-OF-SQUARES CONTOUR TO BE EVALUATED (REQUEST NO.19) BY "/       11130
   *" FIXING ALL EXECEPT 2 PARAMETERS."/                                 11140
   *" A LATTICE SEARCH FOR OPTIMAL INITIAL PARAMETER ESTIMATES CAN"/     11150
```

D02

```
   *" BE MADE BEFORE THE START OF THE FITTING PROCEDURE BY ENTERING"/    11160
   *" 1 AS THE MAX. NO. OF SUM-OF-SQUARES FUNCTION EVALUATIONS"/         11170
   *" ALLOWED IN REQUEST NO.5. "/                                        11180
   *" THE DEFAULT VALUE SPECIFIED IN SOME REQUESTS CAN BE ENTERED IF"/   11190
   *" THERE IS DOUBT ABOUT WHICH VALUE TO ENTER."/                       11200
   *" FOR MORE DETAILED INFORMATION ABOUT THE PROGRAM CONTACT"/          11210
   *" P.VENG PEDERSEN. //)                                              11220
520 FORMAT(" (2) ENTER A HEADING BUT NOT MORE THAN ONE LINE"// )         11230
525 FORMAT(80H                                                           11240
   *                              )                                     11250
530 FORMAT(" (3) DO YOU WANT STATISTICAL EVALUATIONS OF THE",            11260
   *" PARAMETERS (DEFAULT=YES)"// )                                      11270
535 FORMAT(" ENTER THE EXPANSION CRITERION TO AVOID ROUNDING-OFF",       11280
   *" ERRORS"/" IN THE STATISTICAL EVALUATION OF THE PARAMETERS",        11290
   *" (DEFAULT=1E-6)" // )                                               11300
540 FORMAT(" (4) DO YOU WANT AN ANALYSIS OF THE RESIDUALS",              11310
   *" (DEFAULT=NO)" // )                                                11320
541 FORMAT(" DO YOU WANT PLOTS OF THE RESIDUALS"//  )                    11330
545 FORMAT(" (5) HOW MANY SUM-OF-SQUARES FUNCTION EVALUATIONS WILL",     11340
   *" YOU ALLOW"/" TO REACH CONVERGENCE ON MINIMUM (DEFAULT=800)"// )     11350
550 FORMAT(" (6) THE CHECK FOR FINAL CONVERGENCE IS DONE EVERY NLOOP"/   11360
   *" TIMES THAT THE PARAMETERS ARE CHANGED BY THE ITERATIVE PROCESS"/   11370
   *" ENTER THE VALUE YOU WANT FOR NLOOP (DEFAULT=20)"//)                11380
555 FORMAT(" (7) ENTER THE DEGREE OF OUTPUT YOU WANT FROM"/              11390
   *" THE FUNCTION MINIMIZATION PROCESS (DEFAULT=-1)"//                  11400
   *1X," NUMBER= FULL PROGRESS REPORT EVERY *NUMBER* FUNCTION",          11410
   *" EVALUATIONS"/                                                     11420
   *5X,"  0= PARTIAL REPORT"/                                           11430
   *5X," -1= NO OUTPUT" //)                                             11440
560 FORMAT("*(8) HOW MANY FUNCTIONS OR RESPONSE SYSTEMS ARE TO BE",      11450
   *" FITTED"/"     SIMULTANEOUSLY BY LEAST SQUARES (DEFAULT=1)"// )     11460
561 FORMAT(" ENTER",I2,"  LINES CONTAINING THESE TWO NUMBERS IN THE",    11470
   *" FOLLOWING ORDER"//5X," (I) THE NUMBER OF THE FUNCTION TO BE",      11480
   *" FITTED (I.E. 1,2,3...)"/5X," (II) THE NUMBER OF OBSERVATIONS",     11490
   *" FOR THE FUNCTION"// )                                              11500
562 FORMAT(" HOW MANY OBSERVATIONS DO YOU HAVE"// )                      11510
565 FORMAT("*(9) HOW MANY PARAMETERS (I.E. PARAMETERS+CONSTANTS)",       11520
   *" DOES THE FUNCTION(S)"/"     HAVE.  (A CONSTANT IS A PARAMETER",    11530
   *" WITH LOWER LIMIT = UPPER LIMIT)"//  )                             11540
570 FORMAT("*(10) ENTER",I2," LINES EACH CONTAINING THESE 3 NUMBERS",    11550
   *" IN THE FOLLOWING ORDER"//                                         11560
   *,5X," (I) THE INDEX OF THE PARAMETER (I.E. 1,2,3..ETC)"/            11570
   *,5X," (II) ITS LOWER LIMIT"/5X," (III) ITS UPPER LIMIT"//)          11580
571 FORMAT(/" YOU FORGOT TO ENTER THE LAST PARAMETER (NO.",I2,           11590
   *") -TRY AGAIN-"/ )                                                  11600
575 FORMAT(" (11) DO YOU WANT TO GIVE AN INITIAL ESTIMATE OF ONE OR"/    11610
   *" MORE OF YOUR PARAMETERS (DEFAULT=NO)"//)                          11620
580 FORMAT(" FOR HOW MANY PARAMETERS"//)                                 11630
585 FORMAT(" ENTER O N E LINE CONTAINING THE INDEX OF THE PARAMETER"/    11640
   *" FOLLOWED BY ITS VALUE."//)                                        11650
590 FORMAT(" ENTER FOR EACH OF THESE",I2," PARAMETERS ONE"/              11660
   *" LINE CONTAINING ITS INDEX FOLLOWED BY ITS VALUE."//)              11670
595 FORMAT(" (12) DO YOU WANT TO GIVE A STARTING STEP SIZE FOR ONE"/     11680
   *" OR MORE OF YOUR PARAMETERS (DEFAULT=NO)"//)                       11690
600 FORMAT(" FOR HOW MANY PARAMETERS"//)                                 11700
605 FORMAT(" ENTER O N E LINE CONTAINING THE INDEX OF THE PARAMETER "/   11710
   *" FOLLOWED BY ITS STEP SIZE."//)                                    11720
610 FORMAT(" ENTER FOR EACH OF THESE",I2," DIFFERENT PARAMETERS ONE"/    11730
   *" LINE CONTAINING ITS INDEX FOLLOWED BY ITS STEP SIZE."//)          11740
615 FORMAT(" (13) ENTER THE CONVERGENCE CRITERION FOR THE SUM OF",       11750
   *" SQUARES"/"     MINIMIZATION PROCESS  (DEFAULT=0.001)" // )         11760
620 FORMAT(" (14) HOW MANY INDEPENDENT VARIABLES DOES YOUR FUNCTION",    11770
```

E02

```
   *" HAVE (DEFAULT=1)" // )                                             11780
625 FORMAT(" (15) DO YOU WANT A PLOT OF THE FITTED CURVE (DEFAULT=NO)"   11790
   *//  )                                                               11800
635 FORMAT(" (16) DO YOU WANT TO WEIGHT YOUR DATA (DEFAULT=NO)" //)      11810
636 FORMAT( " ENTER ONE OF FOLLOWING 4 DIGITS ACCORDING TO THE",         11820
   *" WEIGHTING YOU PREFER "/" (Y= DEPENDENT VARIABLE. YOU CAN",         11830
   *" SPECIFY R IN NEXT INPUT REQUEST)"//                                11840
   *"   1     THE WEIGHT TO BE SPECIFIED FOR EACH OBSERVATION"          11850
   */"   2     WEIGHT = Y**R        ( R FOR EXAMPLE = -2,1,1/2 ETC. )"/ 11860
   *"   3     WEIGHT = LOG( R*Y )                                  "/   11870
   *"   4     WEIGHT = EXP( R*Y )                                  "/)  11880
637 FORMAT(" ENTER THE VALUE FOR R IN THE PREFERRED WEIGHTING SCHEME",   11890
   *" NO.",I2," ABOVE" //  )                                           11900
640 FORMAT("*(17) ENTER A LINE FOR EACH OF YOUR",I3," OBSERVATIONS."/    11910
   *" EACH LINE MUST CONTAIN",I2," NUMBERS IN THE FOLLOWING ORDER"       11920
   *//10X," (I) THE OBSERVATION NUMBER"/                                 11930
```

```
      IF(IJK.EQ.3) GO TO 718                                                20210
      IF(IPRINT.GE.0) WRITE(IW,714)(VC(J),J=I1,I2)                          20220
      IF(IJK.NE.1) GO TO 713                                                20230
      DO 500 J=I1,I2                                                        20240
  500 VARCOV(J)=VC(J)                                                       20250
      GO TO 713                                                             20260
  718 IF(IPRINT.GE.0) WRITE(IW,714)(BMAT(J),J=I1,I2)                        20270
  713 CONTINUE                                                              20280
  714 FORMAT(1X,6D13.6)                                                     20290
      WRITE(IW,715)                                                         20300
  715 FORMAT(1H ,/)                                                         20310
      L=L+6                                                                 20320
      GO TO 716                                                             20330
      END                                                                   20340
C     ******************************************************************* 20350
      SUBROUTINE CHOL  (A,N,U,NULLTY,IFAULT)                                20360
      DIMENSION A(55),U(55)                                                 20370
      DOUBLE PRECISION A,U,W,ETA                                            20380
      ETA=1D-16                                                             20390
      NN=0                                                                  20400
      IFAULT=1                                                              20410
      IF (N.LE.0) GO TO 100                                                 20420
      IFAULT=2                                                              20430
      NULLTY=0                                                              20440
      J=1                                                                   20450
```

D03

```
      K=0                                                                   20460
      DO 10 ICOL = 1,N                                                      20470
      L=0                                                                   20480
      DO 11 IROW = 1,ICOL                                                   20490
      K=K+1                                                                 20500
      W=A(K)                                                                20510
      M=J                                                                   20520
      DO 12 I = 1,IROW                                                      20530
      L=L+1                                                                 20540
      IF (I.EQ.IROW) GO TO 13                                               20550
      W=W-U(L)*U(M)                                                         20560
      M=M+1                                                                 20570
   12 CONTINUE                                                              20580
   13 IF (IROW.EQ.ICOL) GO TO 14                                            20590
      IF(U(L).EQ.0D0) GO TO 21                                              20600
      U(K)=W/U(L)                                                           20610
      GO TO 11                                                              20620
   21 U(K)=0D0                                                              20630
   11 CONTINUE                                                              20640
   14 IF(W.LE.DABS(ETA*A(K)))GO TO 20                                       20650
      U(K)=DSQRT(W)                                                         20660
      GO TO 15                                                              20670
   20 U(K)=0D0                                                              20680
      IF(W.LT.0D0)NN=NN+1                                                   20690
      NULLTY=NULLTY+1                                                       20700
   15 J=J+ICOL                                                              20710
   10 CONTINUE                                                              20720
      IFAULT=0                                                              20730
      IF(N.LE.4.AND.NN.GT.1) IFAULT=2                                       20740
      IF(N.LE.6.AND.NN.GT.2) IFAULT=2                                       20750
      IF(NN.GT.3.OR.N.EQ.NN) IFAULT=2                                       20760
  100 RETURN                                                                20770
      END                                                                   20780
C     ******************************************************************* 20790
      SUBROUTINE SYMINV (A,N,C,W,NULLTY,IFAULT)                             20800
      DIMENSION A(55),C(55),W(20)                                           20810
      DOUBLE PRECISION A,C,W,X                                              20820
      NROW=N                                                                20830
      IFAULT=1                                                              20840
      IF (NROW.LE.0) GO TO 100                                              20850
      IFAULT=0                                                              20860
      CALL CHOL(A,NROW,C,NULLTY,IFAULT)                                     20870
      IF (IFAULT.NE.0) GO TO 100                                            20880
      NN=(NROW*(NROW+1))/2                                                  20890
      IROW=NROW                                                             20900
      NDIAG=NN                                                              20910
   16 IF(C(NDIAG).EQ.0D0) GO TO 11                                          20920
      L=NDIAG                                                               20930
      DO 10 I=IROW,NROW                                                     20940
      W(I)=C(L)                                                             20950
      L=L+I                                                                 20960
   10 CONTINUE                                                              20970
      ICOL=NROW                                                             20980
      JCOL=NN                                                               20990
      MDIAG=NN                                                              21000
   15 L=JCOL                                                                21010
      X=0D0                                                                 21020
      IF(ICOL.EQ.IROW)X=1D0/W(IROW)                                         21030
      K=NROW                                                                21040
   13 IF (K.EQ.IROW) GO TO 12                                               21050
      X=X-W(K)*C(L)                                                         21060
      K=K-1                                                                 21070
```

E03

```
      L=L-1                                                                 21080
      IF (L.GT.MDIAG) L=L-K+1                                               21090
      GO TO 13                                                              21100
   12 C(L)=X/W(IROW)                                                        21110
      IF (ICOL.EQ.IROW) GO TO 14                                            21120
      MDIAG=MDIAG-ICOL                                                      21130
      ICOL=ICOL-1                                                           21140
      JCOL=JCOL-1                                                           21150
      GO TO 15                                                              21160
   11 L=NDIAG                                                               21170
      DO 17 J=IROW,NROW                                                     21180
      C(L)=0D0                                                              21190
      L=L+J                                                                 21200
   17 CONTINUE                                                              21210
   14 NDIAG=NDIAG-IROW                                                      21220
      IROW=IROW-1                                                           21230
```

```
      GO TO 58                                                          2240
   57 NAGR=9                                                            2250
      GO TO 1008                                                        2260
   58 IF(N11.GT.1) GO TO 59                                             2270
      WRITE(6,585)                                                      2280
      GO TO 60                                                          2290
   59 WRITE(6,590)N11                                                   2300
   60 CONTINUE                                                          2310
      DO 65 I=1,N11                                                     2320
      CALL READ(2,AR)                                                   2330
      IF(IGO.EQ.2) GO TO 1001                                           2340
      IF(ANS.EQ.4HBACK) GO TO 1004                                      2350
      J=IFIX(ANS+.1)                                                    2360
      IF(J.LT.1) GO TO 1007                                             2370
      IF(J.GT.NOP) GO TO 61                                             2380
      GO TO 62                                                          2390
   61 NAGR=9                                                            2400
      GO TO 1008                                                        2410
   62 IF(AR(2).LT.F1(J).OR.AR(2).GT.F2(J)) GO TO 63                     2420
      GO TO 64                                                          2430
   63 NAGR=10                                                           2440
      GO TO 1008                                                        2450
   64 NF(I)=J                                                           2460
   65 FS(J)=AR(2)                                                       2470
```

E01

```
   66 IF(NS.GT.NQ) GO TO 1006                                           2480
   70 NQ=12                                                             2490
      WRITE(6,595)                                                      2500
      CALL READ(1,AR)                                                   2510
      GO TO (1000,1001) IGO                                             2520
   71 ANS12=ANS                                                         2530
      IF(ANS.EQ.2HNO) GO TO 81                                          2540
      WRITE(6,600)                                                      2550
      CALL READ(1,AR)                                                   2560
      IF(IGO.EQ.2) GO TO 1001                                           2570
      IF(ANS.EQ.4HBACK) GO TO 1004                                      2580
      N12=IFIX(ANS+.1)                                                  2590
      IF(N12.LT.1) GO TO 1007                                           2600
      IF(N12.GT.NOP) GO TO 72                                           2610
      GO TO 73                                                          2620
   72 NAGR=9                                                            2630
      GO TO 1008                                                        2640
   73 IF(N12.GT.1) GO TO 74                                             2650
      WRITE(6,605)                                                      2660
      GO TO 75                                                          2670
   74 WRITE(6,610)N12                                                   2680
   75 CONTINUE                                                          2690
      DO 80 I=1,N12                                                     2700
      CALL READ(2,AR)                                                   2710
      IF(IGO.EQ.2) GO TO 1001                                           2720
      IF(ANS.EQ.4HBACK) GO TO 1004                                      2730
      J=IFIX(ANS+.1)                                                    2740
      IF(J.LT.1) GO TO 1007                                             2750
      IF(J.GT.NOP) GO TO 76                                             2760
      GO TO 77                                                          2770
   76 NAGR=9                                                            2780
      GO TO 1008                                                        2790
   77 IF(ABS(AR(2)).GE.(F2(J)-F1(J))) GO TO 78                          2800
      GO TO 79                                                          2810
   78 NAGR=10                                                           2820
      GO TO 1008                                                        2830
   79 NT(I)=J                                                           2840
   80 STPS(J)=AR(2)                                                     2850
   81 IF(NS.GT.NQ) GO TO 1006                                           2860
   85 NQ=13                                                             2870
      WRITE(6,615)                                                      2880
      CALL READ(1,AR)                                                   2890
      GO TO (1003,1001)IGO                                              2900
   86 IF(ANS.LT.0..OR.ANS.GT.1.) GO TO 1007                             2910
      STOPCR=ANS                                                        2920
      IF(NS.GT.NQ) GO TO 1006                                           2930
   90 NQ=14                                                             2940
      WRITE(6,620)                                                      2950
   92 CALL READ(1,AR)                                                   2960
      GO TO (1003,1001)IGO                                              2970
   91 NIND=IFIX(ANS+.1)                                                 2980
      IF(NIND.LT.1.OR.NIND.GT.9) GO TO 93                               2990
      GO TO 94                                                          3000
   93 WRITE(6,1024)                                                     3010
      GO TO 92                                                          3020
   94 IF(NS.GT.NQ) GO TO 1006                                           3030
   95 NQ=15                                                             3040
      IF(NIND.GT.1) GO TO 97                                            3050
      WRITE(6,625)                                                      3060
      CALL READ(1,AR)                                                   3070
      GO TO (1000,1001) IGO                                             3080
   96 PLT=ANS                                                           3090
```

F01

```
   97 IF(NS.GT.NQ) GO TO 1006                                           3100
  100 NQ=16                                                             3110
      WRITE(6,635)                                                      3120
      CALL READ(1,AR)                                                   3130
      GO TO (1000,1001) IGO                                             3140
  101 ANS16=ANS                                                         3150
      IWGHT=0                                                           3160
      IF(ANS16.EQ.2HNO) GO TO 102                                       3170
      WRITE(6,636)                                                      3180
      CALL READ(1,AR)                                                   3190
      IF(IGO.EQ.2) GO TO 1001                                           3200
      IF(ANS.EQ.4HBACK) GO TO 1004                                      3210
      IWGHT=IFIX(ANS+.1)                                                3220
      IF(IWGHT.LT.1.OR.IWGHT.GT.4) GO TO 1007                           3230
      IF(IWGHT.EQ.1) GO TO 102                                          3240
      WRITE(6,637)IWGHT                                                 3250
      CALL READ(1,AR)                                                   3260
```

```
     *" WITH LOWER LIMIT = UPPER LIMIT)"//  )                              11540
  570 FORMAT("*(10) ENTER",I2," LINES EACH CONTAINING THESE 3 NUMBERS",     11550
     *" IN THE FOLLOWING ORDER"//                                          11560
     *,5X," (I) THE INDEX OF THE PARAMETER (I.E. 1,2,3..ETC)"/             11570
     *,5X," (II) ITS LOWER LIMIT"/5X," (III) ITS UPPER LIMIT"//)           11580
  571 FORMAT(/" YOU FORGOT TO ENTER THE LAST PARAMETER (NO.",I2,            11590
     *") -TRY AGAIN-"/ )                                                   11600
  575 FORMAT(" (11) DO YOU WANT TO GIVE AN INITIAL ESTIMATE OF ONE OR"/     11610
     *" MORE OF YOUR PARAMETERS (DEFAULT=NO)"//)                           11620
  580 FORMAT(" FOR HOW MANY PARAMETERS"//)                                  11630
  585 FORMAT(" ENTER O N E LINE CONTAINING THE INDEX OF THE PARAMETER"/     11640
     *" FOLLOWED BY ITS VALUE."//)                                         11650
  590 FORMAT(" ENTER FOR EACH OF THESE",I2," PARAMETERS ONE"/               11660
     *" LINE CONTAINING ITS INDEX FOLLOWED BY ITS VALUE."//)               11670
  595 FORMAT(" (12) DO YOU WANT TO GIVE A STARTING STEP SIZE FOR ONE"/      11680
     *" OR MORE OF YOUR PARAMETERS (DEFAULT=NO)"//)                        11690
  600 FORMAT(" FOR HOW MANY PARAMETERS"//)                                  11700
  605 FORMAT(" ENTER O N E LINE CONTAINING THE INDEX OF THE PARAMETER "/    11710
     *" FOLLOWED BY ITS STEP SIZE."//)                                     11720
  610 FORMAT(" ENTER FOR EACH OF THESE",I2," DIFFERENT PARAMETERS ONE"/     11730
     *" LINE CONTAINING ITS INDEX FOLLOWED BY ITS STEP SIZE."//)           11740
  615 FORMAT(" (13) ENTER THE CONVERGENCE CRITERION FOR THE SUM OF",        11750
     *" SQUARES"/"      MINIMIZATION PROCESS  (DEFAULT=0.001)" // )         11760
  620 FORMAT(" (14) HOW MANY INDEPENDENT VARIABLES DOES YOUR FUNCTION",     11770
```

E02

```
     *" HAVE (DEFAULT=1)" // )                                             11780
  625 FORMAT(" (15) DO YOU WANT A PLOT OF THE FITTED CURVE (DEFAULT=NO)"    11790
     *//  )                                                               11800
  635 FORMAT(" (16) DO YOU WANT TO WEIGHT YOUR DATA (DEFAULT=NO)" //)       11810
  636 FORMAT( " ENTER ONE OF FOLLOWING 4 DIGITS ACCORDING TO THE",          11820
     *" WEIGHTING YOU PREFER "/" (Y= DEPENDENT VARIABLE. YOU CAN",         11830
     *" SPECIFY R IN NEXT INPUT REQUEST)"//                                11840
     *"   1     THE WEIGHT TO BE SPECIFIED FOR EACH OBSERVATION"           11850
     */"   2     WEIGHT = Y**R       ( R FOR EXAMPLE = -2,1,1/2 ETC. )"/  11860
     *"   3     WEIGHT = LOG( R*Y )                                  "/    11870
     *"   4     WEIGHT = EXP( R*Y )                                  "/)   11880
  637 FORMAT(" ENTER THE VALUE FOR R IN THE PREFERRED WEIGHTING SCHEME",    11890
     *" NO.",I2," ABOVE" // )                                              11900
  640 FORMAT("*(17) ENTER A LINE FOR EACH OF YOUR",I3," OBSERVATIONS."/     11910
     *" EACH LINE MUST CONTAIN",I2," NUMBERS IN THE FOLLOWING ORDER"       11920
     *//10X," (I) THE OBSERVATION NUMBER"/                                 11930
     *10X," (II) THE VALUE OF THE DEPENDENT VARIABLE (RESPONSE)"/          11940
     *10X," (III) THE WEIGHT OF THE OBSERVATION"/                          11950
     *10X," (IV) THE VALUE(S) OF THE",I2," INDEPENDENT VARIABLE(S)"// )    11960
  641 FORMAT(/" ENTER NOW THE",I3," OBSERVATIONS FOR RESPONSE SYSTEM",      11970
     *" NO.",I2/" REMEMBER THE FIRST OBSERVATION FOR THIS RESPONSE",       11980
     *" SYSTEM MUST BE NUMBERED",I3/" IN CONTINUATION OF THE PREVIOUS",    11990
     *" OBSERVATION NUMBER."//  )                                          12000
  642 FORMAT(/" YOU FORGOT TO ENTER THE LAST OBSERVATION (NO.",I3,")"/      12010
     *" REENTER YOUR DATA STRICTLY ACCORDING TO THE FOLLOWING REQUEST"/)    12020
  655 FORMAT("*(17) ENTER A LINE FOR EACH OF YOUR",I3," OBSERVATIONS."/     12030
     *" EACH LINE MUST CONTAIN",I2," NUMBERS IN THE FOLLOWING ORDER"       12040
     *//10X," (I) THE OBSERVATION NUMBER"/                                 12050
     *10X," (II) THE VALUE OF THE DEPENDENT VARIABLE (RESPONSE)"/          12060
     *10X," (III) THE VALUE(S) OF THE",I2," INDEPENDENT VARIABLE(S)"// )   12070
  660 FORMAT(" (18) ENTER-BACK-FOR CORRECTION OR-OK-FOR A FINAL",           12080
     *" CHECK ON ALL INPUTS"// )                                           12090
  662 FORMAT( " YOUR INPUTS HAVE BEEN CHECKED. DO YOU WANT A SUMMARY",      12100
     *" OF THEM"//  )                                                      12110
  663 FORMAT(" DO YOU WANT A PLOT OF YOUR DATA POINTS INCLUDED IN",         12120
     *" THE SUMMARY"// )                                                   12130
  664 FORMAT( " ENTER -BACK- FOR CORRECTION OR -OK- FOR ACTIVATION OF",     12140
     *" THE COMPUTATIONS" //)                                              12150
  670 FORMAT(" WAIT NOW FOR THE RESULTS"//)                                 12160
 6700 FORMAT(//" (19) DO YOU WANT A ",I1,"-DIMENSIONAL SEARCH" // )         12170
 6705 FORMAT(" HOW MANY FUNCTION EVALUATIONS DO YOU ALLOW IN THE",          12180
     *" SEARCH (DEFAULT=",I4,")" //  )                                    12190
 6710 FORMAT(//20(1H*),I2,"-D I M E N S I O N A L  S E A R C H ",          12200
     *20(1H*)  )                                                          12210
 6715 FORMAT( // I5," LATTICE POINTS HAVE BEEN EVALUATED (",               12220
     *I2," POINTS PER PARAMETER )"  )                                     12230
 6720 FORMAT(//  " THE SMALLEST WEIGHTED RESIDUAL SUM OF SQUARES VALUE"/   12240
     *" FOUND IN SEARCH =",E10.4," AT"/ )                                 12250
 6725 FORMAT(//  " THE SMALLEST RESIDUAL SUM OF SQUARES VALUE",           12260
     *" FOUND IN THE SEARCH =",E10.4," AT" / )                            12270
 6730 FORMAT( 5X," PAR.NO. ",I1," =",E11.4,5X,40A1  )                      12280
 6731 FORMAT(/" WHICH IS SMALLER THAN THE VALUE (",E10.4,") OBTAINED",     12290
     *" USING THE FOLLOWING"/" INITIAL PARAMETER ESTIMATES"/  )            12300
 6735 FORMAT(/  " WHICH IS SMALLER THAN THE VALUE (",E10.4,") FOUND",      12310
     *" BY THE MINIMIZATION"/" PROCESS.THE MINIMIZATION IS THEREFORE",     12320
     *" RESTARTED IN NEW MINIMUM REGION"/" USING THE PARAMETERS ABOVE"/)   12330
 6740 FORMAT(/ " WHICH IS LARGER THAN THE VALUE (",E10.4,") FOUND BY",     12340
     *" THE MINIMIZATION PROCESS"/" AT"  )                                12350
 6741 FORMAT(/" WHICH IS LARGER THAN THE VALUE (",E10.4,") OBTAINED",      12360
     *" USING THE FOLLOWING"/" INITIAL PARAMETER ESTIMATES"/  )            12370
 6745 FORMAT(// 77(1H*) )                                                  12380
 6747 FORMAT(/ " DO YOU WANT TO MAKE ANY ADDITIONAL CORRECTIONS BEFORE",    12390
```

F02

```
     *" THE RESTART"// )                                                   12400
 6748 FORMAT(//" DO YOU WANT TO START THE MINIMIZATION PROCESS"/           12410
     *" USING THE PARAMETER VALUES FOUND IN THE SEARCH ABOVE" // )         12420
  675 FORMAT(//" (20) DO YOU WANT TO REPEAT THE COMPUTATIONS ABOVE BUT"/   12430
     *" WITH SOME CHANGE IN THE INPUTS"//)                                 12440
  680 FORMAT(" DO YOU WANT ANOTHER RUN WITH A NEW SET OF OBSERVATIONS"//    12450
     *)                                                                    12460
  685 FORMAT(" I HOPE YOU HAVE BEEN PLEASED WITH THE PROGRAM."/            12470
     *" IF YOU HAVE ANY SUGGESTIONS FOR ALTERATIONS PLEASE CONTACT ME"//    12480
     *"                                        REGARDS"//                 12490
     *"                                    P.VENG PEDERSEN"//)            12500
  690 FORMAT(" (21) DO YOU WANT TO USE THE FINAL COMPUTED PARAMETERS"/      12510
     *" FROM THE RUN ABOVE AS STARTING PARAMETERS FOR THE NEXT RUN."//)     12520
  692 FORMAT(" YOU CAN NOW MAKE ALTERATIONS IN PREVIOUS INPUTS ",           12530
     *"FOR THE NEXT RUN."  )                                               12540
  694 FORMAT(" ( 22) DO YOU WANT TO MAKE MORE ALTERATIONS"//)               12550
```

```
      IFAULT=1                                                          20840
      IF (NROW.LE.0) GO TO 100                                          20850
      IFAULT=0                                                          20860
      CALL CHOL(A,NROW,C,NULLTY,IFAULT)                                 20870
      IF (IFAULT.NE.0) GO TO 100                                        20880
      NN=(NROW*(NROW+1))/2                                              20890
      IROW=NROW                                                         20900
      NDIAG=NN                                                          20910
   16 IF(C(NDIAG).EQ.0D0) GO TO 11                                      20920
      L=NDIAG                                                           20930
      DO 10 I=IROW,NROW                                                 20940
      W(I)=C(L)                                                         20950
      L=L+I                                                             20960
   10 CONTINUE                                                          20970
      ICOL=NROW                                                         20980
      JCOL=NN                                                           20990
      MDIAG=NN                                                          21000
   15 L=JCOL                                                            21010
      X=0D0                                                             21020
      IF(ICOL.EQ.IROW)X=1D0/W(IROW)                                     21030
      K=NROW                                                            21040
   13 IF (K.EQ.IROW) GO TO 12                                           21050
      X=X-W(K)*C(L)                                                     21060
      K=K-1                                                             21070
```

E03

```
      L=L-1                                                             21080
      IF (L.GT.MDIAG) L=L-K+1                                           21090
      GO TO 13                                                          21100
   12 C(L)=X/W(IROW)                                                    21110
      IF (ICOL.EQ.IROW) GO TO 14                                        21120
      MDIAG=MDIAG-ICOL                                                  21130
      ICOL=ICOL-1                                                       21140
      JCOL=JCOL-1                                                       21150
      GO TO 15                                                          21160
   11 L=NDIAG                                                           21170
      DO 17 J=IROW,NROW                                                 21180
      C(L)=0D0                                                          21190
      L=L+J                                                             21200
   17 CONTINUE                                                          21210
   14 NDIAG=NDIAG-IROW                                                  21220
      IROW=IROW-1                                                       21230
      IF (IROW.NE.0) GO TO 16                                           21240
  100 RETURN                                                            21250
      END                                                               21260
C     ******************************************************************21270
      SUBROUTINE LSQ(F,FUNC)                                            21280
      DIMENSION F(20),F1(20),F2(20),Y(100),WYN(100),XX(9,100),A(9),     21290
     *LABEL(100)                                                        21300
      DOUBLE PRECISION FUNC,YOBS,YCAL,W                                 21310
      COMMON /DATA/ XX,Y,WYN,NOBS                                       21320
      COMMON /PARLIM/ F1,F2                                             21330
      COMMON /CONSTR/LIMITS                                             21340
      COMMON /FUNNUM/ITHFUN                                             21350
      COMMON /B4/ NIND                                                  21360
      COMMON /B6/ NOP                                                   21370
      COMMON /B10/LABEL                                                 21380
      COMMON /B14/NVIOL                                                 21390
      S=0.                                                              21400
      FUNC=0D0                                                          21410
      IF(LIMITS.EQ.0) GO TO 6                                           21420
      DO 5 I=1,NOP                                                      21430
      IF(F(I).GE.F1(I).AND.F(I).LE.F2(I).OR.F1(I).EQ.F2(I)) GO TO 5     21440
      B=ABS(F1(I)-F(I))                                                 21450
      B2=ABS(F2(I)-F(I))                                                21460
      B=1.E50*(1.+AMIN1(B,B2)/AMAX1(B,B2))                              21470
      IF(B.GT.S)S=B                                                     21480
    5 CONTINUE                                                          21490
      IF(S.GE.1E50) GO TO 20                                            21500
    6 CONTINUE                                                          21510
      DO 15 I=1,NOBS                                                    21520
      DO 10 J=1,NIND                                                    21530
   10 A(J)=XX(J,I)                                                      21540
      ITHFUN=LABEL(I)                                                   21550
      S=Y(I)                                                            21560
      IF(NIND.EQ.1) CALL MODEL(S,A(1),F,0)                              21570
      IF(NIND.GT.1) CALL MODEL(S,A,F,0)                                 21580
      IF(S.GE.1E50) GO TO 20                                            21590
      YCAL=S                                                            21600
      YOBS=Y(I)                                                         21610
      W=WYN(I)                                                          21620
   15 FUNC=FUNC+W*(YOBS-YCAL)*(YOBS-YCAL)                               21630
      RETURN                                                            21640
   20 FUNC=S*1.000000001                                                21650
      NVIOL=NVIOL+1                                                     21660
      RETURN                                                            21670
      END                                                               21680
C     ******************************************************************21690
```

F03

```
      SUBROUTINE READ(N,AR)                                             21700
      DIMENSION A(78),AR(12),AL(10)                                     21710
      COMMON/B5/NQ,NS,IGO                                               21720
      COMMON /B15/SKIP                                                  21730
      LOGICAL SKIP                                                      21740
      DATA YES,ANO,BACK,AO,AK,BLANK,PLUS,AMINUS,COMMA,DOT,E             21750
     */1HY,1HN,1HB,1HO,1HK,1H ,1H+,1H-,1H,,1H.,1HE/                     21760
      DATA AL/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9 /                 21770
    5 IGO=1                                                             21780
      SKIP=.FALSE.                                                      21790
      DO 6 L=1,78                                                       21800
    6 A(L)=BLANK                                                        21810
      DO 7 L=1,12                                                       21820
    7 AR(L)=0.0                                                         21830
      READ(5,200)A                                                      21840
      IF(EOF(5))8,9,8                                                   21850
    8 WRITE(6,225)                                                      21860
```

```
 90 NQ=14                                                              2940
    WRITE(6,620)                                                       2950
 92 CALL READ(1,AR)                                                    2960
    GO TO (1003,1001)IGO                                               2970
 91 NIND=IFIX(ANS+.1)                                                  2980
    IF(NIND.LT.1.OR.NIND.GT.9) GO TO 93                                2990
    GO TO 94                                                           3000
 93 WRITE(6,1024)                                                      3010
    GO TO 92                                                           3020
 94 IF(NS.GT.NQ) GO TO 1006                                            3030
 95 NQ=15                                                              3040
    IF(NIND.GT.1) GO TO 97                                             3050
    WRITE(6,625)                                                       3060
    CALL READ(1,AR)                                                    3070
    GO TO (1000,1001) IGO                                              3080
 96 PLT=ANS                                                            3090
```

F01

```
 97 IF(NS.GT.NQ) GO TO 1006                                            3100
100 NQ=16                                                              3110
    WRITE(6,635)                                                       3120
    CALL READ(1,AR)                                                    3130
    GO TO (1000,1001) IGO                                              3140
101 ANS16=ANS                                                          3150
    IWGHT=0                                                            3160
    IF(ANS16.EQ.2HNO) GO TO 102                                        3170
    WRITE(6,636)                                                       3180
    CALL READ(1,AR)                                                    3190
    IF(IGO.EQ.2) GO TO 1001                                            3200
    IF(ANS.EQ.4HBACK) GO TO 1004                                       3210
    IWGHT=IFIX(ANS+.1)                                                 3220
    IF(IWGHT.LT.1.OR.IWGHT.GT.4) GO TO 1007                            3230
    IF(IWGHT.EQ.1) GO TO 102                                           3240
    WRITE(6,637)IWGHT                                                  3250
    CALL READ(1,AR)                                                    3260
    IF(IGO.EQ.2) GO TO 1001                                            3270
    IF(ANS.EQ.4HBACK) GO TO 1004                                       3280
    ANS16B=ANS                                                         3290
102 IF(NS.GT.NQ) GO TO 1006                                            3300
105 NQ=17                                                              3310
    NSAVE=1                                                            3320
    J=0                                                                3330
    NOBS2=NOBS+5                                                       3340
    IF(ANS16.EQ.2HNO.OR.IWGHT.GT.1) GO TO 120                          3350
106 NN=NIND+3                                                          3360
    WRITE(6,640)NOBS,NN,NIND                                           3370
    IW=1                                                               3380
    DO 109 I=1,NOBS2                                                   3390
    J=J+1                                                              3400
    N=LABEL(J)                                                         3410
    IF(N.EQ.NSAVE+1) WRITE(6,641)ITHSET(N),N,J                         3420
    NSAVE=N                                                            3430
1060 CALL READ(NN,AR)                                                  3440
    IF(IGO.EQ.2) GO TO 1001                                            3450
    IF(ANS.EQ.4HBACK) GO TO 1004                                       3460
    IF(AR(3).LT.0.) WRITE(6,1033)                                      3470
    IF(AR(3).LT.0.) GO TO 1060                                         3480
    J=IFIX(ANS+.1)                                                     3490
    IF(J.LT.1) GO TO 1007                                              3500
    IF(J.GT.NOBS) GO TO 107                                            3510
    GO TO 108                                                          3520
107 NAGR=8                                                             3530
    GO TO 1008                                                         3540
108 CONTINUE                                                           3550
    DO 1080 L=1,NIND                                                   3560
1080 XX(L,J)=AR(L+3)                                                   3570
    INDEX(J)=J                                                         3580
    Y(J)=AR(2)                                                         3590
    WY(J)=AR(3)                                                        3600
    IF(J.EQ.NOBS)GO TO 125                                             3610
109 CONTINUE                                                           3620
    WRITE(6,642)NOBS                                                   3630
    GO TO 105                                                          3640
120 NN=NIND+2                                                          3650
    WRITE(6,655)NOBS,NN,NIND                                           3660
    IW=2                                                               3670
    DO 123 I=1,NOBS2                                                   3680
    J=J+1                                                              3690
    N=LABEL(J)                                                         3700
    IF(N.EQ.NSAVE+1) WRITE(6,641)ITHSET(N),N,J                         3710
```

G01

```
    NSAVE=N                                                            3720
    CALL READ(NN,AR)                                                   3730
    IF(IGO.EQ.2) GO TO 1001                                            3740
    IF(ANS.EQ.4HBACK) GO TO 1004                                       3750
    J=IFIX(ANS+.1)                                                     3760
    IF(J.LT.1) GO TO 1007                                              3770
    IF(J.GT.NOBS) GO TO 121                                            3780
    GO TO 122                                                          3790
121 NAGR=8                                                             3800
    GO TO 1008                                                         3810
122 CONTINUE                                                           3820
    DO 1220 L=1,NIND                                                   3830
1220 XX(L,J)=AR(L+2)                                                   3840
    INDEX(J)=J                                                         3850
    Y(J)=AR(2)                                                         3860
    IF(J.EQ.NOBS) GO TO 125                                            3870
123 CONTINUE                                                           3880
    WRITE(6,642)NOBS                                                   3890
    GO TO 120                                                          3900
125 CONTINUE                                                           3910
    DO 126 I=1,NOBS                                                    3920
    X(I)=XX(1,I)                                                       3930
126 IF(NIND.GT.1) Z(I)=XX(2,I)                                         3940
    IF(NS.GT.NQ) GO TO 1006                                            3950
130 NQ=18                                                              3960
```

```
6720 FORMAT(// " THE SMALLEST WEIGHTED RESIDUAL SUM OF SQUARES VALUE"/      12240
   *" FOUND IN SEARCH =",E10.4," AT"/ )                                    12250
6725 FORMAT(// " THE SMALLEST RESIDUAL SUM OF SQUARES VALUE",              12260
   *" FOUND IN THE SEARCH =",E10.4," AT" / )                               12270
6730 FORMAT( 5X," PAR.NO. ",I1," =",E11.4,5X,40A1  )                       12280
6731 FORMA1(/" WHICH IS SMALLER THAN THE VALUE (",E10.4,") OBTAINED",      12290
   *" USING THE FOLLOWING"/" INITIAL PARAMETER ESTIMATES"/  )              12300
6735 FORMAT(/ " WHICH IS SMALLER THAN THE VALUE (",E10.4,") FOUND",        12310
   *" BY THE MINIMIZATION"/" PROCESS.THE MINIMIZATION IS THEREFORE",       12320
   *" RESTARTED IN NEW MINIMUM REGION"/" USING THE PARAMETERS ABOVE"/)     12330
6740 FORMAT(/ " WHICH IS LARGER THAN THE VALUE (",E10.4,") FOUND BY",      12340
   *" THE MINIMIZATION PROCESS"/" AT"  )                                   12350
6741 FORMAT(/" WHICH IS LARGER THAN THE VALUE (",E10.4,") OBTAINED",       12360
   *" USING THE FOLLOWING"/" INITIAL PARAMETER ESTIMATES"/  )              12370
6745 FORMAT(// 77(1H*) )                                                   12380
6747 FORMAT(/ " DO YOU WANT TO MAKE ANY ADDITIONAL CORRECTIONS BEFORE",    12390
```

F02

```
   *" THE RESTART"// )                                                     12400
6748 FORMAT(//" DO YOU WANT TO START THE MINIMIZATION PROCESS"/            12410
   *" USING THE PARAMETER VALUES FOUND IN THE SEARCH ABOVE" // )            12420
 675 FORMAT(//" (20) DO YOU WANT TO REPEAT THE COMPUTATIONS ABOVE BUT"/     12430
   *" WITH SOME CHANGE IN THE INPUTS"//)                                   12440
 680 FORMAT(" DO YOU WANT ANOTHER RUN WITH A NEW SET OF OBSERVATIONS"//     12450
   *)                                                                      12460
 685 FORMAT(" I HOPE YOU HAVE BEEN PLEASED WITH THE PROGRAM."/             12470
   *" IF YOU HAVE ANY SUGGESTIONS FOR ALTERATIONS PLEASE CONTACT ME"//      12480
   *"                                          REGARDS"//                  12490
   *"                                      P.VENG PEDERSEN"//)             12500
 690 FORMAT(" (21) DO YOU WANT TO USE THE FINAL COMPUTED PARAMETERS"/       12510
   *" FROM THE RUN ABOVE AS STARTING PARAMETERS FOR THE NEXT RUN."//)       12520
 692 FORMAT(" YOU CAN NOW MAKE ALTERATIONS IN PREVIOUS INPUTS ",            12530
   *"FOR THE NEXT RUN."  )                                                 12540
 694 FORMAT(" ( 22) DO YOU WANT TO MAKE MORE ALTERATIONS"//)               12550
 750 FORMAT(5(/),1X,77(1H*)/ )                                             12560
 752 FORMAT( /1X,29(1H*)," I N P U T  D A T A ",28(1H*)///                 12570
   *" NUMBER OF VARIABLE PARAMETERS      =",I4/                            12580
   *" NUMBER OF PARAMETERS HELD CONSTANT  =",I4/                           12590
   *" NUMBER OF OBSERVATIONS              =",I4/                           12600
   *" NUMBER OF DUMMY OBSERVATIONS        =",I4/                           12610
   *" NUMBER OF RESPONSE SYSTEMS          =",I4/                           12620
   *" NUMBER OF INDEPENDENT VARIABLES     =",I4///                         12630
   *" PAR.NO.",7X," ALLOWED RANGE",9X," INITIAL ESTIMATE ",                 12640
   *" INITIAL STEP SIZE"/1X,77(1H-)  )                                     12650
 754 FORMAT(I4,4(6X,E10.4))                                                12660
 756 FORMAT(1X,77(1H-)///" OBS.NO.    X1",12X," X2",10X," RESPONSE",7X,      12670
   *" WEIGHT", 4X," NORM.WEIGHT" /1X,77(1H-) )                             12680
 758 FORMAT(I4,4X,E13.4,15X,E13.4 )                                        12690
 760 FORMAT(I4,4X,E13.4,15X,3(E13.4,1X) )                                  12700
 762 FORMAT(I4,4X,3(E13.4,1X) )                                            12710
 764 FORMAT(I4,4X,5(E13.4,1X) )                                            12720
 765 FORMAT(/9X,                                                           12730
   *" PLOT OF RESPONSE VERSUS INDEPENDENT VARIABLE X1 (ARB.UNITS)" )       12740
 766 FORMAT(//                                                             12750
   *" MAXIMUM NUMBER OF FUNCTION EVALUATIONS ALLOWED =",I4/                12760
   *" STOPPING CRITERION FOR CONVERGENCE             =",E10.4)             12770
 767 FORMAT(///4X,                                                         12780
   *" PLOT OF RESPONSE VERSUS SECOND INDEPENDENT VARIABLE X2",             12790
   *"  (ARB.UNITS)"  )                                                     12800
 768 FORMAT(   STATISTICS ON THE PARAMETERS HAVE BEEN REQUESTED"/          12810
   *" CRITERION FOR EXPANSION OF SIMPLEX BEFORE QUADRATIC FITTING =",      12820
   *E10.4)                                                                 12830
 770 FORMAT(" ANALYSIS OF RESIDUALS HAS BEEN REQUESTED")                   12840
 772 FORMAT(" RESIDUAL PLOTS HAVE BEEN REQUESTED")                         12850
 774 FORMAT(                                                               12860
   *" NO OUTPUT FROM THE FUNCTION MINIMIZATION HAS BEEN REQUESTED")        12870
 776 FORMAT(" PARTIAL PROGRESS REPORT OF THE FUNCTION MINIMIZATION",       12880
   *" HAS BEEN REQUESTED")                                                 12890
 778 FORMAT(" FULL PROGRESS REPORT OF THE FUNCTION MINIMIZATION",          12900
   *" HAS BEEN REQUESTED")                                                 12910
 780 FORMAT(" PLOT OF THE FITTED CURVE AND DATA HAS BEEN REQUESTED")       12920
 781 FORMAT( " WEIGHTING HAS BEEN DONE ACCORDING TO SCHEME NO.",I2,        12930
   *" (WITH R=",E10.4,")"  )                                               12940
 789 FORMAT(/1X,16(1H*)," SUM-OF-SQUARES FUNCTION MINIMIZATION REPORT "    12950
   *,16(1H*)//  )                                                         12960
 790 FORMAT(/1X,77(1H*)// )                                                12970
 791 FORMAT(/ " IT IS NECESSARY TO MAKE CORRECTION IN YOUR INPUT",         12980
   *" TO REACH CONVERGENCE. "  )                                           12990
 792 FORMAT(" PROBABLY BECAUSE OF POSSIBLE CONSTRAIN(S) AS SEEN",          13000
   *" FROM" )                                                              13010
```

G02

```
 793 FORMAT(/ " THE REASON CAN ALSO BE UNSUITABLE VALUE(S) OF"  )          13020
 794 FORMAT(/" TRY NEW INITIAL ESTIMATES OR RESCALE THE PARAMETERS",       13030
   *" OR CHANGE THE VALUE OF" )                                            13040
 795 FORMAT(/" OR CHANGE THE LIMITS OF ONE OR MORE OF THE PARAMETERS" )    13050
 800 FORMAT(/1X,31(1H*)," R E S U L T S ",31(1H*)//                        13060
   *1X,77(1H*)////                                                         13070
   *" PAR.NO.   ALLOWED RANGE",5X," INITIAL VALUE FINAL VALUE STD."        13080
   *,"DEV.",4X," C.V.(PCT)"/1X,77(1H-))                                    13090
8010 FORMAT(/1X,31(1H*)," R E S U L T S ",31(1H*)//                        13100
   *1X,77(1H*)////                                                         13110
   *" PAR.NO.   ALLOWED RANGE",5X," INITIAL VALUE   FINAL VALUE"/          13120
   *1X,57(1H-))                                                            13130
8011 FORMAT(I4,3X,4(E11.5,1X))                                             13140
8012 FORMAT(1X,57(1H-))                                                    13150
 802 FORMAT(I4,3X,6(E11.5,1X) )                                            13160
 803 FORMAT(1X,77(1H-)///                                                  13170
   *" PAR.NO.  APPROX. 95 PCT. CONF.LIMITS",/1X,38(1H-)  )                 13180
8030 FORMAT( / " THE MINIMUM FOUND BY THE MINIMIZATION",                   13190
   *" PROCESS APPEARS"/" TO BE CONSTRAINED BY THE ",A5,                    13200
   *" LIMIT OF PARAMETER NUMBER ",I2  )                                    13210
8031 FORMAT(/                                                              13220
   *" *** ENTER B10 OR B11 IF YOU WANT A NEW RUN WITH DIFFERENT"/           13230
   *" *** PARAMETER LIMITS OR DIFFERENT PARAMETER ESTIMATES"/               13240
   *" *** OTHERWISE TYPE -OK- TO CONTINUE"// )                              13250
 804 FORMAT(//                                                             13260
```

```
      ITHFUN=LABEL(I)                                                   21550
      S=Y(I)                                                            21560
      IF(NIND.EQ.1) CALL MODEL(S,A(1),F,0)                              21570
      IF(NIND.GT.1) CALL MODEL(S,A,F,0)                                 21580
      IF(S.GE.1E50) GO TO 20                                            21590
      YCAL=S                                                            21600
      YOBS=Y(I)                                                         21610
      W=WYN(I)                                                          21620
   15 FUNC=FUNC+W*(YOBS-YCAL)*(YOBS-YCAL)                               21630
      RETURN                                                            21640
   20 FUNC=S*1.000000001                                                21650
      NVIOL=NVIOL+1                                                     21660
      RETURN                                                            21670
      END                                                               21680
C     ****************************************************************  21690
```

F03

```
      SUBROUTINE READ(N,AR)                                             21700
      DIMENSION A(78),AR(12),AL(10)                                     21710
      COMMON/B5/NQ,NS,IGO                                               21720
      COMMON /B15/SKIP                                                  21730
      LOGICAL SKIP                                                      21740
      DATA YES,ANO,BACK,AO,AK,BLANK,PLUS,AMINUS,COMMA,DOT,E             21750
     */1HY,1HN,1HB,1HO,1HK,1H ,1H+,1H-,1H,,1H.,1HE/                     21760
      DATA AL/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9 /                 21770
    5 IGO=1                                                             21780
      SKIP=.FALSE.                                                      21790
      DO 6 L=1,78                                                       21800
    6 A(L)=BLANK                                                        21810
      DO 7 L=1,12                                                       21820
    7 AR(L)=0.0                                                         21830
      READ(5,200)A                                                      21840
      IF(EOF(5))8,9,8                                                   21850
    8 WRITE(6,225)                                                      21860
      GO TO 5                                                           21870
    9 CONTINUE                                                          21880
      DO 11 I=1,78                                                      21890
      IF(A(I).EQ.1HR) GO TO 139                                         21900
      IF(A(I).EQ.BACK) GO TO 140                                        21910
   11 CONTINUE                                                          21920
      IF(A(1).EQ.YES) GO TO 130                                         21930
      IF(A(1).EQ.ANO) GO TO 135                                         21940
      IF(A(1).EQ.AO.AND.A(2).EQ.AK) GO TO 145                           21950
      I=0                                                               21960
      J=1                                                               21970
      K=0                                                               21980
      FAC=1.                                                            21990
      GO TO 15                                                          22000
   10 FAC=-1.                                                           22010
   15 I=I+1                                                             22020
      IF(I.EQ.79) GO TO 125                                             22030
      DO 20 L=1,10                                                      22040
      IF(A(I).EQ.AL(L)) GO TO 50                                        22050
   20 CONTINUE                                                          22060
      IF(K.EQ.0) GO TO 30                                               22070
      IF(A(I).EQ.BLANK.OR.A(I).EQ.COMMA.OR.A(I).EQ.DOT.OR.A(I).EQ.E)    22080
     *GO TO 55                                                          22090
   25 WRITE(6,205)A(I)                                                  22100
      GO TO 5                                                           22110
   30 IF(A(I).EQ.BLANK) GO TO 15                                        22120
      IF(J.GT.1.AND.A(I).EQ.COMMA) GO TO 15                             22130
      IF(A(I).EQ.AMINUS) GO TO 10                                       22140
      IF(A(I).EQ.DOT) GO TO 71                                          22150
      GO TO 25                                                          22160
   50 IF(K.EQ.0) IFLAG=I                                                22170
      AR(J)=AR(J)+FLOAT(L-1)*(10.**(-K))                                22180
      K=K+1                                                             22190
      GO TO 15                                                          22200
   55 AR(J)=FAC*AR(J)*(10.**(I-IFLAG-1))                                22210
      IF(A(I).EQ.DOT.AND.A(I+1).EQ.COMMA.OR.A(I+1).EQ.BLANK) GO TO 60   22220
      IF(A(I).EQ.DOT) GO TO 70                                          22230
      IF(A(I).EQ.E) GO TO 90                                            22240
   60 J=J+1                                                             22250
      IF(J.GT.N) GO TO 65                                               22260
      FAC=1.                                                            22270
      K=0                                                               22280
      GO TO 15                                                          22290
   65 I=I+1                                                             22300
      IF(I.EQ.79) RETURN                                                22310
```

G03

```
      IF(A(I).EQ.BLANK) GO TO 65                                        22320
      WRITE(6,215)N                                                     22330
      GO TO 5                                                           22340
   70 K=0                                                               22350
   71 I=I+1                                                             22360
      K=K+1                                                             22370
      DO 75 L=1,10                                                      22380
      IF(A(I).EQ.AL(L)) GO TO 85                                        22390
   75 CONTINUE                                                          22400
      IF(A(I).EQ.BLANK.OR.A(I).EQ.COMMA) GO TO 60                       22410
      IF(A(I).EQ.E) GO TO 90                                            22420
      GO TO 25                                                          22430
   85 AR(J)=AR(J)+FAC*FLOAT(L-1)*(10.**(-K))                            22440
      GO TO 71                                                          22450
   90 KK=0                                                              22460
      NEXP=0                                                            22470
      NFAC=1                                                            22480
      NCOUNT=0                                                          22490
      GO TO 100                                                         22500
   95 NFAC=-1                                                           22510
  100 I=I+1                                                             22520
      IF(I.EQ.79) GO TO 126                                             22530
      NCOUNT=NCOUNT+1                                                   22540
      IF(NCOUNT.EQ.5) GO TO 126                                         22550
      IF(NCOUNT.GT.1) GO TO 105                                         22560
      IF(A(I).EQ.BLANK.OR.A(I).EQ.PLUS) GO TO 100                       22570
```

```
109 CONTINUE                                                                    3620
    WRITE(6,642)NOBS                                                            3630
    GO TO 105                                                                   3640
120 NN=NIND+2                                                                   3650
    WRITE(6,655)NOBS,NN,NIND                                                    3660
    IW=2                                                                        3670
    DO 123 I=1,NOBS2                                                            3680
    J=J+1                                                                       3690
    N=LABEL(J)                                                                  3700
    IF(N.EQ.NSAVE+1) WRITE(6,641)ITHSET(N),N,J                                  3710
```

G01

```
    NSAVE=N                                                                     3720
    CALL READ(NN,AR)                                                            3730
    IF(IGO.EQ.2) GO TO 1001                                                     3740
    IF(ANS.EQ.4HBACK) GO TO 1004                                                3750
    J=IFIX(ANS+.1)                                                              3760
    IF(J.LT.1) GO TO 1007                                                       3770
    IF(J.GT.NOBS) GO TO 121                                                     3780
    GO TO 122                                                                   3790
121 NAGR=8                                                                      3800
    GO TO 1008                                                                  3810
122 CONTINUE                                                                    3820
    DO 1220 L=1,NIND                                                            3830
1220 XX(L,J)=AR(L+2)                                                            3840
    INDEX(J)=J                                                                  3850
    Y(J)=AR(2)                                                                  3860
    IF(J.EQ.NOBS) GO TO 125                                                     3870
123 CONTINUE                                                                    3880
    WRITE(6,642)NOBS                                                            3890
    GO TO 120                                                                   3900
125 CONTINUE                                                                    3910
    DO 126 I=1,NOBS                                                             3920
    X(I)=XX(1,I)                                                                3930
126 IF(NIND.GT.1) Z(I)=XX(2,I)                                                  3940
    IF(NS.GT.NQ) GO TO 1006                                                     3950
130 NQ=18                                                                       3960
    WRITE(6,660)                                                                3970
    CALL READ(1,AR)                                                             3980
    IF(IGO.EQ.2) GO TO 1001                                                     3990
    IF(ANS.EQ.4HBACK) GO TO 1004                                                4000
    IF(ANS.EQ.2HOK) GO TO 131                                                   4010
    WRITE(6,1027)                                                               4020
    GO TO 130                                                                   4030
131 IF(NFUNC.EQ.-1.OR.NOP.EQ.-1.OR.NOP2.EQ.-1.OR.NOBS2.EQ.-1)GOTO 1013          4040
    DO 133 I=1,NOBS                                                             4050
    IF(ANS16.NE.3HYES) GO TO 1310                                               4060
    IF(IWGHT.GT.1) WY(I)=WEIGHT(Y(I),IWGHT,ANS16B)                              4070
    IF(WY(I).LT.0.) GO TO 1012                                                  4080
1310 CONTINUE                                                                   4090
    DO 132 J=1,NOBS2                                                            4100
    IF(INDEX(J).EQ.I) GO TO 133                                                 4110
132 CONTINUE                                                                    4120
    J=0                                                                         4130
    NSAVE=1                                                                     4140
    WRITE(6,1032)I                                                              4150
    GO TO (106,120) IW                                                          4160
133 CONTINUE                                                                    4170
    IF(NLOOP.GT.MAX.AND.MAX.NE.1) GO TO 1010                                    4180
    DO 144 I=1,NOP                                                              4190
    IF(ANS11.EQ.2HNO) GO TO 143                                                 4200
    DO 141 J=1,N11                                                              4210
    K=NF(J)                                                                     4220
    IF(K.EQ.I) GO TO 142                                                        4230
141 CONTINUE                                                                    4240
    GO TO 143                                                                   4250
142 IF(F1(K).EQ.F2(K)) GO TO 143                                                4260
    IF(FS(K).LT.F1(K).OR.FS(K).GT.F2(K)) GO TO 1420                             4270
    F(K)=FS(K)                                                                  4280
    FSAVE(K)=F(K)                                                               4290
    GO TO 144                                                                   4300
1420 WRITE(6,1030)K                                                             4310
    CALL READ(1,AR)                                                             4320
    IF(IGO.EQ.2) GO TO 1001                                                     4330
```

H01

```
    IF(ANS.EQ.4HBACK) GO TO 1004                                                4340
    IF(ANS.EQ.2HOK) GO TO 143                                                   4350
    WRITE(6,1027)                                                               4360
    GO TO 1420                                                                  4370
143 F(I)=F1(I)+0.5*(F2(I)-F1(I))                                                4380
    FSAVE(I)=F(I)                                                               4390
144 CONTINUE                                                                    4400
    NEWNOP=NOP                                                                  4410
    DO 150 I=1,NOP                                                              4420
    IF(ANS12.EQ.2HNO) GO TO 148                                                 4430
    DO 145 J=1,N12                                                              4440
    K=NT(J)                                                                     4450
    IF(K.EQ.I) GO TO 146                                                        4460
145 CONTINUE                                                                    4470
    GO TO 148                                                                   4480
146 IF((STPS(I)+F(I)).LT.F1(I).OR.(STPS(I)+F(I)).GT.F2(I)) GO TO 147            4490
    STEP(I)=STPS(I)                                                             4500
    GO TO 149                                                                   4510
147 WRITE(6,1031)I                                                              4520
    CALL READ(1,AR)                                                             4530
    IF(IGO.EQ.2) GO TO 1001                                                     4540
    IF(ANS.EQ.4HBACK) GO TO 1004                                                4550
    IF(ANS.EQ.2HOK) GO TO 148                                                   4560
    WRITE(6,1027)                                                               4570
    GO TO 147                                                                   4580
148 DEL1=F(I)-F1(I)                                                             4590
    DEL2=F2(I)-F(I)                                                             4600
    IF(DEL1.GT.DEL2) STEP(I)=-(0.3+0.1/FLOAT(I))*DEL1                           4610
    IF(DEL1.LE.DEL2) STEP(I)=(0.3+0.1/FLOAT(I))*DEL2                            4620
    STPS(I)=STEP(I)                                                             4630
149 STEPS(I)=STEP(I)                                                            4640
    IF(F1(I).EQ.F2(I)) STEP(I)=0                                                4650
```

```
 781 FORMAT( " WEIGHTING HAS BEEN DONE ACCORDING TO SCHEME NO.",I2,      12930
    *" (WITH R=",E10.4,")"  )                                           12940
 789 FORMAT(/1X,16(1H*)," SUM-OF-SQUARES FUNCTION MINIMIZATION REPORT "  12950
    *,16(1H*)//  )                                                      12960
 790 FORMAT(/1X,77(1H*)// )                                             12970
 791 FORMAT(/ " IT IS NECESSARY TO MAKE CORRECTION IN YOUR INPUT",       12980
    *" TO REACH CONVERGENCE. "  )                                       12990
 792 FORMAT(" PROBABLY BECAUSE OF POSSIBLE CONSTRAIN(S) AS SEEN",       13000
    *" FROM" )                                                          13010
```

G02

```
 793 FORMAT(/ " THE REASON CAN ALSO BE UNSUITABLE VALUE(S) OF"  )        13020
 794 FORMAT(/" TRY NEW INITIAL ESTIMATES OR RESCALE THE PARAMETERS",     13030
    *" OR CHANGE THE VALUE OF" )                                        13040
 795 FORMAT(/" OR CHANGE THE LIMITS OF ONE OR MORE OF THE PARAMETERS" )  13050
 800 FORMAT(/1X,31(1H*)," R E S U L T S ",31(1H*)//                      13060
    *1X,77(1H*)////                                                     13070
    *" PAR.NO.   ALLOWED RANGE",5X," INITIAL VALUE FINAL VALUE STD."    13080
    *," DEV.",4X," C.V.(PCT)"/1X,77(1H-))                               13090
8010 FORMAT(/1X,31(1H*)," R E S U L T S ",31(1H*)//                     13100
    *1X,77(1H*)////                                                     13110
    *" PAR.NO.   ALLOWED RANGE",5X," INITIAL VALUE   FINAL VALUE"/      13120
    *1X,57(1H-))                                                        13130
8011 FORMAT(I4,3X,4(E11.5,1X))                                          13140
8012 FORMAT(1X,57(1H-))                                                 13150
 802 FORMAT(I4,3X,6(E11.5,1X) )                                         13160
 803 FORMAT(1X,77(1H-)///                                               13170
    *" PAR.NO.  APPROX. 95 PCT. CONF.LIMITS",/1X,38(1H-)  )             13180
8030 FORMAT( / " THE MINIMUM FOUND BY THE MINIMIZATION",                13190
    *" PROCESS APPEARS"/" TO BE CONSTRAINED BY THE ",A5,                13200
    *" LIMIT OF PARAMETER NUMBER ",I2  )                                13210
8031 FORMAT(/                                                           13220
    *" *** ENTER B10 OR B11 IF YOU WANT A NEW RUN WITH DIFFERENT"/       13230
    *" *** PARAMETER LIMITS OR DIFFERENT PARAMETER ESTIMATES"/           13240
    *" *** OTHERWISE TYPE -OK- TO CONTINUE"// )                          13250
 804 FORMAT(//                                                          13260
    *" RES. SUM OF SQUARES",4X,E12.6,5X," REG. SUM OF SQUARES",5X,       13270
    *E12.6/                                                             13280
    *" TOT. SUM OF SQUARES",4X,E12.6,5X," SUM OF SQUARED RESPONSE ",     13290
    *E12.6/                                                             13300
    *" MEAN OF RESPONSE",7X,E12.6,5X," RES. MEAN SQUARE ",7X,E12.6/     13310
    *" REG. MEAN SQUARE",7X,E12.6,5X," MEAN OF RESIDUALS",7X,E12.6/     13320
    *" STD. DEV. OF RESIDUALS ",E12.6,5X," PCT. RESPONSE EXPLAINED ",   13330
    *E12.6/" CORR. COEFFICIENT R ",2X,E12.6/                            13340
 805 FORMAT(I4,6X,2E13.4 )                                              13350
8050 FORMAT(1X,38(1H-))                                                 13360
8051 FORMAT(/ " RELATIVE POSITION OF THE CALCULATED PARAMETER",         13370
    *" IN ITS ALLOWED RANGE."/ )                                        13380
8052 FORMAT(/ " RELATIVE POSITIONS OF THE CALCULATED PARAMETERS",       13390
    *" IN THEIR ALLOWED RANGE."/)                                       13400
8053 FORMAT( " PAR.NO.",I2,3X,65A1 )                                    13410
8054 FORMAT(/" RESIDUAL SUM OF SQUARES =",E12.6/ )                      13420
8055 FORMAT(/" WEIGHTED RESIDUAL SUM OF SQUARES =",E12.6/ )             13430
 806 FORMAT(                                                            13440
    *" WGT. RES. SUM OF SQRES ",E12.6,5X," WGT. RES. MEAN SQUARE",3X,   13450
    *E12.6/                                                             13460
    *" MEAN OF WGT. RESIDUALS ",E12.6,5X," STD. DEV. OF WGT. RES.",2X,  13470
    *E12.6/ )                                                           13480
 807 FORMAT( " THE RESIDUAL SUM OF SQUARES FOR RESPONSE SYSTEM NO."      13490
    *,I2," IS ",E12.6 )                                                 13500
 808 FORMAT( " THE WEIGHTED RESIDUAL SUM OF SQUARES FOR RESPONSE",      13510
    *" SYSTEM NO.",I2," IS",E12.6)                                      13520
 810 FORMAT(" ---THE REGRESSION IS NOT SIGNIFICANT---")                 13530
 812 FORMAT( /" OBS.NO.  X1",9X,"X2   RESPONSE  CALC.RESPONSE DIFF." ,  13540
    *" DIFF(PCT) NORM.DEVIATE" /1X,77(1H-))                             13550
 813 FORMAT(/28X," RESPONSE SYSTEM NO.",I2/ )                           13560
 814 FORMAT( I4,3X,E10.4,10X,5E10.4 )                                   13570
 816 FORMAT( I4, 3X,7E10.4 )                                            13580
 817 FORMAT(/" OBS.NO",7("    X",I1,4X)/1X,77(1H-))                     13590
 818 FORMAT( 1X,77(1H-)/ )                                              13600
 819 FORMAT(I5,2X,7E10.4)                                               13610
 820 FORMAT(" ASSUMING A NORMAL DISTRIBUTION OF ERRORS,THE UNIT NORMAL"  13620
    *," DEVIATE FORM"/" OF THE RESIDUALS TOGETHER WITH THE T-DIS",      13630
```

H02

```
    *"TRIBUTION WITH",I3," DEGREES OF FREEDOM"/" INDICATES THAT")        13640
 822 FORMAT(20X," OBSERVATION NUMBER ",I3)                              13650
 824 FORMAT(/ " IS AN OUTLIER (P.LT.0.05) "/  )                         13660
8240 FORMAT(/21X," RESPONSE SYSTEM NO.",I2/ )                           13670
8241 FORMAT( /" OBS.NO.  WEIGHT  NORM.WEIGHT  WEIGHTED RESIDUAL",       13680
    *" UNIT NORMAL DEVIATE"/47X," (CRIT.VALUES=+-",F4.2,")"/            13690
    *1X,67(1H-) )                                                       13700
8242 FORMAT(I4,A2,2E12.4,2E15.4 )                                       13710
8243 FORMAT(I4,2X,2E12.4,2E15.4 )                                       13720
8244 FORMAT(1X,67(1H-)/  )                                              13730
8245 FORMAT(  " THE OBSERVATION MARKED * IN THE TABLE ABOVE"/           13740
    *" APPEARS TO BE AN OUTLIER."/ )                                    13750
8246 FORMAT(  " THE OBSERVATIONS MARKED * IN THE TABLE ABOVE"/          13760
    *" APPEAR TO BE OUTLIERS."/ )                                       13770
 826 FORMAT(/" ARE OUTLIERS (P.LT.0.05)"/ )                             13780
8261 FORMAT(" THE FITTED CURVE CAN BE PLOTTED FROM FOLLOWING 51 POINTS" 13790
    *//2(" PT.NO.   X",8X," RESPONSE")/1X,58(1H-))                      13800
8262 FORMAT(2(I4,3X,2E11.4))                                            13810
8263 FORMAT(1X,58(1H-) )                                                13820
8264 FORMAT( /18X," RESPONSE SYSTEM NO.",I2/ )                          13830
 827 FORMAT(/ " VARIANCE-COVARIANCE MATRIX OF THE PARAMETERS." /)       13840
8270 FORMAT(/" CORRELATION MATRIX "/  )                                 13850
 829 FORMAT(1X,6E13.6)                                                  13860
 830 FORMAT(/" DURBIN WATSON STATISTIC FOR SERIAL CORRELATION OF",       13870
    *" RESIDUALS =",E10.4/" (REF. J.DURBIN AND G.S.WATSON,"             13880
    *" BIOMETRICA,PP 157-178,1951)"/  )                                 13890
 832 FORMAT( " THE PROBABILITY OF RANDOMNESS OF RUNS =",E12.4/          13900
    *" (REF. F.S.SWED AND C.EISENHART, ANN. OF MATH. STATS.,14,",       13910
    *"PP66-87 ?43)" /)                                                  13920
8320 FORMAT(" ASSUMING EQUAL PROBABILITY (0.5) OF GETTING NEGATIVE AND" 13930
    *," POSITIVE"/" RESIDUALS, THE PROBABILITY OF GETTING ",I2,          13940
    *" OR LESS RESIDUALS OF SAME SIGN"/" OF A TOTAL OF",I3,              13950
```

```
      IF(A(I).EQ.DOT) GO TO 70                                          22230
      IF(A(I).EQ.E) GO TO 90                                            22240
   60 J=J+1                                                             22250
      IF(J.GT.N) GO TO 65                                               22260
      FAC=1.                                                            22270
      K=0                                                               22280
      GO TO 15                                                          22290
   65 I=I+1                                                             22300
      IF(I.EQ.79) RETURN                                                22310
```

G03

```
      IF(A(I).EQ.BLANK) GO TO 65                                        22320
      WRITE(6,215)N                                                     22330
      GO TO 5                                                           22340
   70 K=0                                                               22350
   71 I=I+1                                                             22360
      K=K+1                                                             22370
      DO 75 L=1,10                                                      22380
      IF(A(I).EQ.AL(L)) GO TO 85                                        22390
   75 CONTINUE                                                          22400
      IF(A(I).EQ.BLANK.OR.A(I).EQ.COMMA) GO TO 60                       22410
      IF(A(I).EQ.E) GO TO 90                                            22420
      GO TO 25                                                          22430
   85 AR(J)=AR(J)+FAC*FLOAT(L-1)*(10.**(-K))                            22440
      GO TO 71                                                          22450
   90 KK=0                                                              22460
      NEXP=0                                                            22470
      NFAC=1                                                            22480
      NCOUNT=0                                                          22490
      GO TO 100                                                         22500
   95 NFAC=-1                                                           22510
  100 I=I+1                                                             22520
      IF(I.EQ.79) GO TO 126                                             22530
      NCOUNT=NCOUNT+1                                                   22540
      IF(NCOUNT.EQ.5) GO TO 126                                         22550
      IF(NCOUNT.GT.1) GO TO 105                                         22560
      IF(A(I).EQ.BLANK.OR.A(I).EQ.PLUS) GO TO 100                       22570
      IF(A(I).EQ.AMINUS) GO TO 95                                       22580
  105 CONTINUE                                                          22590
      DO 110 L=1,10                                                     22600
      IF(A(I).EQ.AL(L)) GO TO 115                                       22610
  110 CONTINUE                                                          22620
      IF(KK.EQ.0) GO TO 25                                              22630
      IF(A(I).EQ.BLANK.OR.A(I).EQ.COMMA) GO TO 120                      22640
      GO TO 25                                                          22650
  115 IF(KK.EQ.1) NEXP=10*NEXP                                          22660
      NEXP=NEXP+NFAC*(L-1)                                              22670
      KK=KK+1                                                           22680
      GO TO 100                                                         22690
  120 AR(J)=AR(J)*(10.**(NEXP))                                         22700
      GO TO 60                                                          22710
  125 WRITE(6,210)N                                                     22720
      GO TO 5                                                           22730
  126 WRITE(6,220)                                                      22740
      GO TO 5                                                           22750
  130 AR(1)=3HYES                                                       22760
      RETURN                                                            22770
  135 AR(1)=2HNO                                                        22780
      RETURN                                                            22790
  139 SKIP=.TRUE.                                                       22800
  140 AR(1)=4HBACK                                                      22810
      J1=I+1                                                            22820
      J2=I+2                                                            22830
      IF(A(I+3).NE.1HK) GO TO 141                                       22840
      J1=I+4                                                            22850
      J2=I+5                                                            22860
  141 N1=11                                                             22870
      N2=11                                                             22880
      DO 142 I=1,10                                                     22890
      IF(A(J1).EQ.AL(I)) N1=I-1                                         22900
  142 IF(A(J2).EQ.AL(I)) N2=I-1                                         22910
      IF(N1.EQ.11.OR.N2.EQ.11.AND.A(J2).NE.BLANK.OR.                    22920
     *N1.EQ.0.AND.N2.EQ.0) RETURN                                       22930
```

H03

```
      NS=NQ                                                             22940
      NQ=10*N1+N2                                                       22950
      IF(A(J2).EQ.BLANK) NQ=N1                                          22960
      IGO=2                                                             22970
      IF(SKIP.AND.NQ.GE.1.AND.NQ.LE.22) RETURN                          22980
      IF(NQ.GE.1.AND.NQ.LE.NS) RETURN                                   22990
      NQ=NS                                                             23000
      WRITE(6,230)                                                      23010
      GO TO 5                                                           23020
  145 AR(1)=2HOK                                                        23030
      RETURN                                                            23040
  200 FORMAT(78A1)                                                      23050
  205 FORMAT(/" INPUT ERROR. UNRECOGNIZED CHARACTER ( ",A1," )"/        23060
     *" REENTER LAST LINE IN CORRECT FORM."//)                          23070
  210 FORMAT(/" INPUT ERROR. TOO FEW NUMBERS ON LINE ABOVE.",I2,        23080
     *" ARE EXPECTED."/" REENTER LAST LINE IN CORRECT FORM ACCORDING",   23090
     *" TO REQUEST."//)                                                 23100
  215 FORMAT(/" INPUT ERROR. TOO MANY NUMBERS ON LINE ABOVE.",I2,       23110
     *" EXPECTED."/" REENTER LAST LINE IN CORRECT FORM ACCORDING",       23120
     *" TO REQUEST."//)                                                 23130
  220 FORMAT(/" INPUT ERROR. -WRONG EXPONENT-"/                         23140
     *" REENTER LAST LINE IN CORRECT FORM ACCORDING TO REQUEST."//)      23150
  225 FORMAT(/" INPUT ERROR, YOU FORGOT TO ENTER DATA.  -TRY AGAIN-" / ) 23160
  230 FORMAT(/" INPUT ERROR,ILLIGAL TRANSFER COMMAND   - TRY AGAIN -"/) 23170
      END                                                               23180
C     ****************************************************************  23190
      SUBROUTINE PLACE(F1,F2,F,NP,NOP,A)                                23200
      DIMENSION A(650),F1(20),F2(20),F(20)                              23210
      DATA DOT,X,BLANK /1H-,1HX,1H /                                    23220
      DO 20 J=1,NOP                                                     23230
      N1=NP*(J-1)+1                                                     23240
      N2=N1+NP-1                                                        23250
```

```
      IF(FS(K).LT.F1(K).OR.FS(K).GT.F2(K)) GO TO 1420                    4270
      F(K)=FS(K)                                                         4280
      FSAVE(K)=F(K)                                                      4290
      GO TO 144                                                          4300
 1420 WRITE(6,1030)K                                                     4310
      CALL READ(1,AR)                                                    4320
      IF(IGO.EQ.2) GO TO 1001                                            4330
```

H01

```
      IF(ANS.EQ.4HBACK) GO TO 1004                                       4340
      IF(ANS.EQ.2HOK) GO TO 143                                          4350
      WRITE(6,1027)                                                      4360
      GO TO 1420                                                         4370
  143 F(I)=F1(I)+0.5*(F2(I)-F1(I))                                       4380
      FSAVE(I)=F(I)                                                      4390
  144 CONTINUE                                                           4400
      NEWNOP=NOP                                                         4410
      DO 150 I=1,NOP                                                     4420
      IF(ANS12.EQ.2HNO) GO TO 148                                        4430
      DO 145 J=1,N12                                                     4440
      K=NT(J)                                                            4450
      IF(K.EQ.I) GO TO 146                                               4460
  145 CONTINUE                                                           4470
      GO TO 148                                                          4480
  146 IF((STPS(I)+F(I)).LT.F1(I).OR.(STPS(I)+F(I)).GT.F2(I)) GO TO 147   4490
      STEP(I)=STPS(I)                                                    4500
      GO TO 149                                                          4510
  147 WRITE(6,1031)I                                                     4520
      CALL READ(1,AR)                                                    4530
      IF(IGO.EQ.2) GO TO 1001                                            4540
      IF(ANS.EQ.4HBACK) GO TO 1004                                       4550
      IF(ANS.EQ.2HOK) GO TO 148                                          4560
      WRITE(6,1027)                                                      4570
      GO TO 147                                                          4580
  148 DEL1=F(I)-F1(I)                                                    4590
      DEL2=F2(I)-F(I)                                                    4600
      IF(DEL1.GT.DEL2) STEP(I)=-(0.3+0.1/FLOAT(I))*DEL1                  4610
      IF(DEL1.LE.DEL2) STEP(I)=(0.3+0.1/FLOAT(I))*DEL2                   4620
      STPS(I)=STEP(I)                                                    4630
  149 STEPS(I)=STEP(I)                                                   4640
      IF(F1(I).EQ.F2(I)) STEP(I)=0.                                      4650
      IF(F1(I).EQ.F2(I)) NEWNOP=NEWNOP-1                                 4660
  150 CONTINUE                                                           4670
      IF(NEWNOP.GT.NOBS) GO TO 1011                                      4680
      NOWGHT=0                                                           4690
      IF(ANS16.NE.3HYES) GO TO 1525                                      4700
      IF(IWGHT.NE.1) GO TO 1510                                          4710
      N=0                                                                4720
      NDUMMY=0                                                           4730
      DO 1500 I=1,NOBS                                                   4740
      IF(WY(I).EQ.0.) NDUMMY=NDUMMY+1                                    4750
 1500 IF(WY(I).EQ.0..OR.WY(I).EQ.1.) N=N+1                               4760
      IF(NDUMMY.EQ.0.OR.N.NE.NOBS) GO TO 1510                            4770
      NOWGHT=1                                                           4780
      DO 1505 I=1,NOBS                                                   4790
 1505 WYN(I)=WY(I)                                                       4800
      GO TO 155                                                          4810
 1510 SUMW=0.                                                            4820
      DO 1515 I=1,NOBS                                                   4830
 1515 SUMW=SUMW+WY(I)                                                    4840
      FACTOR=FLOAT(NOBS-NDUMMY)/SUMW                                     4850
      DO 1520 I=1,NOBS                                                   4860
 1520 WYN(I)=FACTOR*WY(I)                                                4870
      GO TO 155                                                          4880
 1525 DO 1530 I=1,NOBS                                                   4890
      WY(I)=1.                                                           4900
 1530 WYN(I)=1.                                                          4910
  155 CONTINUE                                                           4920
  157 WRITE(6,662)                                                       4930
      CALL READ(1,AR)                                                    4940
      IF(IGO.EQ.2) GO TO 1001                                            4950
```

I01

```
      IF(ANS.EQ.4HBACK) GO TO 1004                                       4960
      IF(ANS.EQ.3HYES) GO TO 300                                         4970
      IF(ANS.EQ.2HNO) GO TO 159                                          4980
      WRITE(6,1020)                                                      4990
      GO TO 157                                                          5000
  158 WRITE(6,664)                                                       5010
      CALL READ(1,AR)                                                    5020
      IF(IGO.EQ.2) GO TO 1001                                            5030
      IF(ANS.EQ.4HBACK) GO TO 1004                                       5040
      IF(ANS.EQ.2HOK) GO TO 159                                          5050
      WRITE(6,1027)                                                      5060
      GO TO 158                                                          5070
  159 IF(NRUN.EQ.0.AND.MAX.NE.1) WRITE(6,670)                            5080
      GO TO 350                                                          5090
  160 NQ=19                                                              5100
      NEWNOP=NOP                                                         5110
      DO 1600 I=1,NOP                                                    5120
 1600 IF(F1(I).EQ.F2(I)) NEWNOP=NEWNOP-1                                 5130
      IF(NEWNOP.EQ.0.OR.NOP.GT.7) GO TO 1666                             5140
      IF(MAX.EQ.1) GO TO 162                                             5150
      WRITE(6,6700)NEWNOP                                                5160
      CALL READ(1,AR)                                                    5170
      GO TO (1000,1001) IGO                                              5180
  161 IF(ANS.EQ.2HNO) GO TO 1666                                         5190
  162 IF(NEWNOP.LT.3) GO TO 165                                          5200
      WRITE(6,6705) NRECOM(NEWNOP)                                       5210
  163 CALL READ(1,AR)                                                    5220
      IF(IGO.EQ.2) GO TO 1001                                            5230
      IF(ANS.EQ.4HBACK) GO TO 1004                                       5240
      NPOINT=IFIX(ANS+.1)                                                5250
      IF(NPOINT.LT.NCRIT(NEWNOP).OR.NPOINT.GT.2187) GO TO 164            5260
      GO TO 165                                                          5270
  164 WRITE(6,1024)                                                      5280
      GO TO 163                                                          5290
```

```
 814 FORMAT( I4,3X,E10.4,10X,5E10.4 )                                   13570
 816 FORMAT( I4, 3X,7E10.4 )                                            13580
 817 FORMAT(/" OBS.NO",7("    X",I1,4X)/1X,77(1H-))                     13590
 818 FORMAT( 1X,77(1H-)/ )                                              13600
 819 FORMAT(I5,2X,7E10.4)                                               13610
 820 FORMAT(" ASSUMING A NORMAL DISTRIBUTION OF ERRORS,THE UNIT NORMAL" 13620
    *," DEVIATE FORM"/" OF THE RESIDUALS TOGETHER WITH THE T-DIS",      13630
```

H02

```
    *"TRIBUTION WITH",I3," DEGREES OF FREEDOM"/" INDICATES THAT")       13640
 822 FORMAT(20X," OBSERVATION NUMBER ",I3)                              13650
 824 FORMAT(/ " IS AN OUTLIER (P.LT.0.05) "/ )                          13660
8240 FORMAT(/21X," RESPONSE SYSTEM NO.",I2/ )                           13670
8241 FORMAT( /" OBS.NO.  WEIGHT  NORM.WEIGHT  WEIGHTED RESIDUAL",      13680
    *" UNIT NORMAL DEVIATE"/47X," (CRIT.VALUES=+-",F4.2,")"/            13690
    *1X,67(1H-) )                                                       13700
8242 FORMAT(I4,A2,2E12.4,2E15.4 )                                       13710
8243 FORMAT(I4,2X,2E12.4,2E15.4 )                                       13720
8244 FORMAT(1X,67(1H-)/ )                                               13730
8245 FORMAT( " THE OBSERVATION MARKED * IN THE TABLE ABOVE"/            13740
    *" APPEARS TO BE AN OUTLIER."/ )                                    13750
8246 FORMAT( " THE OBSERVATIONS MARKED * IN THE TABLE ABOVE"/           13760
    *" APPEAR TO BE OUTLIERS."/ )                                       13770
 826 FORMAT(/" ARE OUTLIERS (P.LT.0.05)"/ )                             13780
8261 FORMAT(" THE FITTED CURVE CAN BE PLOTTED FROM FOLLOWING 51 POINTS" 13790
    *//2("  PT.NO.    X",8X," RESPONSE")/1X,58(1H-))                    13800
8262 FORMAT(2(I4,3X,2E11.4))                                            13810
8263 FORMAT(1X,58(1H-) )                                                13820
8264 FORMAT( /18X," RESPONSE SYSTEM NO.",I2/ )                          13830
 827 FORMAT(/ " VARIANCE-COVARIANCE MATRIX OF THE PARAMETERS." /)       13840
8270 FORMAT(/" CORRELATION MATRIX "/  )                                 13850
 829 FORMAT(1X,6E13.6)                                                  13860
 830 FORMAT(/" DURBIN WATSON STATISTIC FOR SERIAL CORRELATION OF",      13870
    *" RESIDUALS =",E10.4/" (REF. J.DURBIN AND G.S.WATSON,"             13880
    *" BIOMETRICA,PP 157-178,1951)"/  )                                 13890
 832 FORMAT( " THE PROBABILITY OF RANDOMNESS OF RUNS =",E12.4/          13900
    *" (REF. W.S.SWED AND C.EISENHART, ANN. OF MATH. STATS.,14,",       13910
    *"PP66-87,1943)" /)                                                 13920
8320 FORMAT(" ASSUMING EQUAL PROBABILITY (0.5) OF GETTING NEGATIVE AND" 13930
    *," POSITIVE"/" RESIDUALS, THE PROBABILITY OF GETTING ",I2,          13940
    *" OR LESS RESIDUALS OF SAME SIGN"/" OF A TOTAL OF",I3,              13950
    *" RESIDUALS IS =",E11.4 / )                                         13960
 833 FORMAT(" THIS INDICATES THAT THE ERRORS ARE NOT RANDOM AT THE",    13970
    *" MINIMUM FOUND"/ )                                                13980
 834 FORMAT(" THE RATIO OF THE DEGREES OF FREEDOM OF RESIDUALS TO THE", 13990
    *" NUMBER"/" OF RESIDUALS =",E10.4,". A SIGNIFICANT CORRELATION"     14000
    *," BETWEEN THE RESIDUALS"/" IS THEREFORE EXPECTED,VIOLATING THE",   14010
    *" ASSUMPTION BEHIND THE STATISTICAL"/" EVALUATION OF THE",          14020
    *" PARAMETERS."  )                                                  14030
 836 FORMAT(//8X,                                                       14040
    *" PLOT OF RESIDUALS VERSUS INDEPENDENT VARIABLE X1 (SCL.UNITS)" )   14050
8360 FORMAT(//2X," PLOT OF WEIGHTED RESIDUALS VERSUS",                  14060
    *" INDEPENDENT VARIABLE X1 (SCL.UNITS)"  )                          14070
 838 FORMAT(//11X,                                                      14080
    *" PLOT OF RESIDUALS VERSUS RESPONSE ESTIMATE (SCL.UNITS)")         14090
8380 FORMAT(//5X," PLOT OF WEIGHTED RESIDUALS",                         14100
    *" VERSUS RESPONSE ESTIMATE (SCL.UNITS)")                           14110
 840 FORMAT(//5X," PLOT OF RESIDUALS VERSUS 1ST. INDEPENDENT",          14120
    *" VARIABLE  X1  (SCL.UNITS)" )                                     14130
8400 FORMAT(//2X," PLOT OF WEIGHTED RESIDUALS VERSUS",                  14140
    *" 1ST. INDEPENDENT VARIABLE X1 (SCL.UNITS)")                       14150
 842 FORMAT(//5X," PLOT OF RESIDUALS VERSUS 2ND. INDEPENDENT VARIABLE", 14160
    *" X2   (SCL.UNITS) " )                                             14170
8420 FORMAT(//2X," PLOT OF WEIGHTED RESIDUALS VERSUS",                  14180
    *" 2ND. INDEPENDENT VARIABLE X2 (SCL.UNITS)"  )                     14190
8401 FORMAT(" CONVERGENCE CRITERION ",E10.4 )                           14200
8402 FORMAT(" EXPANSION CRITERION ",E12.4 )                             14210
8403 FORMAT(" MINIMUM OF FITTED QUADRATIC SURFACE IS  ",E15.8/          14220
    *" COMPARE WITH MINIMUM FOUND BY ITERATION ",E15.8/                 14230
    *" IF DIFFERENCE IS LARGE THE STATISTICAL EVALUATION OF THE",       14240
    *" PARAMETERS IS"/" INACCURATE AND A NEW RUN WITH A DIFFERENT",     14250
```

I02

```
    *" STOPPING CRITERION,EXPANSION"/" CRITERION OR DIFFERENT RANGE",   14260
    *" OF THE PARAMETERS IS RECOMMENDED."  )                            14270
8404 FORMAT(/" TYPE BACK (8**) IF YOU WANT TO MAKE CORRECTIONS OR",     14280
    *" OK IF YOU WANT"/" TO CONTINUE BUT WITHOUT STATISTICAL",          14290
    *" EVALUATION OF THE PARAMETERS"// )                                14300
8422 FORMAT(" NO. OF SS EVALUATIONS ",I4,17X,                           14310
    *" NO. OF CONSTR. VIOLATIONS ",I4 )                                 14320
8424 FORMAT( " THE QUADRATIC SURFACE FITTING REQUIRED A FURTHER",I4,    14330
    *" SS EVALUATIONS" )                                                14340
8425 FORMAT(/1H  )                                                      14350
 844 FORMAT(                                                            14360
    *//1X,27(1H*)," END OF FITTING NO.",I2,2X,26(1H*) // )               14370
1000 IF(ANS.EQ.3HYES.OR.ANS.EQ.2HNO) GO TO 1002                         14380
     IF(ANS.EQ.4HBACK) GO TO 1004                                       14390
     WRITE(6,1020)                                                      14400
1001 IF(SKIP) NS=NQ                                                     14410
     GO TO (5,10,15,20,25,30,35,40,45,50,55,70,85,90,95,100,105,        14420
    *130,160,170,175,185          )NQ                                   14430
1002 GO TO (6,11,16,21,26,31,36,41,46,51,56,71,86,91,96,101,105,        14440
    *130,161,171,175,185          )NQ                                   14450
1003 IF(ANS.EQ.4HBACK) GO TO 1004                                       14460
     GO TO 1002                                                         14470
1004 NS=NQ                                                              14480
     WRITE(6,1021)                                                      14490
     SKIPS=SKIP                                                         14500
     CALL READ(1,AR)                                                    14510
     IF(ANS.EQ.4HBACK) GO TO 1004                                       14520
     SKIP=SKIPS                                                         14530
     NQ=IFIX(ANS+.1)                                                    14540
     IF(SKIP.AND.NQ.GE.1.AND.NQ.LE.22) GO TO 1001                       14550
     DO 1005 I=1,NS                                                     14560
     IF(I.EQ.NQ) GO TO 1001                                             14570
1005 CONTINUE                                                           14580
     WRITE(6,1022)                                                      14590
```

```
 814 FORMAT( I4,3X,E10.4,10X,5E10.4 )                                   13570
 816 FORMAT( I4, 3X,7E10.4 )                                            13580
 817 FORMAT(/" OBS.NO",7("    X",I1,4X)/1X,77(1H-))                     13590
 818 FORMAT( 1X,77(1H-)/ )                                              13600
 819 FORMAT(I5,2X,7E10.4)                                               13610
 820 FORMAT(" ASSUMING A NORMAL DISTRIBUTION OF ERRORS,THE UNIT NORMAL" 13620
    *," DEVIATE FORM"/" OF THE RESIDUALS TOGETHER WITH THE T-DIS",      13630
```

H02

```
    *"TRIBUTION WITH",I3," DEGREES OF FREEDOM"/" INDICATES THAT")       13640
 822 FORMAT(20X," OBSERVATION NUMBER ",I3)                              13650
 824 FORMAT(/ " IS AN OUTLIER (P.LT.0.05) "/ )                          13660
8240 FORMAT(/21X," RESPONSE SYSTEM NO.",I2/ )                           13670
8241 FORMAT( /" OBS.NO.  WEIGHT  NORM.WEIGHT  WEIGHTED RESIDUAL",      13680
    *" UNIT NORMAL DEVIATE"/47X," (CRIT.VALUES=+-",F4.2,")"/            13690
    *1X,67(1H-) )                                                       13700
8242 FORMAT(I4,A2,2E12.4,2E15.4 )                                       13710
8243 FORMAT(I4,2X,2E12.4,2E15.4 )                                       13720
8244 FORMAT(1X,67(1H-)/ )                                               13730
8245 FORMAT( " THE OBSERVATION MARKED * IN THE TABLE ABOVE"/            13740
    *" APPEARS TO BE AN OUTLIER."/ )                                    13750
8246 FORMAT( " THE OBSERVATIONS MARKED * IN THE TABLE ABOVE"/           13760
    *" APPEAR TO BE OUTLIERS."/ )                                       13770
 826 FORMAT(/" ARE OUTLIERS (P.LT.0.05)"/ )                             13780
8261 FORMAT(" THE FITTED CURVE CAN BE PLOTTED FROM FOLLOWING 51 POINTS" 13790
    *//2("  PT.NO.    X",8X," RESPONSE")/1X,58(1H-))                    13800
8262 FORMAT(2(I4,3X,2E11.4))                                            13810
8263 FORMAT(1X,58(1H-) )                                                13820
8264 FORMAT( /18X," RESPONSE SYSTEM NO.",I2/ )                          13830
 827 FORMAT(/ " VARIANCE-COVARIANCE MATRIX OF THE PARAMETERS." /)       13840
8270 FORMAT(/" CORRELATION MATRIX "/  )                                 13850
 829 FORMAT(1X,6E13.6)                                                  13860
 830 FORMAT(/" DURBIN WATSON STATISTIC FOR SERIAL CORRELATION OF",      13870
    *" RESIDUALS =",E10.4/" (REF. J.DURBIN AND G.S.WATSON,"             13880
    *" BIOMETRICA,PP 157-178,1951)"/  )                                 13890
 832 FORMAT( " THE PROBABILITY OF RANDOMNESS OF RUNS =",E12.4/          13900
    *" (REF. W.S.SWED AND C.EISENHART, ANN. OF MATH. STATS.,14,",       13910
    *"PP66-87,1943)" /)                                                 13920
8320 FORMAT(" ASSUMING EQUAL PROBABILITY (0.5) OF GETTING NEGATIVE AND" 13930
    *," POSITIVE"/" RESIDUALS, THE PROBABILITY OF GETTING ",I2,          13940
    *" OR LESS RESIDUALS OF SAME SIGN"/" OF A TOTAL OF",I3,              13950
    *" RESIDUALS IS =",E11.4 / )                                         13960
 833 FORMAT(" THIS INDICATES THAT THE ERRORS ARE NOT RANDOM AT THE",    13970
    *" MINIMUM FOUND"/ )                                                13980
 834 FORMAT(" THE RATIO OF THE DEGREES OF FREEDOM OF RESIDUALS TO THE", 13990
    *" NUMBER"/" OF RESIDUALS =",E10.4,". A SIGNIFICANT CORRELATION"     14000
    *," BETWEEN THE RESIDUALS"/" IS THEREFORE EXPECTED,VIOLATING THE",   14010
    *" ASSUMPTION BEHIND THE STATISTICAL"/" EVALUATION OF THE",          14020
    *" PARAMETERS."  )                                                  14030
 836 FORMAT(//8X,                                                       14040
    *" PLOT OF RESIDUALS VERSUS INDEPENDENT VARIABLE X1 (SCL.UNITS)" )   14050
8360 FORMAT(//2X," PLOT OF WEIGHTED RESIDUALS VERSUS",                  14060
    *" INDEPENDENT VARIABLE X1 (SCL.UNITS)"  )                          14070
 838 FORMAT(//11X,                                                      14080
    *" PLOT OF RESIDUALS VERSUS RESPONSE ESTIMATE (SCL.UNITS)")         14090
8380 FORMAT(//5X," PLOT OF WEIGHTED RESIDUALS",                         14100
    *" VERSUS RESPONSE ESTIMATE (SCL.UNITS)")                           14110
 840 FORMAT(//5X," PLOT OF RESIDUALS VERSUS 1ST. INDEPENDENT",          14120
    *" VARIABLE  X1  (SCL.UNITS)" )                                     14130
8400 FORMAT(//2X," PLOT OF WEIGHTED RESIDUALS VERSUS",                  14140
    *" 1ST. INDEPENDENT VARIABLE X1 (SCL.UNITS)")                       14150
 842 FORMAT(//5X," PLOT OF RESIDUALS VERSUS 2ND. INDEPENDENT VARIABLE", 14160
    *" X2   (SCL.UNITS) " )                                             14170
8420 FORMAT(//2X," PLOT OF WEIGHTED RESIDUALS VERSUS",                  14180
    *" 2ND. INDEPENDENT VARIABLE X2 (SCL.UNITS)"  )                     14190
8401 FORMAT(" CONVERGENCE CRITERION ",E10.4 )                           14200
8402 FORMAT(" EXPANSION CRITERION ",E12.4 )                             14210
8403 FORMAT(" MINIMUM OF FITTED QUADRATIC SURFACE IS  ",E15.8/          14220
    *" COMPARE WITH MINIMUM FOUND BY ITERATION ",E15.8/                 14230
    *" IF DIFFERENCE IS LARGE THE STATISTICAL EVALUATION OF THE",       14240
    *" PARAMETERS IS"/" INACCURATE AND A NEW RUN WITH A DIFFERENT",     14250
```

I02

```
    *" STOPPING CRITERION,EXPANSION"/" CRITERION OR DIFFERENT RANGE",   14260
    *" OF THE PARAMETERS IS RECOMMENDED."  )                            14270
8404 FORMAT(/" TYPE BACK (8**) IF YOU WANT TO MAKE CORRECTIONS OR",     14280
    *" OK IF YOU WANT"/" TO CONTINUE BUT WITHOUT STATISTICAL",          14290
    *" EVALUATION OF THE PARAMETERS"// )                                14300
8422 FORMAT(" NO. OF SS EVALUATIONS ",I4,17X,                           14310
    *" NO. OF CONSTR. VIOLATIONS ",I4 )                                 14320
8424 FORMAT( " THE QUADRATIC SURFACE FITTING REQUIRED A FURTHER",I4,    14330
    *" SS EVALUATIONS" )                                                14340
8425 FORMAT(/1H  )                                                      14350
 844 FORMAT(                                                            14360
    *//1X,27(1H*)," END OF FITTING NO.",I2,2X,26(1H*) // )               14370
1000 IF(ANS.EQ.3HYES.OR.ANS.EQ.2HNO) GO TO 1002                         14380
     IF(ANS.EQ.4HBACK) GO TO 1004                                       14390
     WRITE(6,1020)                                                      14400
1001 IF(SKIP) NS=NQ                                                     14410
     GO TO (5,10,15,20,25,30,35,40,45,50,55,70,85,90,95,100,105,        14420
    *130,160,170,175,185          )NQ                                   14430
1002 GO TO (6,11,16,21,26,31,36,41,46,51,56,71,86,91,96,101,105,        14440
    *130,161,171,175,185          )NQ                                   14450
1003 IF(ANS.EQ.4HBACK) GO TO 1004                                       14460
     GO TO 1002                                                         14470
1004 NS=NQ                                                              14480
     WRITE(6,1021)                                                      14490
     SKIPS=SKIP                                                         14500
     CALL READ(1,AR)                                                    14510
     IF(ANS.EQ.4HBACK) GO TO 1004                                       14520
     SKIP=SKIPS                                                         14530
     NQ=IFIX(ANS+.1)                                                    14540
     IF(SKIP.AND.NQ.GE.1.AND.NQ.LE.22) GO TO 1001                       14550
     DO 1005 I=1,NS                                                     14560
     IF(I.EQ.NQ) GO TO 1001                                             14570
1005 CONTINUE                                                           14580
     WRITE(6,1022)                                                      14590
```

```
      N2=11                                                             22880
      DO 142 I=1,10                                                     22890
      IF(A(J1).EQ.AL(I)) N1=I-1                                         22900
  142 IF(A(J2).EQ.AL(I)) N2=I-1                                         22910
      IF(N1.EQ.11.OR.N2.EQ.11.AND.A(J2).NE.BLANK.OR.                    22920
     *N1.EQ.0.AND.N2.EQ.0) RETURN                                       22930
```

H03

```
      NS=NQ                                                             22940
      NQ=10*N1+N2                                                       22950
      IF(A(J2).EQ.BLANK) NQ=N1                                          22960
      IGO=2                                                             22970
      IF(SKIP.AND.NQ.GE.1.AND.NQ.LE.22) RETURN                          22980
      IF(NQ.GE.1.AND.NQ.LE.NS) RETURN                                   22990
      NQ=NS                                                             23000
      WRITE(6,230)                                                      23010
      GO TO 5                                                           23020
  145 AR(1)=ZHOK                                                        23030
      RETURN                                                            23040
  200 FORMAT(78A1)                                                      23050
  205 FORMAT(/" INPUT ERROR. UNRECOGNIZED CHARACTER ( ",A1," )"/        23060
     *" REENTER LAST LINE IN CORRECT FORM."//)                          23070
  210 FORMAT(/" INPUT ERROR. TOO FEW NUMBERS ON LINE ABOVE.",I2,        23080
     *" ARE EXPECTED."/" REENTER LAST LINE IN CORRECT FORM ACCORDING",   23090
     *" TO REQUEST."//)                                                 23100
  215 FORMAT(/" INPUT ERROR. TOO MANY NUMBERS ON LINE ABOVE.",I2,       23110
     *" EXPECTED."/" REENTER LAST LINE IN CORRECT FORM ACCORDING",       23120
     *" TO REQUEST."//)                                                 23130
  220 FORMAT(/" INPUT ERROR. -WRONG EXPONENT-"/                         23140
     *" REENTER LAST LINE IN CORRECT FORM ACCORDING TO REQUEST."//)      23150
  225 FORMAT(/" INPUT ERROR, YOU FORGOT TO ENTER DATA.  -TRY AGAIN-" / ) 23160
  230 FORMAT(/" INPUT ERROR,ILLIGAL TRANSFER COMMAND   - TRY AGAIN -"/) 23170
      END                                                               23180
C     ****************************************************************  23190
      SUBROUTINE PLACE(F1,F2,F,NP,NOP,A)                                23200
      DIMENSION A(650),F1(20),F2(20),F(20)                              23210
      DATA DOT,X,BLANK /1H-,1HX,1H /                                    23220
      DO 20 J=1,NOP                                                     23230
      N1=NP*(J-1)+1                                                     23240
      N2=N1+NP-1                                                        23250
      IF(F1(J).EQ.F2(J)) GO TO 10                                       23260
      DO 5 I=N1,N2                                                      23270
    5 A(I)=DOT                                                          23280
      NX=N1+IFIX((NP-1)*(F(J)-F1(J))/(F2(J)-F1(J))+.5)                  23290
      IF(NX.GT.N2.OR.NX.LT.N1) GO TO 20                                 23300
      A(NX)=X                                                           23310
      GO TO 20                                                          23320
   10 CONTINUE                                                          23330
      DO 15 I=N1,N2                                                     23340
   15 A(I)=BLANK                                                        23350
   20 CONTINUE                                                          23360
      RETURN                                                            23370
      END                                                               23380
C     ****************************************************************  23390
      FUNCTION WEIGHT(Y,IWGHT,R)                                        23400
      I=IWGHT-1                                                         23410
      GO TO (5,10,15)I                                                  23420
    5 IF(Y.LT.0..OR.Y.EQ.0..AND.R.LT.0.) GO TO 20                       23430
      WEIGHT=Y**R                                                       23440
      RETURN                                                            23450
   10 IF(R*Y.LT.1) GO TO 20                                             23460
      WEIGHT=ALOG(R*Y)                                                  23470
      RETURN                                                            23480
   15 WEIGHT=EXP(R*Y)                                                   23490
      RETURN                                                            23500
   20 WEIGHT=-1.                                                        23510
      RETURN                                                            23520
      END                                                               23530
C     ****************************************************************  23540
      SUBROUTINE XYPLOT(F)                                              23550
```

I03

```
      DIMENSION X(100),Y(100),WYN(100),LX(100),LY(100),LIM1(10),LIM2(10)  23560
     *,LXF(75),LYF(75),XF(75),YF(75),F(20),A(75),XX(9,100)              23570
      COMMON /DATA/ XX,Y,WYN,NOBS                                       23580
      COMMON /FUNNUM/ITHFUN                                             23590
      COMMON /LXLY/LX,LY                                                23600
      COMMON /B1/ X                                                     23610
      COMMON /B4/ NIND                                                  23620
      COMMON /B6/ NOP                                                   23630
      COMMON /B8/NFUNC                                                  23640
      COMMON /B9/LIM1,LIM2                                              23650
      DATA BLANK,POINT1,POINT2,DOT,STAR /1H ,1H+,1HO,1H.,1HX/           23660
      DO 45 II=1,NFUNC                                                  23670
      N1=LIM1(II)                                                       23680
      N2=N1+1                                                           23690
      N3=LIM2(II)                                                       23700
      NN=N3-1                                                           23710
      XMIN=X(N1)                                                        23720
      XMAX=X(N1)                                                        23730
      YMIN=Y(N1)                                                        23740
      YMAX=Y(N1)                                                        23750
      IF(II.EQ.1) WRITE(6,100)                                          23760
      IF(II.GT.1) WRITE(6,101)II                                        23770
      DO 5 I=N2,N3                                                      23780
      IF(X(I).GT.XMAX) XMAX=X(I)                                        23790
      IF(X(I).LT.XMIN) XMIN=X(I)                                        23800
      IF(Y(I).GT.YMAX) YMAX=Y(I)                                        23810
    5 IF(Y(I).LT.YMIN)YMIN=Y(I)                                         23820
      SPANX=XMAX-XMIN                                                   23830
      IF(SPANX.EQ.0.)RETURN                                             23840
      DELX=SPANX/74.                                                    23850
      DO 10 I=1,75                                                      23860
      XF(I)=XMIN+FLOAT(I-1)*DELX                                        23870
      DO 7 L=N1,NN                                                      23880
    7 IF(XF(I).GT.X(L).AND.XF(I).LE.X(L+1))                             23890
     *YF(I)=Y(L)+(Y(L+1)-Y(L))*(XF(I)-X(L))/(X(L+1)-X(L))               23900
```

```
      FACTOR=FLOAT(NOBS-NDUMMY)/SUMW                                    4850
      DO 1520 I=1,NOBS                                                  4860
 1520 WYN(I)=FACTOR*WY(I)                                               4870
      GO TO 155                                                         4880
 1525 DO 1530 I=1,NOBS                                                  4890
      WY(I)=1.                                                          4900
 1530 WYN(I)=1.                                                         4910
  155 CONTINUE                                                          4920
  157 WRITE(6,662)                                                      4930
      CALL READ(1,AR)                                                   4940
      IF(IGO.EQ.2) GO TO 1001                                           4950
```

I01

```
      IF(ANS.EQ.4HBACK) GO TO 1004                                      4960
      IF(ANS.EQ.3HYES) GO TO 300                                        4970
      IF(ANS.EQ.2HNO) GO TO 159                                         4980
      WRITE(6,1020)                                                     4990
      GO TO 157                                                         5000
  158 WRITE(6,664)                                                      5010
      CALL READ(1,AR)                                                   5020
      IF(IGO.EQ.2) GO TO 1001                                           5030
      IF(ANS.EQ.4HBACK) GO TO 1004                                      5040
      IF(ANS.EQ.2HOK) GO TO 159                                         5050
      WRITE(6,1027)                                                     5060
      GO TO 158                                                         5070
  159 IF(NRUN.EQ.0.AND.MAX.NE.1) WRITE(6,670)                           5080
      GO TO 350                                                         5090
  160 NQ=19                                                             5100
      NEWNOP=NOP                                                        5110
      DO 1600 I=1,NOP                                                   5120
 1600 IF(F1(I).EQ.F2(I)) NEWNOP=NEWNOP-1                                5130
      IF(NEWNOP.EQ.0.OR.NOP.GT.7) GO TO 1666                            5140
      IF(MAX.EQ.1) GO TO 162                                            5150
      WRITE(6,6700)NEWNOP                                               5160
      CALL READ(1,AR)                                                   5170
      GO TO (1000,1001) IGO                                             5180
  161 IF(ANS.EQ.2HNO) GO TO 1666                                        5190
  162 IF(NEWNOP.LT.3) GO TO 165                                         5200
      WRITE(6,6705) NRECOM(NEWNOP)                                      5210
  163 CALL READ(1,AR)                                                   5220
      IF(IGO.EQ.2) GO TO 1001                                           5230
      IF(ANS.EQ.4HBACK) GO TO 1004                                      5240
      NPOINT=IFIX(ANS+.1)                                               5250
      IF(NPOINT.LT.NCRIT(NEWNOP).OR.NPOINT.GT.2187) GO TO 164           5260
      GO TO 165                                                         5270
  164 WRITE(6,1020)                                                     5280
      GO TO 163                                                         5290
  165 WRITE(6,6710)NEWNOP                                               5300
      IF(MAX.EQ.1) CALL LSQ(F,FUNC1)                                    5310
      IF(MAX.EQ.1) FUNC=FUNC1                                           5320
      CALL SEARCH(FNEW,STPNEW,SSMIN,ANS16,NPOINT,NEVAL,NEWNOP,STEP,NP)  5330
      IF(NEWNOP.GT.2) WRITE(6,6715)NEVAL,NP                             5340
      IF(ANS16.EQ.3HYES) WRITE(6,6720)SSMIN                             5350
      IF(ANS16.EQ.2HNO) WRITE(6,6725)SSMIN                              5360
      CALL PLACE(F1,F2,FNEW,40,NPR,A)                                   5370
      DO 1650 J=1,NPR                                                   5380
      IF(F1(J).EQ.F2(J)) GO TO 1650                                     5390
      N1=40*(J-1)+1                                                     5400
      N2=N1+39                                                          5410
      WRITE(6,6730)J,FNEW(J),(A(I),I=N1,N2)                             5420
 1650 CONTINUE                                                          5430
      IF(SSMIN.GE.FUNC) GO TO 1653                                      5440
      IF(MAX.EQ.1) GO TO 1652                                           5450
      WRITE(6,6735) FUNC                                                5460
      DO 1651 I=1,NOP                                                   5470
      F(I)=FNEW(I)                                                      5480
      FSAVE(I)=F(I)                                                     5490
      IF(F(I)+STPNEW(I).GT.F2(I)) STPNEW(I)=-STPNEW(I)                  5500
 1651 STEP(I)=STPNEW(I)                                                 5510
      GO TO 1664                                                        5520
 1652 WRITE(6,6731) FUNC                                                5530
      GO TO 1655                                                        5540
 1653 IF(MAX.EQ.1) GO TO 1654                                           5550
      WRITE(6,6740) FUNC                                                5560
      CALL PLACE(F1,F2,FMIN,40,NPR,A)                                   5570
```

J01

```
      GO TO 1656                                                        5580
 1654 WRITE(6,6741) FUNC                                                5590
 1655 CALL PLACE(F1,F2,F,40,NPR,A)                                      5600
 1656 CONTINUE                                                          5610
      DO 1657 J=1,NPR                                                   5620
      IF(F1(J).EQ.F2(J)) GO TO 1657                                     5630
      N1=40*(J-1)+1                                                     5640
      N2=N1+39                                                          5650
      IF(MAX.EQ.1) WRITE(6,6730)J,F(J),(A(I),I=N1,N2)                   5660
      IF(MAX.NE.1) WRITE(6,6730)J,FMIN(J),(A(I),I=N1,N2)                5670
 1657 CONTINUE                                                          5680
      WRITE(6,6745)                                                     5690
 1660 WRITE(6,6748)                                                     5700
      CALL READ(1,AR)                                                   5710
      IF(IGO.EQ.2) GO TO 1001                                           5720
      IF(ANS.EQ.4HBACK) GO TO 1004                                      5730
      IF(ANS.EQ.3HYES) GO TO 1661                                       5740
      IF(ANS.EQ.2HNO.AND.MAX.EQ.1) GO TO 1663                           5750
      IF(ANS.EQ.2HNO) GO TO 1666                                        5760
      WRITE(6,1020)                                                     5770
      GO TO 1660                                                        5780
 1661 ANS11=3HYES                                                       5790
      ANS12=3HYES                                                       5800
      N11=NOP                                                           5810
      N12=NOP                                                           5820
      DO 1662 I=1,NOP                                                   5830
      NF(I)=I                                                           5840
      NT(I)=I                                                           5850
      F(I)=FNEW(I)                                                      5860
      FS(I)=F(I)                                                        5870
```

```
     *" 1ST. INDEPENDENT VARIABLE X1 (SCL.UNITS)")                          14150
  842 FORMAT(//5X," PLOT OF RESIDUALS VERSUS 2ND. INDEPENDENT VARIABLE",   14160
     *" X2   (SCL.UNITS) " )                                               14170
 8420 FORMAT//2X," PLOT OF WEIGHTED RESIDUALS VERSUS",                     14180
     *" 2ND. INDEPENDENT VARIABLE X2 'SCL.UNITS)"  )                       14190
 8401 FORMAT(," CONVERGENCE CRITERION ",E10.4 )                            14200
 8402 FORMAT(" EXPANSION CRITERION ",E12.4 )                               14210
 8403 FORMAT(" MINIMUM OF FITTED QUADRATIC SURFACE IS  ",E15.8/            14220
     *" COMPARE WITH MINIMUM FOUND BY ITERATION ",E15.8/                   14230
     *" IF DIFFERENCE IS LARGE THE STATISTICAL EVALUATION OF THE",         14240
     *" PARAMETERS IS"/" INACCURATE AND A NEW RUN WITH A DIFFERENT",       14250
```

I02

```
     *" STOPPING CRITERION,EXPANSION"/" CRITERION OR DIFFERENT RANGE",     14260
     *" OF THE PARAMETERS IS RECOMMENDED."  )                              14270
 8404 FORMAT(/" TYPE BACK (B**) IF YOU WANT TO MAKE CORRECTIONS OR",       14280
     *" OK IF YOU WANT"/" TO CONTINUE BUT WITHOUT STATISTICAL",            14290
     *" EVALUATION OF THE PARAMETERS"// )                                  14300
 8422 FORMAT(" NO. OF SS EVALUATIONS ",I4,17X,                             14310
     *" NO. OF CONSTR. VIOLATIONS ",I4 )                                   14320
 8424 FORMAT( " THE QUADRATIC SURFACE FITTING REQUIRED A FURTHER",I4,      14330
     *" SS EVALUATIONS" )                                                  14340
 8425 FORMAT(/1H  )                                                        14350
  844 FORMAT(                                                              14360
     *//1X,27(1H*)," END OF FITTING NO.",I2,2X,26(1H*) // )                14370
 1000 IF(ANS.EQ.3HYES.OR.ANS.EQ.2HNO) GO TO 1002                           14380
      IF(ANS.EQ.4HBACK) GO TO 1004                                         14390
      WRITE(6,1020)                                                        14400
 1001 IF(SKIP) NS=NQ                                                       14410
      GO TO (5,10,15,20,25,30,35,40,45,50,55,70,85,90,95,100,105,          14420
     *130,160,170,175,185          )NQ                                    14430
 1002 GO TO (6,11,16,21,26,31,36,41,46,51,56,71,86,91,96,101,105,          14440
     *130,161,171,175,185          )NQ                                    14450
 1003 IF(ANS.EQ.4HBACK) GO TO 1004                                         14460
      GO TO 1002                                                           14470
 1004 NS=NQ                                                                14480
      WRITE(6,1021)                                                        14490
      SKIPS=SKIP                                                           14500
      CALL READ(1,AR)                                                      14510
      IF(ANS.EQ.4HBACK) GO TO 1004                                         14520
      SKIP=SKIPS                                                           14530
      NQ=IFIX(ANS+.1)                                                      14540
      IF(SKIP.AND.NQ.GE.1.AND.NQ.LE.22) GO TO 1001                         14550
      DO 1005 I=1,NS                                                       14560
      IF(I.EQ.NQ) GO TO 1001                                               14570
 1005 CONTINUE                                                             14580
      WRITE(6,1022)                                                        14590
      NQ=NS                                                                14600
      GO TO 1004                                                           14610
 1006 NCORR=NCORR+1                                                        14620
      IF(NCORR.EQ.1.AND.NS.NE.NQ.AND.NS.NE.22) WRITE(6,1023)NS             14630
      NQ=NS                                                                14640
      GO TO 1001                                                           14650
 1007 WRITE(6,1024)                                                        14660
      GO TO 1001                                                           14670
 1008 WRITE(6,1025) NAGR                                                   14680
      GO TO 1001                                                           14690
 1009 WRITE(6,1026)                                                        14700
      GO TO 1001                                                           14710
 1010 WRITE(6,1028)                                                        14720
      GO TO 1004                                                           14730
 1011 WRITE(6,1029)                                                        14740
      GO TO 1004                                                           14750
 1012 WRITE(6,1034)I,Y(I),IWGHT                                            14760
      GO TO 1004                                                           14770
 1013 WRITE(6,1035)                                                        14780
      GO TO 1004                                                           14790
 1020 FORMAT(/," INPUT ERROR, YOUR ANSWER MUST BE ONE OF THE FOLLOWING"/   14800
     *" FOUR,   Y   N   B**   R**      - TRY AGAIN -"/ )                   14810
 1021 FORMAT(/" WHAT REQUEST NUMBER DO YOU WANT TO GO TO"// )              14820
 1035 FORMAT(" YOU FORGOT AN ESSENTIAL REQUEST (I.E. 8,9,10 OR 17)" )      14830
 1022 FORMAT(/," INPUT ERROR, THE ENTERED NUMBER IS NOT AMONG"/            14840
     *" THE REQUEST NUMBERS ABOVE.     - TRY AGAIN -"/)                   14850
 1023 FORMAT(" CONTINUE NOW AFTER CORRECTION FROM REQUEST NO."I3/)         14860
 1024 FORMAT(/," INPUT ERROR, THE ENTERED NUMBER IS NOT",                  14870
```

J02

```
     *" IN THE ALLOWED RANGE. - TRY AGAIN -"/ )                           14880
 1025 FORMAT(/," INPUT ERROR, YOUR INPUT DOES NOT AGREE WITH YOUR"/        14890
     *" INPUT ABOVE UNDER REQUEST NUMBER",I3,"      - TRY AGAIN -"/)       14900
 1026 FORMAT(/," INPUT ERROR, THE INPUT IS NOT IN THE RIGHT ORDER.  ",     14910
     *" - TRY AGAIN -" / )                                                14920
 1027 FORMAT(/ " INPUT ERROR, YOUR ANSWER MUST BE  OK  B  OR  R" ,         14930
     *"- TRY AGAIN -"/ )                                                  14940
 1028 FORMAT(" INPUT ERROR, YOUR INPUT UNDER REQUEST NUMBER 6 DOES"/       14950
     *" NOT AGREE WITH THE INPUT UNDER REQUEST NUMBER 5")                  14960
 1029 FORMAT(" INPUT ERROR,YOUR INPUT UNDER REQUEST NUMBER 9 DOES"/        14970
     *" NOT AGREE WITH THE INPUT UNDER REQUEST NUMBER 8")                  14980
 1030 FORMAT(/" INPUT ERROR, THE INITIAL ESTIMATE OF PARAMETER NO.",       14990
     *I2," IS NOT WITHIN"/" ITS LIMITS. ENTER -BACK- IF YOU WANT TO",      15000
     *" MAKE CORRECTION OR -OK- IF YOU"/" WANT THE PROGRAM TO CHOOSE",     15010
     *" AN ACCEPTABLE INITIAL PARAMETER ESTIMATE" //  )                   15020
 1031 FORMAT(/" INPUT ERROR, THE STEP SIZE FOR PARAMETER NO.",I2,          15030
     *" DOES NOT AGREE WITH"/" THE LIMITS FOR THIS PARAMETER. ENTER",      15040
     *" -BACK- IF YOU WANT TO MAKE CORRECTION"/" OR -OK- IF YOU WANT",     15050
     *" THE PROGRAM TO CHOOSE AN ACCEPTABLE STEP SIZE" //  )               15060
 1032 FORMAT(" INPUT ERROR, YOU FORGOT TO ENTER OBSERVATION NUMBER",I3/    15070
     *" REENTER YOUR DATA STRICTLY ACCORDING TO FOLLOWING REQUEST "/)      15080
 1033 FORMAT( " INPUT ERROR,THE WEIGHT FOR THE OBSERVATION ",              15090
     *"IS NEGATIVE."/" REENTER THE LAST LINE IN CORRECT FORM."//)          15100
 1034 FORMAT(/" INPUT ERROR, THE WEIGHT FOR OBSERVATION NO.",I3,           15110
     *" (Y=",E10.4,") "/" IS NOT DEFINED BY THE WEIGHTING SCHEME NO.",     15120
     *I2  )                                                               15130
      END                                                                 15140
C     ********************************************************************15150
      SUBROUTINE NELDR(F,STEP,NOP,FUNC,MAX,IPRINT,STOPCR,NLOOP,IQUAD,     15160
     *    SIMP,VAR,FUNCTN,IFAULT)                                          15170
```

```
   10 IF(R*Y.LT.1) GO TO 20                                                23460
      WEIGHT=ALOG(R*Y)                                                     23470
      RETURN                                                               23480
   15 WEIGHT=EXP(R*Y)                                                      23490
      RETURN                                                               23500
   20 WEIGHT=-1.                                                           23510
      RETURN                                                               23520
      END                                                                  23530
C     ****************************************************************** 23540
      SUBROUTINE XYPLOT(F)                                                 23550
```

I03

```
      DIMENSION X(100),Y(100),WYN(100),LX(100),LY(100),LIM1(10),LIM2(10)   23560
     *,LXF(75),LYF(75),XF(75),YF(75),F(20),A(75),XX(9,100)                 23570
      COMMON /DATA/ XX,Y,WYN,NOBS                                          23580
      COMMON /FUNNUM/ITHFUN                                                23590
      COMMON /LXLY/LX,LY                                                   23600
      COMMON /B1/ X                                                        23610
      COMMON /B4/ NIND                                                     23620
      COMMON /B6/ NOP                                                      23630
      COMMON /B8/NFUNC                                                     23640
      COMMON /B9/LIM1,LIM2                                                 23650
      DATA BLANK,POINT1,POINT2,DOT,STAR /1H ,1H*,1HO,1H.,1HX/              23660
      DO 45 II=1,NFUNC                                                     23670
      N1=LIM1(II)                                                          23680
      N2=N1+1                                                              23690
      N3=LIM2(II)                                                          23700
      NN=N3-1                                                              23710
      XMIN=X(N1)                                                           23720
      XMAX=X(N1)                                                           23730
      YMIN=Y(N1)                                                           23740
      YMAX=Y(N1)                                                           23750
      IF(II.EQ.1) WRITE(6,100)                                             23760
      IF(II.GT.1) WRITE(6,101)II                                           23770
      DO 5 I=N2,N3                                                         23780
      IF(X(I).GT.XMAX) XMAX=X(I)                                           23790
      IF(X(I).LT.XMIN) XMIN=X(I)                                           23800
      IF(Y(I).GT.YMAX) YMAX=Y(I)                                           23810
    5 IF(Y(I).LT.YMIN)YMIN=Y(I)                                            23820
      SPANX=XMAX-XMIN                                                      23830
      IF(SPANX.EQ.0.)RETURN                                                23840
      DELX=SPANX/74.                                                       23850
      DO 10 I=1,75                                                         23860
      XF(I)=XMIN+FLOAT(I-1)*DELX                                           23870
      DO 7 L=N1,NN                                                         23880
    7 IF(XF(I).GT.X(L).AND.XF(I).LE.X(L+1))                                23890
     *YF(I)=Y(L)+(Y(L+1)-Y(L))*(XF(I)-X(L))/(X(L+1)-X(L))                   23900
      ITHFUN=II                                                            23910
      CALL MODEL(YF(I),XF(I),F,0)                                          23920
      IF(YF(I).GT.YMAX) YMAX=YF(I)                                         23930
   10 IF(YF(I).LT.YMIN) YMIN=YF(I)                                         23940
      SPANY=YMAX-YMIN                                                      23950
      IF(SPANY.EQ.0.)RETURN                                                23960
      DO 15 I=N1,N3                                                        23970
      IF(WYN(I).EQ.0.) GO TO 12                                            23980
      LY(I)=IFIX(49.*(YMAX-Y(I))/SPANY+.5) +1                              23990
      LX(I)=IFIX(74.*(X(I)-XMIN)/SPANX+.5) +1                              24000
      GO TO 15                                                             24010
   12 LY(I)=0                                                              24020
      LX(I)=0                                                              24030
   15 CONTINUE                                                             24040
      DO 20 I=1,75                                                         24050
      LYF(I)=IFIX(49.*(YMAX-YF(I))/SPANY+.5) +1                            24060
   20 LXF(I)=IFIX(74.*(XF(I)-XMIN)/SPANX+.5) +1                            24070
      NZERO=IFIX(49.*YMAX/SPANY+.5) +1                                     24080
      IF(NZERO.EQ.50.AND.YMIN.GT.0.)NZERO=51                               24090
      IF(NZERO.EQ.1.AND.YMAX.LT.0.) NZERO=0                                24100
      DO 40 I=1,50                                                         24110
      DO 35 J=1,75                                                         24120
      A(J)=BLANK                                                           24130
      IF(I.EQ.NZERO)A(J)=DOT                                               24140
      N=0                                                                  24150
      DO 25 K=1,75                                                         24160
   25 IF(LXF(K).EQ.J.AND.LYF(K).EQ.I) A(J)=POINT1                          24170
```

J03

```
      IF(A(J).EQ.POINT1)N=1                                                24180
      DO 30 L=N1,N3                                                        24190
   30 IF(LX(L).EQ.J.AND.LY(L).EQ.I) A(J)=POINT2                            24200
   35 IF(N.EQ.1.AND.A(J).EQ.POINT2)A(J)=STAR                               24210
   40 WRITE(6,105)A                                                        24220
      WRITE(6,110)                                                         24230
   45 CONTINUE                                                             24240
      WRITE(6,115)                                                         24250
  100 FORMAT(//15X," PLOT OF FITTED CURVE AND DATA POINTS (SCL.UNITS)"//   24260
     *1X,77(1H-) )                                                         24270
  101 FORMAT(/26X," RESPONSE SYSTEM NO.",I2//1X,77(1H-)  )                 24280
  105 FORMAT(2H I,75A1,1HI )                                               24290
  110 FORMAT (1X,77(1H-)/" O =OBSERVED POINTS   * =CALCULATED POINTS",     24300
     *"   X =CALCULATED AND OBSERVED POINTS"/ )                            24310
  115 FORMAT( " THE ACCURACY OF THE ABOVE PLOT(S) IS +-1/2 CHARACTER",     24320
     *" AND LINE DISTANCE."//  )                                           24330
      RETURN                                                               24340
      END                                                                  24350
C     ****************************************************************** 24360
      SUBROUTINE PLOT(X,Y,NPOINT,LINES)                                    24370
      DIMENSION X(100),Y(100),A(75),LX(100),LY(100),LIM1(10),LIM2(10)      24380
      COMMON /LXLY/LX,LY                                                   24390
      COMMON /B7/ ITABLE                                                   24400
      COMMON /B8/NFUNC                                                     24410
      COMMON /B9/LIM1,LIM2                                                 24420
      COMMON /B13/IGNORE                                                   24430
      DATA BLANK,POINT,DOT/1H ,1HO,1H./                                    24440
      N1=1                                                                 24450
      N2=2                                                                 24460
      N3=NPOINT                                                            24470
      N4=1                                                                 24480
```

```
      F(I)=FNEW(I)                                                      5480
      FSAVE(I)=F(I)                                                     5490
      IF(F(I)+STPNEW(I).GT.F2(I)) STPNEW(I)=-STPNEW(I)                  5500
 1651 STEP(I)=STPNEW(I)                                                 5510
      GO TO 1664                                                        5520
 1652 WRITE(6,6731) FUNC                                                5530
      GO TO 1655                                                        5540
 1653 IF(MAX.EQ.1) GO TO 1654                                           5550
      WRITE(6,6740) FUNC                                                5560
      CALL PLACE(F1,F2,FMIN,40,NPR,A)                                   5570
```

J01

```
      GO TO 1656                                                        5580
 1654 WRITE(6,6741) FUNC                                                5590
 1655 CALL PLACE(F1,F2,F,40,NPR,A)                                      5600
 1656 CONTINUE                                                          5610
      DO 1657 J=1,NPR                                                   5620
      IF(F1(J).EQ.F2(J)) GO TO 1657                                     5630
      N1=40*(J-1)+1                                                     5640
      N2=N1+39                                                          5650
      IF(MAX.EQ.1) WRITE(6,6730) J,F(J),(A(I),I=N1,N2)                  5660
      IF(MAX.NE.1) WRITE(6,6730) J,FMIN(J),(A(I),I=N1,N2)               5670
 1657 CONTINUE                                                          5680
      WRITE(6,6745)                                                     5690
 1660 WRITE(6,6748)                                                     5700
      CALL READ(1,AR)                                                   5710
      IF(IGO.EQ.2) GO TO 1001                                           5720
      IF(ANS.EQ.4HBACK) GO TO 1004                                      5730
      IF(ANS.EQ.3HYES) GO TO 1661                                       5740
      IF(ANS.EQ.2HNO.AND.MAX.EQ.1) GO TO 1663                           5750
      IF(ANS.EQ.2HNO) GO TO 1666                                        5760
      WRITE(6,1020)                                                     5770
      GO TO 1660                                                        5780
 1661 ANS11=3HYES                                                       5790
      ANS12=3HYES                                                       5800
      N11=NOP                                                           5810
      N12=NOP                                                           5820
      DO 1662 I=1,NOP                                                   5830
      NF(I)=I                                                           5840
      NT(I)=I                                                           5850
      F(I)=FNEW(I)                                                      5860
      FS(I)=F(I)                                                        5870
      FSAVE(I)=F(I)                                                     5880
      IF(F(I)+STPNEW(I).GT.F2(I)) STPNEW(I)=-STPNEW(I)                  5890
      STEP(I)=STPNEW(I)                                                 5900
      IF(F1(I).EQ.F2(I)) STEP(I)=0.                                     5910
 1662 STPS(I)=STEP(I)                                                   5920
      IF(MAX.NE.1) GO TO 1664                                           5930
 1663 NS=18                                                             5940
      GO TO 25                                                          5950
 1664 WRITE(6,6747)                                                     5960
      CALL READ(1,AR)                                                   5970
      IF(IGO.EQ.2) GO TO 1001                                           5980
      IF(ANS.EQ.4HBACK) GO TO 1004                                      5990
      IF(ANS.EQ.2HNO) GO TO 350                                         6000
      IF(ANS.NE.3HYES) GO TO 1665                                       6010
      NQ=22                                                             6020
      GO TO 1004                                                        6030
 1665 WRITE(6,1020)                                                     6040
      GO TO 1664                                                        6050
 1666 IF(NS.GT.NQ) GO TO 1006                                           6060
  170 NQ=20                                                             6070
      WRITE(6,675)                                                      6080
      CALL READ(1,AR)                                                   6090
      IF(IGO.EQ.2) GO TO 1001                                           6100
      IF(ANS.EQ.4HBACK) GO TO 1004                                      6110
      IF(ANS.EQ.2HNO) GO TO 171                                         6120
      IF(ANS.EQ.3HYES) GO TO 173                                        6130
      WRITE(6,1020)                                                     6140
      GO TO 170                                                         6150
  171 WRITE(6,680)                                                      6160
      CALL READ(1,AR)                                                   6170
      IF(IGO.EQ.2) GO TO 1001                                           6180
      IF(ANS.EQ.4HBACK) GO TO 1004                                      6190
```

K01

```
      IF(ANS.EQ.3HYES) NS=8                                             6200
      IF(ANS.EQ.3HYES) GO TO 9                                          6210
      IF(ANS.EQ.2HNO) GO TO 172                                         6220
      WRITE(6,1020)                                                     6230
      GO TO 171                                                         6240
  172 WRITE(6,685)                                                      6250
      STOP                                                              6260
  173 IF(NS.GT.NQ) GO TO 1006                                           6270
  175 NQ=21                                                             6280
      WRITE(6,690)                                                      6290
      CALL READ(1,AR)                                                   6300
      IF(IGO.EQ.2) GO TO 1001                                           6310
      IF(ANS.EQ.4HBACK) GO TO 1004                                      6320
      IF(ANS.EQ.3HYES) GO TO 178                                        6330
      IF(ANS.EQ.2HNO) GO TO 176                                         6340
      WRITE(6,1020)                                                     6350
      GO TO 175                                                         6360
  176 CONTINUE                                                          6370
      ANS11=3HYES                                                       6380
      ANS12=2HNO                                                        6390
      N11=NOP                                                           6400
      DO 177 I=1,NOP                                                    6410
      NF(I)=I                                                           6420
  177 FS(I)=FSAVE(I)                                                    6430
      GO TO 179                                                         6440
  178 ANS12=2HNO                                                        6450
      ANS11=3HYES                                                       6460
      N11=NOP                                                           6470
      DO 1780 I=1,NOP                                                   6480
      NF(I)=I                                                           6490
 1780 FS(I)=F(I)                                                        6500
```

```
      GO TO 1004                                                        14790
 1020 FORMAT(/," INPUT ERROR, YOUR ANSWER MUST BE ONE OF THE FOLLOWING"/  14800
     *" FOUR,   Y   N   B**   R**      - TRY AGAIN -"/ )                 14810
 1021 FORMAT(/" WHAT REQUEST NUMBER DO YOU WANT TO GO TO"// )            14820
 1035 FORMAT(" YOU FORGOT AN ESSENTIAL REQUEST (I.E. 8,9,10 OR 17)" )    14830
 1022 FORMAT(/," INPUT ERROR, THE ENTERED NUMBER IS NOT AMONG"/          14840
     *" THE REQUEST NUMBERS ABOVE.     - TRY AGAIN -"/)                  14850
 1023 FORMAT(" CONTINUE NOW AFTER CORRECTION FROM REQUEST NO."I3/)        14860
 1024 FORMAT(/," INPUT ERROR, THE ENTERED NUMBER IS NOT",                 14870
```

J02

```
     *" IN THE ALLOWED RANGE. - TRY AGAIN -"/ )                         14880
 1025 FORMAT(/," INPUT ERROR, YOUR INPUT DOES NOT AGREE WITH YOUR"/      14890
     *" INPUT ABOVE UNDER REQUEST NUMBER",I3,"     - TRY AGAIN -"/)      14900
 1026 FORMAT(/," INPUT ERROR, THE INPUT IS NOT IN THE RIGHT ORDER.  ",   14910
     *" - TRY AGAIN -" / )                                              14920
 1027 FORMAT(/ " INPUT ERROR, YOUR ANSWER MUST BE  OK  B  OR  R" ,       14930
     *"- TRY AGAIN -"/ )                                                14940
 1028 FORMAT(" INPUT ERROR, YOUR INPUT UNDER REQUEST NUMBER 6 DOES"/     14950
     *" NOT AGREE WITH THE INPUT UNDER REQUEST NUMBER 5")                14960
 1029 FORMAT(" INPUT ERROR,YOUR INPUT UNDER REQUEST NUMBER 9 DOES"/      14970
     *" NOT AGREE WITH THE INPUT UNDER REQUEST NUMBER 8")                14980
 1030 FORMAT(/" INPUT ERROR, THE INITIAL ESTIMATE OF PARAMETER NO.",     14990
     *I2," IS NOT WITHIN"/" ITS LIMITS. ENTER -BACK- IF YOU WANT TO",    15000
     *" MAKE CORRECTION OR -OK- IF YOU"/" WANT THE PROGRAM TO CHOOSE",   15010
     *" AN ACCEPTABLE INITIAL PARAMETER ESTIMATE" //  )                  15020
 1031 FORMAT(/" INPUT ERROR, THE STEP SIZE FOR PARAMETER NO.",I2,        15030
     *" DOES NOT AGREE WITH"/" THE LIMITS FOR THIS PARAMETER. ENTER",    15040
     *" -BACK- IF YOU WANT TO MAKE CORRECTION"/" OR -OK- IF YOU WANT",   15050
     *" THE PROGRAM TO CHOOSE AN ACCEPTABLE STEP SIZE" //  )             15060
 1032 FORMAT(" INPUT ERROR, YOU FORGOT TO ENTER OBSERVATION NUMBER",I3/  15070
     *" REENTER YOUR DATA STRICTLY ACCORDING TO FOLLOWING REQUEST "/)    15080
 1033 FORMAT( " INPUT ERROR,THE WEIGHT FOR THE OBSERVATION ",            15090
     *"IS NEGATIVE."/" REENTER THE LAST LINE IN CORRECT FORM."//)        15100
 1034 FORMAT(/" INPUT ERROR, THE WEIGHT FOR OBSERVATION NO.",I3,         15110
     *" (Y=",E10.4,") "/" IS NOT DEFINED BY THE WEIGHTING SCHEME NO.",   15120
     *I2  )                                                             15130
      END                                                               15140
C     ********************************************************************15150
      SUBROUTINE NELDR(F,STEP,NOP,FUNC,MAX,IPRINT,STOPCR,NLOOP,IQUAD,   15160
     *    SIMP,VAR,FUNCTN,IFAULT)                                       15170
      DOUBLE PRECISION DMEAN,DMEANS,DHCV,FUNC,FUNCS,H,HMIN,HMAX,        15180
     *HSTAR,HSTST,YMIN,AO,AVAL,BMAT,VC,TEMP,QMIN,PBAR                   15190
      DIMENSION F(20),STEP(20),AR(12),F1(20),F2(20)                     15200
      DIMENSION G(11,20),H(11),PBAR(20),PSTAR(20),PSTST(20),VARCOV(55)  15210
      DIMENSION AVAL(20),BMAT(55),PMIN(20),VC(55),VAR(10),TEMP(20)      15220
      EQUIVALENCE(PMIN,PSTAR),(AVAL,PBAR)                               15230
      COMMON /PARLIM/F1,F2                                              15240
      COMMON /CONSTR/LIMITS                                             15250
      COMMON/B3/VARCOV,VC,NEVAL1,NEVAL2,AMINQ                           15260
      COMMON /B4/ NIND                                                  15270
      COMMON /B5/NQ,NS,IGO                                              15280
      COMMON /B11/AR                                                    15290
      COMMON /B12/NPR                                                   15300
      COMMON /B14/NVIOL                                                 15310
      COMMON /B15/ SKIP                                                 15320
      LOGICAL SKIP                                                      15330
      A=1.                                                              15340
      B=0.5                                                             15350
      C=2.5                                                             15360
      LIMITS=1                                                          15370
      CALL FUNCTN(F,FUNC)                                               15380
      NEVAL=1                                                           15390
      IF(FUNC.GT.1D50) WRITE(6,1000)                                    15400
 1000 FORMAT(/" ONE OR MORE OF THE INITIAL PARAMETERS DOES NOT LIE"/    15410
     *" WITHIN THE SPECIFIED CONSTRAINED PARAMETER SPACE."/  )          15420
      IW=6                                                              15430
      IF(IPRINT)5,5,55                                                  15440
   55 WRITE(IW,100)IPRINT                                               15450
  100 FORMAT(" PROGRESS REPORT EVERY",I4," FUNCTION EVALUATIONS."//     15460
     *" EVAL.NO.  FUNC.VALUE              PARAMETERS" /)                15470
    5 IFAULT = 0                                                        15480
      IF(NOP.LE.0) IFAULT=3                                             15490
```

K02

```
      IF(NLOOP.LE.0) IFAULT=4                                           15500
      IF (IFAULT.NE.0) RETURN                                           15510
      NAP=0                                                             15520
      LOOP=0                                                            15530
      IFLAG=0                                                           15540
      DO 1 I=1,NOP                                                      15550
      IF(STEP(I).NE.0.0) NAP=NAP+1                                      15560
    1 CONTINUE                                                          15570
      IF(NAP.EQ.0)RETURN                                                15580
      MORE=0.20+((FLOAT(NAP+1))**2.5)                                   15590
      MORE=MIN0( 50,MORE)                                               15600
      MORE=MAX0(20,MORE)                                                15610
      DO 6 I=1,NOP                                                      15620
    6 G(1,I)=F(I)                                                       15630
      IROW=2                                                            15640
      DO 7 I=1,NOP                                                      15650
      IF(STEP(I).EQ.0.) GO TO 7                                         15660
      DO 9 J = 1,NOP                                                    15670
    9 G(IROW,J)=F(J)                                                    15680
      G(IROW,I)=G(IROW,I)+STEP(I)                                       15690
      IROW=IROW+1                                                       15700
    7 CONTINUE                                                          15710
      NP1=NAP+1                                                         15720
      NEVAL=0                                                           15730
      DO 10 I=1,NP1                                                     15740
      DO 11 J=1,NOP                                                     15750
   11 F(J)=G(I,J)                                                       15760
      CALL FUNCTN(F,H(I))                                               15770
      NEVAL=NEVAL+1                                                     15780
      IF(IPRINT)10,10,12                                                15790
   12 IF(NPR.LE.5)WRITE(IW,1010)NEVAL,H(I),(F(J),J=1,NPR)               15800
```

```
      IF(NZERO.EQ.50.AND.YMIN.GT.0.)NZERO=51                              24090
      IF(NZERO.EQ.1.AND.YMAX.LT.0.) NZERO=0                               24100
      DO 40 I=1,50                                                        24110
      DO 35 J=1,75                                                        24120
      A(J)=BLANK                                                          24130
      IF(I.EQ.NZERO)A(J)=DOT                                              24140
      N=0                                                                 24150
      DO 25 K=1,75                                                        24160
   25 IF(LXF(K).EQ.J.AND.LYF(K).EQ.I) A(J)=POINT1                         24170
```

J03

```
      IF(A(J).EQ.POINT1)N=1                                               24180
      DO 30 L=N1,N3                                                       24190
   30 IF(LX(L).EQ.J.AND.LY(L).EQ.I) A(J)=POINT2                           24200
   35 IF(N.EQ.1.AND.A(J).EQ.POINT2)A(J)=STAR                              24210
   40 WRITE(6,105)A                                                       24220
      WRITE(6,110)                                                        24230
   45 CONTINUE                                                            24240
      WRITE(6,115)                                                        24250
  100 FORMAT(//15X," PLOT OF FITTED CURVE AND DATA POINTS (SCL.UNITS)"//  24260
     *1X,77(1H-) )                                                        24270
  101 FORMAT(/26X," RESPONSE SYSTEM NO.",I2//1X,77(1H-)  )                24280
  105 FORMAT(2H I,75A1,1HI )                                              24290
  110 FORMAT (1X,77(1H-)/" O =OBSERVED POINTS   * =CALCULATED POINTS",    24300
     *"   X =CALCULATED AND OBSERVED POINTS"/ )                           24310
  115 FORMAT( " THE ACCURACY OF THE ABOVE PLOT(S) IS +-1/2 CHARACTER",    24320
     *" AND LINE DISTANCE."//  )                                          24330
      RETURN                                                              24340
      END                                                                 24350
C     ********************************************************************24360
      SUBROUTINE PLOT(X,Y,NPOINT,LINES)                                   24370
      DIMENSION X(100),Y(100),A(75),LX(100),LY(100),LIM1(10),LIM2(10)     24380
      COMMON /LXLY/LX,LY                                                  24390
      COMMON /B7/ ITABLE                                                  24400
      COMMON /B8/NFUNC                                                    24410
      COMMON /B9/LIM1,LIM2                                                24420
      COMMON /B13/IGNORE                                                  24430
      DATA BLANK,POINT,DOT/1H ,1HO,1H./                                   24440
      N1=1                                                                24450
      N2=2                                                                24460
      N3=NPOINT                                                           24470
      N4=1                                                                24480
      IF(ITABLE.EQ.0) N4=NFUNC                                            24490
      DO 25 II=1,N4                                                       24500
      IF(ITABLE.EQ.1) GO TO 1                                             24510
      IF(II.GT.1) WRITE(6,125)II                                          24520
      N1=LIM1(II)                                                         24530
      N2=N1+1                                                             24540
      N3=LIM2(II)                                                         24550
    1 XMIN=X(N1)                                                          24560
      XMAX=X(N1)                                                          24570
      YMIN=Y(N1)                                                          24580
      YMAX=Y(N1)                                                          24590
      IF(IGNORE.EQ.0) GO TO 3                                             24600
      DO 2 I=N2,N3                                                        24610
      IF(Y(I).EQ.0.) GO TO 2                                              24620
      XMIN=X(I)                                                           24630
      XMAX=X(I)                                                           24640
      YMIN=Y(I)                                                           24650
      YMAX=Y(I)                                                           24660
    2 CONTINUE                                                            24670
    3 CONTINUE                                                            24680
      DO 5 I=N2,N3                                                        24690
      IF(Y(I).EQ.0..AND.IGNORE.NE.0) GO TO 5                              24700
      IF(X(I).GT.XMAX) XMAX=X(I)                                          24710
      IF(X(I).LT.XMIN) XMIN=X(I)                                          24720
      IF(Y(I).GT.YMAX) YMAX=Y(I)                                          24730
      IF(Y(I).LT.YMIN) YMIN=Y(I)                                          24740
    5 CONTINUE                                                            24750
      SPANX=XMAX-XMIN                                                     24760
      SPANY=YMAX-YMIN                                                     24770
      IF(SPANX.EQ.0..OR.SPANY.EQ.0.)RETURN                                24780
      B=FLOAT(LINES-1)                                                    24790
```

K03

```
      DO 10 I=N1,N3                                                       24800
      IF(Y(I).EQ.0..AND.IGNORE.NE.0) GO TO 9                              24810
      LY(I)=IFIX(B*(YMAX-Y(I))/SPANY+.5)+1                                24820
      LX(I)=IFIX(74.*(X(I)-XMIN)/SPANX+.5)+1                              24830
      GO TO 10                                                            24840
    9 LY(I)=0                                                             24850
      LX(I)=0                                                             24860
   10 CONTINUE                                                            24870
      NZERO=IFIX(B*YMAX/SPANY+.5)+1                                       24880
      IF(NZERO.EQ.LINES.AND.YMIN.GT.0.)NZERO=LINES+1                      24890
      IF(NZERO.EQ.1.AND.YMAX.LT.0.) NZERO=0                               24900
      WRITE(6,110)                                                        24910
      DO 20 I=1,LINES                                                     24920
      DO 15 J=1,75                                                        24930
      A(J)=BLANK                                                          24940
      IF(I.EQ.NZERO)A(J)=DOT                                              24950
      DO 15 K=N1,N3                                                       24960
   15 IF(LX(K).EQ.J.AND.LY(K).EQ.I) A(J)=POINT                            24970
   20 WRITE(6,105)A                                                       24980
      WRITE(6,100)                                                        24990
      WRITE(6,115)YMAX,YMIN,XMAX,XMIN                                     25000
  100 FORMAT(1H ,77(1H-))                                                 25010
  105 FORMAT(2H I,75A1,1HI)                                               25020
  110 FORMAT(/1X,77(1H-))                                                 25030
  115 FORMAT(" YMAX=",E11.4,"  YMIN=",E11.4,"  XMAX=",E11.4,              25040
     *"  XMIN=",E11.4 )                                                   25050
      IF(ITABLE.EQ.0) GO TO 25                                            25060
      WRITE(6,120) NPOINT,(X(I),Y(I),I=N1,N3)                             25070
  120 FORMAT(/" THE COORDINATES OF THE",I3," POINTS ARE"//                25080
     *(3(" (",E9.3,",",E9.3,")",2X)))                                     25090
  125 FORMAT(/ 28X," RESPONSE SYSTEM NO.",I2 )                            25100
   25 CONTINUE                                                            25110
```

```
      NQ=22                                                              6020
      GO TO 1004                                                         6030
 1665 WRITE(6,1020)                                                      6040
      GO TO 1664                                                         6050
 1666 IF(NS.GT.NQ) GO TO 1006                                            6060
  170 NQ=20                                                             6070
      WRITE(6,675)                                                       6080
      CALL READ(1,AR)                                                    6090
      IF(IGO.EQ.2) GO TO 1001                                            6100
      IF(ANS.EQ.4HBACK) GO TO 1004                                       6110
      IF(ANS.EQ.2HNO) GO TO 171                                          6120
      IF(ANS.EQ.3HYES) GO TO 173                                         6130
      WRITE(6,1020)                                                      6140
      GO TO 170                                                          6150
  171 WRITE(6,680)                                                       6160
      CALL READ(1,AR)                                                    6170
      IF(IGO.EQ.2) GO TO 1001                                            6180
      IF(ANS.EQ.4HBACK) GO TO 1004                                       6190
```

K01

```
      IF(ANS.EQ.3HYES) NS=8                                              6200
      IF(ANS.EQ.3HYES) GO TO 9                                           6210
      IF(ANS.EQ.2HNO) GO TO 172                                          6220
      WRITE(6,1020)                                                      6230
      GO TO 171                                                          6240
  172 WRITE(6,685)                                                       6250
      STOP                                                               6260
  173 IF(NS.GT.NQ) GO TO 1006                                            6270
  175 NQ=21                                                             6280
      WRITE(6,690)                                                       6290
      CALL READ(1,AR)                                                    6300
      IF(IGO.EQ.2) GO TO 1001                                            6310
      IF(ANS.EQ.4HBACK) GO TO 1004                                       6320
      IF(ANS.EQ.3HYES) GO TO 178                                         6330
      IF(ANS.EQ.2HNO) GO TO 176                                          6340
      WRITE(6,1020)                                                      6350
      GO TO 175                                                          6360
  176 CONTINUE                                                           6370
      ANS11=3HYES                                                        6380
      ANS12=2HNO                                                         6390
      N11=NOP                                                            6400
      DO 177 I=1,NOP                                                     6410
      NF(I)=I                                                            6420
  177 FS(I)=FSAVE(I)                                                     6430
      GO TO 179                                                          6440
  178 ANS12=2HNO                                                         6450
      ANS11=3HYES                                                        6460
      N11=NOP                                                            6470
      DO 1780 I=1,NOP                                                    6480
      NF(I)=I                                                            6490
 1780 FS(I)=F(I)                                                         6500
  179 IF(NS.GT.NQ) GO TO 1006                                            6510
      WRITE(6,692)                                                       6520
      NQ=22                                                              6530
      GO TO 1004                                                         6540
  185 NQ=22                                                             6550
      WRITE(6,694)                                                       6560
      CALL READ(1,AR)                                                    6570
      IF(IGO.EQ.2) GO TO 1001                                            6580
      IF(ANS.EQ.3HYES.OR.ANS.EQ.4HBACK) GO TO 1004                       6590
      IF(ANS.EQ.2HNO) GO TO 186                                          6600
      WRITE(6,1020)                                                      6610
      GO TO 185                                                          6620
  186 NQ=18                                                             6630
      GO TO 131                                                          6640
  300 WRITE(6,663)                                                       6650
      CALL READ(1,AR)                                                    6660
      IF(IGO.EQ.2) GO TO 1001                                            6670
      IF(ANS.EQ.4HBACK) GO TO 1004                                       6680
      IF(ANS.NE.3HYES.AND.ANS.NE.2HNO) GO TO 301                         6690
      PLOTDP=ANS                                                         6700
      GO TO 302                                                          6710
  301 WRITE(6,1020)                                                      6720
      GO TO 300                                                          6730
  302 NCON=NOP-NEWNOP                                                    6740
      WRITE(6,750)                                                       6750
      WRITE(6,525)                                                       6760
      WRITE(6,752) NEWNOP,NCON,NOBS,NDUMMY,NFUNC,NIND                    6770
      DO 305 I=1,NOP                                                     6780
  305 WRITE(6,754)I,F1(I),F2(I),F(I),STEP(I)                             6790
      WRITE(6,756)                                                       6800
      NSAVE=1                                                            6810
```

L01

```
      DO 310 I=1,NOBS                                                    6820
      N=LABEL(I)                                                         6830
      IF(N.EQ.NSAVE+1) WRITE(6,813)N                                     6840
      NSAVE=N                                                            6850
      IF(NIND.GT.1) GO TO 308                                            6860
      IF(ANS16.EQ.3HYES) GO TO 307                                       6870
      WRITE(6,758)I,X(I),Y(I)                                            6880
      GO TO 310                                                          6890
  307 WRITE(6,760)I,X(I),Y(I),WY(I),WYN(I)                               6900
      GO TO 310                                                          6910
  308 IF(ANS16.EQ.3HYES) GO TO 309                                       6920
      WRITE(6,762)I,X(I),Z(I),Y(I)                                       6930
      GO TO 310                                                          6940
  309 WRITE(6,764)I,X(I),Z(I),Y(I),WY(I),WYN(I)                          6950
  310 CONTINUE                                                           6960
      WRITE(6,818)                                                       6970
      IF(NIND.LT.3) GO TO 312                                            6980
      WRITE(6,817)(I,I=3,9)                                              6990
      DO 311 I=1,NOBS                                                    7000
  311 WRITE(6,819)I,(XX(J,I),J=3,NIND)                                   7010
      WRITE(6,818)                                                       7020
  312 IF(PLOTDP.EQ.2HNO) GO TO 313                                       7030
      WRITE(6,765)                                                       7040
      CALL PLOT(X,Y,NOBS,50)                                             7050
```

```
      COMMON /B12/NPR                                                   15300
      COMMON /B14/NVIOL                                                 15310
      COMMON /B15/ SKIP                                                 15320
      LOGICAL SKIP                                                      15330
      A=1.                                                              15340
      B=0.5                                                             15350
      C=2.5                                                             15360
      LIMITS=1                                                          15370
      CALL FUNCTN(F,FUNC)                                               15380
      NEVAL=1                                                           15390
      IF(FUNC.GT.1D50) WRITE(6,1000)                                    15400
 1000 FORMAT(/" ONE OR MORE OF THE INITIAL PARAMETERS DOES NOT LIE"/    15410
     *" WITHIN THE SPECIFIED CONSTRAINED PARAMETER SPACE."/  )          15420
      IW=6                                                              15430
      IF(IPRINT)5,5,55                                                  15440
   55 WRITE(IW,100)IPRINT                                               15450
  100 FORMAT(" PROGRESS REPORT EVERY",I4," FUNCTION EVALUATIONS."//     15460
     *" EVAL.NO.  FUNC.VALUE              PARAMETERS" /)                15470
    5 IFAULT = 0                                                        15480
      IF(NOP.LE.0) IFAULT=3                                             15490
```

K02

```
      IF(NLOOP.LE.0) IFAULT=4                                           15500
      IF (IFAULT.NE.0) RETURN                                           15510
      NAP=0                                                             15520
      LOOP=0                                                            15530
      IFLAG=0                                                           15540
      DO 1 I=1,NOP                                                      15550
      IF(STEP(I).NE.0.0) NAP=NAP+1                                      15560
    1 CONTINUE                                                          15570
      IF(NAP.EQ.0)RETURN                                                15580
      MORE=0.20*((FLOAT(NAP+1))**2.5)                                   15590
      MORE=MIN0( 50,MORE)                                               15600
      MORE=MAX0(20,MORE)                                                15610
      DO 6 I=1,NOP                                                      15620
    6 G(1,I)=F(I)                                                       15630
      IROW=2                                                            15640
      DO 7 I=1,NOP                                                      15650
      IF(STEP(I).EQ.0.) GO TO 7                                         15660
      DO 9 J = 1,NOP                                                    15670
    9 G(IROW,J)=F(J)                                                    15680
      G(IROW,I)=G(IROW,I)+STEP(I)                                       15690
      IROW=IROW+1                                                       15700
    7 CONTINUE                                                          15710
      NP1=NAP+1                                                         15720
      NEVAL=0                                                           15730
      DO 10 I=1,NP1                                                     15740
      DO 11 J=1,NOP                                                     15750
   11 F(J)=G(I,J)                                                       15760
      CALL FUNCTN(F,H(I))                                               15770
      NEVAL=NEVAL+1                                                     15780
      IF(IPRINT)10,10,12                                                15790
   12 IF(NPR.LE.5)WRITE(IW,1010)NEVAL,H(I),(F(J),J=1,NPR)               15800
 1010 FORMAT(1X,I4,D16.8,1X,5E11.5)                                     15810
      IF(NPR.GT.5) WRITE(IW,101)NEVAL,H(I),(F(J),J=1,NPR)               15820
  101 FORMAT(1X,I4,D16.8,1X,5E11.5/(22X,5E11.5))                        15830
   10 CONTINUE                                                          15840
   45 LOOP=LOOP+1                                                       15850
      IMAX=1                                                            15860
      IMIN=1                                                            15870
      DO 13 I=2,NP1                                                     15880
      IF(H(I)-H(IMAX))15,15,14                                          15890
   14 IMAX=I                                                            15900
   15 IF(H(I)-H(IMIN))16,13,13                                          15910
   16 IMIN = I                                                          15920
   13 CONTINUE                                                          15930
      HMAX = H(IMAX)                                                    15940
      HMIN = H(IMIN)                                                    15950
      DO 17 I=1,NOP                                                     15960
   17 PBAR(I)=0.0                                                       15970
      DO 18 I=1,NP1                                                     15980
      IF(I-IMAX)19,18,19                                                15990
   19 DO 20 J=1,NOP                                                     16000
   20 PBAR(J) = PBAR(J)+G(I,J)                                          16010
   18 CONTINUE                                                          16020
      DO 602 J=1,NOP                                                    16030
  602 PBAR(J) = PBAR(J)/NAP                                             16040
      DO 21 I=1,NOP                                                     16050
   21 PSTAR(I)=A*(PBAR(I)-G(IMAX,I))+PBAR(I)                            16060
      CALL FUNCTN (PSTAR,HSTAR)                                         16070
      NEVAL=NEVAL+1                                                     16080
      IF(IPRINT)57,57,56                                                16090
   56 IF(NPR.LE.5.AND.MOD(NEVAL,IPRINT).EQ.0) WRITE(IW,1010)            16100
     *NEVAL,HSTAR,(PSTAR(J),J=1,NPR)                                    16110
```

L02

```
      IF(NPR.GT.5.AND.MOD(NEVAL,IPRINT).EQ.0) WRITE(IW,101)             16120
     *NEVAL,HSTAR,(PSTAR(J),J=1,NPR)                                    16130
   57 IF(HSTAR-HMIN)22,23,23                                            16140
   22 DO 24 I=1,NOP                                                     16150
   24 PSTST(I)=C*(PSTAR(I)-PBAR(I))+PBAR(I)                             16160
      CALL FUNCTN (PSTST,HSTST)                                         16170
      NEVAL=NEVAL+1                                                     16180
      IF(IPRINT)60,60,59                                                16190
   59 IF(NPR.LE.5.AND.MOD(NEVAL,IPRINT).EQ.0) WRITE(IW,1010)            16200
     *NEVAL,HSTST,(PSTST(J),J=1,NPR)                                    16210
      IF(NPR.GT.5.AND.MOD(NEVAL,IPRINT).EQ.0) WRITE(IW,101)             16220
     *NEVAL,HSTST,(PSTST(J),J=1,NPR)                                    16230
   60 IF(HSTST-HMIN)25,26,26                                            16240
   25 DO 27 I=1,NOP                                                     16250
      IF(STEP(I).NE.0.0) G(IMAX,I)=PSTST(I)                             16260
   27 CONTINUE                                                          16270
      H(IMAX)=HSTST                                                     16280
      GO TO 41                                                          16290
   23 DO 28 I=1,NP1                                                     16300
      IF(I-IMAX)29,28,29                                                16310
   29 IF(HSTAR-H(I))26,28,28                                            16320
```

```
      IF(IGNORE.EQ.0) GO TO 3                                          24600
      DO 2 I=N2,N3                                                     24610
      IF(Y(I).EQ.0.) GO TO 2                                           24620
      XMIN=X(I)                                                        24630
      XMAX=X(I)                                                        24640
      YMIN=Y(I)                                                        24650
      YMAX=Y(I)                                                        24660
    2 CONTINUE                                                         24670
    3 CONTINUE                                                         24680
      DO 5 I=N2,N3                                                     24690
      IF(Y(I).EQ.0..AND.IGNORE.NE.0) GO TO 5                           24700
      IF(X(I).GT.XMAX) XMAX=X(I)                                       24710
      IF(X(I).LT.XMIN) XMIN=X(I)                                       24720
      IF(Y(I).GT.YMAX) YMAX=Y(I)                                       24730
      IF(Y(I).LT.YMIN) YMIN=Y(I)                                       24740
    5 CONTINUE                                                         24750
      SPANX=XMAX-XMIN                                                  24760
      SPANY=YMAX-YMIN                                                  24770
      IF(SPANX.EQ.0..OR.SPANY.EQ.0.)RETURN                             24780
      B=FLOAT(LINES-1)                                                 24790
```

K03

```
      DO 10 I=N1,N3                                                    24800
      IF(Y(I).EQ.0..AND.IGNORE.NE.0) GO TO 9                           24810
      LY(I)=IFIX(B*(YMAX-Y(I))/SPANY+.5)+1                             24820
      LX(I)=IFIX(74.*(X(I)-XMIN)/SPANX+.5)+1                           24830
      GO TO 10                                                         24840
    9 LY(I)=0                                                          24850
      LX(I)=0                                                          24860
   10 CONTINUE                                                         24870
      NZERO=IFIX(B*YMAX/SPANY+.5)+1                                    24880
      IF(NZERO.EQ.LINES.AND.YMIN.GT.0.)NZERO=LINES+1                   24890
      IF(NZERO.EQ.1.AND.YMAX.LT.0.) NZERO=0                            24900
      WRITE(6,110)                                                     24910
      DO 20 I=1,LINES                                                  24920
      DO 15 J=1,75                                                     24930
      A(J)=BLANK                                                       24940
      IF(I.EQ.NZERO)A(J)=DOT                                           24950
      DO 15 K=N1,N3                                                    24960
   15 IF(LX(K).EQ.J.AND.LY(K).EQ.I) A(J)=POINT                         24970
   20 WRITE(6,105)A                                                    24980
      WRITE(6,100)                                                     24990
      WRITE(6,115)YMAX,YMIN,XMAX,XMIN                                  25000
  100 FORMAT(1H ,77(1H-))                                              25010
  105 FORMAT(2H I,75A1,1HI)                                            25020
  110 FORMAT(/1X,77(1H-))                                              25030
  115 FORMAT(" YMAX=",E11.4," YMIN=",E11.4," XMAX=",E11.4,             25040
     *" XMIN=",E11.4 )                                                 25050
      IF(ITABLE.EQ.0) GO TO 25                                         25060
      WRITE(6,120) NPOINT,(X(I),Y(I),I=N1,N3)                          25070
  120 FORMAT(/" THE COORDINATES OF THE",I3," POINTS ARE"//              25080
     *(3(" (",E9.3,",",E9.3,")",2X)))                                  25090
  125 FORMAT(/ 28X," RESPONSE SYSTEM NO.",I2 )                         25100
   25 CONTINUE                                                         25110
      RETURN                                                           25120
      END                                                              25130
C     ******************************************************************25140
      SUBROUTINE KOLMIR(Z,N,DN,CRIT,IPRINT)                            25150
      DIMENSION X(100),Z(100),DNCRIT(30)                               25160
      DATA DNCRIT /0.,0.,0.,.381,.337,.319,.300,.285,.271,.258,.249,   25170
     *.242,.234,.227,.220,.213,.206,.200,.195,.190,.188,186,.184,.182, 25180
     *.180,.176,.172,.168,.164,.161 /                                  25190
      IF(N.LT.31) CRIT=DNCRIT(N)                                       25200
      IF(N.GT.30) CRIT=.886/SQRT(FLOAT(N))                             25210
      SX=0.                                                            25220
      SXX=0.                                                           25230
      DO 1 I=1,N                                                       25240
      X(I)=Z(I)                                                        25250
      SX=SX+X(I)                                                       25260
    1 SXX=SXX + X(I)*X(I)                                              25270
      AN=N                                                             25280
      BAR=SX/AN                                                        25290
      SIGMA=SQRT((SXX-AN*BAR*BAR)/FLOAT(N-1))                          25300
      IF(SIGMA.EQ.0.)RETURN                                            25310
      DO 6 I=2,N                                                       25320
      IF(X(I)-X(I-1))2,6,6                                             25330
    2 TEMP=X(I)                                                        25340
      IM=I-1                                                           25350
      DO 4 J=1,IM                                                      25360
      L=I-J                                                            25370
      IF(TEMP-X(L))3,5,5                                               25380
    3 X(L+1)=X(L)                                                      25390
    4 CONTINUE                                                         25400
      X(1)=TEMP                                                        25410
```

L03

```
      GO TO 6                                                          25420
    5 X(L+1)=TEMP                                                      25430
    6 CONTINUE                                                         25440
      AN=N                                                             25450
      DN=0.                                                            25460
      DO 7 I=1,N                                                       25470
      XT=(X(I)-BAR)/SIGMA                                              25480
      FSTAR=SDF(XT)                                                    25490
      D1=ABS(FSTAR-FLOAT(I-1)/AN)                                      25500
      D2=ABS(FSTAR-FLOAT(I)/AN)                                        25510
    7 DN=AMAX1(DN,D1,D2)                                               25520
      IF(IPRINT.EQ.0) RETURN                                           25530
      WRITE(6,100) DN,CRIT                                             25540
      IF(DN.GT.CRIT) WRITE(6,101)                                      25550
  100 FORMAT(" KOLMOGOROV-SMIRNOV STATISTIC FOR TEST OF NORMALITY",    25560
     *" OF THE RESIDUALS =",F7.4/" (CRIT.VALUE =",F6.3,                25570
     *" ,P.LT.0.05--REF. H.W.LILLIEFORS,J.A.S.A.,62,P399,1967)"/ )      25580
  101 FORMAT(" THE STATISTIC INDICATES THAT THE RESIDUALS ARE NOT",     25590
     *" NORMALLY DISTRIBUTED."/ )                                      25600
      RETURN                                                           25610
      END                                                              25620
C     ******************************************************************25630
```

```
      IF(IGO.EQ.2) GO TO 1001                                          6670
      IF(ANS.EQ.4HBACK) GO TO 1004                                     6680
      IF(ANS.NE.3HYES.AND.ANS.NE.2HNO) GO TO 301                       6690
      PLOTDP=ANS                                                       6700
      GO TO 302                                                        6710
  301 WRITE(6,1020)                                                    6720
      GO TO 300                                                        6730
  302 NCON=NOP-NEWNOP                                                  6740
      WRITE(6,750)                                                     6750
      WRITE(6,525)                                                     6760
      WRITE(6,752) NEWNOP,NCON,NOBS,NDUMMY,NFUNC,NIND                  6770
      DO 305 I=1,NOP                                                   6780
  305 WRITE(6,754)I,F1(I),F2(I),F(I),STEP(I)                           6790
      WRITE(6,756)                                                     6800
      NSAVE=1                                                          6810
```

L01

```
      DO 310 I=1,NOBS                                                  6820
      N=LABEL(I)                                                       6830
      IF(N.EQ.NSAVE+1) WRITE(6,813)N                                   6840
      NSAVE=N                                                          6850
      IF(NIND.GT.1) GO TO 308                                          6860
      IF(ANS16.EQ.3HYES) GO TO 307                                     6870
      WRITE(6,758)I,X(I),Y(I)                                          6880
      GO TO 310                                                        6890
  307 WRITE(6,760)I,X(I),Y(I),WY(I),WYN(I)                             6900
      GO TO 310                                                        6910
  308 IF(ANS16.EQ.3HYES) GO TO 309                                     6920
      WRITE(6,762)I,X(I),Z(I),Y(I)                                     6930
      GO TO 310                                                        6940
  309 WRITE(6,764)I,X(I),Z(I),Y(I),WY(I),WYN(I)                        6950
  310 CONTINUE                                                         6960
      WRITE(6,818)                                                     6970
      IF(NIND.LT.3) GO TO 312                                          6980
      WRITE(6,817)(I,I=3,9)                                            6990
      DO 311 I=1,NOBS                                                  7000
  311 WRITE(6,819)I,(XX(J,I),J=3,NIND)                                 7010
      WRITE(6,818)                                                     7020
  312 IF(PLOTDP.EQ.2HNO) GO TO 313                                     7030
      WRITE(6,765)                                                     7040
      CALL PLOT(X,Y,NOBS,50)                                           7050
      IF(NIND.GT.1) WRITE(6,767)                                       7060
      IF(NIND.GT.1)CALL PLOT(Z,Y,NOBS,50)                              7070
  313 WRITE(6,766)MAX,STOPCR                                           7080
      IF(IQUAD.EQ.1) WRITE(6,768)SIMP                                  7090
      IF(ANS4.EQ.3HYES) WRITE(6,770)                                   7100
      IF(PLOTRS.EQ.3HYES) WRITE(6,772)                                 7110
      IF(IPRINT.EQ.-1) WRITE(6,774)                                    7120
      IF(IPRINT.EQ.0) WRITE(6,776)                                     7130
      IF(IPRINT.GT.0) WRITE(6,778)                                     7140
      IF(PLT.EQ.3HYES)WRITE(6,780)                                     7150
      IF(ANS16.EQ.3HYES.AND.IWGHT.GT.1) WRITE(6,781)IWGHT,ANS16B       7160
      WRITE(6,790)                                                     7170
      GO TO 158                                                        7180
  350 NPR=NOP                                                          7190
      DO 3500 I=1,NOP                                                  7200
      N=NOP+1-I                                                        7210
      IF(STEP(N).NE.0.) GO TO 3501                                     7220
 3500 NPR=NPR-1                                                        7230
 3501 IF(MAX.EQ.1) GO TO 160                                           7240
      NRUN=NRUN+1                                                      7250
      NVIOL=0                                                          7260
      IF(IPRINT.EQ.-1) GO TO 351                                       7270
      WRITE(6,790)                                                     7280
      WRITE(6,525)                                                     7290
      WRITE(6,789)                                                     7300
  351 CALL NELDR(F,STEP,NOP,FUNC,MAX,IPRINT,STOPCR,NLOOP,IQUAD,SIMP,   7310
     *VAR,LSQ,IFAULT)                                                  7320
      SKIP=.FALSE.                                                     7330
      IF(IFAULT.EQ.5.AND.IGO.EQ.2) GO TO 1001                          7340
      IF(IFAULT.EQ.5.AND.AR(1).EQ.4HBACK) GO TO 1004                   7350
      LIMITS=0                                                         7360
      DO 3510 I=1,NOP                                                  7370
 3510 FMIN(I)=F(I)                                                     7380
      IF(IFAULT.EQ.3) GO TO 3514                                       7390
      IF(IFAULT.NE.1.AND.IFAULT.NE.2) GO TO 352                        7400
      ANS12=2HNO                                                       7410
      IF(IFAULT.EQ.2) GO TO 3511                                       7420
      WRITE(6,791)                                                     7430
```

M01

```
      NQ=18                                                            7440
      GO TO 1004                                                       7450
 3511 IFLAG=1                                                          7460
      DO 3512 I=1,NOP                                                  7470
      IF(F2(I).EQ.F1(I)) GO TO 3512                                    7480
      DEL=(F2(I)-F1(I))*0.001                                          7490
      IF(F(I)-F1(I).GT.DEL.AND.F2(I)-F(I).GT.DEL) GO TO 3512           7500
      IFLAG=2                                                          7510
      GO TO 3601                                                       7520
 3512 CONTINUE                                                         7530
      GO TO 3602                                                       7540
 3513 IF(IFLAG.EQ.2) WRITE(6,793)                                      7550
      IF(IFLAG.EQ.1) WRITE(6,794)                                      7560
      WRITE(6,8401)STOPCR                                              7570
      WRITE(6,8402)SIMP                                                7580
      IF(IFLAG.EQ.1) WRITE(6,795)                                      7590
      IF(IFAULT.NE.2) GO TO 3514                                       7600
 3520 WRITE(6,8404)                                                    7610
      CALL READ(1,AR)                                                  7620
      IF(IGO.EQ.2) GO TO 1001                                          7630
      IF(ANS.EQ.4HBACK.OR.ANS.EQ.2HNO) GO TO 3514                      7640
      IF(ANS.EQ.2HOK.OR.ANS.EQ.3HYES) GO TO 3521                       7650
      WRITE(6,1020)                                                    7660
      GO TO 3520                                                       7670
 3521 IQUAD=0                                                          7680
      GO TO 352                                                        7690
```

```
      DO 17 I=1,NOP                                                    15960
   17 PBAR(I)=0.0                                                      15970
      DO 18 I=1,NP1                                                    15980
      IF(I-IMAX)19,18,19                                               15990
   19 DO 20 J=1,NOP                                                    16000
   20 PBAR(J) = PBAR(J)+G(I,J)                                         16010
   18 CONTINUE                                                         16020
      DO 602 J=1,NOP                                                   16030
  602 PBAR(J) = PBAR(J)/NAP                                            16040
      DO 21 I=1,NOP                                                    16050
   21 PSTAR(I)=A*(PBAR(I)-G(IMAX,I))+PBAR(I)                           16060
      CALL FUNCTN (PSTAR,HSTAR)                                        16070
      NEVAL=NEVAL+1                                                    16080
      IF(IPRINT)57,57,56                                               16090
   56 IF(NPR.LE.5.AND.MOD(NEVAL,IPRINT).EQ.0) WRITE(IW,1010)           16100
     *NEVAL,HSTAR,(PSTAR(J),J=1,NPR)                                   16110
```

L02

```
      IF(NPR.GT.5.AND.MOD(NEVAL,IPRINT).EQ.0) WRITE(IW,101)            16120
     *NEVAL,HSTAR,(PSTAR(J),J=1,NPR)                                   16130
   57 IF(HSTAR-HMIN)22,23,23                                           16140
   22 DO 24 I=1,NOP                                                    16150
   24 PSTST(I)=C*(PSTAR(I)-PBAR(I))+PBAR(I)                            16160
      CALL FUNCTN (PSTST,HSTST)                                        16170
      NEVAL=NEVAL+1                                                    16180
      IF(IPRINT)60,60,59                                               16190
   59 IF(NPR.LE.5.AND.MOD(NEVAL,IPRINT).EQ.0) WRITE(IW,1010)           16200
     *NEVAL,HSTST,(PSTST(J),J=1,NPR)                                   16210
      IF(NPR.GT.5.AND.MOD(NEVAL,IPRINT).EQ.0) WRITE(IW,101)            16220
     *NEVAL,HSTST,(PSTST(J),J=1,NPR)                                   16230
   60 IF(HSTST-HMIN)25,26,26                                           16240
   25 DO 27 I=1,NOP                                                    16250
      IF(STEP(I).NE.0.0) G(IMAX,I)=PSTST(I)                            16260
   27 CONTINUE                                                         16270
      H(IMAX)=HSTST                                                    16280
      GO TO 41                                                         16290
   23 DO 28 I=1,NP1                                                    16300
      IF(I-IMAX)29,28,29                                               16310
   29 IF(HSTAR-H(I))26,28,28                                           16320
   28 CONTINUE                                                         16330
      IF(HSTAR-HMAX)30,30,31                                           16340
   30 DO 32 I=1,NOP                                                    16350
      IF(STEP(I).NE.0.0) G(IMAX,I)=PSTAR(I)                            16360
   32 CONTINUE                                                         16370
      HMAX=HSTAR                                                       16380
      H(IMAX)=HSTAR                                                    16390
   31 DO 33 I=1,NOP                                                    16400
   33 PSTST(I)=B*G(IMAX,I)+(1.0-B)*PBAR(I)                             16410
      CALL FUNCTN(PSTST,HSTST)                                         16420
      NEVAL=NEVAL+1                                                    16430
      IF(IPRINT) 63,63,62                                              16440
   62 IF(NPR.LE.5.AND.MOD(NEVAL,IPRINT).EQ.0)WRITE(IW,1010)            16450
     *NEVAL,HSTST,(PSTST(J),J=1,NPR)                                   16460
      IF(NPR.GT.5.AND.MOD(NEVAL,IPRINT).EQ.0)WRITE(IW,101)             16470
     *NEVAL,HSTST,(PSTST(J),J=1,NPR)                                   16480
   63 IF(HSTST-HMAX)35,35,34                                           16490
   35 DO 36 I=1,NOP                                                    16500
      IF(STEP(I).NE.0.0) G(IMAX,I)=PSTST(I)                            16510
   36 CONTINUE                                                         16520
      H(IMAX)=HSTST                                                    16530
      GO TO 41                                                         16540
   34 DO 38 I=1,NP1                                                    16550
      IF(I.EQ.IMIN) GO TO 38                                           16560
      DO 39 J=1,NOP                                                    16570
      IF(STEP(J).NE.0.0) G(I,J)=(G(I,J)+G(IMIN,J))/2.0                 16580
   39 F(J) = G(I,J)                                                    16590
      CALL FUNCTN (F,H(I))                                             16600
      NEVAL=NEVAL+1                                                    16610
      IF(IPRINT)38,38,65                                               16620
   65 IF(NPR.LE.5.AND.MOD(NEVAL,IPRINT).EQ.0)WRITE(IW,1010)            16630
     *NEVAL,H(I),(F(J),J=1,NPR)                                        16640
      IF(NPR.GT.5.AND.MOD(NEVAL,IPRINT).EQ.0)WRITE(IW,101)             16650
     *NEVAL,H(I),(F(J),J=1,NPR)                                        16660
   38 CONTINUE                                                         16670
      GO TO 41                                                         16680
   26 DO 40 I=1,NOP                                                    16690
      IF(STEP(I).NE.0.0) G(IMAX,I)=PSTAR(I)                            16700
   40 CONTINUE                                                         16710
      H(IMAX)=HSTAR                                                    16720
   41 IF(NEVAL-MAX.GT.0) GO TO 46                                      16730
```

M02

```
      IF(IFLAG.NE.1) GO TO 4100                                        16740
      IF(LOOP-MAX0(NLOOP,MORE))45,46,45                                16750
 4100 IF(LOOP-NLOOP)45,46,45                                           16760
   46 DHCV=0D0                                                         16770
      DMEAN=0D0                                                        16780
      DO 42 I=1,NP1                                                    16790
      IF(H(I).GE.1D50) GO TO 45                                        16800
   42 DMEAN=DMEAN+H(I)                                                 16810
      DMEAN=DMEAN/FLOAT(NP1)                                           16820
      DO 601 I=1,NP1                                                   16830
  601 DHCV=DHCV+(H(I)-DMEAN)*(H(I)-DMEAN)                              16840
      DHCV=1D2*DSQRT(DHCV/FLOAT(NP1))/DMEAN                            16850
      DO 53 I=1,NOP                                                    16860
      IF(STEP(I).EQ.0.0) GO TO 53                                      16870
      F(I)=0.0                                                         16880
      DO 54 J=1,NP1                                                    16890
   54 F(I)=F(I)+G(J,I)                                                 16900
      F(I)=F(I)/NP1                                                    16910
   53 CONTINUE                                                         16920
      CALL FUNCTN (F,FUNC)                                             16930
      NEVAL=NEVAL+1                                                    16940
      IF(IPRINT.LE.0) GO TO 700                                        16950
      IF(NPR.LE.5.AND.MOD(NEVAL,IPRINT).EQ.0)WRITE(IW,1010)            16960
     *NEVAL,FUNC,(F(J),J=1,NPR)                                        16970
      IF(NPR.GT.5.AND.MOD(NEVAL,IPRINT).EQ.0)WRITE(IW,101)             16980
```

```
      SX=SX+X(I)                                                        25260
    1 SXX=SXX + X(I)*X(I)                                               25270
      AN=N                                                              25280
      BAR=SX/AN                                                         25290
      SIGMA=SQRT((SXX-AN*BAR*BAR)/FLOAT(N-1))                           25300
      IF(SIGMA.EQ.0.)RETURN                                             25310
      DO 6 I=2,N                                                        25320
      IF(X(I)-X(I-1))2,6,6                                              25330
    2 TEMP=X(I)                                                         25340
      IM=I-1                                                            25350
      DO 4 J=1,IM                                                       25360
      L=I-J                                                             25370
      IF(TEMP-X(L))3,5,5                                                25380
    3 X(L+1)=X(L)                                                       25390
    4 CONTINUE                                                          25400
      X(1)=TEMP                                                         25410
```

L03

```
      GO TO 6                                                           25420
    5 X(L+1)=TEMP                                                       25430
    6 CONTINUE                                                          25440
      AN=N                                                              25450
      DN=0.                                                             25460
      DO 7 I=1,N                                                        25470
      XT=(X(I)-BAR)/SIGMA                                               25480
      FSTAR=SDF(XT)                                                     25490
      D1=ABS(FSTAR-FLOAT(I-1)/AN)                                       25500
      D2=ABS(FSTAR-FLOAT(I)/AN)                                         25510
    7 DN=AMAX1(DN,D1,D2)                                                25520
      IF(IPRINT.EQ.0) RETURN                                            25530
      WRITE(6,100) DN,CRIT                                              25540
      IF(DN.GT.CRIT) WRITE(6,101)                                       25550
  100 FORMAT(" KOLMOGOROV-SMIRNOV STATISTIC FOR TEST OF NORMALITY",     25560
     *" OF THE RESIDUALS =",F7.4/" (CRIT.VALUE =",F6.3,                 25570
     *" ,P.LT.0.05--REF. H.W.LILLIEFORS,J.A.S.A.,62,P399,1967)"/  )      25580
  101 FORMAT(" THE STATISTIC INDICATES THAT THE RESIDUALS ARE NOT",      25590
     *" NORMALLY DISTRIBUTED."/  )                                      25600
      RETURN                                                            25610
      END                                                               25620
C     ******************************************************************25630
      FUNCTION SEQPRO(E,NE)                                             25640
      DIMENSION E(100)                                                  25650
      N=0                                                               25660
      NU=1                                                              25670
      NFU=0                                                             25680
      IF(E(1).GT.0.)N=1                                                 25690
      DO 5 J=2,NE                                                       25700
      IF(E(J).GT.0.)N=N+1                                               25710
    5 IF(E(J).LT.0..AND.E(J-1).GT.0.                                    25720
     *.OR.E(J).GT.0..AND.E(J-1).LT.0.)NU=NU+1                           25730
      IF(NU.EQ.1) GO TO 20                                              25740
      IF(N.LT.NE-N)N=NE-N                                               25750
      M=NE-N                                                            25760
      MM=M-1                                                            25770
      NN=N-1                                                            25780
      IF(NE.GT.25) GO TO 25                                             25790
      DO 15 I=2,NU                                                      25800
      IF(MOD(I,2).EQ.0) GO TO 10                                        25810
      KK=(I+1)/2-1                                                      25820
      NFU=NFU+NBC(MM,KK)*NBC(NN,KK-1)+NBC(MM,KK-1)*NBC(NN,KK)           25830
      GO TO 15                                                          25840
   10 KK=I/2-1                                                          25850
      NFU=NFU+2*NBC(MM,KK)*NBC(NN,KK)                                   25860
   15 CONTINUE                                                          25870
      SEQPRO=FLOAT(NFU)/FLOAT(NBC(NE,M))                                25880
      RETURN                                                            25890
   20 SEQPRO=1.                                                         25900
      RETURN                                                            25910
   25 AM=M                                                              25920
      AN=N                                                              25930
      BAR=2.*AM*AN/(AM+AN)+1.                                           25940
      SIGMA=(BAR-1.)*(BAR-2.)/(AM+AN-1.)                                25950
      Z= (FLOAT(NU)+.5-BAR)/SIGMA                                       25960
      SEQPRO=SDF(Z)                                                     25970
      RETURN                                                            25980
      END                                                               25990
C     ******************************************************************26000
      FUNCTION NBC(N,M)                                                 26010
      DOUBLE PRECISION NC,ND,NE,NF                                      26020
      IF(M.GT.N) GO TO 6                                                26030
```

M03

```
      IF(N.EQ.M.OR.M.EQ.0) GO TO 5                                      26040
      IF(M.EQ.1.OR.N-M.EQ.1) GO TO 4                                    26050
      IF(2*M-N.GT.0) GO TO 1                                            26060
      NC=N-M                                                            26070
      ND=NC+1                                                           26080
      K=M-1                                                             26090
      GO TO 2                                                           26100
    1 NC=M                                                              26110
      ND=M+1                                                            26120
      K=N-M-1                                                           26130
    2 NE=1                                                              26140
      DO 3 J=1,K                                                        26150
      NF=J+1                                                            26160
      ND=ND*(NC+NF)                                                     26170
    3 NE=NE*NF                                                          26180
      NBC=ND/NE                                                         26190
      RETURN                                                            26200
    4 NBC=N                                                             26210
      RETURN                                                            26220
    5 NBC=1                                                             26230
      RETURN                                                            26240
    6 NBC=0                                                             26250
      RETURN                                                            26260
      END                                                               26270
C     ******************************************************************26280
      FUNCTION SDF(X)                                                   26290
```

```
      IF(IFAULT.EQ.5.AND.IGO.EQ.2) GO TO 1001                          7340
      IF(IFAULT.EQ.5.AND.AR(1).EQ.4HBACK) GO TO 1004                   7350
      LIMITS=0                                                         7360
      DO 3510 I=1,NOP                                                  7370
 3510 FMIN(I)=F(I)                                                     7380
      IF(IFAULT.EQ.3) GO TO 3514                                       7390
      IF(IFAULT.NE.1.AND.IFAULT.NE.2) GO TO 352                        7400
      ANS12=2HNO                                                       7410
      IF(IFAULT.EQ.2) GO TO 3511                                       7420
      WRITE(6,791)                                                     7430
```

M01

```
      NQ=18                                                            7440
      GO TO 1004                                                       7450
 3511 IFLAG=1                                                          7460
      DO 3512 I=1,NOP                                                  7470
      IF(F2(I).EQ.F1(I)) GO TO 3512                                    7480
      DEL=(F2(I)-F1(I))*0.001                                          7490
      IF(F(I)-F1(I).GT.DEL.AND.F2(I)-F(I).GT.DEL) GO TO 3512           7500
      IFLAG=2                                                          7510
      GO TO 3601                                                       7520
 3512 CONTINUE                                                         7530
      GO TO 3602                                                       7540
 3513 IF(IFLAG.EQ.2) WRITE(6,793)                                      7550
      IF(IFLAG.EQ.1) WRITE(6,794)                                      7560
      WRITE(6,8401)STOPCR                                              7570
      WRITE(6,8402)SIMP                                                7580
      IF(IFLAG.EQ.1) WRITE(6,795)                                      7590
      IF(IFAULT.NE.2) GO TO 3514                                       7600
 3520 WRITE(6,8404)                                                    7610
      CALL READ(1,AR)                                                  7620
      IF(IGO.EQ.2) GO TO 1001                                          7630
      IF(ANS.EQ.4HBACK.OR.ANS.EQ.2HNO) GO TO 3514                      7640
      IF(ANS.EQ.2HOK.OR.ANS.EQ.3HYES) GO TO 3521                       7650
      WRITE(6,1020)                                                    7660
      GO TO 3520                                                       7670
 3521 IQUAD=0                                                          7680
      GO TO 352                                                        7690
 3514 NQ=18                                                            7700
      GO TO 1004                                                       7710
  352 SSRES=0.                                                         7720
      SUMRES=0.                                                        7730
      WSUMRS=0.                                                        7740
      SSY=0.                                                           7750
      SY=0.                                                            7760
      SSYC=0.                                                          7770
      SYC=0.                                                           7780
      SYCY=0.                                                          7790
      IFLAG=0                                                          7800
      DO 355 II=1,NFUNC                                                7810
      RSS(II)=0.                                                       7820
      WRSS(II)=0.                                                      7830
      N1=LIM1(II)                                                      7840
      N2=LIM2(II)                                                      7850
      DO 355 I=N1,N2                                                   7860
      DO 353 J=1,NIND                                                  7870
  353 G(J)=XX(J,I)                                                     7880
      B=Y(I)                                                           7890
      ITHFUN=II                                                        7900
      IF(NIND.EQ.1) CALL MODEL(B,G(1),F,0)                             7910
      IF(NIND.GT.1) CALL MODEL(B,G,F,0)                                7920
      YEST(I)=B                                                        7930
      RES(I)=Y(I)-YEST(I)                                              7940
      IF(WYN(I).EQ.0.)RES(I)=0.                                        7950
      WRES(I)=RES(I)*SQRT(WYN(I))                                      7960
      SUMRES=SUMRES+RES(I)                                             7970
      WSUMRS=WSUMRS+WRES(I)                                            7980
      RESQ=RES(I)*RES(I)                                               7990
      SSRES=SSRES+RESQ                                                 8000
      RSS(II)=RSS(II)+RESQ                                             8010
      WRSS(II)=WRSS(II)+WYN(I)*RESQ                                    8020
      IF(ABS(YEST(I)).LT.1E-50) DIFPCT(I)=9.999E+98                    8030
      IF(ABS(YEST(I)).GE.1E-50) DIFPCT(I)=100.*RES(I)/YEST(I)          8040
      IF(WYN(I).EQ.0.) GO TO 355                                       8050
```

M01

```
      SSY=SSY+Y(I)*Y(I)                                                8060
      SY=SY+Y(I)                                                       8070
      SSYC=SSYC+B*B                                                    8080
      SYC=SYC+B                                                        8090
      SYCY=SYCY+B*Y(I)                                                 8100
  355 CONTINUE                                                         8110
      NOBSS=NOBS                                                       8120
      NOBS=NOBS-NDUMMY                                                 8130
      YBAR=SY/FLOAT(NOBS)                                              8140
      SSTOT=SSY-SY*SY/FLOAT(NOBS)                                      8150
      B=NOBS                                                           8160
      R=ABS(SYCY-SY*SYC/B)/SQRT(ABS((SSY-SY*SY/B)*(SSYC-SYC*SYC/B)))   8170
      RSQ=100*R*R                                                      8180
      SSREG=SSTOT-SSRES                                                8190
      RESBAR=SUMRES/FLOAT(NOBS)                                        8200
      WRSBAR=WSUMRS/FLOAT(NOBS)                                        8210
      NRGDF=NEWNOP-1                                                   8220
      IF(NRGDF.LE.0) NRGDF=1                                           8230
      NRSDF=NOBS-NEWNOP                                                8240
      NTODF=NOBS-1                                                     8250
      REGMSQ=SSREG/FLOAT(NRGDF)                                        8260
      RESMSQ=SSRES/FLOAT(NRSDF)                                        8270
      WRSMSQ=FUNC/FLOAT(NRSDF)                                         8280
      SDVRES=SQRT(RESMSQ)                                              8290
      SDVRS=SQRT(WRSMSQ)                                               8300
      OVERF=REGMSQ/RESMSQ                                              8310
      T=1.95                                                           8320
      IF(NRSDF.LE.30[illegible])                                       8330
      WRITE(6,790)                                                     8340
      WRITE(6,525)                                                     8350
      NOBS=NOBSS                                                       8360
```

```
      IF(NPR.GT.5.AND.MOD(NEVAL,IPRINT).EQ.0)WRITE(IW,101)                  16650
     *NEVAL,H(I),(F(J),J=1,NPR)                                             16660
   38 CONTINUE                                                              16670
      GO TO 41                                                              16680
   26 DO 40 I=1,NOP                                                         16690
      IF(STEP(I).NE.0.0) G(IMAX,I)=PSTAR(I)                                 16700
   40 CONTINUE                                                              16710
      H(IMAX)=HSTAR                                                         16720
   41 IF(NEVAL-MAX.GT.0) GO TO 46                                           16730
```



```
      IF(IFLAG.NE.1) GO TO 4100                                             16740
      IF(LOOP-MAX0(NLOOP,MORE))45,46,45                                     16750
 4100 IF(LOOP-NLOOP)45,46,45                                                16760
   46 DHCV=0D0                                                              16770
      DMEAN=0D0                                                             16780
      DO 42 I=1,NP1                                                         16790
      IF(H(I).GE.1D50) GO TO 45                                             16800
   42 DMEAN=DMEAN+H(I)                                                      16810
      DMEAN=DMEAN/FLOAT(NP1)                                                16820
      DO 601 I=1,NP1                                                        16830
  601 DHCV=DHCV+(H(I)-DMEAN)*(H(I)-DMEAN)                                   16840
      DHCV=1D2*DSQRT(DHCV/FLOAT(NP1))/DMEAN                                 16850
      DO 53 I=1,NOP                                                         16860
      IF(STEP(I).EQ.0.0) GO TO 53                                           16870
      F(I)=0.0                                                              16880
      DO 54 J=1,NP1                                                         16890
   54 F(I)=F(I)+G(J,I)                                                      16900
      F(I)=F(I)/NP1                                                         16910
   53 CONTINUE                                                              16920
      CALL FUNCTN (F,FUNC)                                                  16930
      NEVAL=NEVAL+1                                                         16940
      IF(IPRINT.LE.0) GO TO 700                                             16950
      IF(NPR.LE.5.AND.MOD(NEVAL,IPRINT).EQ.0)WRITE(IW,1010)                 16960
     *NEVAL,FUNC,(F(J),J=1,NPR)                                             16970
      IF(NPR.GT.5.AND.MOD(NEVAL,IPRINT).EQ.0)WRITE(IW,101)                  16980
     *NEVAL,FUNC,(F(J),J=1,NPR)                                             16990
  700 IF(NEVAL-MAX) 44,44,43                                                17000
   43 GO TO 67                                                              17010
   67 WRITE(IW,102)MAX                                                      17020
  102 FORMAT(40H NUMBER OF FUNCTION EVALUATIONS EXCEEDS ,I4)                 17030
      IF(NVIOL.GT.10) WRITE(IW,8031)NVIOL                                   17040
      WRITE(IW,103)DHCV                                                     17050
  103 FORMAT(" COEFFICIENT OF VARIATION (PCT) OF FUNCTION VALUES",          17060
     *" OF LAST SIMPLEX" ,D10.4 )                                           17070
      IF(NPR.LE.5.)WRITE(IW,1040)(F(I),I=1,NPR)                             17080
 1040 FORMAT(" CENTROID    ",5E12.6)                                        17090
      IF(NPR.GT.5)WRITE(IW,104)(F(I),I=1,NPR)                               17100
  104 FORMAT( " CENTROID    ",5E12.6/(12X,5E12.6))                          17110
      WRITE(IW,105)FUNC                                                     17120
  105 FORMAT(" FUNCTION VALUE AT CENTROID    ",D15.8 / )                    17130
   68 IFAULT = 1                                                            17140
      RETURN                                                                17150
   44 IF(DHCV-DBLE(STOPCR))72,48,48                                         17160
   48 IFLAG=0                                                               17170
      LOOP=0                                                                17180
      GO TO 45                                                              17190
   72 IF(IPRINT)47,70,70                                                    17200
   70 WRITE(IW,106)DHCV                                                     17210
  106 FORMAT(" * INITIAL EVIDENCE OF CONVERGENCE   COEFF. OF VARIATION",    17220
     *" (PCT) ",D10.4 )                                                     17230
      IF(NPR.LE.5) WRITE(IW,1010)NEVAL,FUNC,(F(I),I=1,NPR)                  17240
      IF(NPR.GT.5) WRITE(IW,101)NEVAL,FUNC,(F(I),I=1,NPR)                   17250
   47 IF(IFLAG)49,50,49                                                     17260
   50 IFLAG=1                                                               17270
   51 DMEANS=DMEAN                                                          17280
      FUNCS=FUNC                                                            17290
      LOOP=0                                                                17300
      GO TO 45                                                              17310
   49 IF(DABS(1D1*(DMEANS-DMEAN)/DMEANS).GT.DBLE(STOPCR).OR.                17320
     *DABS(1D2*(FUNC-FUNCS)/FUNCS).GT.DBLE(STOPCR)) GO TO 51                17330
      NEVAL1=NEVAL                                                          17340
      IF(IQUAD.NE.0.OR.H(IMIN).GT.FUNC) GO TO 4905                          17350
```



```
      FUNC=H(IMIN)                                                          17360
      DO 4900 J=1,NOP                                                       17370
 4900 F(I)=G(IMIN,J)                                                        17380
 4905 CONTINUE                                                              17390
      IF(IPRINT)74,73,73                                                    17400
   73 IF(NVIOL.GT.0) WRITE(IW,107)NEVAL,NVIOL                               17410
  107 FORMAT(/" PROCESS CONVERGES AFTER",I4,"-",I3," FUNCTION",             17420
     *" EVALUATIONS" / )                                                    17430
      IF(NPR.LE.5)WRITE(IW,1080)(F(I),I=1,NPR)                              17440
 1080 FORMAT(" MINIMUM AT ",5E12.6)                                         17450
      IF(NPR.GT.5) WRITE(IW,108)(F(I),I=1,NPR)                              17460
  108 FORMAT(  " MINIMUM AT ",5E12.6/(12X,5E12.6))                          17470
      WRITE(IW,109)FUNC                                                     17480
      WRITE(IW,103)DHCV                                                     17490
  109 FORMAT(/ 26H MINIMUM FUNCTION VALUE   ,D15.8/ )                       17500
   74 CONTINUE                                                              17510
      IF(IPRINT.GT.0.AND.NVIOL.GT.NPR) WRITE(IW,8035)                       17520
 8035 FORMAT(" (FUNC.VALUES GREATER THAN 1D+50 INDICATE VIOLATION",         17530
     *" OF PARAMETER CONSTRAINS)" )                                         17540
      IF(IPRINT.GE.0.AND.NVIOL.GT.1) WRITE(IW,8031)NVIOL                    17550
 8031 FORMAT(" * NO. OF VIOLATIONS OF PARAMETER CONSTRAINS",                17560
     *" DURING MINIMIZATION"  ,I4 )                                         17570
      IF(IQUAD)200,75,200                                                   17580
   75 RETURN                                                                17590
  200 LIMITS=0                                                              17600
      ALOW=5HLOWER                                                          17610
      AUPP=5HUPPER                                                          17620
      ICNSTR=0                                                              17630
      DO 806 I=1,NOP                                                        17640
      IF(F1(I).EQ.F2(I)) GO TO 806                                          17650
      DEL=(F2(I)-F1(I))*0.001                                               17660
      IF(F(I)-F1(I).GT.DEL) GO TO
```

```
      SIGMA=(BAR-1.)*(BAR-2.)/(AM+AN-1.)                                26950
      Z= (FLOAT(NU)+.5-BAR)/SIGMA                                       26960
      SEQPRO=SDF(Z)                                                     26970
      RETURN                                                            26980
      END                                                               26990
C     ******************************************************************26000
      FUNCTION NBC(N,M)                                                 26010
      DOUBLE PRECISION NC,ND,NE,NF                                      26020
      IF(M.GT.N) GO TO 6                                                26030
```



```
      IF(N.EQ.M.OR.M.EQ.0) GO TO 5                                      26040
      IF(M.EQ.1.OR.N-M.EQ.1) GO TO 4                                    26050
      IF(2*M-N.GT.0) GO TO 1                                            26060
      NC=N-M                                                            26070
      ND=NC+1                                                           26080
      K=M-1                                                             26090
      GO TO 2                                                           26100
    1 NC=M                                                              26110
      ND=M+1                                                            26120
      K=N-M-1                                                           26130
    2 NE=1                                                              26140
      DO 3 J=1,K                                                        26150
      NF=J+1                                                            26160
      ND=ND*(NC+NF)                                                     26170
    3 NE=NE*NF                                                          26180
      NBC=ND/NE                                                         26190
      RETURN                                                            26200
    4 NBC=N                                                             26210
      RETURN                                                            26220
    5 NBC=1                                                             26230
      RETURN                                                            26240
    6 NBC=0                                                             26250
      RETURN                                                            26260
      END                                                               26270
C     ******************************************************************26280
      FUNCTION SDF(X)                                                   26290
      AX=ABS(X)                                                         26300
      T=1./(1.+.2316419*AX)                                             26310
      D=.3989423*EXP(-X*X/2.)                                           26320
      SDF=1.-D*T*((((1.330274*T-1.821256)*T+1.781478)*T-                26330
     *.3565638)*T+.3193815)                                             26340
      IF(X)1,2,2                                                        26350
    1 SDF=1.-SDF                                                        26360
    2 RETURN                                                            26370
      END                                                               26380
C     ******************************************************************26390
      SUBROUTINE PROB1(E,NE,M,P)                                        26400
      DIMENSION E(100)                                                  26410
      NS=0                                                              26420
      N=0                                                               26430
      DO 5 J=1,NE                                                       26440
    5 IF(E(J).GT.0.) N=N+1                                              26450
      IF(N.LT.NE-N) N=NE-N                                              26460
      M=NE-N                                                            26470
      MM=M+1                                                            26480
      IF(NE.GT.25) GO TO 15                                             26490
      DO 10 J=1,MM                                                      26500
   10 NS=NS+NBC(NE,J-1)                                                 26510
      P=FLOAT(NS)*(2.**(-NE))                                           26520
      RETURN                                                            26530
   15 Z=FLOAT(2*M+1-NE)/SQRT(FLOAT(NE))                                 26540
      P=SDF(Z)                                                          26550
      RETURN                                                            26560
      END                                                               26570
C     ******************************************************************26580
      SUBROUTINE SEARCH(FNEW,STPNEW,SSMIN,ANS16,NPOINT,NEVAL,           26590
     *NEWNOP,STEP,NP)                                                   26600
      DIMENSION FNEW(7),F1(20),F2(20),STPNEW(7),P1(75),P2(75),XX(9,100),26610
     *SS(75),SSS(10,10),PDEV(10,20),FF(7),Y(100),LABEL(100),            26620
     *WYN(100),DEL(7),PP(7,12),IS(7),STEP(20),A(10),AL(9),CON(10,10)    26630
      COMMON /DATA/ XX,Y,WYN,NOBS                                       26640
      COMMON /CONSTR/LIMITS                                             26650
```



```
      COMMON /PARLIM/ F1,F2                                             26660
      COMMON /FUNNUM/ITHFUN                                             26670
      COMMON /B4/ NIND                                                  26680
      COMMON /B6/ NOP                                                   26690
      COMMON /B10/LABEL                                                 26700
      DATA AL/3H1  ,3H2  ,3H3  ,3H4  ,3H5  ,3H6  ,3H7  ,3H8  ,3H9   /   26710
      LIMITS=0                                                          26720
      IF(NEWNOP-2) 5,50,135                                             26730
    5 CONTINUE                                                          26740
      DO 10 I=1 NOP                                                     26750
      FNEW(I)=F1(I)                                                     26760
      STPNEW(I)=STEP(I)                                                 26770
   10 IF(F1(I).NE.F2(I)) IP=I                                           26780
      DEL(IP)=(F2(IP)-F1(IP))/74.                                       26790
      DO 20 I=1,75                                                      26800
      FNEW(IP)=F1(IP)+FLOAT(I-1)*DEL(IP)                                26810
      CALL LSQ(FNEW,SS(I))                                              26820
   20 P1(I)=FNEW(IP)                                                    26830
      IF(ANS16.EQ.2HNO) WRITE(6,1000)IP,F1(IP),F2(IP)                   26840
      IF(ANS16.EQ.3HYES) WRITE(6,1010)IP,F1(IP),F2(IP)                  26850
      CALL PLOT(P1,SS,75,50)                                            26860
      IF(ANS16.EQ.2HNO) WRITE(6,1020)                                   26870
      IF(ANS16.EQ.3HYES) WRITE(6,1025)                                  26880
      DO 30 I=1,37                                                      26890
      J=I+37                                                            26900
   30 WRITE(6,1026)I,SS(I),P1(I),J,SS(J),P1(J)                          26910
      WRITE(6,1027)                                                     26920
      ISAVE=1                                                           26930
      SSMIN=SS(1)                                                       26940
      DO 40 I=2,75                                                      26950
      IF(SS(I).GE.SSMIN) GO TO 40                                       26960
      SSMIN=SS(I)                                                       26970
```

```
      YEST(I)=B                                                          7930
      RES(I)=Y(I)-YEST(I)                                                7940
      IF(WYN(I).EQ.0.)RES(I)=0.                                          7950
      WRES(I)=RES(I)*SQRT(WYN(I))                                        7960
      SUMRES=SUMRES+RES(I)                                               7970
      WSUMRS=WSUMRS+WRES(I)                                              7980
      RESQ=RES(I)*RES(I)                                                 7990
      SSRES=SSRES+RESQ                                                   8000
      RSS(II)=RSS(II)+RESQ                                               8010
      WRSS(II)=WRSS(II)+WYN(I)*RESQ                                      8020
      IF(ABS(YEST(I)).LT.1E-50) DIFPCT(I)=9.999E+98                      8030
      IF(ABS(YEST(I)).GE.1E-50) DIFPCT(I)=100.*RES(I)/YEST(I)            8040
      IF(WYN(I).EQ.0.) GO TO 355                                         8050
```

N01

```
      SSY=SSY+Y(I)*Y(I)                                                  8060
      SY=SY+Y(I)                                                         8070
      SSYC=SSYC+B*B                                                      8080
      SYC=SYC+B                                                          8090
      SYCY=SYCY+B*Y(I)                                                   8100
  355 CONTINUE                                                           8110
      NOBSS=NOBS                                                         8120
      NOBS=NOBS-NDUMMY                                                   8130
      YBAR=SY/FLOAT(NOBS)                                                8140
      SSTOT=SSY-SY*SY/FLOAT(NOBS)                                        8150
      B=NOBS                                                             8160
      R=ABS(SYCY-SY*SYC/B)/SQRT(ABS((SSY-SY*SY/B)*(SSYC-SYC*SYC/B)))     8170
      RSQ=100*R*R                                                        8180
      SSREG=SSTOT-SSRES                                                  8190
      RESBAR=SUMRES/FLOAT(NOBS)                                          8200
      WRSBAR=WSUMRS/FLOAT(NOBS)                                          8210
      NRGDF=NEWNOP-1                                                     8220
      IF(NRGDF.LE.0) NRGDF=1                                             8230
      NRSDF=NOBS-NEWNOP                                                  8240
      NTODF=NOBS-1                                                       8250
      REGMSQ=SSREG/FLOAT(NRGDF)                                          8260
      RESMSQ=SSRES/FLOAT(NRSDF)                                          8270
      WRSMSQ=FUNC/FLOAT(NRSDF)                                           8280
      SDVRES=SQRT(RESMSQ)                                                8290
      WSDVRS=SQRT(WRSMSQ)                                                8300
      OVERF=REGMSQ/RESMSQ                                                8310
      T=1.95                                                             8320
      IF(NRSDF.LE.30) T=TTT(NRSDF)                                       8330
      WRITE(6,790)                                                       8340
      WRITE(6,525)                                                       8350
      NOBS=NOBSS                                                         8360
      IF(IQUAD.EQ.0) GO TO 3610                                          8370
      WRITE(6,800)                                                       8380
      FAC=SQRT(2.*FUNC/FLOAT(NRSDF))                                     8390
      DO 360 I=1,NOP                                                     8400
      SDV=FAC*SQRT(VAR(I))                                               8410
      IF(ABS(F(I)).LT.1E-50) CV=9.999E+98                                8420
      IF(ABS(F(I)).GE.1E-50) CV=100.*SDV/ABS(F(I))                       8430
      IF(SDV.EQ.0.) CV=0.                                                8440
  360 WRITE(6,802)I,F1(I),F2(I),FSAVE(I),F(I),SDV,CV                     8450
      WRITE(6,803)                                                       8460
      DO 361 I=1,NOP                                                     8470
      E=T*FAC*SQRT(VAR(I))                                               8480
      E1=F(I)-E                                                          8490
      E2=F(I)+E                                                          8500
  361 IF(F1(I).NE.F2(I)) WRITE(6,805)I,E1,E2                             8510
      WRITE(6,8050)                                                      8520
 3600 CONTINUE                                                           8530
      IFLAG=0                                                            8540
      GO TO 3602                                                         8550
 3601 WRITE(6,792)                                                       8560
 3602 CALL PLACE(F1,F2,F,65,NPR,A)                                       8570
      IF(NPR.EQ.1) WRITE(6,8051)                                         8580
      IF(NPR.GT.1) WRITE(6,8052)                                         8590
      DO 3605 J=1,NPR                                                    8600
      IF(F1(J).EQ.F2(J)) GO TO 3605                                      8610
      N1=65*(J-1)+1                                                      8620
      N2=N1+64                                                           8630
      WRITE(6,8053) J,(A(I),I=N1,N2)                                     8640
 3605 CONTINUE                                                           8650
      IF(IFLAG.NE.0.AND.ANS16.EQ.2HNO) WRITE(6,8054)FUNC                 8660
      IF(IFLAG.NE.0.AND.ANS16.NE.2HNO) WRITE(6,8055)FUNC                 8670
```

001

```
      GO TO 3612                                                         8680
 3610 WRITE(6,8010)                                                      8690
      DO 3611 I=1,NOP                                                    8700
 3611 WRITE(6,8011)I,F1(I),F2(I),FSAVE(I),F(I)                           8710
      WRITE(6,8012)                                                      8720
      GO TO 3600                                                         8730
 3612 CONTINUE                                                           8740
      ALOW=5HLOWER                                                       8750
      AUPP=5HUPPER                                                       8760
      ICNSTR=0                                                           8770
      DO 3621 I=1,NOP                                                    8780
      IF(F1(I).EQ.F2(I)) GO TO 3621                                      8790
      DEL=(F2(I)-F1(I))*0.001                                            8800
      IF(F(I)-F1(I).GT.DEL) GO TO 3620                                   8810
      ICNSTR=ICNSTR+1                                                    8820
      WRITE(6,8030) ALOW,I                                               8830
 3620 IF(F2(I)-F(I).GT.DEL) GO TO 3621                                   8840
      ICNSTR=ICNSTR+1                                                    8850
      WRITE(6,8030) AUPP,I                                               8860
 3621 CONTINUE                                                           8870
      IF(ICNSTR.EQ.0) GO TO 3622                                         8880
      WRITE(6,8031)                                                      8890
      CALL READ(1,AR)                                                    8900
      IF(IGO.EQ.2.OR.ANS.EQ.4HBACK) WRITE(6,844) NRUN                    8910
      IF(IGO.EQ.2) GO TO 1001                                            8920
      IF(ANS.EQ.4HBACK) GO TO 1004                                       8930
 3622 CONTINUE                                                           8940
      IF(IFLAG.NE.0) GO TO 3513                                          8950
      WRITE(6,80)SSRES,SSREG,SSTOT,SSY,YBAR,RESMSQ,REGMSQ,RESBAR        8960
```

```
      IF(NPR.LE.5) WRITE(IW,1010)NEVAL,FUNC,(F(I),I=1,NPR)                17240
      IF(NPR.GT.5) WRITE(IW,101)NEVAL,FUNC,(F(I),I=1,NPR)                 17250
   47 IF(IFLAG)49,50,49                                                   17260
   50 IFLAG=1                                                             17270
   51 DMEANS=DMEAN                                                        17280
      FUNCS=FUNC                                                          17290
      LOOP=0                                                              17300
      GO TO 45                                                            17310
   49 IF(DABS(1D1*(DMEANS-DMEAN)/DMEANS).GT.DBLE(STOPCR).OR.              17320
     *DABS(1D2*(FUNC-FUNCS)/FUNCS).GT.DBLE(STOPCR)) GO TO 51              17330
      NEVAL1=NEVAL                                                        17340
      IF(IQUAD.NE.0.OR.H(IMIN).GT.FUNC) GO TO 4905                        17350
```

W02

```
      FUNC=H(IMIN)                                                        17360
      DO 4900 J=1,NOP                                                     17370
 4900 F(I)=G(IMIN,J)                                                      17380
 4905 CONTINUE                                                            17390
      IF(IPRINT)74,73,73                                                  17400
   73 IF(NVIOL.GT.0) WRITE(IW,107)NEVAL,NVIOL                             17410
  107 FORMAT(/" PROCESS CONVERGES AFTER",I4,"-",I3," FUNCTION",           17420
     *" EVALUATIONS" / )                                                  17430
      IF(NPR.LE.5)WRITE(IW,1080)(F(I),I=1,NPR)                            17440
 1080 FORMAT(" MINIMUM AT  ",5E12.6)                                      17450
      IF(NPR.GT.5) WRITE(IW,108)(F(I),I=1,NPR)                            17460
  108 FORMAT(" MINIMUM AT ",5E12.6/(12X,5E12.6))                          17470
      WRITE(IW,109)FUNC                                                   17480
      WRITE(IW,103)DHCV                                                   17490
  109 FORMAT(/ 26H MINIMUM FUNCTION VALUE   ,D15.8/ )                     17500
   74 CONTINUE                                                            17510
      IF(IPRINT.GT.0.AND.NVIOL.GT.NPR) WRITE(IW,8035)                     17520
 8035 FORMAT(" (FUNC.VALUES GREATER THAN 1D+50 INDICATE VIOLATION",       17530
     *" OF PARAMETER CONSTRAINS)" )                                       17540
      IF(IPRINT.GE.0.AND.NVIOL.GT.1) WRITE(IW,8031)NVIOL                  17550
 8031 FORMAT(" * NO. OF VIOLATIONS OF PARAMETER CONSTRAINS",              17560
     *" DURING MINIMIZATION" ,I4 )                                        17570
      IF(IQUAD)200,75,200                                                 17580
   75 RETURN                                                              17590
  200 LIMITS=0                                                            17600
      ALOW=5HLOWER                                                        17610
      AUPP=5HUPPER                                                        17620
      ICNSTR=0                                                            17630
      DO 806 I=1,NOP                                                      17640
      IF(F1(I).EQ.F2(I)) GO TO 806                                        17650
      DEL=(F2(I)-F1(I))*0.001                                             17660
      IF(F(I)-F1(I).GT.DEL) GO TO 8                                       17670
      ICNSTR=ICNSTR+1                                                     17680
      IF(ICNSTR.NE.1) GO TO 800                                           17690
      WRITE(6,8000) FUNC,(F(L),L=1,NPR)                                   17700
      WRITE(6,8005)                                                       17710
  800 WRITE(6,8010) ALOW,F1(I),I,F(I)                                     17720
  802 IF(F2(I)-F(I).GT.DEL) GO TO 806                                     17730
      ICNSTR=ICNSTR+1                                                     17740
      IF(ICNSTR.NE.1) GO TO 804                                           17750
      WRITE(6,8000) FUNC,(F(L),L=1,NPR)                                   17760
      WRITE(6,8005)                                                       17770
  804 WRITE(6,8010) AUPP,F2(I),I,F(I)                                     17780
  806 CONTINUE                                                            17790
      IF(ICNSTR.EQ.0) GO TO 814                                           17800
      IF(IPRINT.EQ.-1.AND.NVIOL.GT.*) WRITE(IW,8031)NVIOL                 17810
      WRITE(6,8015)                                                       17820
  808 CALL READ(1,AR)                                                     17830
      IF(AR(1).EQ.4HBACK.OR.IGO.EQ.2) GO TO 810                           17840
      IF(AR(1).EQ.2HNO) GO TO 812                                         17850
      IF(AR(1).EQ.3HYES) GO TO 814                                        17860
      WRITE(6,8030)                                                       17870
      GO TO 808                                                           17880
  810 IFAULT=5                                                            17890
      WRITE(6,8020)                                                       17900
      RETURN                                                              17910
  812 IQUAD=0                                                             17920
      WRITE(6,8025)                                                       17930
      RETURN                                                              17940
  814 CONTINUE                                                            17950
 8000 FORMAT(" * THE RESIDUAL SUM OF SQUARES MINIMUM (",D12.6,            17960
     * ) AT THE PARAMETER VALUES"/ *"/(" *",7E10.4))                      17970
```

002

```
 8005 FORMAT(" *"/" * SEEM TO BE CONSTRAINED BY THE"/" *" )               17980
 8010 FORMAT(" *",5X,A5," LIMIT (",E11.5,") OF PAR.NO.",I2,               17990
     *" (",E11.5,")" )                                                    18000
 8015 FORMAT(" *"/" * A STATISTICAL EVALUATION OF THE PARAMETERS UNDER",  18010
     *" THESE CONDITIONS MAY FAIL."/                                      18020
     *" * ENTER B10 OR B11 IF YOU WANT A NEW RUN WITH DIFFERENT"/         18030
     *" * PARAMETER LIMITS OR DIFFERENT INITIAL PARAMETER ESTIMATES"/     18040
     *" * ENTER -YES- IF YOU WANT TO CONTINUE OR"/                        18050
     *" * ENTER -NO- IF YOU WANT TO CONTINUE BUT WITHOUT A STATISTICAL",  18060
     *" EVALUATION"//  )                                                  18070
 8020 FORMAT(/1X,77(1H-)// )                                              18080
 8025 FORMAT(/" * REMEMBER YOUR INPUT UNDER REQUEST NO.3 HAS NOW",        18090
     *" BEEN CHANGED TO -NO-"/ )                                          18100
 8030 FORMAT(/" INPUT ERROR, YOUR ANSWER MUST BE ONE OF THE FOLLOWING"/   18110
     *" FOUR,   Y   N   B   R     - TRY AGAIN -"/  )                     18120
      IF(IPRINT)233,232,232                                               18130
  232 WRITE(IW,301)                                                       18140
  301 FORMAT(/1X,13(1H*)," FITTING OF QUADRATIC SURFACE IN REGION OF",     18150
     *" MINIMUM ",13(1H*) / )                                             18160
  233 NEVAL=0                                                             18170
      NFIX=0                                                              18180
      B=SIMP/100.                                                         18190
      SIMP2=FUNC*B                                                        18200
      SIMP2=AMAX1(B,SIMP2)                                                18210
      DO 201 I=1,NP1                                                      18220
  204 IF(H(I).GE.1D50) GO TO 2030                                         18230
      TEST=DABS(H(I)-FUNC)                                                18240
      IF(TEST-SIMP2)202,201,201                                           18250
  202 DO 203 J=1,NOP                                                      18260
```

```
   15 Z=FLOAT(2*M+1-NE)/SQRT(FLOAT(NE))                                  26540
      P=SDF(Z)                                                           26550
      RETURN                                                             26560
      END                                                                26570
C     ******************************************************************  26580
      SUBROUTINE SEARCH(FNEW,STPNEW,SSMIN,ANS16,NPOINT,NEVAL,            26590
     *NEWNOP,STEP,NP)                                                    26600
      DIMENSION FNEW(7),F1(20),F2(20),STPNEW(7),P1(75),P2(75),XX(9,100), 26610
     *SS(75),SSS(10,10),PDEV(10,20),FF(7),Y(100),LABEL(100),             26620
     *WYN(100),DEL(7),PP(7,12),IS(7),STEP(20),A(10),AL(9),CON(10,10)     26630
      COMMON /DATA/ XX,Y,WYN,NOBS                                        26640
      COMMON /CONSTR/LIMITS                                              26650
```

N03

```
      COMMON /PARLIM/ F1,F2                                              26660
      COMMON /FUNNUM/ITHFUN                                              26670
      COMMON /B4/ NIND                                                   26680
      COMMON /B6/ NOP                                                    26690
      COMMON /B10/LABEL                                                  26700
      DATA AL/3H1  ,3H2  ,3H3  ,3H4  ,3H5  ,3H6  ,3H7  ,3H8  ,3H9   /   26710
      LIMITS=0                                                           26720
      IF(NEWNOP-2) 5,50,135                                              26730
    5 CONTINUE                                                           26740
      DO 10 I=1,NOP                                                      26750
      FNEW(I)=F1(I)                                                      26760
      STPNEW(I)=STEP(I)                                                  26770
   10 IF(F1(I).NE.F2(I)) IP=I                                            26780
      DEL(IP)=(F2(IP)-F1(IP))/74.                                        26790
      DO 20 I=1,75                                                       26800
      FNEW(IP)=F1(IP)+FLOAT(I-1)*DEL(IP)                                 26810
      CALL LSQ(FNEW,SS(I))                                               26820
   20 P1(I)=FNEW(IP)                                                     26830
      IF(ANS16.EQ.2HNO) WRITE(6,1000)IP,F1(IP),F2(IP)                    26840
      IF(ANS16.EQ.3HYES) WRITE(6,1010)IP,F1(IP),F2(IP)                   26850
      CALL PLOT(P1,SS,75,50)                                             26860
      IF(ANS16.EQ.2HNO) WRITE(6,1020)                                    26870
      IF(ANS16.EQ.3HYES) WRITE(6,1025)                                   26880
      DO 30 I=1,37                                                       26890
      J=I+37                                                             26900
   30 WRITE(6,1026)I,SS(I),P1(I),J,SS(J),P1(J)                           26910
      WRITE(6,1027)                                                      26920
      ISAVE=1                                                            26930
      SSMIN=SS(1)                                                        26940
      DO 40 I=2,75                                                       26950
      IF(SS(I).GE.SSMIN) GO TO 40                                        26960
      SSMIN=SS(I)                                                        26970
      ISAVE=I                                                            26980
   40 CONTINUE                                                           26990
      FNEW(IP)=P1(ISAVE)                                                 27000
      STPNEW(IP)=DEL(IP)/2.                                              27010
      RETURN                                                             27020
   50 IP1=7                                                              27030
      DO 55 I=1,NOP                                                      27040
      FNEW(I)=F1(I)                                                      27050
      STPNEW(I)=STEP(I)                                                  27060
      IF(I.GT.IP1) GO TO 55                                              27070
      IF(F1(I).NE.F2(I)) IP1=I                                           27080
   55 IF(F1(I).NE.F2(I)) IP2=I                                           27090
      DEL(IP1)=(F2(IP1)-F1(IP1))/9.                                      27100
      DEL(IP2)=(F2(IP2)-F1(IP2))/9.                                      27110
      ISAVE=1                                                            27120
      JSAVE=1                                                            27130
      SSMIN=1.E99                                                        27140
      SSMAX=0.                                                           27150
      DO 60 J=1,10                                                       27160
      FNEW(IP1)=F1(IP1)+FLOAT(J-1)*DEL(IP1)                              27170
      P1(J)=FNEW(IP1)                                                    27180
      DO 60 I=1,10                                                       27190
      FNEW(IP2)=F1(IP2)+FLOAT(I-1)*DEL(IP2)                              27200
      P2(I)=FNEW(IP2)                                                    27210
      CALL LSQ(FNEW,SSS(I,J))                                            27220
      IF(SSS(I,J).GE.SSMIN) GO TO 60                                     27230
      SSMIN=SSS(I,J)                                                     27240
      ISAVE=I                                                            27250
      JSAVE=J                                                            27260
   60 IF(SSS(I,J).GT.SSMAX) SSMAX=SSS(I,J)                               27270
```

003

```
      FNEW(IP1)=P1(JSAVE)                                                27280
      FNEW(IP2)=P2(ISAVE)                                                27290
      STPNEW(IP1)=DEL(IP1)/2.                                            27300
      STPNEW(IP2)=DEL(IP2)/2.                                            27310
      D=(SSMAX-SSMIN)/9.                                                 27320
      DO 65 I=1,10                                                       27330
   65 A(I)=SSMIN+(I-1)*D                                                 27340
      A(10)=A(10)+0.01*D                                                 27350
      WRITE(6,1030)IP1,(I,P1(I),I=1,10)                                  27360
      WRITE(6,1040)IP2,(I,P2(I),I=1,10)                                  27370
      IF(ANS16.EQ.2HNO) WRITE(6,1050)                                    27380
      IF(ANS16.EQ.3HYES) WRITE(6,1060)                                   27390
      WRITE(6,1070)IP1,IP2                                               27400
      WRITE(6,1075)(I,I=1,10)                                            27410
      DO 67 I=1,10                                                       27420
      WRITE(6,1080)(SSS(I,J),J=1,9,2)                                    27430
   67 WRITE(6,1085)(SSS(I,J),J=2,10,2)                                   27440
      DO 120 J=1,10                                                      27450
      DO 120 I=1,10                                                      27460
      IF(J.EQ.1) GO TO 70                                                27470
      IF(J.EQ.10) GO TO 80                                               27480
      PDEV(I,J)=SIGNOF(SSS(I,J+1)-SSS(I,J-1))                            27490
      GO TO 90                                                           27500
   70 PDEV(I,J)=SIGNOF(4.*SSS(I,J+1)-3.*SSS(I,J)-SSS(I,J+2))             27510
      GO TO 90                                                           27520
   80 PDEV(I,J)=SIGNOF(-4.*SSS(I,J-1)+3.*SSS(I,J)+SSS(I,J-2))            27530
   90 K=J+10                                                             27540
      IF(I.EQ.1) GO TO 100                                               27550
      IF(I.EQ.10) GO TO 110                                              27560
```

```
      IF(SDV.EQ.0.) CV=0.                                                8440
  360 WRITE(6,802)I,F1(I),F2(I),FSAVE(I),F(I),SDV,CV                     8450
      WRITE(6,803)                                                       8460
      DO 361 I=1,NOP                                                     8470
      E=T*FAC*SQRT(VAR(I))                                               8480
      E1=F(I)-E                                                          8490
      E2=F(I)+E                                                          8500
  361 IF(F1(I).NE.F2(I)) WRITE(6,805)I,E1,E2                             8510
      WRITE(6,8050)                                                      8520
 3600 CONTINUE                                                           8530
      IFLAG=0                                                            8540
      GO TO 3602                                                         8550
 3601 WRITE(6,792)                                                       8560
 3602 CALL PLACE(F1,F2,F,65,NPR,A)                                       8570
      IF(NPR.EQ.1) WRITE(6,8051)                                         8580
      IF(NPR.GT.1) WRITE(6,8052)                                         8590
      DO 3605 J=1,NPR                                                    8600
      IF(F1(J).EQ.F2(J)) GO TO 3605                                      8610
      N1=65*(J-1)+1                                                      8620
      N2=N1+64                                                           8630
      WRITE(6,8053) J,(A(I),I=N1,N2)                                     8640
 3605 CONTINUE                                                           8650
      IF(IFLAG.NE.0.AND.ANS16.EQ.2HNO) WRITE(6,8054)FUNC                 8660
      IF(IFLAG.NE.0.AND.ANS16.NE.2HNO) WRITE(6,8055)FUNC                 8670
```

001

```
      GO TO 3612                                                         8680
 3610 WRITE(6,8010)                                                      8690
      DO 3611 I=1,NOP                                                    8700
 3611 WRITE(6,8011)I,F1(I),F2(I),FSAVE(I),F(I)                           8710
      WRITE(6,8012)                                                      8720
      GO TO 3600                                                         8730
 3612 CONTINUE                                                           8740
      ALOW=5HLOWER                                                       8750
      AUPP=5HUPPER                                                       8760
      ICNSTR=0                                                           8770
      DO 3621 I=1,NOP                                                    8780
      IF(F1(I).EQ.F2(I)) GO TO 3621                                      8790
      DEL=(F2(I)-F1(I))*0.001                                            8800
      IF(F(I)-F1(I).GT.DEL) GO TO 3620                                   8810
      ICNSTR=ICNSTR+1                                                    8820
      WRITE(6,8030) ALOW,I                                               8830
 3620 IF(F2(I)-F(I).GT.DEL) GO TO 3621                                   8840
      ICNSTR=ICNSTR+1                                                    8850
      WRITE(6,8030) AUPP,I                                               8860
 3621 CONTINUE                                                           8870
      IF(ICNSTR.EQ.0) GO TO 3622                                         8880
      WRITE(6,8031)                                                      8890
      CALL READ(1,AR)                                                    8900
      IF(IGO.EQ.2.OR.ANS.EQ.4HBACK) WRITE(6,844) NRUN                    8910
      IF(IGO.EQ.2) GO TO 1001                                            8920
      IF(ANS.EQ.4HBACK) GO TO 1004                                       8930
 3622 CONTINUE                                                           8940
      IF(IFLAG.NE.0) GO TO 3513                                          8950
      WRITE(6,804)SSRES,SSREG,SSTOT,SSY,YBAR,RESMSQ,REGMSQ,RESBAR,       8960
     *SDVRES,RSQ,R                                                       8970
      IF(ANS16.EQ.3HYES.AND.NOWGHT.NE.1) WRITE(6,806)FUNC,WRSMSQ,        8980
     *WRSBAR,WSDVRS                                                      8990
      IF(NFUNC.EQ.1) GO TO 364                                           9000
      DO 363 II=1,NFUNC                                                  9010
      WRITE(6,807)II,RSS(II)                                             9020
  363 IF(ANS16.EQ.3HYES.AND.NOWGHT.NE.1) WRITE(6,808)II,WRSS(II)         9030
  364 IF(RSQ.LT.50.) WRITE(6,810)                                        9040
      WRITE(6,812)                                                       9050
      NOUTL=0                                                            9060
      NSAVE=1                                                            9070
      DO 365 I=1,NOBS                                                    9080
      N=LABEL(I)                                                         9090
      IF(N.EQ.NSAVE+1) WRITE(6,813)N                                     9100
      NSAVE=N                                                            9110
      ANRDV=RES(I)/SDVRES                                                9120
      IF(NIND.EQ.1)WRITE(6,814)I,X(I),Y(I),YEST(I),RES(I),DIFPCT(I)      9130
     *,ANRDV                                                             9140
      IF(NIND.GT.1)WRITE(6,816)I,X(I),Z(I),Y(I),YEST(I),RES(I),          9150
     *DIFPCT(I),ANRDV                                                    9160
      IF(ABS(ANRDV).LT.T) GO TO 365                                      9170
      NOUTL=NOUTL+1                                                      9180
      IOUTL(NOUTL)=I                                                     9190
  365 CONTINUE                                                           9200
      WRITE(6,818)                                                       9210
      IF(NIND.LT.3) GO TO 367                                            9220
      WRITE(6,817)(I,I=3,9)                                              9230
      DO 366 I=1,NOBS                                                    9240
  366 WRITE(6,819)I,(XX(J,I),J=3,NIND)                                   9250
      WRITE(6,818)                                                       9260
  367 IF(NOUTL.EQ.0) GO TO 3701                                          9270
      WRITE(6,820) NRSDF                                                 9280
      DO 370 I=1,NOUTL                                                   9290
```

P01

```
      IF(ICNSTR.NE.1) GO TO 804                                          17750
      WRITE(6,8000) FUNC,(F(L),L=1,NPR)                                  17760
      WRITE(6,8005)                                                      17770
  804 WRITE(6,8010) AUPP,F2(I),I,F(I)                                    17780
  806 CONTINUE                                                           17790
      IF(ICNSTR.EQ.0) GO TO 814                                          17800
      IF(IPRINT.EQ.-1.AND.NVIOL.GT.1) WRITE(IW,8031)NVIOL                17810
      WRITE(6,8015)                                                      17820
  808 CALL READ(1,AR)                                                    17830
      IF(AR(1).EQ.4HBACK.OR.IGO.EQ.2) GO TO 810                          17840
      IF(AR(1).EQ.2HNO) GO TO 812                                        17850
      IF(AR(1).EQ.3HYES) GO TO 814                                       17860
      WRITE(6,8030)                                                      17870
      GO TO 808                                                          17880
  810 IFAULT=5                                                           17890
      WRITE(6,8020)                                                      17900
      RETURN                                                             17910
  812 IQUAD=0                                                            17920
      WRITE(6,8025)                                                      17930
      RETURN                                                             17940
  814 CONTINUE                                                           17950
 8000 FORMAT(" * THE RESIDUAL SUM OF SQUARES MINIMUM (",D12.6,           17960
     *") AT THE PARAMETER VALUES"/" *"/(" *",7E10.4))                    17970
```

002

```
 8005 FORMAT(" *"/" * SEEM TO BE CONSTRAINED BY THE"/" *"  )             17980
 8010 FORMAT(" *",5X,A5," LIMIT (",E11.5,") OF PAR.NO.",I2,               17990
     *"  (",E11.5,")"  )                                                18000
 8015 FORMAT(" *"/" * A STATISTICAL EVALUATION OF THE PARAMETERS UNDER",  18010
     *" THESE CONDITIONS MAY FAIL."/                                     18020
     *" * ENTER B10 OR B11 IF YOU WANT A NEW RUN WITH DIFFERENT"/        18030
     *" * PARAMETER LIMITS OR DIFFERENT INITIAL PARAMETER ESTIMATES"/    18040
     *" * ENTER -YES- IF YOU WANT TO CONTINUE OR"/                       18050
     *" * ENTER -NO- IF YOU WANT TO CONTINUE BUT WITHOUT A STATISTICAL", 18060
     *" EVALUATION"//  )                                                18070
 8020 FORMAT(/1X,77(1H-)// )                                            18080
 8025 FORMAT(/" * REMEMBER YOUR INPUT UNDER REQUEST NO.3 HAS NOW",      18090
     *" BEEN CHANGED TO -NO-"/ )                                        18100
 8030 FORMAT(/" INPUT ERROR, YOUR ANSWER MUST BE ONE OF THE FOLLOWING"/ 18110
     *" FOUR,   Y   N   B   R     - TRY AGAIN -"/  )                   18120
      IF(IPRINT)233,232,232                                             18130
  232 WRITE(IW,301)                                                     18140
  301 FORMAT(/1X,13(1H*)," FITTING OF QUADRATIC SURFACE IN REGION OF",   18150
     *" MINIMUM ",13(1H*) / )                                           18160
  233 NEVAL=0                                                           18170
      NFIX=0                                                            18180
      B=SIMP/100.                                                       18190
      SIMP2=FUNC+B                                                      18200
      SIMP2=AMAX1(B,SIMP2)                                              18210
      DO 201 I=1,NP1                                                    18220
  204 IF(H(I).GE.1D50) GO TO 2030                                       18230
      TEST=DABS(H(I)-FUNC)                                              18240
      IF(TEST-SIMP2)202,201,201                                         18250
  202 DO 203 J=1,NOP                                                    18260
      IF(STEP(J).NE.0.0) G(I,J)=(G(I,J)-F(J))+G(I,J)                    18270
  203 PSTST(J)=G(I,J)                                                   18280
      CALL FUNCTN(PSTST,H(I))                                           18290
      NEVAL=NEVAL+1                                                     18300
      GO TO 204                                                         18310
 2030 DO 2031 J=1,NOP                                                   18320
      IF(STEP(J).EQ.0.) GO TO 2031                                      18330
      IF(G(I,J).GT.F2(J)) G(I,J)=F2(J)                                  18340
      IF(G(I,J).LT.F1(J)) G(I,J)=F1(J)                                  18350
 2031 PSTST(J)=G(I,J)                                                   18360
      CALL FUNCTN(PSTST,H(I))                                           18370
      NEVAL=NEVAL+1                                                     18380
      NFIX=NFIX+1                                                       18390
  201 CONTINUE                                                          18400
      A0 = H(1)                                                         18410
      DO 205 I=1,NAP                                                    18420
      I1=I+1                                                            18430
      DO 206 J=1,NOP                                                    18440
  206 PSTAR(J)=(G(1,J)+G(I1,J))/2.0                                     18450
      CALL FUNCTN(PSTAR,AVAL(I))                                        18460
      IF(AVAL(I).GT.(H(1)+H(I1))/2D0)AVAL(I)=H(1)+H(I1)-AVAL(I)         18470
  205 NEVAL=NEVAL+1                                                     18480
      DO 207 I=1,NAP                                                    18490
      I1=I-1                                                            18500
      I2=I+1                                                            18510
      IF(I1.LT.1)GOTO207                                                18520
      DO 208 J=1,I1                                                     18530
      J1=J+1                                                            18540
      DO 209 K=1,NOP                                                    18550
  209 PSTST(K)=(G(I2,K)+G(J1,K))/2.0                                    18560
      CALL FUNCTN(PSTST,HSTST)                                          18570
      IF(HSTST.GT.(H(I2)+H(J1))/2D0)HSTST=H(I2)+H(J1)-HSTST             18580
      NEVAL=NEVAL+1                                                     18590
```

P02

```
      FNEW(I)=F1(I)                                                     27050
      STPNEW(I)=STEP(I)                                                 27060
      IF(I.GT.IP1) GO TO 55                                             27070
      IF(F1(I).NE.F2(I)) IP1=I                                          27080
   55 IF(F1(I).NE.F2(I)) IP2=I                                          27090
      DEL(IP1)=(F2(IP1)-F1(IP1))/9.                                     27100
      DEL(IP2)=(F2(IP2)-F1(IP2))/9.                                     27110
      ISAVE=1                                                           27120
      JSAVE=1                                                           27130
      SSMIN=1.E99                                                       27140
      SSMAX=0.                                                          27150
      DO 60 J=1,10                                                      27160
      FNEW(IP1)=F1(IP1)+FLOAT(J-1)*DEL(IP1)                             27170
      P1(J)=FNEW(IP1)                                                   27180
      DO 60 I=1,10                                                      27190
      FNEW(IP2)=F1(IP2)+FLOAT(I-1)*DEL(IP2)                             27200
      P2(I)=FNEW(IP2)                                                   27210
      CALL LSQ(FNEW,SSS(I,J))                                           27220
      IF(SSS(I,J).GE.SSMIN) GO TO 60                                    27230
      SSMIN=SSS(I,J)                                                    27240
      ISAVE=I                                                           27250
      JSAVE=J                                                           27260
   60 IF(SSS(I,J).GT.SSMAX) SSMAX=SSS(I,J)                              27270
```



```
      FNEW(IP1)=P1(JSAVE)                                               27280
      FNEW(IP2)=P2(ISAVE)                                               27290
      STPNEW(IP1)=DEL(IP1)/2.                                           27300
      STPNEW(IP2)=DEL(IP2)/2.                                           27310
      D=(SSMAX-SSMIN)/9.                                                27320
      DO 65 I=1,10                                                      27330
   65 A(I)=SSMIN+(I-1)*D                                                27340
      A(10)=A(10)+0.01*D                                                27350
      WRITE(6,1030)IP1,(I,P1(I),I=1,10)                                 27360
      WRITE(6,1040)IP2,(I,P2(I),I=1,10)                                 27370
      IF(ANS16.EQ.2HNO) WRITE(6,1050)                                   27380
      IF(ANS16.EQ.3HYES) WRITE(6,1060)                                  27390
      WRITE(6,1070)IP1,IP2                                              27400
      WRITE(6,1075)(I,I=1,10)                                           27410
      DO 67 I=1,10                                                      27420
      WRITE(6,1080)(SSS(I,J),J=1,9,2)                                   27430
   67 WRITE(6,1085)(SSS(I,J),J=2,10,2)                                  27440
      DO 120 J=1,10                                                     27450
      DO 120 I=1,10                                                     27460
      IF(J.EQ.1) GO TO 70                                               27470
      IF(J.EQ.10) GO TO 80                                              27480
      PDEV(I,J)=SIGNOF(SSS(I,J+1)-SSS(I,J-1))                           27490
      GO TO 90                                                          27500
   70 PDEV(I,J)=SIGNOF(4.*SSS(I,J+1)-3.*SSS(I,J)-SSS(I,J+2))            27510
      GO TO 90                                                          27520
   80 PDEV(I,J)=SIGNOF(-4.*SSS(I,J-1)+3.*SSS(I,J)+SSS(I,J-2))           27530
   90 K=J+10                                                            27540
      IF(I.EQ.1) GO TO 100                                              27550
      IF(I.EQ.10) GO TO 110                                             27560
      PDEV(I,K)=SIGNOF(SSS(I+1,J)-SSS(I-1,J))                           27570
      GO TO 115                                                         27580
  100 PDEV(I,K)=SIGNOF(4.*SSS(I+1,J)-3.*SSS(I,J)-SSS(I+2,J))            27590
      GO TO 115                                                         27600
  110 PDEV(I,K)=SIGNOF(-4.*SSS(I-1,J)+3.*SSS(I,J)+SSS(I-2,J))           27610
  115 CONTINUE                                                          27620
      DO 120 L=1,9                                                      27630
  120 IF(SSS(I,J).GE.A(L).AND.SSS(I,J).LE.A(L+1)) CON(I,J)=AL(L)        27640
      WRITE(6,1090)IP1,IP2                                              27650
      WRITE(6,1100)((PDEV(I,J),J=1,20),I=1,10)                          27660
      WRITE(6,1110)                                                     27670
      DO 130 I=1,10                                                     27680
      WRITE(6,1120)(CON(I,J),J=1,10)                                    27690
      IF(I.EQ.10) GO TO 130                                             27700
      WRITE(6,1130)I,A(I),A(I+1)                                        27710
  130 CONTINUE                                                          27720
      RETURN                                                            27730
  135 NP=IFIX(NPOINT**(1./FLOAT(NEWNOP))+.001)                          27740
  140 NEVAL=NP**NEWNOP                                                  27750
      IF(NEVAL.LE.2187) GO TO 150                                       27760
      NP=NP-1                                                           27770
      GO TO 140                                                         27780
  150 CONTINUE                                                          27790
      IF(NOP.EQ.7) GO TO 170                                            27800
      K=NOP+1                                                           27810
      DO 160 I=K,7                                                      27820
      F1(I)=0.                                                          27830
  160 F2(I)=0.                                                          27840
  170 CONTINUE                                                          27850
      DO 180 I=1,7                                                      27860
  180 DEL(I)=(F2(I)-F1(I))/FLOAT(NP-1)                                  27870
      SSMIN=1.E98                                                       27880
      NEVAL=0                                                           27890
```

```
      FNEW(I)=F1(I)                                                     27050
      STPNEW(I)=STEP(I)                                                 27060
      IF(I.GT.IP1) GO TO 55                                             27070
      IF(F1(I).NE.F2(I)) IP1=I                                          27080
   55 IF(F1(I).NE.F2(I)) IP2=I                                          27090
      DEL(IP1)=(F2(IP1)-F1(IP1))/9.                                     27100
      DEL(IP2)=(F2(IP2)-F1(IP2))/9.                                     27110
      ISAVE=1                                                           27120
      JSAVE=1                                                           27130
      SSMIN=1.E99                                                       27140
      SSMAX=0.                                                          27150
      DO 60 J=1,10                                                      27160
      FNEW(IP1)=F1(IP1)+FLOAT(J-1)*DEL(IP1)                             27170
      P1(J)=FNEW(IP1)                                                   27180
      DO 60 I=1,10                                                      27190
      FNEW(IP2)=F1(IP2)+FLOAT(I-1)*DEL(IP2)                             27200
      P2(I)=FNEW(IP2)                                                   27210
      CALL LSQ(FNEW,SSS(I,J))                                           27220
      IF(SSS(I,J).GE.SSMIN) GO TO 60                                    27230
      SSMIN=SSS(I,J)                                                    27240
      ISAVE=I                                                           27250
      JSAVE=J                                                           27260
   60 IF(SSS(I,J).GT.SSMAX) SSMAX=SSS(I,J)                              27270
```



```
      FNEW(IP1)=P1(JSAVE)                                               27280
      FNEW(IP2)=P2(ISAVE)                                               27290
      STPNEW(IP1)=DEL(IP1)/2.                                           27300
      STPNEW(IP2)=DEL(IP2)/2.                                           27310
      D=(SSMAX-SSMIN)/9.                                                27320
      DO 65 I=1,10                                                      27330
   65 A(I)=SSMIN+(I-1)*D                                                27340
      A(10)=A(10)+0.01*D                                                27350
      WRITE(6,1030)IP1,(I,P1(I),I=1,10)                                 27360
      WRITE(6,1040)IP2,(I,P2(I),I=1,10)                                 27370
      IF(ANS16.EQ.2HNO) WRITE(6,1050)                                   27380
      IF(ANS16.EQ.3HYES) WRITE(6,1060)                                  27390
      WRITE(6,1070)IP1,IP2                                              27400
      WRITE(6,1075)(I,I=1,10)                                           27410
      DO 67 I=1,10                                                      27420
      WRITE(6,1080)(SSS(I,J),J=1,9,2)                                   27430
   67 WRITE(6,1085)(SSS(I,J),J=2,10,2)                                  27440
      DO 120 J=1,10                                                     27450
      DO 120 I=1,10                                                     27460
      IF(J.EQ.1) GO TO 70                                               27470
      IF(J.EQ.10) GO TO 80                                              27480
      PDEV(I,J)=SIGNOF(SSS(I,J+1)-SSS(I,J-1))                           27490
      GO TO 90                                                          27500
   70 PDEV(I,J)=SIGNOF(4.*SSS(I,J+1)-3.*SSS(I,J)-SSS(I,J+2))            27510
      GO TO 90                                                          27520
   80 PDEV(I,J)=SIGNOF(-4.*SSS(I,J-1)+3.*SSS(I,J)+SSS(I,J-2))           27530
   90 K=J+10                                                            27540
      IF(I.EQ.1) GO TO 100                                              27550
      IF(I.EQ.10) GO TO 110                                             27560
      PDEV(I,K)=SIGNOF(SSS(I+1,J)-SSS(I-1,J))                           27570
      GO TO 115                                                         27580
  100 PDEV(I,K)=SIGNOF(4.*SSS(I+1,J)-3.*SSS(I,J)-SSS(I+2,J))            27590
      GO TO 115                                                         27600
  110 PDEV(I,K)=SIGNOF(-4.*SSS(I-1,J)+3.*SSS(I,J)+SSS(I-2,J))           27610
  115 CONTINUE                                                          27620
      DO 120 L=1,9                                                      27630
  120 IF(SSS(I,J).GE.A(L).AND.SSS(I,J).LE.A(L+1)) CON(I,J)=AL(L)        27640
      WRITE(6,1090)IP1,IP2                                              27650
      WRITE(6,1100)((PDEV(I,J),J=1,20),I=1,10)                          27660
      WRITE(6,1110)                                                     27670
      DO 130 I=1,10                                                     27680
      WRITE(6,1120)(CON(I,J),J=1,10)                                    27690
      IF(I.EQ.10) GO TO 130                                             27700
      WRITE(6,1130)I,A(I),A(I+1)                                        27710
  130 CONTINUE                                                          27720
      RETURN                                                            27730
  135 NP=IFIX(NPOINT**(1./FLOAT(NEWNOP))+.001)                          27740
  140 NEVAL=NP**NEWNOP                                                  27750
      IF(NEVAL.LE.2187) GO TO 150                                       27760
      NP=NP-1                                                           27770
      GO TO 140                                                         27780
  150 CONTINUE                                                          27790
      IF(NOP.EQ.7) GO TO 170                                            27800
      K=NOP+1                                                           27810
      DO 160 I=K,7                                                      27820
      F1(I)=0.                                                          27830
  160 F2(I)=0.                                                          27840
  170 CONTINUE                                                          27850
      DO 180 I=1,7                                                      27860
  180 DEL(I)=(F2(I)-F1(I))/FLOAT(NP-1)                                  27870
      SSMIN=1.E98                                                       27880
      NEVAL=0                                                           27890
```