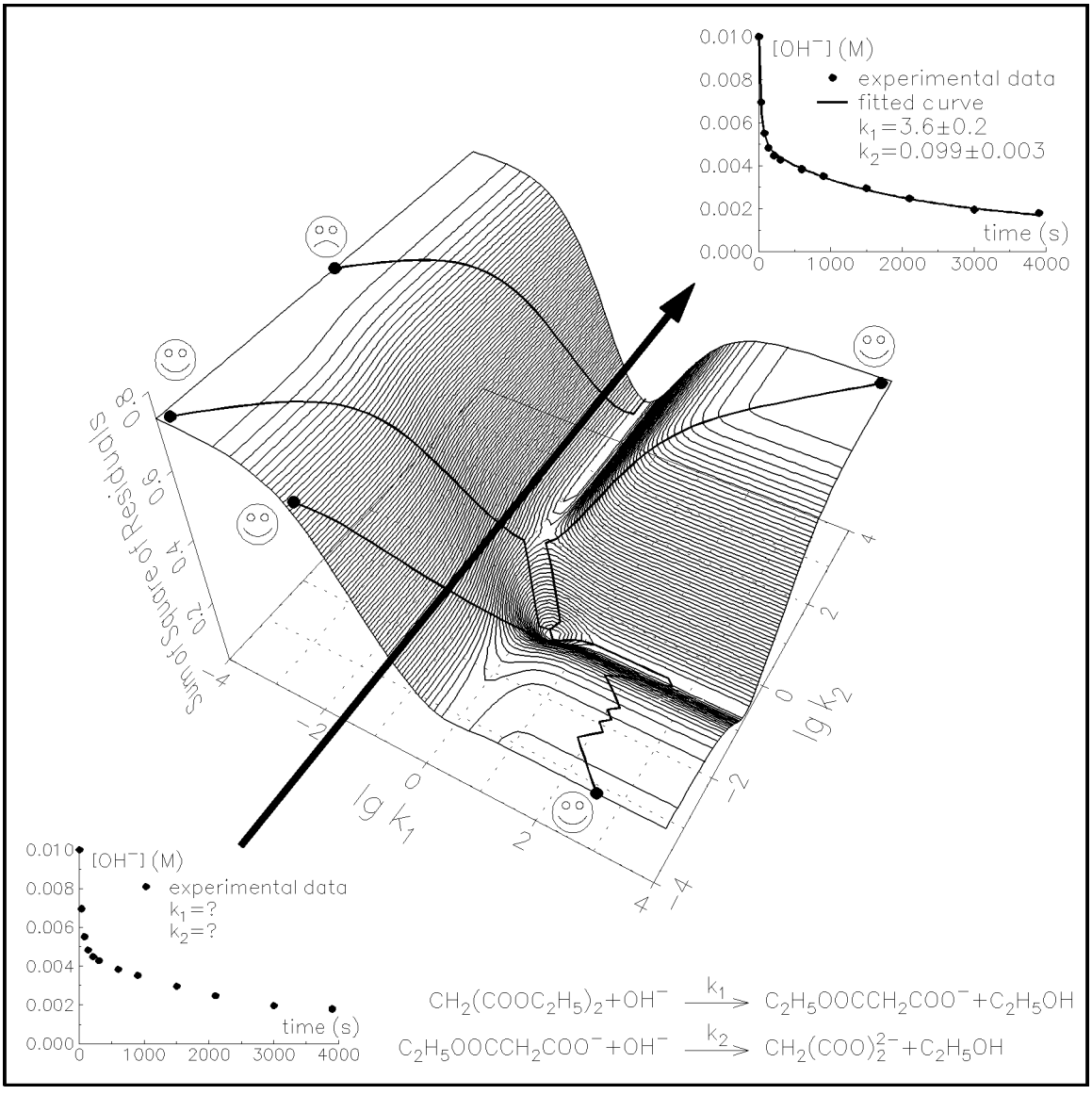
A Comprehensive Program Package for Fitting Parameters of Chemical Reaction Mechanisms.

ZITA
Version 4.1



"If a program is useful, it will have to be changed.
If a program is useless, it will have to be documented."

The Laws of Computer Programming [17]

Reference Manual

ZITA Version 4.1

A Comprehensive Program Package for
Fitting Parameters of Chemical
Reaction Mechanisms.

Developed by Gábor Peintler

Institute of Physical Chemistry
Attila József University
H-6701 Szeged, P. O. Box 105.
HUNGARY

THIRD EDITION
May 1997
Printed in Hungary

This documentation was prepared with LaTeX (emTeX version 3.1415) using graphs created in Axum and ZeTeA.

Warning!

Since this program package uses some files from Turbo Pascal Version 5.5 or higher only the owners of this compiler are able to use it!

Gábor Peintler
Institute of Physical Chemistry, Attila József University
H-6720 Szeged, Rerrich Béla tér 1., HUNGARY; (or H-6701, Szeged, P. O. Box 105.)
Telephone/FAX: (36)62-311-943
E-mail: PEINTLER@CHEM.U-SZEGED.HU

Preface

Dear User,

I send my greetings to you as an owner of this Program Package. I hope it will help you in your work. This Manual helps you in getting acquainted with using the Program Package.

Before installing these programs you should read the first chapter of the User's Guide. There are some pieces of information in this chapter which make your decisions easier during installation.

Before setting up your own models and calculating with them, you should go through the tutorials in Chapter 3. Once you have tried out and comprehended the examples included in this chapter, you can be sure that you understand how the Program Package works. Additional examples help your work through the Reference Guide to solve your more complicated problems. If problems arise during the usage of the Program Package, please, let me know!

Finally, I would like to express my gratitude to Tita Tóth, Prof. István Nagypál, Prof. Irving R. Epstein, Prof. Kenneth Kustin, Sándor Kádár and Attila Horváth for their help.

Yours Sincerely,

Gábor Peintler

Contents

List of Figures

List of Tables

List of Texts

Introduction

This program package[1] has been developed for fitting parameters in models of chemical reaction mechanisms. This is a new method for evaluating almost arbitrary combinations of kinetic experimental data.

Summary of the Evaluating Methods in Kinetics

Nowadays, several methods are frequently used to get the rate constants and a possible mechanism from the experiments:

- Probably, the *method of initial rates* is used most frequently. This method might not reveal the full mechanism mostly because the products and intermediates themselves can also affect the rate. A reaction can usually be followed experimentally through arbitrary time interval but the whole time-series cannot be evaluated with this method. It uses only the experimental data measured at the beginning of the reactions. Furthermore, the determination of the initial slope is very uncertain procedure. If every mechanistic step plays proportional role during the whole reaction, this method may be adequate. Only a few reactions, however, have this feature.

- Sometimes *analytical integration* of the ordinal differential equation(s) can be used. The parameters of the integrated equation(s) can be fitted based on the experimental data. This is an exact method which uses all the experimental information; however, this method can only be applied if the supposed mechanism is quite simple. Since this method works very well, many articles detail mathematical procedures for integrating particular differential equations in the literature. These papers show clearly that this method is inapplicable to more complex reactions for either theoretical or practical reasons.

- In principle, *simulation* is able to handle the most complex ordinary differential equations if the user can choose appropriate simulation conditions (e.g. step size, error handling, etc.). But in setting up the possible mechanism the user must employ chemical knowledge, intuition and the results of mathematical sensitivity analysis. These tools are often not enough; in practice, it is almost impossible to get the best-fitted parameter set for a more complicated supposed mechanism if the chemist uses only the results of simulations.

[1] The name $\mathcal{ZITA}$ means nothing in chemistry. It is just a beautiful Hungarian first name for females.

- The transformation of differential equations to a simpler set of nonlinear algebraic and/or differential equation(s) by the *steady-state approximation* can also be useful. This method is often used to find a complicated rate law from a supposed mechanism. Frequently this approximation is used with either the initial rate method or simulation. However, this procedure is useless if the intermediates are not reactive enough.

- The *sensitivity analysis* [23]1 is frequently considered as a method to support a mechanism for a reaction. *It is not true!* For carrying out this mathematical procedure, all the elementary and mechanistic steps as well as their rate constants have to be well known. If it is so, the sensitivity analysis can quantitatively describe the effect of changing parameter values on the simulated concentration *vs.* time curves. In other words, the *properties of a theoretical model* can be studied by the sensitivity analysis.

  However, if we want to find a mechanism for a reaction, initially we know only experimental concentration *vs.* time curves. Setting up a chemically acceptable mechanism and computing its parameter values is the real goal in this case. In spite of this problem, these things are initially given for a sensitivity analysis. Its real question is the stability of the calculated concentration *vs.* time curves as a function of rate constants.

Features of the Program Package

As it can be seen from the previous summary, the evaluating methods in kinetics cannot always help to find a chemically possible mechanism within the experimental errors. This program package attempts to exploit the advantages of these methods and to avoid their disadvantages. Of course, this package is not the final solution, but it offers a more efficient and much more precise way to get reliable kinetic data.

What can the user do with this package? The answer is simple: if the user has practically any kind of kinetic experimental data and a supposed mechanism, s/he is able to calculate the best-fitted parameter set for that model. $\mathcal{ZITA}$ uses simulations and fitting methods together for computing this parameter set, thus also providing the user with the statistical data of the parameters. In this way the results from different models become truly comparable.

The following incomplete list contains the main features of the program package:

- Besides the concentrations, the *primary measured experimental data can also be used as input*. In this way *other chemical parameters* (e.g. molar absorption, standard electrode potential) *can also be fitted* together with the rate constants. This feature makes it possible to evaluate those kinds of experiments from which the concentrations cannot be calculated explicitly. Another advantage is that the experimental errors become comparable to the differences between experiments and calculated values of the supposed model.

- The *number of experimental curves used for fitting is practically unlimited* — only the size of the hard drive determines the upper limit (e.g. the writer used $\mathcal{ZITA}$ to evaluate

~20000 experimental data pairs simultaneously). $\mathcal{ZITA}$ is also able to evaluate different kinds of experimental data together. In this way the *concentration dependence* of the rate equations can be handled correctly. The *data files have a free form*; they can contain columns of several measured quantities, and the package can handle any missing values as well. The only restriction is that one column of an experimental data file cannot be longer than 15400 data. More data can also be used but they have to be placed into several files.

- *There is no restriction concerning the ordinary differential equation system.* The numbers of the species and equations are limited only by the size of the computer's memory. The user can enter either elementary and mechanistic steps or more complex rate equations derived from elsewhere (e.g. from the steady-state approximation).

- Using only elementary or mechanistic steps in the model, the program package offers a *very efficient way to create the subroutines associated with the ordinary differential equation system* in question. Using this method *the user need not know anything about the programming language*. S/he needs only to know the stoichiometry and order numbers of the species in the model. In this case not only the batch reactor but also *several other types of reactors* are handled automatically.

- Depending on the applied experimental method, the program package can handle some specialities like *dead* and/or *mixing time* in stopped-flow measurements, *changing volumes* in gas reactions, etc. Of course, $\mathcal{ZITA}$ is not able to handle all the specialities automatically, however, *the user is able to teach the program package* for doing it.

- $\mathcal{ZITA}$ includes a *simple and very fast graphics program* to help the user work more efficiently. This program can also be used independently from the program package.

- If the fitting takes a long time (e.g. overnight) and the user is not present when an error occurs, the *error detecting feature* helps to find the reasons of the errors.

- If the user's task is not very demanding, the program package can be used even faster. In this case *most of the necessary files are created automatically*, *several models can be evaluated in a single run* and *the user can interrupt and restart the fitting process* as many times as s/he wants.

- The user can employ *either fixed or fitted parameters*. The number of the fixed parameters are unlimited, however, the number of the fitted values cannot be higher than about 100 (depending on the computer's memory). The program package gives the user a *detailed statistical description* of the results after every iteration.

- The fitting process is divided into parts. This feature gives the user *complete control over the evaluation process*.

- The full installation requires only about 700 Kbytes space on a hard drive (the minimum is about 500 Kbytes). The program package can work with 384 Kbytes conventional memory but at least 480 Kbytes are recommended.

These features make possible the evaluation of virtually any kind of experimental data and model — but using the program package can be difficult. The next few paragraphs outline the difficulties and ways the user can overcome them.

- *The user cannot use ZTIA as a 'black box'!* Some kinds of program packages (e.g. spreadsheets, word processors, etc.) help the user to carry out tasks much *more easily* than it is possible without the packages. These tasks usually have a well-defined algorithm and a program of this type can be used as a 'black box', in which a user places information and gets out results without having to participate very much in the in-between steps. ZTIA has another purpose: it helps to solve problems which are *impossible* to solve without automated assistance. Since ZTIA is not a foolproof evaluating tool (like the others mentioned above), the *user must have control over the calculations*, and must also have some idea of how the package works. A very detailed tutorial, reference book and several key examples help the user to learn how to use the package. *The user must devote sufficient time to these learning aids!*
- Because of the many options in ZTIA the user may get lost along the way while creating the necessary files. In this case s/he should study either the appropriate syntax in the Reference Guide or the examples. Almost every possibility is illustrated by an example.
- Most users will not use all the options of this package but rather only a few ones. In this way solving the first problem requires much more time than the later ones. In other words, the beginning will probably be difficult, but with experience the user will enjoy the real efficiency of this package.
- Only that user can use the program package efficiently who knows the basic DOS command and the usage of environment variables.

Even if the data files have a complicated syntax (which is certainly possible), solving any problem follows this general outline:

- The user creates several types of files:
  - the files containing the experimental curves and their initial conditions;
  - the file containing the user's model;
  - the data files containing the parameters and conditions for the simulation and fitting; and
  - the organizer file which creates an iteration loop using the data files and executable programs.
- The user must run the organizer file to compute the best-fitted parameter set. They are written in the `RESULT.*` or `MODEL*.RES` files.
- With the help of these results, the user must make the necessary modifications into the files created earlier and use the organizer file again, until the final result is reached.

Part I

User's Guide

"It is better to solve a problem with a crude approximation and know the truth, $\pm 10\,\%$, than to demand exact solution and not know the truth at all."

Thumb's First Postulate [17]

Chapter 1

Installation and Some Advice

This chapter contains information with which the program package ($\mathcal{ZTA}$) can be successfully installed on a floppy diskette or on a hard disk. It is essential to read this material before installation! The reading of the other chapters is recommended only after the successful installation, because the user cannot study the first example of the User's Guide without it.

1.1 Hardware and Software Environment

$\mathcal{ZTA}$ has been developed on the IBM PC, because this is the most commonly used computer all over the world. Its most important operating system is DOS and one of the most frequently used program languages is PASCAL. One of the PASCAL's implementations is called Turbo Pascal (denoted as TP) [1], which is able to cooperate with DOS [2] and which creates fast executable codes. For these reasons, the programs were developed entirely in TP Version 5.5. The user is not required to know this language; however, in some cases the syntax of assignment statements in PASCAL should be known. These cases will be discussed in detail later on. The necessary parts of TP *are not included* in $\mathcal{ZTA}$, so *you need to obtain them*! *Section 1.2* contains all the necessary information concerning these parts. $\mathcal{ZTA}$ requires DOS Version 3.00 or higher. An ASCII-editor is also needed for constructing data files. Any editor can be used, but a more efficient editor, e.g. Kedit, Norton Editor, etc. is recommended.

What is necessary for the usage of $\mathcal{ZTA}$ regarding hardware? At least an IBM XT with a hard disk or a floppy diskette (which has more than 660 Kbytes free space) is essential. A math coprocessor is highly recommended, because there are many floating point operations.

$\mathcal{ZTA}$ assumes at least 384 Kbytes memory, but for solving more complicated problems larger memory size is recommended. The best situation occurs when the user's computer has 640 Kbytes memory size or more. This is enough for evaluating data from several hundred experimental curves. The extended memory is used only if its size is bigger than 280 Kbytes, however, neither EMS nor XMS memory [2] is necessary to use the program package.

The computer must have a graphic card if the user wants to represent the curves on figures, too. *Section 6.1* details each usable graphics card. The figures can be plotted on

	Name	Size in bytes
1.	INSTALL.EXE	59984
2.	GOZITA.EXE	31328
3.	NEWEXE.BAT	979
4.	COMPILE.BAT	1165
5.	FINDER.BAT	1322
6.	ZITA4ALL.PAS	7612
7.	MAKEODE.PAS	20419

	Name	Size in bytes
8.	ZITA.PAS	41206
9.	JC.PAS	13465
10.	GNM.PAS	19546
11.	TASK.PAS	31605
12.	ODES.SPL	3330
13.	DRAWER.EXE	132192
14.	EXAMPLES.EXE	163651

Table 1.1: Files of the installation disk(s).

different types of screens. It is possible to create a hardcopy on any kind of a printer if the user has a resident program for doing screen capture, e.g. Pizazz, Hijaak, etc.

1.2 Before Installation

The original diskette(s) of ZITA contain(s) the files listed in *Table 1.1*. If the original version happens to be on two 360 Kbytes floppy diskettes, these files are divided into two parts: the second diskette contains `DRAWER.EXE` and `EXAMPLES.EXE` files[1] and all the other files are stored on the first diskette. Otherwise all the files are included on one floppy diskette. The user should make a copy of original diskette(s) before installation, e.g., using the `DISKCOPY` DOS command. In this way the user is able to avoid major problems if the original diskette(s) go(es) wrong.

```
COPY␣c:\tp\TPC.EXE␣a:*.*
COPY␣c:\tp\TURBO.TPL␣a:*.*
c:\tp\TPUMOVER.EXE␣a:\TURBO.TPL␣/-OVERLAY␣/-PRINTER[2]
```

Text 1.1: DOS commands for completing the first installation disk.

Two further files of TP are necessary for working with ZITA. These files are the original form of `TPC.EXE` and a modified form of `TURBO.TPL` of either Version 5.5 or higher. Furthermore the program named `TPUMOVER.EXE` is also needed temporarily. These files should be copied to the first disk before installation[3]. This can be carried out by the three DOS

[1] The second diskette may also include a utility named `ODELATEX.EXE`. It is not a necessary part of ZITA but it can be useful for LaTeX users. This program transforms a `*.ODE` file into a LaTeX file. The `*.ODE` file contains the actual mechanism in matrix form which cannot be studied easily by a human being. The LaTeX file presents the mechanism in the traditional way so it can be checked fast.

There is also another utility called `PCXTIMES.EXE`. It enlarges the PCX-files generated by DRAWER.EXE. The utilities are not detailed in this book since their usage is written on the screen when they started without any DOS parameter.

[2] With Turbo Pascal Version 6.0 or higher the '/' signs must be omitted from this line!

[3] The original disk cannot contain these two files, because it would directly violate the license agreement and any copyright laws of the USA.

commands listed in *Text 1.1*[4]. It is assumed that during the execution of these commands the user's disk is placed in the **A** drive and the three necessary files (`TPC.EXE`, `TURBO.TPL` and `TPUMOVER.EXE` are in the `c:\tp` directory. If this is not the case, the user must substitute the appropriate drive letter and directory names for the ones given. These names are written with small letters. If the commands are carried out without any error message, the installation disk comprises all the necessary files so the installation can be started. The installation disk (or disk #1) must be placed into the A or B drive and the following command begins the installation:

`A:\INSTALL` or `B:\INSTALL` and <ENTER>

1.3 Process of Installation

WARNING! If You are installing version 5.0 or higher, skip the rest of this chapter and follow the instructions of `INSTALL.EXE`*!*

ZITA is a program package that can be used by several users who work with the same installed version but who have their own libraries[5] independent of each other. This is the reason why `INSTALL.EXE` works in two steps.

During the first step the program creates a common library containing those files that can be used by every user. The common library connected to the root directory of a hard disk is a suitable location for this library. In this case accessing files is not very complicated, but the files of ZITA's common library are separated from the other contents of the actual disk. The full pathname of the common library cannot be longer than 15 characters.

The user's libraries can be defined in the second step. It is no use choosing the root directory of a drive for a user's library because the number of files is limited between 64 and 1024. This restriction does not apply to subdirectories. Every user's library contains seven files immediately after installation. Three files (`ZITA4#1.BAT`, `CONFZITA.4#1` and `AUTOZITA.4#1`) set up the configuration wanted by the user. The other files are named `*.BAT`[6], and users can use the programs of the common library through these. It is not possible to change `*.BAT` files, but there are no any other restrictions concerning the user's work. The full pathname of the user's library cannot be longer than 15 characters.

If the user does not understand a question, does not know an answer or the definition of a notion, he should look it up in [2] or in the User's Manual of his computer. The user can continue installation once the answer is known.

During the installation, `INSTALL.EXE` reads the necessary information from the computer's `CONFIG.SYS` and `AUTOEXEC.BAT` files. However, the lines over the first 100 lines are regarded in `CONFIG.SYS`.

[4] The important spaces are indicated with '␣' sign in this book.

[5] A library means a DOS directory where files associated with ZITA are stored. It can be either common or user's library.

[6] Two special DOS characters will be used in filenames where necessary. These characters are the asterisk (*) and the question mark (?) and they are called wildcards. Their detailed explanation can be found in [2].

As the original disk(s) contain(s) only the source texts of some necessary executable programs, `INSTALL.EXE` tries to compile them. These programs are named `MAKEODE.EXE`, `JC.EXE`, `GNM.EXE` and `TASK.EXE`. If there is not enough free space on the destination disk for the resulting `*.EXE` files, `INSTALL.EXE` sends a warning message but does not interrupt the installation. The same thing happens if the memory of the computer is too small to store `INSTALL.EXE`, the compiler and the program to be compiled. In this case the installed version cannot be used until the executable programs are created by means of `NEWEXE.BAT` (it is discussed in *Section 6.2*). However, the use of this program is somewhat complicated. Therefore another way is offered for the user who is not very familiar with DOS. On receiving a warning message, the user should interrupt installation, create more free space on the destination disk or remove resident programs from memory. After this, the installation can be started again.

`INSTALL.EXE` writes some pieces of information on the screen during installation. These refer to the necessary data or to the use of ZITA. It is worth printing them to the printer by the help of 'Print Screen' key. The data printed out can help to repeat the installation if necessary.

If the user has installed the common and user's libraries successfully, `INSTALL.EXE` writes on the screen those commands that start ZITA. It is highly recommended to print out or write down them by the help of 'Print Screen' key! If the user forgets them, he must install the user's library again! These commands can also be placed in a `*.BAT` file. In this way ZITA can be started by typing the name of this `*.BAT` file.

If a common library has been installed, any number of the user's libraries can be defined, which use the same common library. Naturally, each user's library should be activated with different starting commands.

1.4 After Installation

If the installation was successful, the common library and the optional user's library(ies) contain the files listed in *Table 1.2*.

The exact roles of the files are detailed in the Reference Guide[8]. The user will also encounter these files in *Chapter 3*. The aims of four files (`GOZITA.EXE`, `ZITA4#1.BAT`, `CONFZITA.4#1` and `AUTOZITA.4#1`) are detailed in *Section 1.5*, because they start ZITA.

Several parts of ZITA can be used independently. This is important in practice in two cases. The first case is that the *ODEs*[9] solver can also be used as an independent program for simulation. The other is that ZITA contains a particular drawing program for displaying two dimensional figures. Their use is detailed in the Reference Guide.

It should now be clear that the user will be able to get along with ZITA only if the basic notions of DOS are known. If ZITA has already been installed, the user has to know only

[7] These files do not exist naturally if `INSTALL.EXE` has not made them.

[8] `ODELATEX.EXE` remains on the diskette during the installation. This file has to be copied into the common library in order to use it. Starting this program without any DOS parameter details its usage.

[9] See the definition in *Chapter 2*!

Files in common library		*Files in user's library*
GOZITA.EXE	.\PROGRAMS\TPC.EXE	ZITA4#1.BAT
NEWEXE.BAT	.\PROGRAMS\TURBO.TPL	CONFZITA.4#1
COMPILE.BAT	.\PROGRAMS\ZITA4ALL.PAS	AUTOZITA.4#1
FINDER.BAT	.\PROGRAMS\MAKEODE.PAS	NEWEXE.BAT
MAKEODE.EXE[7]	.\PROGRAMS\ZITA.PAS	COMPILE.BAT
JC.EXE[7]	.\PROGRAMS\JC.PAS	FINDER.BAT
GNM.EXE[7]	.\PROGRAMS\GNM.PAS	TASK.BAT
TASK.EXE[7]	.\PROGRAMS\TASK.PAS	DRAWER.BAT
DRAWER.EXE	.\PROGRAMS\ODES.SPL	EXAMPLE.BAT
EXAMPLES.EXE		

Table 1.2: Contents of common and user's library after the installation.

DOS commands concerning files and it is not necessary to understand the structure of DOS. However, in the latter case, the user is more likely to make a mistake for which he is not able to find the reason.

1.5 Starting ZITA

It has been mentioned in *Section 1.2* that `INSTALL.EXE` writes the starting commands on the screen. This section describes what happens when ZITA is started. If the user does not want to understand this procedure, he does not need to read this section after the second paragraph. In this case the user should remember the following: the files named `GOZITA.EXE`, `ZITA4#1.BAT`, `CONFZITA.4#1` and `AUTOZITA.4#1` must not be erased, and the latter three files can be revised only as shown by `INSTALL.EXE`. If the user notices an error later and does not understand the reason, this section should be read at that time!

If the user wants to use ZITA with the usual configuration of the computer, he has to carry out the '`ZITA4#1.BAT`' DOS command in the user's library. This command cannot be used from any utilities only directly from the DOS cursor. The `ZITA4#1.BAT` file sets up the value of ZITA's DOS environment variables. Since ZITA requires at least 233 bytes memory for its own environment variables, the user's `CONFIG.SYS` file must contain the '`Shell=Command.com /P /E:300`' line. If other programs also require memory for their environment variables, the user must increase the value of the number behind the colon. The detailed description of `Shell` command can be found in [2].

When ZITA is working, the part of the computer's memory over 384 Kbytes can be used as a RAM disk. ZITA can require such a configuration, which cannot be used by other programs. Some additional files are required to start ZITA in this case. These files are named `GOZITA.EXE`, `CONFZITA.4#1` and `AUTOZITA.4#1`. The usage of `GOZITA.EXE` is detailed on the screen when the installation is finished (it is worth making a copy of this screen on the printer). When `GOZITA.EXE` is called the following happens:

- The user can define (in a DOS parameter [2]) which drive (A or C) is to include the special `CONFIG.SYS` and `AUTOEXEC.BAT` files to start ZITA. If the root directory of this drive contains files with the same name, the original files are renamed to `ORIGIN.CFG` and `ORIGIN.ATX`. After this the `CONFZITA.4#1` and `AUTOZITA.4#1` files are copied from the base library to the root directory of the chosen drive as `CONFIG.SYS` and `AUTOEXEC.BAT`.
- `GOZITA.EXE` calls the normal restart, warm restart or hardware reset interrupt. The latter two are used only if there is an 'w', 'W'; 'h' or 'H' DOS parameter behind '`GOZITA`'. For example, if the user uses the `EMM386.EXE` program of MS-DOS 5.0, ZITA can only be started by a '`GOZITA`' or '`GOZITA H`' command. '`GOZITA N`' will not work in this case!
- `AUTOEXEC.BAT` and `GOZITA.EXE` recognize the last physically existing drive letter (this drive is a RAM disk usually for storing the temporary results). Consequently, two memory disks should be defined in `CONFZITA.4#1` if ZITA is used on a memory disk. The first RAM disk comprises the necessary libraries and files of ZITA and it is only the second which can be used to the temporary results. No further memory disks can be defined, because the programs of ZITA are able to recognize only the last drive letter.
- The values of the required DOS environment variables are set up in `AUTOEXEC.BAT` file.
- The special `AUTOEXEC.BAT` and `CONFIG.SYS` are deleted. Finally, `ORIGIN.ATX` and `ORIGIN.CFG` are renamed to `AUTOEXEC.BAT` and `CONFIG.SYS`.
- By pressing the **CTRL-ALT-DEL** keys together ZITA can be left and the original configuration can be restored.

During installation certain parts of `ZITA4#1.BAT`, `CONFZITA.4#1` and `AUTOZITA.4#1` may be changed by the user. After installation these files can also be changed using any editor. The user can set up the preferred computer configuration by changing these files. The restrictions on these changes are also enumerated by `INSTALL.EXE`. These are:

- If lines 1–14 of `ZITA4#1.BAT` or `AUTOZITA.4#1` are changed, `GOZITA.EXE` will not run!
- If lines 19-43 of `AUTOZITA.4#1` are changed, the RAM disk cannot be recognized!
- The last 13 lines of `AUTOZITA.4#1` cannot be changed and these lines must stay together! Some user's DOS commands can also be placed after these lines.
- It is highly recommended to write the user's DOS commands between lines 49 and 55 of `AUTOZITA.4#1` or lines 42–48 of `ZITA4#1.BAT`. In this way the structure of this file does not change.

The programs of ZITA work well together with other resident programs, but all possibilities could not be tested. Therefore *Appendix E* contains the results of the examples described in *Chapter 7* or their essential parts. If the user runs the examples and gets correct results, the installed version works correctly.

Only one case has been found when ZITA cannot work with a resident program. If the user works with MS-DOS Version 5.0, `FASTOPEN.EXE` cannot be used, because the programs of ZITA will not be able to recognize a renamed file.

1.6 Using ZITA under Windows

The user must not restart the computer under Windows so the program package can be used through `ZITA4#1.BAT` file and the computer configuration cannot be changed.

The best procedure is to call the DOS prompt as a DOS application and to run the `ZITA4#1.BAT` file from the DOS prompt. The package has several requirements concerning the called `*.PIF` file:

- **Optional Parameter** must include the '`/E:1024`' parameter.
- **Start-up Directory** must be the user's directory.
- **KB Required** is 404 Kbyte or larger.
- **Video Memory** is Text.
- **Display Usage** is Full Screen.
- All the **Memory Options** must be cleared.
- **High Graphics** and **Emulate Text Mode** must be set.
- **Working Directory** must be the user's directory in the Program Item Properties window.

All the other settings have optional values.

Since these restrictions do not meet the requirements of other DOS programs, the user must create a copy the original `DOSPRMPT.PIF` file under another name and must change the necessary options in the copy. The user must open a window in the DOS application group for the revised `*.PIF` file.

The method described above is working under any version of DOS. An experienced DOS user can find more efficient ways but they depend on the actual DOS version due to the different DOS parameters of `COMMAND.COM`.

1.7 Learning the Usage of ZITA

Now the user can install ZITA successfully and set up the appropriate configuration. The following chapters of the User's Guide will show the main principles and many possibilities to use ZITA.

Chapter 2 summarizes the main parts of the mathematical background. Information given here will be assumed to be known later.

Chapter 3 illustrates through some simple chemical problems, how to evaluate kinetic experimental data. These examples are not able to show all the possibilities of ZiTiA, but the main roles of the programs and the effective use of the program package can be understood through them. It is useful to devote at least a day to solve these problems. The time put into these studies will bring results later! If the user tries to understand everything immediately, he is likely to be confused!

Chapter 4 summarizes the contents of the User's Guide. It shows the order and the logical connections of all parts of the computational work by aid of a block scheme. Almost every chapter of the Reference Guide contains a detailed description of a part of this scheme.

After understanding everything in the User's Guide, the user can read the Reference Guide. It contains all information concerning the use of ZiTiA. There are also several complicated possibilities, so ZiTiA contains some more difficult examples in the Reference Guide, too. This part presents the pieces of information through these examples in almost every case.

Recently, the most programs have been menu-driven. The advantage of such programs is that the possibilities can be accessed quickly and easily. But they have also a disadvantage: the number of the options is limited, so special problems cannot be handled by them. This is the reason why ZiTiA was not planned as a menu-driven program.

The number of the experimental methods in chemical kinetics is practically unlimited, but the programmer must give a solving method for the user in each case. Therefore, the program package offers two ways to evaluate the user's experiments and his supposed mechanisms. The first way is more difficult and it requires more work and time, however, any kinetic problems can be solved. The second way has the same efficiency and speed as a menu-driven program, but a few special problems cannot be solved in this way.

Either the User's Guide or the Reference Guide presents the first method at first and the second method after this. It seems to be a wrong order because this takes much more learning time. Indeed, if the fitting of kinetic experiments were a usual procedure, this order would be wrong. But nowadays this procedure is rarely used and experience cannot help the user's work, so this book applies the following order:

- The process of the fitting can be expressed by some mathematical expressions. The more difficult solving method follows the logic of these expressions step by step.

  Chapter 2 details the indispensable relations, so *every user must read and understand this chapter*. *Appendix A* details all the mathematics applied by ZiTiA, but this appendix is recommended only for the advanced users and it is not necessary for everybody.

- All the important notions concerning the simulation and the fitting are detailed when the first solving method is explained. When the second method is described, this book only applies these notions.

- The user who has much experience in simulation and fitting can skip *Chapters 3–4* and he can begin to study the Reference Manual.

  Chapter 3 is a tutorial for those users who are practically beginners either in simulation or fitting. This chapter shows the essential parts of the simulation and fitting through some simple examples but does not give a complete description. If someone did not study *Chapter 3* but the Reference Manual seems to be too difficult, s/he should go back to this chapter.

In this way the user can learn the usage of the program package and its mathematics simultaneously and can avoid those difficulties which come up if the programs are used as a 'black box'.

At last a final remark. Signs, like  or  express that the given paragraph makes a point, which is extremely important, very interesting, especially complicated or  easily forgettable. This convention is employed in the Reference Guide, too, because it also contains a lot of new information.

Chapter 2

A Brief Mathematical Summary

The user is not assumed to know the mathematical background in detail. The most important information can be found in this chapter. Further details are given in the relevant sections. *Appendix A* describes in mathematical terms what kind of problems can be solved using $\mathcal{ZIA}$. This appendix is highly recommended to those users who are acquainted with simulations and their application in reaction kinetics, although, $\mathcal{ZIA}$ can be used efficiently without knowledge of the detailed mathematical background.

2.1 The Mathematical Form of Chemical Reactions

$\mathcal{ZIA}$ can be used for fitting parameters occurring in kinetic models of reaction mechanisms. The mathematical formulation of a reaction mechanism is a first order ordinary differential equation system ($\mathcal{ODE}s$). An $\mathcal{ODE}s$ can be solved by numerical integration[1]. The parameters of an $\mathcal{ODE}s$ can be fitted by comparison of experimental and calculated curves. These parameters are usually rate constants, but they can also be molar absorption coefficients, standard electrode potentials, etc., depending on the experimental methods of the measurement. $\mathcal{ZIA}$ is also suitable for solving non-chemical problems which may be formulated as an $\mathcal{ODE}s$. An $\mathcal{ODE}s$ takes the following general form:

$$\frac{\mathrm{d}\vec{y}}{\mathrm{d}t} = \vec{f}(t, \vec{y}, \vec{k}), \quad t_0 \leq t \leq t_{\mathrm{final}} \tag{2.1}$$

$$\vec{y}(t_0) = \vec{y}_0 \tag{2.2}$$

where

- t is the independent variable (time).
- $\vec{y}$ is the column n-vector of the dependent variables $\left(\vec{y} = (y^1, y^2, \ \ldots \ , y^n)^T\right)$ where n is the number of scalar first order $\mathcal{ODE}s$.
- $\vec{k}$ is the vector of parameters.

[1] $\mathcal{ZIA}$ can use the Pascal-version of ten different types of the Adams-, Gear- and ROW4A-algorithms [3, 4, 5, 6] to solve an $\mathcal{ODE}s$. The choice of the most effective method depends on the concrete model, so it is always worth trying several methods.

$\frac{d\vec{y}}{dt}$ is the derivative of $\vec{y}$ with respect to t.

$\vec{f}$ is an n-vector valued function of t, $\vec{y}$ and $\vec{k}$.

t_0 is the initial value of t.

t_{final} is the final value of the interval of integration.

$\vec{y}_0$ is the initial value of the n-vector of dependent variables.

For our chemical applications, the independent variable is the and the i^{th} dependent variable is the i^{th} concentration or the i^{th} function of concentrations (see below).

2.2 Fitting in General

If one wishes to describe the mechanism of a given reaction, then many data should be measured which contain all the attainable experimental information under the given conditions. If the experimental information is not enough, then it cannot be replaced by any calculations! Independent of the method of measurement, the experimental information is embodied in many curves which present the concentration of a species[2] as a function of time. These are called *measured characteristics* or *properties* and the analogous values calculated on the basis of the reaction model are called *calculated characteristics* or *properties*. The values of these characteristics are *data* or *points*. The concentration as a function of time can be calculated from a proposed mechanism and can be transformed into calculated equivalents of experimental curves. The differences between these curves show to what extent the given mechanism is able to describe the behavior of the chemical reaction in question.

Using only the *ODEs*-solver, a given mechanism can be shown to be plausible, but the accuracy, standard deviation and correlations of parameters cannot be determined. Without these quantities it is impossible to choose between mechanisms that give similar results but are derived from different chemical conceptions. For example, values of parameters cannot be reliably compared to each other without the values of deviations if the same mechanistic step occurs in two different sets of reactions.

Fitting of kinetic data provides a solution to this problem. Fitting with a least-square method means that the minimum value of the function defined in *Eq. (2.3)* is searched for:

$$S(\vec{k}) = \sum_{i=1}^{q} \sum_{j=1}^{c_i} \sum_{l=1}^{n_{i,j}} \left(\left(y(t_{i,j,l}^{expr}, \vec{k})_j^{calc} - y_{i,j,l}^{expr} \right) \cdot W_{i,j,l} \right)^2 \tag{2.3}$$

$$S(\vec{k}_{min}) = \min(S(\vec{k})), \quad \vec{k}_{min} = ? \tag{2.4}$$

where

[2] These curves can also be well-known functions of the concentrations of one or more species, for example, redox potential, sum of the light absorption of several species at a given wavelength, etc.

$S(\vec{k})$ is the sum of squares of residuals to be minimized as a function of the parameters. The residuals are the differences between the experimental and calculated points. $\vec{k}$ is discussed in *Eq. (2.5)*.

q is the number of experimental curves measured at different initial conditions.

c_i is the number of measured characteristics in the i^{th} experimental curve.

$n_{i,j}$ is the number of data of the j^{th} measured characteristic in the i^{th} experimental curve.

$t^{\text{expr}}_{i,j,l}$ is the l^{th} time point in the i^{th} experimental curve concerning the j^{th} measured characteristic.

$y^{\text{expr}}_{i,j,l}$ is the l^{th} point of the j^{th} measured characteristic in the i^{th} experimental curve.

$y(t^{\text{expr}}_{i,l}, \vec{k})^{\text{calc}}_{j}$ is the l^{th} point of the j^{th} measured characteristic in the i^{th} calculated curve.

$\mathrm{W}_{i,j,l}$ is the weighting factor for the residuals.

Fitting of kinetic models has several problems in practice. The effective fitting methods demand the values of the derivatives of the calculated characteristics with respect to a given parameter. However, the analytical forms of the concentration vs. time curves are unknown in most cases in kinetics. Therefore numerical differentiation is necessary, but this increases the computational time considerably. Another problem is that several thousands of experimental data would be necessary to determine the probable mechanism of a complicated reaction. Handling so many data requires a huge memory size and a large and fast storage capacity.

These problems have been solved by the computer technology. Computer speed has increased to the point where fitting large kinetic models on PCs became realistic, and the problem of handling large data sets can be solved by using hard disks and RAM disks.

Fitting methods have been used to study the chemical equilibrium for a long time [9]. The kinetic data would be much more usable if they were determined by the help of a fitting method and not only by simulation. The statistical data provided by parameter estimation give more exact information about reactions.

2.3 Summary of the Fitting Method

ZITA uses the Gauss-Newton-Marquardt method [7, 8, 9] for fitting. This method was chosen for several reasons. Experience with problems in chemical equilibrium shows that it has fast convergence and it can solve difficult problems. The convergence of successive iterations can be confirmed by the help of the Marquardt-parameter (λ) even if the surface determined by *Eq. (2.3)* is very complicated. This method, being a quasi Newton-method, requires relatively few calculations of function values. This is especially important for fitting kinetic models. The method has quadratic convergence near the minimum.

The iteration formula of this method is given by *Eq. (2.5)*:

$$\vec{k}_{i+1} = \vec{k}_i + \left(\boldsymbol{\mathcal{J}}^T(\vec{k}_i) \cdot \boldsymbol{\mathcal{W}}^2 \cdot \boldsymbol{\mathcal{J}}(\vec{k}_i) + \lambda_i \cdot \mathbf{I}\right)^{-1} \cdot \boldsymbol{\mathcal{J}}^T(\vec{k}_i) \cdot \boldsymbol{\mathcal{W}}^2 \cdot \left(\vec{Y} - \vec{F}(\vec{k}_i)\right) \tag{2.5}$$

where

$\vec{k}_{i+1}$ is the parameter vector after the $(i+1)^{\text{th}}$ iteration.

$\vec{k}_i$ is the parameter vector after the i^{th} iteration.

$\vec{Y}$ is a vector containing all the values of $y^{\text{expr}}_{i,j,l}$ occurring in *Eq. (2.3)*. The sequence of data is not essential, but the vectors and the columns of matrices (containing calculated data or their difference quotients with respect to a given parameter) must include them in the same sequence.

$\vec{F}(\vec{k}_i)$ is a vector containing all the calculated data of all the characteristics. The sequence of data is the same as in $\vec{Y}$.

$\boldsymbol{\mathcal{W}}$ is a diagonal matrix, the order of which coincides with the number of experimental points. The i^{th} diagonal element is the weighting factor of the difference between the i^{th} calculated and measured points in *Eq. (2.3)*.

$\mathbf{I}$ is an identity matrix, the order of which corresponds to the order of $\boldsymbol{\mathcal{W}}$.

λ_i is the Marquardt-parameter for the i^{th} iteration. A greater value of λ gives smaller relative differences of parameters in two successive iterations. λ tends to zero as the fitting approaches the minimum of *Eq. (2.3)*.

$\boldsymbol{\mathcal{J}}(\vec{k}_i)$ is a matrix. The element in the j^{th} row and the p^{th} column is the difference quotient of the j^{th} calculated point with respect to the p^{th} fitted parameter. This value is determined by numerical differentiation in ZITA.

$\boldsymbol{\mathcal{J}}^T(\vec{k}_i)$ is the transpose of $\boldsymbol{\mathcal{J}}(\vec{k}_i)$.

The fitting algorithm is programmable relatively easily. $\vec{Y}$ vector is known from the measurements. $\vec{F}(\vec{k}_i)$ and $\boldsymbol{\mathcal{J}}(\vec{k}_i)$ can be determined by numerical integration. At least two runs should be carried out with two different values of a given parameter for the latter matrix. In order to fit m parameters and calculated the numerical difference quotients from two points, every experimental curve should be simulated $(m+1)$ times in each iteration. If the numerical difference quotients are calculated from more than two points, the number of the necessary simulations increases proportionally.

The weighting factors depend on the magnitude of the experimental data and the mode of fitting. In the case of fitting based on relative differences, the diagonal elements of $\boldsymbol{\mathcal{W}}$ should be chosen to be 1.0. When absolute fitting is preferred, the weighting factors should be chosen so that their products with the corresponding elements of $\left(\vec{Y} - \vec{F}(\vec{k}_i)\right)$ are all of the same order of magnitude, so that the fitting will be sensitive to all the experimental information.

The choice of the initial value of λ is based on experience. It does not influence the results, and has a minor influence on running time. It is discussed only in the Reference Guide.

Chapter 3

Getting Started

The best method to learn how to use a program is to use it! This chapter shows through some simple exercises how to use ZITA for the evaluation of kinetic experiments. It is essential that you find enough time for this chapter! Sit down in front of your computer and carry out the instructions exactly! This is also important even if you do not want to fit, but only to simulate.

This chapter contains a number of data sequences which have to be typed into different ASCII-files. In this book every data sequence to be typed in contains a heading, e.g. *Text 3.1* on page 27. This is a ruler for the positions of the columns. Every line begins a reference number. Obviously the heading and the reference numbers must not be typed into the file to be created. The symbol '␣' can indicate a necessary space. Even if more spaces are indicated in these *Text*s, only one of them is necessary. The other spaces prove a more clear data form only.

You will understand the structure and role of these files more easily if you create them from the beginning. If you handle your favorite ASCII-editor well, the typing requires only a few minutes. However, these files can also be generated by '`Example T`' DOS command in the user's library. In this chapter the user is supposed to type these files manually. If he does not want to do so, '`Example T`' can be used.

You may think after a few hours that you have to work too much to get a good result from ZITA. This is not true! A smart and fast solution is also presented at the end of this chapter. But it is worth learning step by step. In this way you can get to know how ZITA works and you will use it more efficiently later.

Now, let us study the first problem!

3.1 The First Chemical Problem

The first problem was chosen according to the following considerations:

- It must be a real problem, not an artificially created one! After all, the aim of ZITA is the evaluation of experimental kinetic data!
- The mechanism of the reaction to be studied must be relatively simple.

- The *ODEs* of the chemical system must have an analytical solution. In this way the results of different methods can be compared.

Based on these conditions *the saponification of diethyl malonate and diethyl adipate* will be studied. The following abbreviations will be used for the chemical substances:

DE denotes $CH_3–CH_2–O–\overset{\overset{O}{\|}}{C}–(CH_2)_n–\overset{\overset{O}{\|}}{C}–O–CH_2–CH_3$, a diethyl ester of an α-ω dicarboxylic acid. If n=1, this compound is diethyl malonate. If n=4, this compound is diethyl adipate.

OH is the hydroxide ion (OH^-).

ME denotes $CH_3–CH_2–O–\overset{\overset{O}{\|}}{C}–(CH_2)_n–\overset{\overset{O}{\|}}{C}–O^-$, the deprotonated form of the monoethyl ester of an α-ω dicarboxylic acid.

AL denotes ethyl alcohol ($CH_3–CH_2–OH$).

AC denotes $^-O–\overset{\overset{O}{\|}}{C}–(CH_2)_n–\overset{\overset{O}{\|}}{C}–O^-$, the deprotonated form of an α-ω dicarboxylic acid.

Probably, the saponification of **DE** in an alkaline solution consists of two consecutive second order steps:

$$\begin{aligned} \mathbf{DE} + \mathbf{OH} &\xrightarrow{k_1} \mathbf{ME} + \mathbf{AL} \\ \mathbf{ME} + \mathbf{OH} &\xrightarrow{k_2} \mathbf{AC} + \mathbf{AL} \end{aligned} \tag{3.1}$$

The questions are the following: are there values of k_1 and k_2 with which this mechanism is able to describe the experimental data within the experimental errors? If so, what are these values? If not, what model can describe them?

The following experiments have to be carried out for solving this problem[1]:

- The reaction mixture has to be prepared in a regulated vessel. The reactants are in stoichiometric ratio. Let the initial concentration of **DE** ($[\mathbf{DE}]_0$) be 0.005 M and $[\mathbf{OH}]_0 = 0.01$ M.
- We take a 25.0 cm^3 sample from the mixture at the t^{th} time point. The reaction has to be stopped in these samples immediately after sampling. The unreacted **OH** can be titrated with 0.01 M HCl. The concentration of **OH** ($[\mathbf{OH}]_t$) can be calculated from the consumption of HCl.
- The procedure described above must be repeated at several different time points at least ten times. Each run should reach ∼90–95 % conversion.

Julius Meyer carried out these experiments and published the data in [10]. We will use his data now. The primary experimental data and the calculated $[\mathbf{OH}]_t$ are shown in *Table 3.1.* We will determine the value of k_1 and k_2 based on them.

[1] Naturally, the experiments can also be planned in other ways.

t (s)	0	33	79	132	211	300
V (cm^3)	25.0	17.4	13.8	12.1	11.2	10.7
$[\mathbf{OH}]_t$ (M)	0.01000	0.00696	0.00552	0.00484	0.00448	0.00428
t (s)	600	900	1500	2100	3000	3900
V (cm^3)	9.6	8.8	7.4	6.2	4.9	4.5
$[\mathbf{OH}]_t$ (M)	0.00384	0.00352	0.00296	0.00248	0.00196	0.00180

Table 3.1: Experimental data for the saponification of diethyl malonate.

3.2 Analytical Solution

Now we solve the mathematical model of *Eq. (3.1)* analytically. Naturally, this is not necessary if ZIA is used, but we will refer to some equations obtained here later on. The classical solution shows many properties of the problem as well as the limitations of the exact mathematical solution. These limitations can be avoided by using ZIA. That is why the user should read this section.

The *ODEs* of the chemical mechanism is[2]:

$$\begin{aligned} \frac{\mathrm{d}[\mathbf{DE}]_t}{\mathrm{d}t} &= -k_1 \cdot [\mathbf{DE}]_t \cdot [\mathbf{OH}]_t \\ \frac{\mathrm{d}[\mathbf{OH}]_t}{\mathrm{d}t} &= -k_1 \cdot [\mathbf{DE}]_t \cdot [\mathbf{OH}]_t - k_2 \cdot [\mathbf{ME}]_t \cdot [\mathbf{OH}]_t \end{aligned} \tag{3.2}$$

This *ODEs* can be simplified. If $[\mathbf{OH}]_0 = 2 \cdot [\mathbf{DE}]_0$, the following relationship is valid from the principle of material balance:

$$[\mathbf{ME}]_t = [\mathbf{OH}]_t - 2 \cdot [\mathbf{DE}]_t \tag{3.3}$$

By substitution of *Eq. (3.3)* into *Eq. (3.2)* we get the following *ODEs*:

$$\begin{aligned} \frac{\mathrm{d}[\mathbf{DE}]_t}{\mathrm{d}t} &= -k_1 \cdot [\mathbf{DE}]_t \cdot [\mathbf{OH}]_t \\ \frac{\mathrm{d}[\mathbf{OH}]_t}{\mathrm{d}t} &= (2 \cdot k_2 - k_1) \cdot [\mathbf{DE}]_t \cdot [\mathbf{OH}]_t - k_2 \cdot [\mathbf{OH}]_t^2 \end{aligned}$$

In order to get more simple expressions we introduce the following dimensionless and normalized variables:

$$\tau = [\mathbf{DE}]_0 \cdot k_1 \cdot t, \quad \kappa = \frac{k_2}{k_1}, \quad A_R(\tau, \kappa) = \frac{[\mathbf{OH}]_\tau}{[\mathbf{OH}]_0} \quad \text{and} \quad B_R(\tau, \kappa) = \frac{[\mathbf{DE}]_\tau}{[\mathbf{DE}]_0} \tag{3.4}$$

After the substitutions the *ODEs* contains only the variables defined in *Eq. (3.4)*:

$$\begin{aligned} \frac{\partial B_R(\tau, \kappa)}{\partial \tau} &= -2 \cdot A_R(\tau, \kappa) \cdot B_R(\tau, \kappa) \\ \frac{\partial A_R(\tau, \kappa)}{\partial \tau} &= (2 \cdot \kappa - 1) \cdot A_R(\tau, \kappa) \cdot B_R(\tau, \kappa) - 2 \cdot \kappa \cdot A_R^2(\tau, \kappa) \end{aligned} \tag{3.5}$$

[2] From now on, $[\mathrm{A}]_t$ denotes the concentration of substance A as a function of t.

κ	*Solution*	*Simplified mechanism*
0	$[\mathbf{OH}]_t = \dfrac{[\mathbf{OH}]_0}{2 - e^{-[\mathbf{OH}]_0 \cdot k_1 \cdot t/2}}$	$\mathbf{DE} + \mathbf{OH} \xrightarrow{k_1} \mathbf{ME} + \mathbf{AL}$ $2\,[\mathbf{DE}]_0 = [\mathbf{OH}]_0$
$\frac{1}{2}$	$[\mathbf{OH}]_t = \dfrac{[\mathbf{OH}]_0}{[\mathbf{OH}]_0 \cdot \frac{k_1}{2} \cdot t + 1}$	–
1	$[\mathbf{OH}]_t$ has no explicit form.	–
∞	$[\mathbf{OH}]_t = \dfrac{[\mathbf{OH}]_0}{[\mathbf{OH}]_0 \cdot k_1 \cdot t + 1}$	$\mathbf{DE} + 2\,\mathbf{OH} \xrightarrow{k_1} \mathbf{AC} + 2\,\mathbf{AL}$ $2\,[\mathbf{DE}]_0 = [\mathbf{OH}]_0$

Table 3.2: Analytical solutions of *Eq. (3.2)* in some special cases.

Dividing the second equation of *Eq. (3.5)* by the first:

$$\frac{\mathrm{d}A_R(\tau,\kappa)}{\mathrm{d}B_R(\tau,\kappa)} = \frac{1-2\cdot\kappa}{2} + \kappa \cdot \frac{A_R(\tau,\kappa)}{B_R(\tau,\kappa)} \tag{3.6}$$

Eq. (3.6) is a linear differential equation, so it can be solved relatively easy. The solution is:

$$A_R(\tau,\kappa) = \frac{1-2\cdot\kappa}{2\cdot(1-\kappa)} \cdot B_R(\tau,\kappa) + \frac{B_R^{\kappa}(\tau,\kappa)}{2\cdot(1-\kappa)} \tag{3.7}$$

If we substitute *Eq. (3.7)* into the first equation of *Eq. (3.5)*, we get the following equation:

$$\tau = \frac{1-\kappa}{1-2\cdot\kappa} \cdot \int_{B_R}^{1} \frac{\mathrm{d}B_R}{B_R^2 \cdot \left(1 + \dfrac{B_R^{\kappa}-1}{1-2\cdot\kappa}\right)} \tag{3.8}$$

Eq. (3.8) must be solved for $B_R(\tau,\kappa)$. $B_R(\tau,\kappa)$ can be substituted into *Eq. (3.7)* after integration of *Eq. (3.8)*. In this way $A_R(\tau,\kappa)$ can be given explicitly as a function of τ and κ. If $A_R(\tau,\kappa)$ is known, the mathematical relationship among $[\mathbf{OH}]_t$, t, k_1 and k_2 can be derived from the definitions given in *Eq. (3.4)*. In this way the values of the rate constants can be determined if the $(t,[\mathbf{OH}]_t)$ experimental data pairs are transformed to an $A_R(\tau,\kappa)$ curve.

Solving *Eq. (3.8)* appears to be a very difficult problem, but the solution has been given by A. A. Frost and W. C. Schwemmer [11]. *Table 3.2* summarizes the analytical solutions in special cases. The general solution has a very complicated form, so it cannot be used in practice[3]. For this reason, Frost and Schwemmer developed figures and tables from the analytical solution. The value of k_1 and k_2 can be determined from these relatively quickly,

[3] If you are interested in the detailed mathematics of the integration, you can find it in [11].

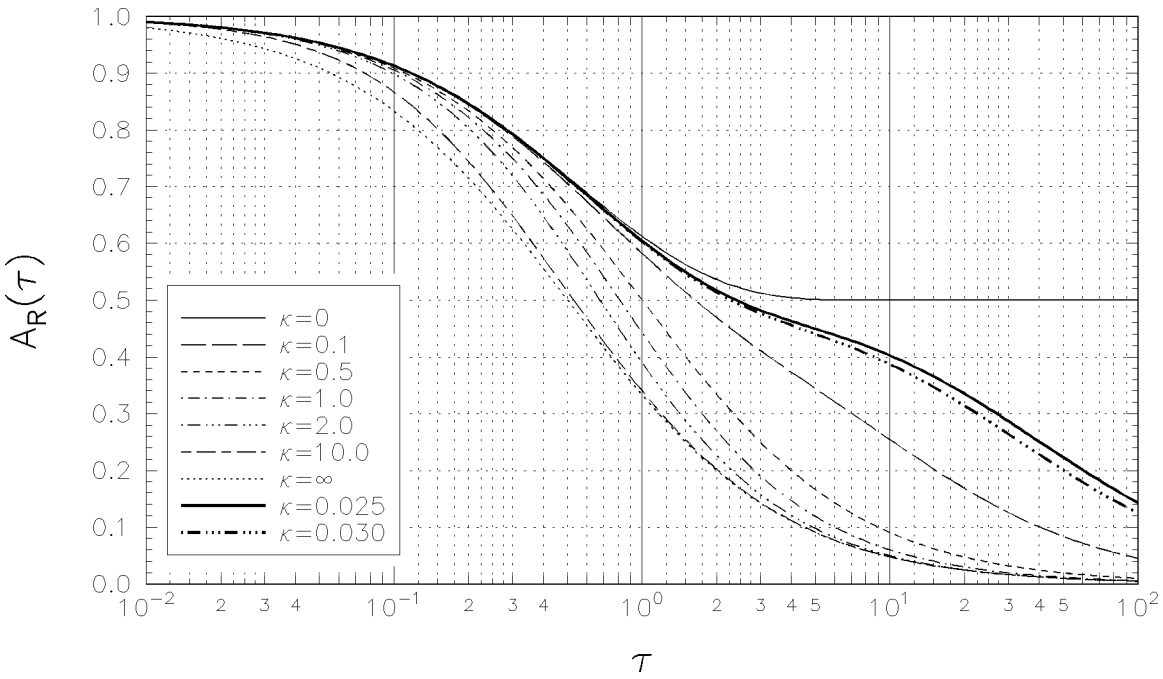


Figure 3.1: Dimensionless $[\mathbf{OH}]_\tau$ as a function of dimensionless time for different ratios of the rate constants.

but two problems arise: this method can be used only in special cases, and its precision is very limited.

Even without reading the original paper you can imagine that many time-consuming calculations are needed to get the appropriate figures and tables from $A_R(\tau, \kappa)$ defined in *(3.7,3.8)*. This work is not detailed now. We employ the figures given in [11] for determining the value of k_1 and k_2.

It can be seen from *(3.7,3.8)* that the values of A_R depend on only τ and κ. Therefore, if A_R is drawn as a function of τ, the shape of this curve depends only on κ. *Figure 3.1* shows $A_R(\tau)$ curves for different values of κ. In principle, the value of κ could be determined by comparing the experimental data with this figure. However, we can get only $A_R(t)$ from the experiments, since the value of k_1 is unknown. Therefore we choose another way to find the value of κ.

We need to find a transformation that can be applied directly to the experimental data. Since $\tau_1 = [\mathbf{DE}]_0 \cdot k_1 \cdot t_1$ and $\tau_2 = [\mathbf{DE}]_0 \cdot k_1 \cdot t_2$, therefore $\tau_1/\tau_2 = t_1/t_2$. If $A_R(\tau)$ is drawn as a function of τ/τ_x where $A_R(\tau_x) = x$ and x is a specific reduced concentration, we get the same curve as if $A_R(t)$ were drawn as a function of t/t_x. Let the third experimental point be chosen[4] so $x = 0.552$. *Figure 3.2* shows the transformed curves for several different κ[5]. This figure also contains the experimental points. The theoretical curve that best describes the

[4]Naturally, we might also choose any other point except $x = 1.0$. If we want to get more values for κ because of statistical purposes, we must carry out this procedure with more values of x.

[5]Frost and Schwemmer gave these data in several tables.

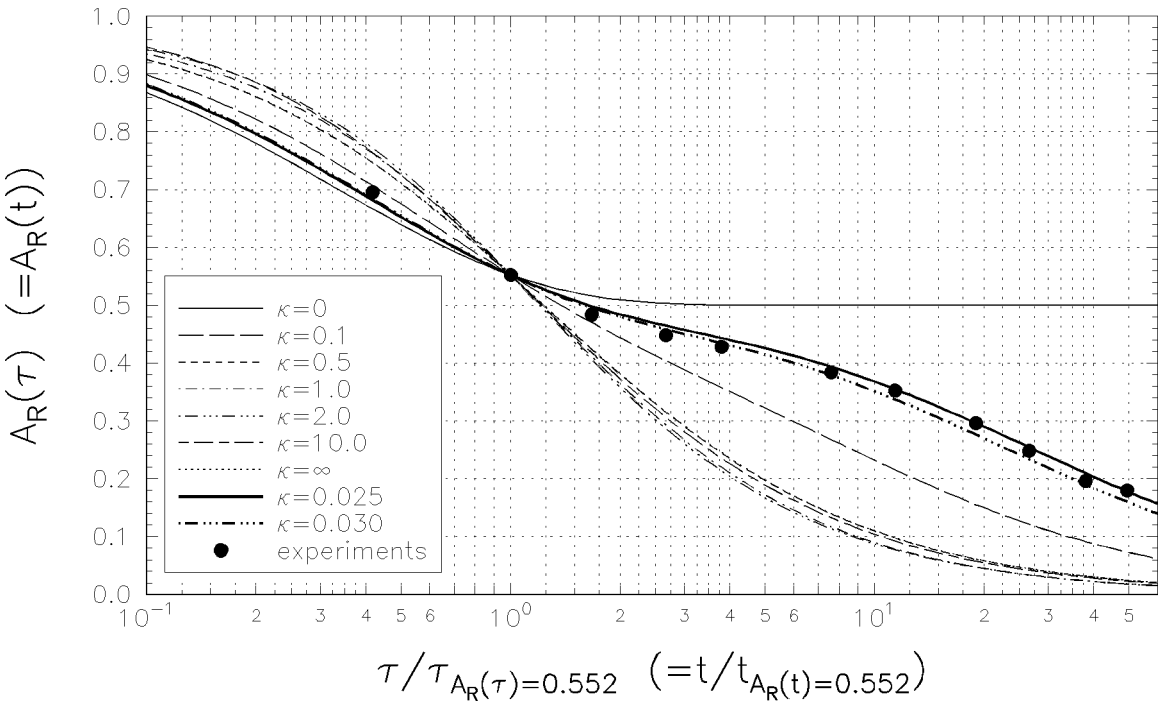


Figure 3.2: Dimensionless $[\mathbf{OH}]_t$ as a function of ratios of time values. The experimental data are derived from *Table 3.1.*

t *(min)*	$A_R(\tau)$	$-\lg\tau$	τ	k_1
33	0.696	–0.254	0.557	3.38
79	0.552	0.186	1.53	3.87
132	0.484	0.471	2.96	4.48
211	0.448	0.692	4.92	4.66
300	0.428	0.847	7.03	4.69
600	0.384	1.07	11.7	3.90
900	0.352	1.20	15.8	3.51
1500	0.296	1.40	25.1	3.35
2100	0.248	1.60	39.8	3.79
3000	0.196	1.75	56.2	3.75
3900	0.180	1.81	64.6	3.31

Table 3.3: Detailed results for calculating k_1 and k_2.

experiment, gives the κ which equals the real k_2/k_1. It is easily seen that κ_{expr} is between 0.025 and 0.030. More precise result cannot be achieved from this figure. Later on, we take $\kappa = 0.0275$.

κ is now known, so τ for experimental points can be determined from *Figure 3.1.* The rate constants can then be calculated from each calculated point (see *Eq. (3.4)*). The partial results of this calculation are given in *Table 3.3.* The final results are the following:

```
      Column
Line  ....5....10....15....20
  1: 0.005 0.01 0.0 0.0 0.0
  2:     0 0.01000 25.0
  3:    33 0.00696 17.4
  4:    79 0.00552 13.8
  5:   132 0.00484 12.1
  6:   211 0.00448 11.2
  7:   300 0.00428 10.7
  8:   600 0.00384  9.6
  9:   900 0.00352  8.8
 10:  1500 0.00296  7.4
 11:  2100 0.00248  6.2
 12:  3000 0.00196  4.9
 13:  3900 0.00180  4.5
```

Text 3.1: Contents of `EXA.EXP`.

```
      Column
Line  ....5....10....15....20
  1: 2 5 12
  2: DE  OH  ME  AL  AC
  3:  1   1   0   0   0  v1
  4:  0   1   1   0   0  v2
  5:
  6:  1   1   1   1   1
  7:
  8: -1  -1   1   1   0  v1
  9:  0  -1  -1   1   1  v2
 10: 0 0
```

Text 3.2: Contents of `EXA1.ODE`.

$k_1 = 3.88 \pm 0.51\,\mathrm{M^{-1}\,s^{-1}}$ and $k_2 = 0.107 \pm 0.014\,\mathrm{M^{-1}\,s^{-1}}$.

If *Figure 3.2* did not contain a calculated curve that describes the experiments well, we would have to search for another chemical mechanism. In view of the mathematical difficulties which are not detailed here, the analytical solution is a difficult approach, and one that cannot be applied to a more complicated mechanism.

Now, we solve the first problem with the help of ZIA.

3.3 Solving the First Problem with ZIA

Sit down, please, in front of your computer and turn it on! Start ZIA with the commands which were written on your screen when `INSTALL.EXE` completed your user's library. If these commands are carried out exactly, your user's library becomes the actual directory.

If `INSTALL.EXE` did not create the executable programs (`*.EXE`) of ZIA during the installation because of insufficient memory, you must do this now with the help of `NEWEXE.BAT` (see in *Section 6.2* on page 70).

First, you need to create those files with whose help the given problem can be solved. The exact contents of these files can be found in several *Text*s of this chapter, so you can type them in easily. However, the structure of these files is not detailed here but only in the Reference Guide. This chapter explains only their main parts.

3.3.1 Input of Experimental Data

The first file to be typed in contains the initial concentrations and the data given in *Table 3.1.* Give the name `EXA.EXP` for this file! The contents of this file are in *Text 3.1.* In this file, the first line contains the initial concentrations of **DE**, **OH**, **ME**, **AL** and **AC**. All the other lines contain one experimental point: the actual values of time, $[\mathbf{OH}]_t$ and the volume of the consumed HCl-solution in $\mathrm{cm^3}$ successively. At first, we use only the data of the first two columns, but we shall need the third column later. Write the contents of *Text 3.1* into

EXA.EXP using your ASCII-editor now! In this way you supply the experimental data to be used for ZITA.

3.3.2 Defining *ODEs*

In order to apply a least squares fitting procedure (i.e. minimization of *Eq. (2.3)*), the calculated analogue of the experimental curve is required. It can be generated by simulation. The simulation requires two things: the *ODEs* of the chemical mechanism and the parameters. This subsection details the first item and the next subsection describes the second.

A chemical mechanism consists of steps having the following two features:

- The rate of every step is directly proportional to the concentrations of reactants raised to various powers. For example, if the step is $aA + bB \rightarrow pP$, the rate is defined by $r = k \cdot A^{\nu_a} \cdot B^{\nu_b}$ equation.
- Every step has a stoichiometry which is independent of the initial concentrations of reactants and the temperature. Every species has two stoichiometric coefficients in each step, the difference of which shows how many molecules of that species are formed or consumed in that step.

Based on these facts a mechanism can be defined by the partial orders of the reactants (ν_a, ν_b) and the change of the stoichiometric coefficients. These numbers can be given in matrix form, so two matrices define a chemical mechanism: an order matrix and a stoichiometric matrix. The matrices of the first problem are shown in *Text 3.2*. The first line contains the number of equations, species and the maximal number of experimental data pairs. Lines 3-4 contain the order matrix and lines 8-9 contain the stoichiometric matrix. Try to produce these two matrices on your own! If you manage to create them, you understand the matrix form! *Subsection 9.1.2* and *Appendix A* contain more information about the matrix-formalism.

The meanings of the other numbers and strings in this file are not detailed here. You can find them in *Subsection 9.1.2*. Write the content of *Text 3.2* into EXA1.ODE! If it is ready, ZITA can create the source text of the actual *ODEs* and the executable program named ZITA.EXE. This program contains the *ODEs*-solver and the concrete *ODEs* together. To make them you have to execute the following DOS command:

COMPILE EXA1 2 and <ENTER>

There are additional effects of this command. First, MAKEODE.EXE creates the Pascal-source text of the *ODEs* for the first problem. The name of the new file is ODES.PAS. The source text of the *ODEs*-solver is in ZITA.PAS. Second, COMPILE.BAT creates the executable program based on these two *.PAS files. As mentioned earlier, the name of the new program is ZITA.EXE. It makes the calculated curves determined by the user. The next subsection explains how the input files must be created.

```
      Column
Line  ....5....10....15....20....25....30....35....40....45....50....
   1: This is the first example to use ZiTa.
   2: Reaction between diethyl malonate and hydroxide ion.
   3: Model: DE + OH -> ME + AL ,  ME + OH -> AC + AL
   4: 6 80 0
   5: 21 1.0E-5 1.0E-5
   6: 5
   7: 1 DE
   8: 2 OH
   9: 3 ME
  10: 4 AL
  11: 5 AC
  12: 0
  13: 2 Exa.exp 5 1 2 3 4 5
  14: 9 5 5 0.0
  15: p Exa.sim
  16: 0
  17: 0
  18: y
  19: 1 1.0 +
  20: 2 Exa.exp 5 1 2 3 4 5
  21: 9 0 0 0.0
  22: p Exa1k1.out
  23: 0
  24: 0
  25: y
  26: 2 1.0 +
  27: 2 Exa.exp 5 1 2 3 4 5
  28: 9 0 0 0.0
  29: p Exa1k2.out
  30: 0
  31: 0
  32: n
```

Text 3.3: Contents of `EXAALL.DAT`.

```
      Column
Line  ....5....10....15....20
   1: 2.0 * p1
   2: 1.0 * p2
   3:
   4: 0.001 0.99
   5: -1.0 1.0E-04 1.0E-04
   6: 12
```

Text 3.4: Contents of `EXAPARAM.DAT`.

3.3.3 Input Data for *ODEs*-solver

`ZITA.EXE` is the most complicated program in the package. This chapter shows only a small fraction of the available options. If you would like to use the *ODEs*-solver more efficiently, you should study the appropriate part of the Reference Guide later.

The input data of `ZITA.EXE` have to be written into two files[6]:

- The name of the first file is `PARAMTRS.DAT`. It contains the initial values of the rate constants and other parameters which are used by `GNM.EXE`[7]. The contents of this file can be found in *Text 3.4*. You must name this file `EXAPARAM.DAT`. The reason for the different name will be explained in *Subsection 3.3.6*. The `EXAPARAM.DAT` file will be copied to the `PARAMTRS.DAT` file before the simulation.

[6] Naturally, the files containing experimental data may also be necessary, if the user wants to use them.
[7] This program is outlined briefly in *Subsection 3.3.6*.

The first two lines of `EXAPARAM.DAT` contain the values of k_1 and k_2 defined in *Eq. (3.1)*. The initial value of k_1 is estimated from the first two data pairs of the experimental curve. If we suppose that k_1 is different from k_2 only because of statistical reasons, then $2 \cdot k_2 = k_1$[8]. The initial value of k_2 is chosen based on this consideration. The asterisks after the numbers mean that the parameter value in question is to be fitted. If there is no asterisk after a parameter value, it will be held fixed during the fitting. The meaning of the other lines is detailed only in *Subsection 3.3.6* but you must write them exactly into the file to be created!

- The other input file contains the parameters of the simulation. Its name is `EXAALL.DAT` in the first problem, but later you can choose any name for it. Its contents can be found in *Text 3.3*. Now, we sketch the purposes of the data in this file:
  - The first three lines contain only remarks.
  - Line 4 determines the form and location of the output file(s). This row in *Text 3.3* means that the accuracy of the numbers is 6 digits and a formatted line can contain a maximum of 80 characters. The third number (0) means that the place of the output file(s) is the `.\SIMU` library.
  - Line 5 contains the index number of the integration method, the relative error bound and the initial step size.
  - Lines 6–11 say that the output file(s) will contain the concentrations of five species (**DE**, **OH**, **ME**, **AL** and **AC**) following the time. The order of the output concentrations is also determined here.
  - Lines 12–18 specify the next output file. Lines 13-14 tell the *ODEs*-solver that `EXA.EXP` contains the initial concentrations and the time points at which we would like to store the calculated concentrations. Lines 15–17 inform the *ODEs*-solver that the name of the output file will be `EXA.SIM` and any other calculated characteristics, e.g. rate values, will not be stored. The '`y`' in line 18 means that this file contains a new task for the *ODEs*.
  - Lines 19–25 and 26–32 have the same meaning as lines 12-18 but there are two differences. The names of the output files will be `EXA1K1.OUT` and `EXA1K2.OUT` instead of `EXA.SIM`. The other difference is that the values of k_1 is increased by 1 % when `EXA1K1.OUT` is created and the values of k_2 is increased by 1 % when `EXA1K2.OUT` is created. *Subsection 3.3.5* explains why we also need to simulate the latter two curves.

When you finish creating `EXAPARAM.DAT` and `EXAALL.DAT`, you can simulate the calculated curves immediately. You should then type the following DOS commands:

`Set ZitaEXPR=.\EXPR` <ENTER>[9]

[8] This equation can be derived from statistical considerations but they are not detailed here.

[9] This command can be carried out directly only from DOS. If you work with a utility, e.g. Norton Commander, PC Shell, etc., your computer cannot carry out this command! In these cases you must exit the utility and return to the DOS cursor!

```
      Column
Line  ....5....10....15....20....25....30....35....40....45....50....55....60....65....70....75....80....85
   1: Experimental (*) and calculated curves: DE(..), OH(____), ME(__ .), AL(___ ), AC(_ _ )
   2: Time (s)
   3: Concentration (M)
   4: 6 1 Z
   5: Exa.exp 1 2 1 0 *
   6: Exa.sim 1 2 0 1 1000100010001000
   7: Exa.sim 1 3 0 1 1111111111111111
   8: Exa.sim 1 4 0 1 1111111000010000
   9: Exa.sim 1 5 0 1 1111111111110000
  10: Exa.sim 1 6 0 1 1110000011100000
  11: 0
  12: n
```

Text 3.5: Contents of `EXA1.DRW`.

```
COPY EXA.EXP .\EXPR\*.*    <ENTER>
COPY EXAPARAM.DAT PARAMTRS.DAT    <ENTER>
ZITA EXAALL    <ENTER>
```

If you type in the contents of the files and enter these commands correctly, `ZITA.EXE` will simulate the calculated curves. The last command requires 1-2 minutes if you have the slowest IBM PC but it takes only 2-3 seconds if you have an AT-386 with a math coprocessor. The output files (`EXA.SIM`, `EXA1K1.OUT` and `EXA1K2.OUT`) can be found in the `.\SIMU\` library. Examine the contents of these files with your editor! You can see that every file contains 6 columns, representing a time point and the concentrations of **DE**, **OH**, **ME**, **AL** and **AC**. You can also see that there are still significant differences between the calculated and experimental concentrations.

We now continue the preparation of the fitting. If the actual directory is not your user's library, change it back now!

3.3.4 Drawing the Experimental and Calculated Curves

Although, drawing is not part of the algorithm of fitting, the easiest way to follow the process of fitting is to create figure(s) after each iteration. ZITA has a drawing program named `DRAWER.EXE`. It takes its input data from specified files and makes two-dimensional figure(s). We can now make a figure from the files containing the experimental and calculated data (these are `.\EXPR\EXA.EXP` and `.\SIMU\EXA.SIM`) and one more file which can be found in *Text 3.5*. The explanation of the drawing program can be found in *Chapter 11*. The structure of `EXA1.DRW` is not analyzed here. Only the following is important: if the files containing data to be plotted and `EXA1.DRW` are ready and your computer has a graphics card, `DRAWER.EXE` will draw the experimental [**OH**] and the calculated concentrations of each species on the screen. Write the contents of *Text 3.5* into `EXA1.DRW`! When it is ready, the following DOS command will generate the desired figure (you can exit by pressing the *'Q'* key):

```
DRAWER EXA1    <ENTER>
```

EXA1K1.JCD

```
      Column
Line  ....5....10....15....20....25....30....35
  1:  0 0
  2:  Exa.exp 2 1 Exa.sim 3 1.0  Exa1k1.out
```

EXA1K2.JCD

```
      Column
Line  ....5....10....15....20....25....30....35
  1:  0 0
  2:  Exa.exp 2 1 Exa.sim 3 1.0  Exa1k2.out
```

Text 3.6: Contents of the `*.JCD` files for the first exercise.

3.3.5 Preparation Files for an Iteration

The experimental and calculated data files are ready. Now, we have to use *Eq. (2.5)*. To employ this equation we must create the $\left(\vec{Y} - \vec{F}(\vec{k}_i)\right)$ vector and the $\boldsymbol{\mathcal{J}}(\vec{k}_i)$ matrix defined in *Eq. (2.5)* on page 20. This can be done by the `JC.EXE` program.

The elements of the vector can be calculated easily from the experimental and calculated data. However, determining the values of the matrix elements requires differentiation. Since the analytical forms of the derivatives are unknown, numerical differentiation must be used. Thus we use difference quotients:

$$\frac{\partial[\mathbf{OH}]_t}{\partial k_1} \approx \frac{\Delta[\mathbf{OH}]_t}{\Delta k_1} \qquad \text{and} \qquad \frac{\partial[\mathbf{OH}]_t}{\partial k_2} \approx \frac{\Delta[\mathbf{OH}]_t}{\Delta k_2}$$

This is the reason that we created `EXA1K1.OUT` and `EXA1K2.OUT` in addition to `EXA.SIM`. We need curves simulated with different rate constants, so the difference quotients can be calculated from the concentrations simulated at the same time point.

`JC.EXE` needs the names of the files that contains the data to be used in the calculation. This information is stored in the files shown in *Text 3.6*. Let us consider `EXA1K1.JCD` (the structure of `EXA1K2.JCD` is the same but it concerns k_2). The first line means that all the files containing calculated data can be found in the `.\SIMU` library. The second line means that:

- The experimental concentrations to be used are in the second column of `EXA.EXP`. The '`1`' after '`2`' means that the initial concentrations are in the first line of `EXA.EXP`.
- The third column of `EXA.SIM` contains the calculated [**OH**] (the first column contains the time and the second contains [**DE**]).
- `EXA1K1.OUT` contains the calculated points computed with an increased value of k_1.

`JC.EXE` stores the results in a set of files labeled `J*.COL`. `J.COL` contains the elements of $\left(\vec{Y} - \vec{F}(\vec{k}_i)\right)$ and `J`n`.COL` contains the elements of n^{th} column of $\boldsymbol{\mathcal{J}}(\vec{k}_i)$. The structures of `*.JCD` and `J*.COL` are detailed in *Section 10.1*.

3.3.6 Execution of an Iteration

After all the previous computations have been completed we can apply *Eq. (2.5)* for computing the new values of the parameters. This is carried out by `GNM.EXE`. Its input data

are in PARAMTRS.DAT and J*.COL. The PARAMTRS.DAT file is ready but its name is still EXAPARAM.DAT. The J*.COL files are not ready, but they will be created by JC.EXE before GNM.EXE is used.

The last three lines of EXAPARAM.DAT contain information for GNM.EXE. Line 4 gives two parameters for the fitting. The first one is the Marquardt parameter. The second one has a complicated meaning that is detailed in *Subsection 10.2.1.* The first data of the 5$^{\text{th}}$ line is the actual value of $S(\vec{k}_i)$ defined in *Eq. (A.7).* This number must be negative before the first iteration. The other two numbers on this line are the terminating conditions for the fitting. These values determine the minimum relative changes in $S(\vec{k}_i)$ and the parameters in order to continue the fitting. The last line contains the maximum number of iterations.

GNM.EXE calculates a better approximation of k_1 and k_2 based on the input files and writes the new values into RES1. The structure of this file is exactly the same as that of PARAMTRS.DAT. There is only one exception: if at least one terminating condition is valid (see them in *Subsection 10.2.2*), RES1 will not be created. If the fitting has not terminated, RES1 must be copied to PARAMTRS.DAT before the new iteration. That is why we stored the contents of PARAMTRS.DAT in EXAPARAM.DAT. Since PARAMTRS.DAT is overwritten in each iteration, we would have to create this file again if an error occurs. It seems much better to store the initial parameter values in another file (concretely in EXAPARAM.DAT) and to copy it to PARAMTRS.DAT before the fitting.

GNM.EXE creates another output file, RESULT.i where i is the serial number of the iteration. This file contains the most recent value of $S(\vec{k}_i)$, the correlation matrix, the new parameter values with their deviations and changes and a Goodness-of-Fit Statistics. The correlation coefficients are given in three ways. The content of the RESULT.* files are detailed in [9, Chapter 8], *Subsection 10.2.3* and *Appendix A.*

The RESULT.* files have enough information to evaluate the fitting. If you would like to analyze the residuals at each point, you should study the last J.COL file, too.

3.3.7 Organizing the Process of Fitting

The previous subsections discussed the programs and data files which realize the parts of the algorithm described in *Chapter 2.* Your last task is to organize these parts into one executable program. This can be realized by EXA1.BAT shown in *Text 3.7*[10]. Copy the contents of *Text 3.7* into EXA1.BAT and carry out the following DOS command:

EXA1 <ENTER>

In this way the fitting can be carried out. Now, analyze the main parts of this file in detail:

- Line 1 writes the initial values of the parameters into PARAMTRS.DAT.

[10] [2] describes the structure of *.BAT and all the commands which can be used within a batch file. If you do not know them, study the chapter entitled *'Working with Batch Programs'* in [2]. Naturally, you can use other books on DOS, too.

```
Line  Column 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75
 1:  Copy Exaparam.dat Paramtrs.dat
 2:  Copy Exa.exp %ZITAEXPR%
 3: :La
 4:  Zita Exaall.dat
 5:  If Errorlevel 1 Goto Lb
 6:  %ZITABASE%\Drawer Exa1.drw
 7:  If Errorlevel 1 Goto Lb
 8:  %ZITABASE%\Jc Exa1k1.jcd r 0
 9:  If Errorlevel 1 Goto Lb
10:  %ZITABASE%\Jc Exa1k1.jcd r 1 + 1.0
11:  If Errorlevel 1 Goto Lb
12:  %ZITABASE%\Jc Exa1k2.jcd r 2 + 1.0
13:  If Errorlevel 1 Goto Lb
14:  %ZITABASE%\Gnm *
15:  If Errorlevel 1 Goto Lb
16:  Echo Off
17:  For %%i In (1 2 3 4 5 6 7 8 9 10 11 12) Do If Exist Result.%%i Type Result.%%i
18:  Pause
19:  Echo On
20:  If Not Exist Res1 Goto Lc
21:  Copy Res1 Paramtrs.dat
22:  Goto La
23: :Lb
24:  Echo There is an error in the calculations!
25: :Lc
```

Text 3.7: Contents of `EXA1.BAT`.

- Line 2 copies that file into the appropriate library which contains the experimental data. `%ZitaEXPR%` is a DOS environment variable. It contains the name of the library in which `ZITA.EXE` searches for the experimental data. If you are not familiar with DOS environment variables, study [2]! ZITA often uses this kind of variable, so you ought to understand their aims and syntax! Every DOS environment variable of ZITA is detailed in *Section 6.1.*
- Line 4 produces the calculated analogue of the experimental curve. The DOS environment variable named `%ZitaBASE%` contains the name of the common library.
- Lines 5, 7, 9, 11, 13 and 15 terminate the fitting if a program makes a mistake.
- Line 6 generates a figure containing the experimental and calculated curves. It is not an essential part of the process, but it is a good idea to show the partial results on some figures as an illustrative exercise. The figure can be left and the fitting can be continued by pressing '*Q*'.
- Lines 8, 10 and 12 calculate the $\left(\vec{Y} - \vec{F}(\vec{k}_i)\right)$ vector and the $\boldsymbol{\mathcal{J}}(\vec{k}_i)$ matrix.
- Line 14 carries out an iteration based on *Eq. (2.5)*.
- Lines 16–19 are included only for the sake of illustration. They list the complete `RESULT.*` files on the screen. The content of the last created file remains on the screen until the user presses a key. The `For-In-Do` statement will not be needed later, so you need not learn it.
- Line 20 leaves the iteration loop if any termination condition is valid.

```
ITERATION 6;     WSTD: 1.154506557E-0002;    NMD:12;    MP: 1.000E-0005
******** Result(s):
       Estimate                    Std. Dev. (abs & %)          dp
  p1  3.62046686229726380E+0000  1.662E-0001  &      4.59    -2.723E-0003
  p2  9.90725183245637818E-0002  3.399E-0003  &      3.43     2.124E-0005
******** Goodness-of-Fit Statistics:
                                Marquardt Parameter: 1.000E-0005
                                      Relative Step: 1.0000
                             Number of Measured Data: 12
                         Number of Fitted Parameters: 2
               Weighted Standard Deviation of Data: 1.154506557424525E-02
             Unweighted Standard Deviation of Data: 1.154506557424525E-02
Correlation between Measured and Calculated Values: 9.992703931556526E-01
                      Coefficient of Determination: 9.984794642794934E-01
                                         R-squared: 9.996908009021721E-01
                           Model Selection Criterion: 6.155359225184052E+00
******** Residual Analysis:
Serial Correlation: 1.521282138686760O3E+0000
Heteroscedactivity:-9.74377687173474701E-0003
          Skewness: 9.16489360635119006E-0001
          Kurtosis:-9.21900655824775644E-0001
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2
  p1 { .118}-.118
  p2  -.118{ .118}
******** 95% Univariate Confidence Interval(s):
  p1  3.25463864868168294E+0000 - 3.98629507591284467E+0000
  p2  9.15917111099496061E-0002 - 1.06553325539177957E-0001
******** 95% Supporting Plane Confidence Interval(s):
  p1  3.14434722776059611E+0000 - 4.09658649683393150E+0000
  p2  8.93363660418857388E-0002 - 1.08808670607241825E-0001
```

Text 3.8: Final result of the first exercise.

- Line 21 copies the new parameter values to `PARAMTRS.DAT` file. Line 22 jumps to the beginning of the iteration loop.

This computation requires about 30–60 minutes on the slowest IBM PC. An AT-386 with math coprocessor carries out the fitting within 1–2 minutes.

The result of the fitting is seven `RESULT.*` files in your user's library. Their detailed interpretation happens in *Section 10.2*. The most important file is the last one, because it contains the final result. `J.COL` can also be important. It contains the residual of each point. You can find the last calculated curves in the `.\SIMU` library, so you can a make figure with the help of `DRAWER.EXE` and `EXA1.DRW` whenever you like.

If you have worked correctly up to now, the contents of your `RESULT.6` file should be the same as *Text 3.8*. There may be some differences between the last digits of the numbers if you used non-extended precision in the calculations. Larger differences can occur if you do not use math coprocessor. The number of the iterations may also be different, but the final results must be the same within the experimental error. If you compare these results

```
        Column
Line    ....5....10....15....20
  1: 2 5 12
  2: DE  OH  ME  AL  AC
  3:  1   1   0   0   0  v1
  4:  0   1   1   0   0  v2
  5:
  6:  1   1   1   1   1
  7:
  8: -1  -1   1   1   0  v1
  9:  0  -1  -1   1   1  v2
 10: 0 1 1
 11: tr[1]:=y[2]/0.01*25.0;
```

Text 3.9: Contents of `EXA2.ODE`.

to the analytical solution described in *Section 3.2* you will see the following:

- The parameter values derived from the numerical solution are in the range determined by the values calculated analytically and their deviations. This range is $[3.37, 4.39]$ for $k_1 \approx 3.6\,\mathrm{M^{-1}\,s^{-1}}$ and $[0.093, 0.121]$ for $k_2 \approx 0.099\,\mathrm{M^{-1}\,s^{-1}}$.
- The parameter values derived from the analytical solution *are not* in the range determined by the values calculated numerically and their deviations. This range is $[3.45, 3.79]$ for $k_1 \approx 3.9\,\mathrm{M^{-1}\,s^{-1}}$ and $[0.096, 0.102]$ for $k_2 \approx 0.11\,\mathrm{M^{-1}\,s^{-1}}$.

We should not be surprised by this outcome. When we solved the problem analytically, we used curves that were derived via many transformations from primary experimental data and the 'fitting' was done only by eye.

I hope, you like ZiTiA now. The next sections show more possibilities for using the package, so you have still time to like it. Finally, there is one more piece of advice. Do not erase the files created up now, because they will be used later!

3.4 Fitting on the Basis of Primary Experimental Data

It is often necessary to carry out a fit using primary experimental data instead of concentration *vs.* time curves. This distinction is not important in the actual problem, because the volume of HCl-solution is directly proportional to [**OH**] in the titration. However, the measured data may be any other characteristics, e.g. electrode potential, which would cause the errors of the measurement to become distorted if you calculate the concentrations from the primary data. So ZiTiA also has to be able to perform fits based on primary experimental data.

The *ODEs*-solver created earlier can simulate only concentration *vs.* time curves. Some changes must be made in `*.ODE` file if volume is to be calculated as a function of time. Write the contents of *Text 3.9* into `EXA2.ODE` file into your user's library! You can see there are only two differences between `EXA1.ODE` and `EXA2.ODE`: the 10^{th} lines contain different data and the new file contains one more line. This line is an assignment statement written in Pascal. It expresses the relationship between the primary experimental data and [**OH**].

EXA2.DRW

```
Line    Column
        ....5....10....15....20....25....30....35....40....45
  1: Experimental (*) and calculated (____) curves
  2: Time (s)
  3: Volume (cm3)
  4: 2 1 Z
  5: Exa.exp 1 3 1 0 *
  6: Exa.sim 1 2 0 1 1111111111111111
  7: 0
  8: n
```

EXA2K1.JCD

```
Line    Column
        ....5....10....15....20....25....30....35
  1: 0 0
  2: Exa.exp 3 1 Exa.sim 2 1.0  Exa1k1.out
```

EXA2K2.JCD

```
Line    Column
        ....5....10....15....20....25....30....35
  1: 0 0
  2: Exa.exp 3 1 Exa.sim 2 1.0  Exa1k2.out
```

EXA2.BAT

```
Line    Column
        ....5....10....15....20....25....30....35....40....45....50....55....60....65....70....75
  1:  Copy Exaparam.dat Paramtrs.dat
  2:  Copy Exa.exp %ZITAEXPR%
  3: :La
  4:  Zita Exaall.dat
  5:  If Errorlevel 1 Goto Lb
  6:  %ZITABASE%\Drawer Exa2.drw
  7:  If Errorlevel 1 Goto Lb
  8:  %ZITABASE%\Jc Exa2k1.jcd r 0
  9:  If Errorlevel 1 Goto Lb
 10:  %ZITABASE%\Jc Exa2k1.jcd r 1 + 1.0
 11:  If Errorlevel 1 Goto Lb
 12:  %ZITABASE%\Jc Exa2k2.jcd r 2 + 1.0
 13:  If Errorlevel 1 Goto Lb
 14:  %ZITABASE%\Gnm *
 15:  If Errorlevel 1 Goto Lb
 16:  Echo Off
 17:  For %%i In (1 2 3 4 5 6 7 8 9 10 11 12) Do If Exist Result.%%i Type Result.%%i
 18:  Pause
 19:  Echo On
 20:  If Not Exist Res1 Goto Lc
 21:  Copy Res1 Paramtrs.dat
 22:  Goto La
 23: :Lb
 24:  Echo There is an error in the calculations!
 25: :Lc
```

Text 3.10: New files necessary for fitting based on the primary experimental data.

If you would like to evaluate only concentration *vs.* time curves, you need not know Pascal at all. If you want to transform simulated curves as well, you have to know the syntax of assignment statements. ZITA does not have an independent notation system to express relationships. It would be redundant, because `MAKEODE.EXE` would become very complicated and the syntax of Pascal is relatively simple. Besides, there are many sources concerning Pascal programing [1].

If `EXA2.ODE` is ready, you can create the *ODEs*-solver by the following DOS command:

`COMPILE EXA2 2` and <ENTER>

The new `ZITA.EXE` is ready now. The other necessary files can be found in *Text 3.10*. Create these files in your user's library! Notice, there are only a few differences between these files and the files created earlier, e.g. `EXA2.BAT` and `EXA1.BAT`. You certainly use your editor perfectly, so you can create the new files from the old ones. What is the reason for these differences? The files containing calculated curves had six columns in the previous exercise, but they have only two columns now. They are the values of time and the calculated volumes of HCl-solution in cm^3. The present fitting uses the second column of the calculated files, while the previous fitting used the third one.

Compare the appropriate files now! `EXA.EXP`, `EXAPARAM.DAT` and `EXAALL.DAT` do not require any modification in the new exercise. The contents of the `*.JCD` files only have different serial numbers for their columns. `EXA2.DRW` differs from `EXA1.DRW` significantly, because we would like to draw only the experimental and calculated volumes on the screen. The `*.BAT` files contain some different filenames, because the input data of the executable programs can be found in other files.

You can see that the creation of files requires much work only at the initial stage. If you would like to execute a parameter or model shift, it requires only a few minutes with a good editor.

If the contents of *Text 3.10* have been typed into your user's library, the fitting can be carried out by the following commands:

`EXA2` and <ENTER>

This fitting also consists of six iterations. After the fitting, `RESULT.6` should correspond to *Text 3.11*. If you compare the result of the previous and new results (*Text 3.8* and *Text 3.11*), you can see that the values of the rate constants are nearly the same. The primary experimental data have only two or three significant figures, which explains the small differences.

3.5 Choosing the Initial Parameter Values and the Fitting Method

Now, let us study the first exercise again! We try to solve the problem with new initial parameter values. First, we have to recreate the older `ZITA.EXE` with a '`COMPILE EXA1 2`'[11] DOS command. Secondly, we have to rewrite `EXAPARAM.DAT`. Replace the contents of this file with *Text 3.12*! As you can see, there are two differences between the old and new files: the initial values of the rate constants are inverted and the maximum number of iterations is increased to twenty. If the new `EXAPARAM.DAT` file is ready, the '`EXA1`' DOS command executes the fitting.

You can see from the results that a relative small change of the initial values causes a huge effect in the results. The last fitting consists of eleven iterations instead of six. The contents of `RESULT.11` are presented in *Text 3.13*.

[11] DOS commands to be executed are written in typewriter style between two single quotes in this book.

```
ITERATION 6;     WSTD: 1.154575758E-0002;    NMD:12;    MP: 1.000E-0005
******** Result(s):
       Estimate                    Std. Dev. (abs & %)          dp
  p1  3.62291421562072282E+0000  1.657E-0001  &      4.57     6.457E-0004
  p2  9.90387750451910579E-0002  3.308E-0003  &      3.34     4.182E-0006
******** Goodness-of-Fit Statistics:
                                Marquardt Parameter: 1.000E-0005
                                      Relative Step: 1.0000
                             Number of Measured Data: 12
                         Number of Fitted Parameters: 2
               Weighted Standard Deviation of Data: 1.154575758207151E-02
             Unweighted Standard Deviation of Data: 1.154575758207151E-02
Correlation between Measured and Calculated Values: 9.992709687231773E-01
                      Coefficient of Determination: 9.984792819931050E-01
                                         R-squared: 9.996907638344557E-01
                           Model Selection Criterion: 6.155239349366413E+00
******** Residual Analysis:
Serial Correlation: 1.52493603949625873E+0000
Heteroscedactivity:-1.07352729581745339E-0002
          Skewness: 9.25452686346566715E-0001
          Kurtosis:-9.19756805032528329E-0001
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2
  p1 { .102}-.102
  p2  -.102{ .102}
******** 95% Univariate Confidence Interval(s):
  p1  3.25818830364219585E+0000 - 3.98764012759924979E+0000
  p2  9.17571197648751036E-0002 - 1.06320430325507012E-0001
******** 95% Supporting Plane Confidence Interval(s):
  p1  3.14822920923204911E+0000 - 4.09759922200939653E+0000
  p2  8.95618158554628616E-0002 - 1.08515734234919254E-0001
```

Text 3.11: Final results for the calculations based on the primary experimental data.

```
        Column
Line    ....5....10....15....20
   1: 1.0 * p1
   2: 2.0 * p2
   3:
   4: 0.001 0.99
   5: -1.0 1.0E-04 1.0E-04
   6: 20
```

Text 3.12: New contents of `EXAPARAM.DAT` file.

If you compare *Text 3.8* and *Text 3.13*, you can see that the last calculation did not find good values for the rate constants. This can be seen better if we make figures. *Figure 3.3* contains figures which were made with '`DRAWER EXA1`' DOS commands after the fitting. The experimental curve plotted with asterisks is almost the same as the calculated one drawn with solid line in figure **a**. This figure was created immediately after the first fitting. Figure **b** was made after the third fitting, and it shows the differences between the experimental and calculated $[\mathbf{OH}]_t$. If you analyze every `RESULT.*` file, you can establish that the algorithm lost its way already at the beginning of the fitting.

```
ITERATION 11;     WSTD: 1.763368910E-0001;    NMD:12;    MP: 1.678E+0000
******** Result(s):
       Estimate                    Std. Dev. (abs & %)          dp
  p1  4.6397613224541084E-0001  4.248E-0001  &     91.56    -3.525E-0004
  p2  1.2622073220996769 4E+0003  5.038E+0004  &   3991.41    -2.227E+0003
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.678E+0000
                                  Relative Step: 1.0000
                           Number of Measured Data: 12
                      Number of Fitted Parameters: 2
              Weighted Standard Deviation of Data: 1.763368910129677E-01
            Unweighted Standard Deviation of Data: 1.763368910129677E-01
Correlation between Measured and Calculated Values: 9.466638807155900E-01
                     Coefficient of Determination: 6.452763226065060E-01
                                        R-squared: 9.278673696717094E-01
                          Model Selection Criterion: 7.030828328028134E-01
******** Residual Analysis:
Serial Correlation: 2.73811021681426713E+0000
Heteroscedactivity:-3.70624349033706278E-0001
          Skewness:-2.30341724146537931E-0001
          Kurtosis:-1.29584919374270723E+0000
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2
  p1 { .958} .958
  p2   .958{ .958}
******** 95% Univariate Confidence Interval(s):
  p1 -4.71005861843509498E-0001 - 1.3989581242925916 7E+0000
  p2 -1.0962305417495306 5E+0005 - 1.1214746881915241 9E+0005
******** 95% Supporting Plane Confidence Interval(s):
  p1 -7.5288814624125145 0E-0001 - 1.6808404086903336 2E+0000
  p2 -1.4305320689293729 8E+0005 - 1.4557762153713665 2E+0005
```

Text 3.13: Result for the first problem based on the wrong initial parameter values.

What is the reason for this phenomenon? We show the function $S(\vec{k}_i)$ defined in *Eq. (2.3)* on *Figure 3.4***a** (page 42) for the explanation. Since we have two parameters and we use relative residuals, so $S(\vec{k}_i) = S_{\text{rel}}(k_1, k_2)$. The figure shows the surface of this function. The k_1- and k_2-axes have logarithmic scales in order to illustrate larger ranges. A contour plot is also used, since the values of k_1 and k_2 can be determined more precisely from this kind of representation. The density of contour lines also characterizes the surface. It can be seen from this figure that the shape of $S_{\text{rel}}(k_1, k_2)$ is similar to a parabolic surface near the minimum. The values of k_1 and k_2 at this minimum constitute the solution. However, this surface also has a saddle point near the minimum. Moreover, there is a long valley after the saddle point. The value of $S_{\text{rel}}(k_1, k_2)$ decreases in this valley as we move away from the saddle point. Therefore, if the actual values of k_1 and k_2 are in this valley, they cannot get out of it, so the fitting will not be successful.

Why are these observations important for a chemist? There are two conclusions which are general relevance:

- Chemical considerations cannot be replaced by a good program for parameter esti-

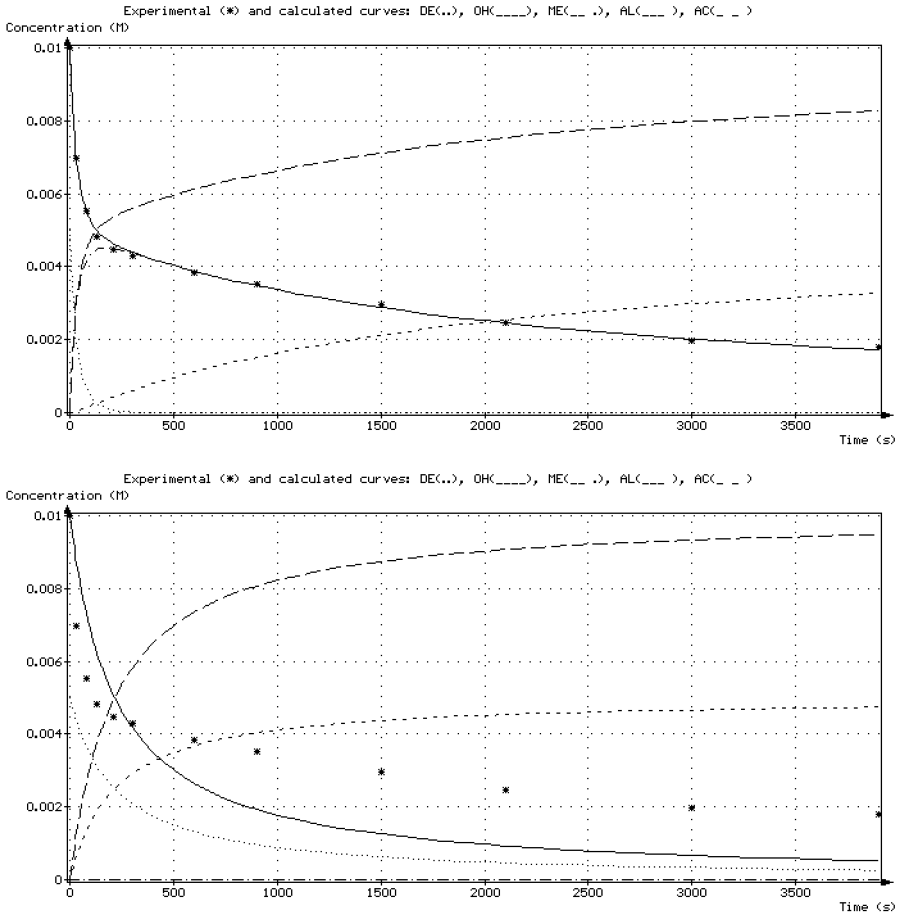


Figure 3.3: The saponification of diethyl malonate. (**a**): $k_{1,0} = 2.0\,\mathrm{M^{-1}\,s^{-1}}$ and $k_{2,0} = 1.0\,\mathrm{M^{-1}\,s^{-1}}$, (**b**): $k_{1,0} = 1.0\,\mathrm{M^{-1}\,s^{-1}}$ and $k_{2,0} = 2.0\,\mathrm{M^{-1}\,s^{-1}}$.

mation! For example, it was a silly idea to choose k_2 larger than k_1 in this chemical system. A good chemist would never do a thing like this! You should always choose the best initial estimate of the parameter values! Chemical considerations help to find these values in many cases.

- The best shape for $S(\vec{k}_i)$ is an n-dimensional parabolic surface near the minimum, because this shape has only one minimum. The shape can be modified by the following:

  - It is generally true that the parabolic characteristic is valid in a larger range, more information is contained by experiments concerning the parameters. So you have to plan and carry out the experimental work precisely!
  - The surface of $S(\vec{k}_i)$ depends on how the residuals are calculated. There are three methods to compute them when ZIA is used: absolute, relative and orthogonal fitting. The earlier calculations were based on relative fitting. The next section shows an example of orthogonal fitting.

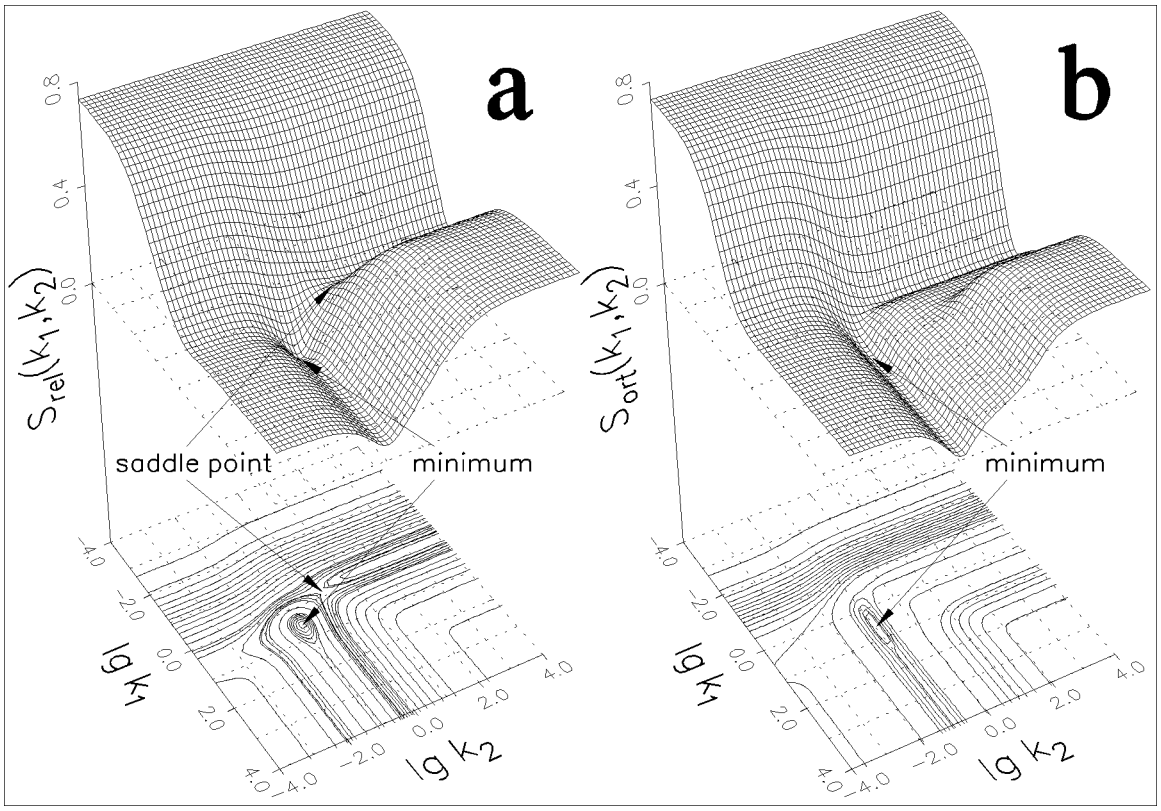


Figure 3.4: The surface of $S(\vec{k}_i)$ in the first example in the cases of relative (**a**) and orthogonal (**b**) residuals.

3.6 Solving the Last Problem by Orthogonal Fitting

We have assumed until now that only the experimental volumes have errors, but the experimental time points are exact. If we want to take into consideration the errors in both data series, the definition of the residual has to be changed. In this section the residual of an experimental point means the smallest distance between the experimental point and the calculated curve. The algorithm based on this definition is called orthogonal fitting. Its exact mathematical background can be found in *Appendix A*.

To change the fitting method is a very simple task with ZITA. Now, we try to solve again the problem described in *Section 3.5*. All the necessary files can be used without any change except `EXA1.BAT`. Lines 8, 10 and 12 of this file contain an '`r`' in their 27$^{\text{th}}$ columns. You should change these letters to '`o`'. After this, the orthogonal fitting can be executed by the '`EXA1`' DOS command. The process requires twelve iterations. The contents of the `RESULT.12` file are indicated in *Text 3.14*. You can see that the task which was unsolvable with the relative method became solvable when orthogonal fitting was used.

If you compare the contents of *Text 3.8* and *Text 3.14* you can see that the values of k_2 are in a good agreement, but this is not true for the values of k_1. However, if you consider

```
ITERATION 12;     WSTD: 6.362216751E-0003;    NMD:12;    MP: 1.000E-0011
******** Result(s):
       Estimate                    Std. Dev. (abs & %)          dp
  p1  5.04175189404187004E+0000  1.573E+0000  &     31.19    -7.194E-0002
  p2  9.73660183067900752E-0002  1.932E-0003  &      1.98    -1.644E-0005
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0011
                                  Relative Step: 1.0000
                            Number of Measured Data: 12
                        Number of Fitted Parameters: 2
               Weighted Standard Deviation of Data: 6.362216750688347E-03
             Unweighted Standard Deviation of Data: 6.362216750688347E-03
Correlation between Measured and Calculated Values: 9.997537637902384E-01
                      Coefficient of Determination: 9.994566115329542E-01
                                        R-squared: 9.999033401199047E-01
                          Model Selection Criterion: 7.184352751555948E+00
******** Residual Analysis:
Serial Correlation: 3.83529364880113186E-0001
Heteroscedactivity: 2.06587683591525346E-0001
          Skewness:-5.57227476640383365E-0001
          Kurtosis:-9.43990907054388175E-0001
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2
  p1 { .036} .036
  p2   .036{ .036}
******** 95% Univariate Confidence Interval(s):
  p1  1.58042425674684800E+0000 - 8.50307953133689208E+0000
  p2  9.31140970108963181E-0002 - 1.01617939602683832E-0001
******** 95% Supporting Plane Confidence Interval(s):
  p1  5.36888714155124094E-0001 - 9.54661507392861598E+0000
  p2  9.18322099832830426E-0002 - 1.02899826630297108E-0001
```

Text 3.14: Final result for the first exercise computed by orthogonal fitting.

the ranges of k_1 determined by their deviations, you can see that the two ranges of k_1 overlap, so both results are consistent.

What is the reason for the large deviation of k_1 in the case of orthogonal fitting? It can be understood from *Figure 3.4***b** in which we plot the orthogonal residuals. It can be seen clearly that the saddle point has disappeared but the parabolic shape is not completely valid for k_1. The minimum lies in a long valley which is parallel to the k_1-axis. This valley does not have a well-defined and sharp minimum with regard to k_1 but it has only a flat one. This is not true in the case of $S_{\mathrm{rel}}(k_1, k_2)$, which has a sharp minimum.

The behavior of these surfaces can be explained if we study *Figure 3.3*. You can see that the experimental curve can be divided into two parts. k_1 determines the first, steeper part and k_2 determines the second, less steep part. If absolute or relative fitting is used, a small change in the calculated curve in the first part increases considerably the value of $S(\vec{k}_i)$. This is not true when orthogonal fitting is used, so the value of k_1 is computed less accurately in the latter case. This means, too, that more experimental points would be needed in the first part to get a more precise k_1! The WIWO principle (Wrong Input

Wrong Output) is also valid in kinetics!

Now, we finished our examination of the reaction between diethyl malonate and hydroxide ion. We begin to study another reaction, namely the saponification of diethyl adipate in an alkaline solution. Two important questions will be answered about ZITA:

- How can you fit on the basis of more experimental curves?
- What is the most efficient and fastest method for fitting when ZITA is used?

3.7 Fitting Based on Several Experimental Curves

We assume that *Eq. (3.1)* is also valid for the saponification of diethyl adipate. The necessary files can be found in *Texts 3.15-3.17*. Write these files into your user's library after reading the following remarks:

- We use two experimental curves for computing k_1 and k_2. Both curves contain time points and [**OH**]-s. The data of `EXB0001.EXP` are derived from [11] and the data of `EXB0002.EXP` were published in [12].
- `EXB.ODE` is almost the same as `EXA1.ODE`. Only the third number of the first line differs. This number is greater or equal to the number of experimental data pairs. If there are several experimental curves, you must choose the highest number! The number of experimental data pairs can be different in the different curves. It is only pure chance that these numbers are the same in the actual curves.
- The values contained in `EXB1PRM.DAT` were chosen on the basis of the considerations detailed in *Subsection 3.3.3*.
- The structure of `EXB.DRW` and `EXA1.DRW` is the same but `EXB.DRW` instructs the drawing program to plot both curves.
- `ZITA.EXE` creates six calculated curves from `EXBALL.DAT`: two for the elements of $\left(\vec{Y} - \vec{F}(\vec{k}_i)\right)$ and four more for the elements of $\boldsymbol{\mathcal{J}}(\vec{k}_i)$. The structure of `EXBALL.DAT` and `EXAALL.DAT` is the same.
- `EXBK*.JCD` files contain information concerning two curves instead of one curve. In this way, `J*.COL` files will contain the residuals and difference quotients of both curves.
- `EXB1.BAT` and `EXA1.BAT` are practically the same, only the filenames differ from one another.

If the files listed in *Texts 3.15-3.17* are written in your user's library, you can create the appropriate `ZITA.EXE` by the '`COMPILE EXB 2`' command. The '`EXB1`' command executes the fitting. It requires only four iterations. The contents of `RESULT.4` are indicated in *Text 3.18*, and *Figure 3.5* shows the experimental and calculated curves after the fitting. They show clearly that the results are acceptable.

EXB0001.EXP

```
Line  Column
      ....5....10....15....20....25
   1: 0.00498 0.00996 0.0 0.0 0.0
   2:     0 0.00996
   3:   165 0.00931
   4:   370 0.00864
   5:   510 0.00822
   6:   628 0.00791
   7:   757 0.00760
   8:   965 0.00719
   9:  1180 0.00676
  10:  1357 0.00648
  11:  1671 0.00599
  12:  2062 0.00553
  13:  2468 0.00510
  14:  3077 0.00456
  15:  3664 0.00420
  16:  4466 0.00376
  17:  5271 0.00340
```

EXB0002.EXP

```
Line  Column
      ....5....10....15....20....25
   1: 0.00451 0.00902 0.0 0.0 0.0
   2:     0 0.00902
   3:   161 0.00849
   4:   309 0.00808
   5:   460 0.00768
   6:   655 0.00724
   7:   860 0.00684
   8:  1056 0.00650
   9:  1359 0.00604
  10:  1660 0.00565
  11:  1956 0.00531
  12:  2259 0.00501
  13:  2658 0.00468
  14:  3161 0.00431
  15:  3659 0.00401
  16:  4158 0.00374
  17:  4660 0.00352
```

EXB.ODE

```
Line  Column
      ....5....10....15....20
   1: 2 5 20
   2: DE  OH  ME  AL  AC
   3:  1   1   0   0   0  v1
   4:  0   1   1   0   0  v2
   5:
   6:  1   1   1   1   1
   7:
   8: -1  -1   1   1   0  v1
   9:  0  -1  -1   1   1  v2
  10: 0 0
```

EXB1PRM.DAT

```
Line  Column
      ....5....10....15....20
   1: 0.08 * p1
   2: 0.04 * p2
   3:
   4: 0.001 0.99
   5: -1.0 1.0E-04 1.0E-04
   6: 12
```

EXB.DRW

```
Line  Column
      ....5....10....15....20....25....30....35....40....45....50....55....60....65....70....75....80....85
   1: Experimental (1) and calculated curves: DE(..), OH(____), ME(__ .), AL(___ ), AC(_ _ )
   2: Time (s)
   3: Concentration (M)
   4: 6 1 Z
   5: Exb0001.exp 1 2 1 0 1
   6: Exb0001.sim 1 2 0 1 1000100010001000
   7: Exb0001.sim 1 3 0 1 1111111111111111
   8: Exb0001.sim 1 4 0 1 1111111000010000
   9: Exb0001.sim 1 5 0 1 1111111111110000
  10: Exb0001.sim 1 6 0 1 1110000011100000
  11: 0
  12: y
  13: Experimental (2) and calculated curves: DE(..), OH(____), ME(__ .), AL(___ ), AC(_ _ )
  14: Time (s)
  15: Concentration (M)
  16: 6 1 Z
  17: Exb0002.exp 1 2 1 0 2
  18: Exb0002.sim 1 2 0 1 1000100010001000
  19: Exb0002.sim 1 3 0 1 1111111111111111
  20: Exb0002.sim 1 4 0 1 1111111000010000
  21: Exb0002.sim 1 5 0 1 1111111111110000
  22: Exb0002.sim 1 6 0 1 1110000011100000
  23: 0
  24: n
```

Text 3.15: Files for determining the rate constants of the saponification of diethyl adipate. Part 1/3.

EXBALL.DAT

```
      Column
Line  ....5....10....15....20....25....30....35....40....45....50
   1: This is the second example to use ZiTa.
   2: Reaction between diethyl adipate and hydroxide ion.
   3: Model: DE + OH -> ME + AL ,  ME + OH -> AC + AL
   4: 6 80 0
   5: 21 1.0E-5 1.0E-5
   6: 5
   7: 1 DE
   8: 2 OH
   9: 3 ME
  10: 4 AL
  11: 5 AC
  12: 0
  13: 2 Exb0001.exp 5 1 2 3 4 5
  14: 9 5 5 0.0
  15: p Exb0001.sim
  16: 0
  17: 0
  18: y
  19: 1 1.0 +
  20: 2 Exb0001.exp 5 1 2 3 4 5
  21: 9 0 0 0.0
  22: p Exb1k1.out
  23: 0
  24: 0
  25: y
  26: 2 1.0 +
  27: 2 Exb0001.exp 5 1 2 3 4 5
  28: 9 0 0 0.0
  29: p Exb1k2.out
  30: 0
  31: 0
  32: y
  33: 0
  34: 2 Exb0002.exp 5 1 2 3 4 5
  35: 9 5 5 0.0
  36: p Exb0002.sim
  37: 0
  38: 0
  39: y
  40: 1 1.0 +
  41: 2 Exb0002.exp 5 1 2 3 4 5
  42: 9 0 0 0.0
  43: p Exb2k1.out
  44: 0
  45: 0
  46: y
  47: 2 1.0 +
  48: 2 Exb0002.exp 5 1 2 3 4 5
  49: 9 0 0 0.0
  50: p Exb2k2.out
  51: 0
  52: 0
  53: n
```

Text 3.16: Files for determining the rate constants of the saponification of diethyl adipate. Part 2/3.

EXBK1.JCD

```
      Column
Line  5   10   15   20   25   30   35   40   45
1:  0 0
2:  Exb0001.exp 2 1 Exb0001.sim 3 1.0  Exb1k1.out
3:  Exb0002.exp 2 1 Exb0002.sim 3 1.0  Exb2k1.out
```

EXBK2.JCD

```
      Column
Line  5   10   15   20   25   30   35   40   45
1:  0 0
2:  Exb0001.exp 2 1 Exb0001.sim 3 1.0  Exb1k2.out
3:  Exb0002.exp 2 1 Exb0002.sim 3 1.0  Exb2k2.out
```

EXB1.BAT

```
      Column
Line  5   10   15   20   25   30   35   40   45   50   55   60   65   70   75
1:   Copy Exb1prm.dat Paramtrs.dat
2:   Copy Exb000?.exp %ZITAEXPR%
3:  :La
4:   Zita Exball.dat
5:   If Errorlevel 1 Goto Lb
6:   %ZITABASE%\Drawer Exb.drw
7:   If Errorlevel 1 Goto Lb
8:   %ZITABASE%\Jc Exbk1.jcd r 0
9:   If Errorlevel 1 Goto Lb
10:  %ZITABASE%\Jc Exbk1.jcd r 1 + 1.0
11:  If Errorlevel 1 Goto Lb
12:  %ZITABASE%\Jc Exbk2.jcd r 2 + 1.0
13:  If Errorlevel 1 Goto Lb
14:  %ZITABASE%\Gnm *
15:  If Errorlevel 1 Goto Lb
16:  Echo Off
17:  For %%i In (1 2 3 4 5 6 7 8 9 10 11 12) Do If Exist Result.%%i Type Result.%%i
18:  Pause
19:  Echo On
20:  If Not Exist Res1 Goto Lc
21:  Copy Res1 Paramtrs.dat
22:  Goto La
23: :Lb
24:  Echo There is an error in the calculations!
25: :Lc
```

Text 3.17: Files for determining the rate constants of the saponification of diethyl adipate. Part 3/3.

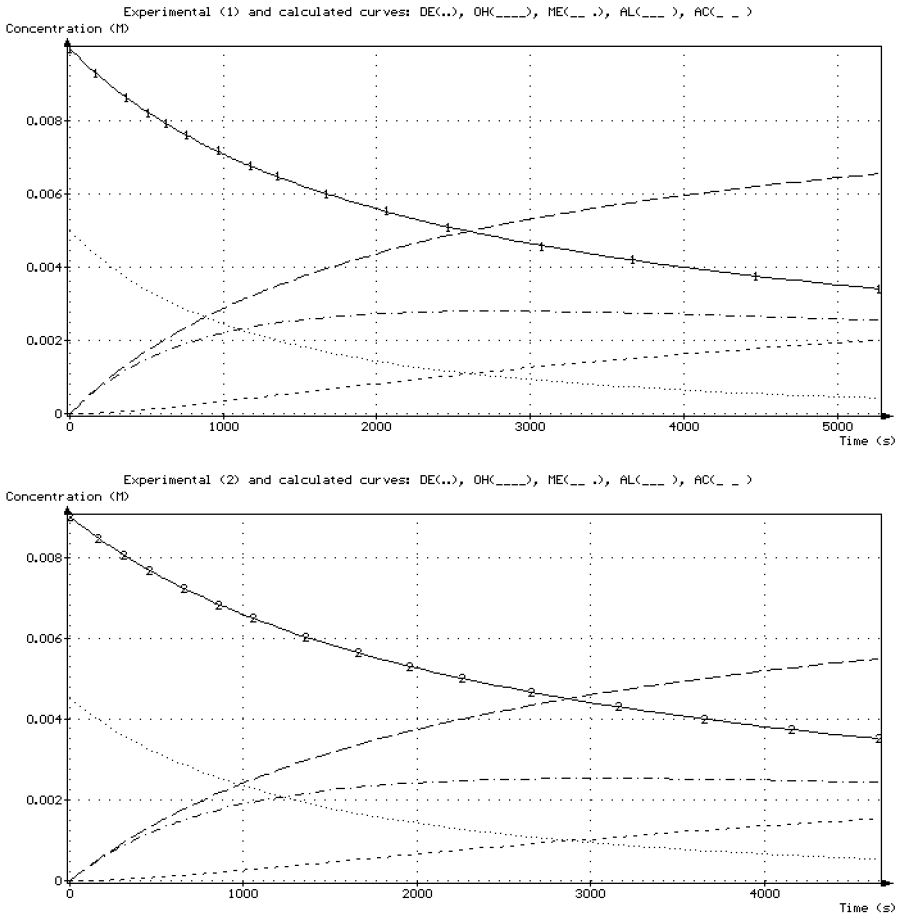


Figure 3.5: The saponification of diethyl adipate in the case of $k_{1,0} = 0.08\,\mathrm{M^{-1}\,s^{-1}}$ and $k_{2,0} = 0.04\,\mathrm{M^{-1}\,s^{-1}}$. (**a**) curves based on `EXB0001.EXP` and (**b**) curves based on `EXB0002.EXP`.

```
ITERATION 4;     WSTD: 1.626533846E-0003;    NMD:32;    MP: 1.000E-0005
******** Result(s):
       Estimate                    Std. Dev. (abs & %)          dp
  p1  8.52731612853312536E-0002  1.950E-0004  &      0.23    -2.239E-0006
  p2  3.08700440473363487E-0002  1.838E-0004  &      0.60     1.844E-0006
******** Goodness-of-Fit Statistics:
                                Marquardt Parameter: 1.000E-0005
                                      Relative Step: 1.0000
                             Number of Measured Data: 32
                          Number of Fitted Parameters: 2
               Weighted Standard Deviation of Data: 1.626533845701158E-03
             Unweighted Standard Deviation of Data: 1.626533845701158E-03
Correlation between Measured and Calculated Values: 9.999869059242988E-01
                       Coefficient of Determination: 9.999736362711452E-01
                                          R-squared: 9.999979069419713E-01
                            Model Selection Criterion: 1.041852139950529E+01
******** Residual Analysis:
Serial Correlation:-8.10426803113466557E-0001
Heteroscedactivity:-8.90764911745593488E-0001
          Skewness: 6.31717780434310146E-0002
          Kurtosis: 2.36139945025268115E+0000
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2
  p1 { .825}-.825
  p2  -.825{ .825}
******** 95% Univariate Confidence Interval(s):
  p1  8.48754854016168195E-0002 - 8.56708371690456877E-0002
  p2  3.04951121652350952E-0002 - 3.12449759294376022E-0002
******** 95% Supporting Plane Confidence Interval(s):
  p1  8.47710318087333825E-0002 - 8.57752907619291247E-0002
  p2  3.03966325144043008E-0002 - 3.13434555802683967E-0002
```

Text 3.18: Final result for the first fitting of the saponification of diethyl adipate.

3.8 An Important Problem Involving Chemical Mechanisms

Execute the last fitting again using different initial parameter values! Write the files given in *Text 3.19* into your user's library! You can also change two old files (`EXB1PRM.DAT` and `EXB1.BAT`) for creating the new ones. The new parameter file (`EXB2PRM.DAT`) has a greater k_2 (1.0 instead of 0.04) and the maximum number of iterations is increased to twenty-five. The new batch file (`EXB2.BAT`) differs from `EXB1.BAT` file in two ways:

- `EXB2PRM.DAT` will become `PARAMTRS.DAT` instead of `EXB1PRM.DAT`.
- Lines 6–7 and 16-19 of `EXB1.BAT` cannot be found in `EXB2.BAT`. This means that figures are not drawn and `RESULT.*` files are not typed on the screen during the calculation. In this way the fitting is executed automatically, because you need not press two keys in each iteration.

EXB2PRM.DAT

```
        Column
Line    ....5....10....15....20
  1:  0.08 * p1
  2:  1.0  * p2
  3:
  4:  0.001 0.99
  5:  -1.0 1.0E-04 1.0E-04
  6:  25
```

EXB2.BAT

```
        Column
Line    ....5....10....15....20....25....30....35....40
  1:   Copy Exb2prm.dat Paramtrs.dat
  2:   Copy Exb000?.exp %ZITAEXPR%
  3:  :La
  4:   Zita Exball.dat
  5:   If Errorlevel 1 Goto Lb
  6:   %ZITABASE%\Jc Exbk1.jcd r 0
  7:   If Errorlevel 1 Goto Lb
  8:   %ZITABASE%\Jc Exbk1.jcd r 1 + 1.0
  9:   If Errorlevel 1 Goto Lb
 10:   %ZITABASE%\Jc Exbk2.jcd r 2 + 1.0
 11:   If Errorlevel 1 Goto Lb
 12:   %ZITABASE%\Gnm *
 13:   If Errorlevel 1 Goto Lb
 14:   If Not Exist Res1 Goto Lc
 15:   Copy Res1 Paramtrs.dat
 16:   Goto La
 17:  :Lb
 18:   Echo There is an error in the calculations!
 19:  :Lc
```

Text 3.19: New files for the second fitting of the saponification of diethyl adipate.

Since `ZITA.EXE` can be the same as earlier, the fitting can be executed now by the '`EXB2`' DOS command. This process gives twenty-five `RESULT.*` files in the user's library. The final result is presented in *Text 3.20*. *Figure 3.6* shows pictures which were made by the '`DRAWER EXB`' command after the fitting had been finished.

If you compare *Text 3.18* to *Text 3.20*, you can see two different results for the same problem. A similar problem already occurred in the case of diethyl malonate, but we got two different calculated $[\mathbf{OH}]_t$-s there. Now, instead, you can see from *Figures 3.5-3.6* (pages 48 and 52) that the calculated [**OH**]-s are almost the same in both cases. In other words, two parameter sets differing significantly from each other give the same calculated curve. This means that ZITA is not able to determine the values of k_1 and k_2 based on the measured experimental data.

What is the reason for this phenomenon? You ought to study *Table 3.2* on page 24 to answer this problem. Compare the analytical solutions in the cases of $\kappa = 1/2$ and $\kappa = \infty$! It can be seen that the forms of these two expressions are almost the same. Only a small difference can be found: the first term of the denominator contains $k_1/2$ in the case of $\kappa = 1/2$ and only k_1 in the other case. This implies that the mechanisms:

$$\begin{aligned} \mathbf{DE} + \mathbf{OH} &\xrightarrow{k} \mathbf{ME} + \mathbf{AL} \\ r_1 &= k\cdot[\mathbf{DE}]\cdot[\mathbf{OH}] \end{aligned}$$

$$\begin{aligned} \mathbf{ME} + \mathbf{OH} &\xrightarrow{k/2} \mathbf{AC} + \mathbf{AL} \\ r_2 &= \frac{k}{2}\cdot[\mathbf{ME}]\cdot[\mathbf{OH}] \end{aligned}$$

and

$$\begin{aligned} \mathbf{DE} + \mathbf{2\,OH} &\xrightarrow{k/2} \mathbf{AC} + \mathbf{2\,AL} \\ r &= \frac{k}{2}\cdot[\mathbf{DE}]\cdot[\mathbf{OH}] \end{aligned}$$

```
ITERATION 25;     WSTD: 1.174913521E-0002;    NMD:32;    MP: 8.590E-0007
******** Result(s):
       Estimate                    Std. Dev. (abs & %)          dp
  p1  3.93459045689613561E-0002  2.667E-0004  &      0.68     3.480E-0004
  p2  5.41262693032373868E+0003  1.252E+0004  &    231.36     4.838E+0003
******** Goodness-of-Fit Statistics:
                                Marquardt Parameter: 8.590E-0007
                                     Relative Step: 1.0000
                             Number of Measured Data: 32
                         Number of Fitted Parameters: 2
               Weighted Standard Deviation of Data: 1.174913520969052E-02
             Unweighted Standard Deviation of Data: 1.174913520969052E-02
Correlation between Measured and Calculated Values: 9.997246333880549E-01
                      Coefficient of Determination: 9.986243991663132E-01
                                         R-squared: 9.998907888794840E-01
                           Model Selection Criterion: 6.463864673353162E+00
******** Residual Analysis:
Serial Correlation: 4.67563335291062825E+0000
Heteroscedactivity:-7.04377767587978886E-0001
          Skewness: 2.08164240287129235E-0001
          Kurtosis:-1.59235170888105898E+0000
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2
  p1 { .213} .213
  p2   .213{ .213}
******** 95% Univariate Confidence Interval(s):
  p1  3.88020316912700104E-0002 - 3.98897774466527018E-0002
  p2 -2.01270066014182350E+0004 - 3.09522604620657123E+0004
******** 95% Supporting Plane Confidence Interval(s):
  p1  3.86591779793071566E-0002 - 4.00326311586155556E-0002
  p2 -2.68352496520136692E+0004 - 3.76605035126611466E+0004
The maximum number of iterations has been reached.
The minimum is not reached probably!
```

Text 3.20: Second result for the saponification of diethyl adipate.

cannot be distinguished from each other if the only measured characteristic is $[\mathbf{OH}]_t$. This is a rigorous result, so there is no help for it. Similar relations may occur in many systems, but there is no general method to find them. These facts emphasize that only well-planned and executed experiments can make a mechanism probable! The affordable ways to distinguish between the possible mechanisms in this case are to measure more than one concentration simultaneously and/or to make the initial concentration range wider.

You can see from *Text 3.18* that $\kappa = k_2/k_1 = 0.36$ in the actual problem. Since there can be some experimental errors, this value approaches the theoretical one ($\kappa = 0.5$). This is the reason why ZIA can determine the value of k_1 and k_2 only if the initial values are very good. The theory is also supported by the result that: $k_1^{\kappa_0>1}/k_1^{\kappa_0<1} = 0.46 \approx 0.5$ (κ_0 means the initial ratio of k_2 and k_1). This was also expected from the functions of *Table 3.2*.

These facts are also illustrated in *Figure 3.7*. $S_{\mathrm{rel}}(k_1, k_2)$ was defined in *Section 3.5*, but now it refers to the diethyl adipate – hydroxide ion reaction. Since the shape of $S_{\mathrm{ort}}(k_1, k_2)$

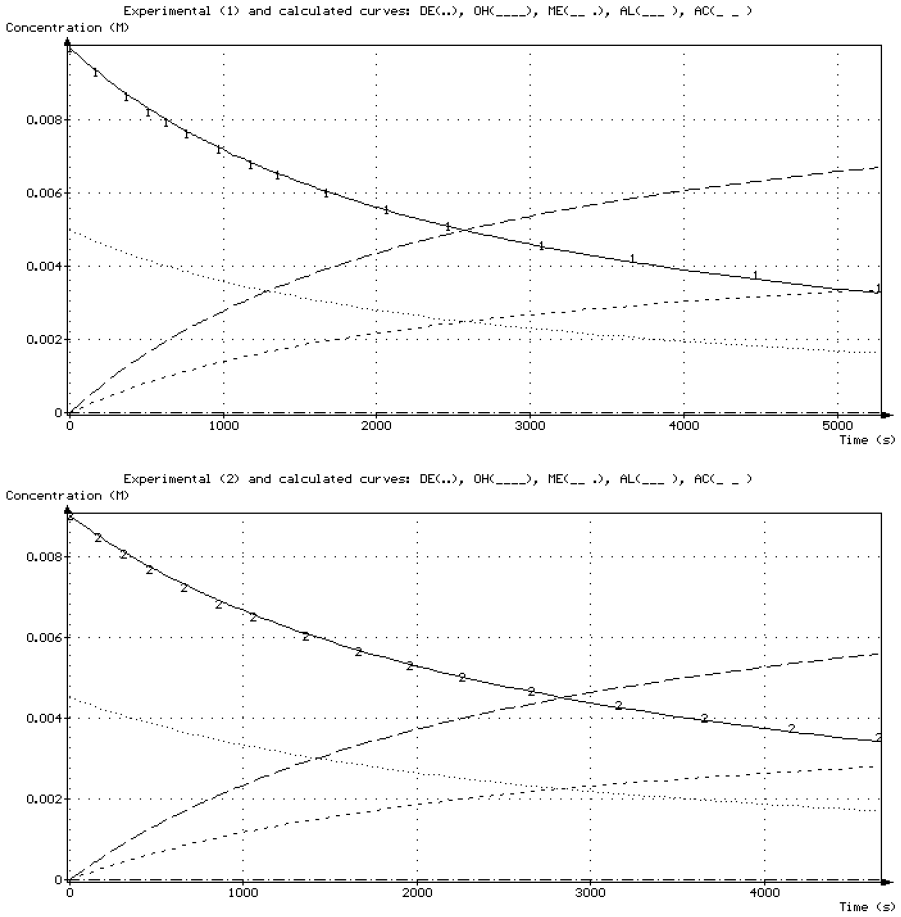


Figure 3.6: The second graphical result in the diethyl adipate – hydroxide ion reaction in the case of $k_{1,0} = 0.08\,\mathrm{M^{-1}\,s^{-1}}$ and $k_{2,0} = 1.0\,\mathrm{M^{-1}\,s^{-1}}$. (**a**) curves based on `EXB0001.EXP` and (**b**) curves based on `EXB0002.EXP`.

does not differ significantly from $S_{\mathrm{rel}}(k_1, k_2)$, only the latter function is studied here.

It can be clearly seen on *Figure 3.4***a** (page 42) that the surface has a global minimum in the range examined. However, the surface plotted on *Figure 3.7* has no global minimum up to four digits! The **view B** in this figure shows that the minimum to be searched for is only a local minimum. You can see a long valley near this minimum. Every point of its bottom and the local minimum have almost the same function value! Thus any point in this bottom can be the solution. Another fact can be seen from **view A** well. There are two valleys on the surface and they are parallel to the k_1-axis. If k_2 is small (less than ~ 0.04) the bottom gives $k_1 \sim 0.08$. However, if k_2 is larger than ~ 0.1 then $k_1 \sim 0.04$. This also illustrates the theoretical prediction.

If you compare *Figure 3.5* and *Figure 3.6*, you can see that the $[\mathbf{OH}]_t$ and $[\mathbf{AL}]_t$ curves are nearly identical in the two pictures. The simplified mechanism does not contain **ME**. So, if we want to determine the real mechanism of the reaction, we must follow [**DE**] or

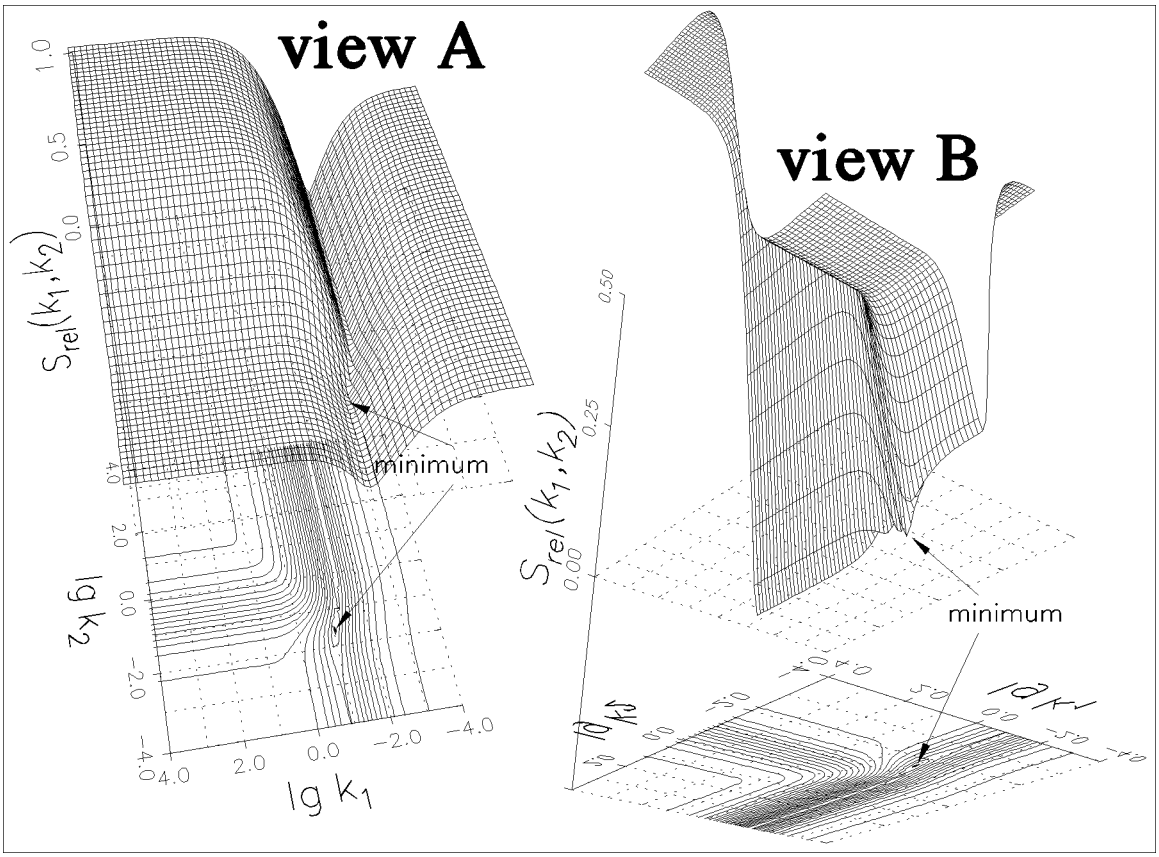


Figure 3.7: The surface of $S(\vec{k}_i)$ in the diethyl adipate – hydroxide ion reaction.

[**AC**] as a function of time.

3.9 A More Powerful Method

You could see from your work up to now that you had to create a lot of files to solve a problem from the beginning. If you made a parameter or model shift, only a few new files were necessary, but many old files had to be revised. The changes were mainly in the filenames, because ZITA does not have any restrictions on filenames. Even the extensions were only recommended, but they are not compulsory. If the user keeps to some restrictions about filenames, the program named `TASK.EXE` generates a part of the necessary file and executes the fitting.

`*.DAT`, `*.JCD` and `*.BAT` can be replaced by `TASK.EXE`, but experimental files, `*.ODE` and `*.DRW` cannot. The user has to determine which experimental data are to be used and what is the supposed chemical mechanism. In addition, ZITA cannot classify the curves to be plotted in a rational manner. `TASK.EXE` requires a new file for its input data. Its recommended extension is `TSK`.

```
      Column
Line  ....5....10....15....20....25....30....35....40....45....50....55....60....
  1: ******+ the name of OTHER TASK(s).
  2:
  3: + the BASENAME of the task. (Base)
  4: Exb
  5: + the number of SIGNIFICANT DIGITS in files. (6)
  6: 6
  7: + the NUMBER OF CHARACTERS in a line of files. (80)
  8: 80
  9: + the PLACE OF THE RESULT FILES after simulation. (0)
 10: 0
 11: + the number of INTEGRATION METHOD. (21)
 12: 21
 13: + the RELATIVE ERROR bound during simulation. (1E-8)
 14: 1.0E-05
 15: + the INITIAL STEP size during simulation. (1E-10)
 16: 1.0E-05
 17: + the ACCOMPLISHED RANGE is displayed /1/ or not /0/. (0)
 18: 0
 19: + the number of INNER POINTs. (0)
 20: 0
 21: + the number of LINES MUST BE SKIPPED in experimental files. (1)
 22: 1
 23: + ABSOLUTE, RELATIVE or orthogonal fitting. (R)
 24: R
 25: + this character signs the DIFFERENTIATION METHOD. (+)
 26: +
 27: + the modifying PERCENT. (1.0)
 28: 1.0
 29: + J*.COL files are erased /0/ or not /1/ after the fitting. (0)
 30: 0
 31: + the serial numbers of EXPERIMENTAL CURVES.
 32: 1 2 -1
 33: + the CONTENT OF THE RESULT files.
 34: 5
 35: 1
 36: 2
 37: 3
 38: 4
 39: 5
 40: + the FIGURES are saved /a number/ or not /0/ to the disk. (0)
 41: 0
 42: + the INITIAL CONCENTRATIONs in experimental files.
 43: 5 1 2 3 4 5
 44: + the columns of the CHARACTERISTICs to be compared.
 45: 2 3 1.0
 46: -1
 47: + the form of the OUTPUT FILES (c)
 48: c
 49: + SPEED OPTIMALIZATION (no)
 50: no
 51: + the content of paramtrs.dat file for MODEL 1.
 52: 0.08 * p1
 53: 0.04 * p2
 54:
 55: 0.001 0.99
 56: -1.0 1.0E-04 1.0E-04
 57: 12
 58: + the content of paramtrs.dat file for MODEL 2.
 59: 0.08 * p1
 60: 1.0  * p2
 61:
 62: 0.001 0.99
 63: -1.0 1.0E-04 1.0E-04
 64: 25
```

Text 3.21: Contents of `EXB.TSK` file.

Carry out the last two tasks using TASK.EXE! To do this the following files are needed: EXP0001.EXP, EXP0002.EXP and EXB.ODE. All the other files can be erased (but this is not necessary). TASK.EXE does not create EXB.DRW, but this file is not necessary during the computation. The contents of the necessary new file is in *Text 3.21*. Write this file into your user's library!

TASK.EXE works in the following way:

1. At first, the parameters of the fitting are read from EXB.TSK. This file contains that information which was contained in *.DAT, *.JCD and *.BAT files until now. The syntax of the *.TSK file is not detailed here. Only the main structure is discussed now (see *Chapter 12* to get more information).

   *.TSK contains logical units. The first line of a unit has to begin with a '+' sign and has to contain a *keystring*. Keystrings are written with capital letters in EXB.TSK but this is not necessary. A keystring shows what kind of information is stored in the lines after the first one. If you study *Text 3.21*, you can see there are some unknown logical units, but there are some familiar ones, too. These pieces of information have already occurred in *.DAT, *.JCD and *.BAT. For example, the keystring named 'MODEL' gives the contents of PARAMTRS.DAT.

2. TASK.EXE creates the necessary *.DAT and *.JCD files based on EXB.TSK. The appropriate batch program is not created because TASK.EXE carries out the functions of *.BAT files, too. In this way, you need not know the syntax of batch programs perfectly.

3. TASK.EXE carries out the fitting. More initial parameter series can be given in a *.TSK file, so TASK.EXE can execute several fittings successively. The computation can be interrupted at any time by pressing **CTRL-Break** keys and can be restarted if your user's library is not changed during the interruption. For example, if five iterations have been performed and you terminate the run during the sixth one, the computation will be continued with the sixth iteration after restarting. There is another way as well which searches for the partial result of the sixth iteration, but it is not detailed until *Chapter 12*.

4. If a fitting is complete, TASK.EXE creates a MODELi.RES file where i is the serial number of the task. This file contains the contents of RESULT.*, the last PARAMTRS.DAT and the most important information from the last J*.COL files. TASK.EXE erases the temporary files before termination.

The general working method with TASK.EXE is the following[12]:

1. First, a basename must be chosen for the task. It contains maximum four alphanumeric characters and we indicate it with 'name' later.

   This basename is EXB in the present task.

[12]The longer paragraphs give general information and the shorter ones concern the actual problem.

2. The experimental data have to be written into files. The extension of these files must be `EXP`. A filename is started with the basename and continued the serial number of the file. This number always consists of four digits, so it contains leading zeroes as well. For example, the name of the file which contains the 13^{th} experimental curve will be `name0013.EXP`.

   These files are ready. Their names are `EXB0001.EXP` and `EXB0002.EXP`.

3. The programs of ZITA search the experimental files in a predefined library. The `%ZitaEXPR%` DOS environment variable contains the name of the library. The experimental files have to be moved into this library before the computation.

   The '`Set ZitaEXPR`' DOS command shows the name of this library. If it does not contain the files in question, you have to move them into the appropriate library.

4. The mathematical model of the mechanism has to be written into `name.ODE` file. If it is ready, `ZITA.EXE` must be created using the '`COMPILE name 2`' command.

   `EXB.ODE` is ready. If you have erased or changed the last `ZITA.EXE`, you have to recreate it with the '`COMPILE EXB 2`' command.

5. A `name.TSK` file has to be produced. It contains all the necessary information. Its name is the DOS parameter of `TASK.EXE`. If the user starts this program with the DOS parameter, the task will be carried out and the results are placed into `MODEL*.RES` files.

   If `EXB.TSK` is placed into your user's library, '`TASK EXB`' command executes the fitting. The result can be found in the `MODEL1.RES` and `MODEL2.RES` files. However, the old `RESULT.*` and `*.COL` files must be erased from the user's library before starting `TASK.EXE`!

Chapter 4

Summary of the User's Guide

Now you have finished studying the exercises. I hope you have taken a fancy to this program package and can find enough time for the Reference Guide. Probably you are not able to solve your own problems with ZITA yet because you do not know the exact structure of the input files. But you have accomplished two very important things:

- You are able to install ZITA.
- You have a more or less clean picture of how ZITA works, what the minimal possibilities to evaluate kinetic experiments are.

This chapter contains three sections. The first summarizes what you now know about using ZITA. The second gives the block scheme for fitting data. It will be useful if you study the Reference Guide. The third section gives you some advice concerning how to use the Reference Guide.

4.1 What Do You Know about Using ZITA?

The input data must be given in ASCII-files and the results appear in the same type of files after runs. The filenames can possess controlled extensions depending on their aims and they can be created or changed by any kind of ASCII-editor.

Fitting a kinetic model is not achieved by a single program; instead a given partial result is produced by a given program. The programs of ZITA call each other depending on the partial results. On the one hand, this is necessary because all of these programs together would occupy too much memory; on the other hand, the computation can be controlled much more easily this way. To perform automatic fitting, the programs should be organized into a `*.BAT` file or the user can use the facilities of `TASK.EXE`. The examples and tests of ZITA (see *Chapter 7*) also present the application of batch files and `TASK.EXE`.

The general technique of solving a fitting task is the following (the meaning of filenames is given in detail in the Reference Guide):

1. Compilation of the `MAKEODE.EXE`, `JC.EXE`, `GNM.EXE` and `TASK.EXE` programs is needed if the installation program has not created them. It can be achieved by using the command file `NEWEXE.BAT`.

2. The points and the initial parameters of the experimental curves should be written into data files using the syntax given in *Chapter 8*.
3. If the *ODEs* of a kinetic model is supplied in matrix form, it must be written into a `*.ODE` file using the conventions described in *Chapter 9*. The mathematical relationships between measured data and experimental concentrations are stored in the same file.

   If the source text of *ODEs* is to be written manually, it must be written into the file `%ZitaTEMP%\ODES.PAS`. This requires knowledge of the syntax of PASCAL assignment statements.

   The *ODEs* solver named `ZITA.EXE` should be created with the help of the actual `*.ODE` and `COMPILE.BAT` files.
4. The data files which contain the data of simulations must be set up (see below in *Chapter 9*).
5. The temporary calculated files (`J*.COL`) for an iteration and some other data files should be calculated or created from the experimental and calculated curves. The structure of these files is given in *Chapter 10*.
6. The programs of ZITA can run either interactively or with DOS parameters. This makes it possible to create a DOS command file that allows one or more iterations to be carried with a given model. Naturally, the user can use `TASK.EXE` instead of batch programs.

If the chemical mechanism needs some changes then only the third and sixth parts of the algorithm need to be repeated if the data files are organized correctly.

4.2 Block Scheme for Fitting

Figure 4.1 presents the block scheme of ZITA-s work process. It shows the relationships between the subprocesses of fitting. This scheme also contains those parts which have not appeared in *Chapter 3*. These are described only in the Reference Guide.

What is the aim of this scheme? This scheme helps us to understand the structure of ZITA. It may be useful to review some of the previous material in order to recall the key points of the relevant chapters. In this way the essence of the subprocess in question becomes easily understandable.

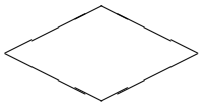
The symbols used in the scheme are the following:

The purpose of a task within a frame like this is the preparation for fitting. The files belonging to these tasks do not change during fitting or model shifts.

Tasks within a frame like this are the components of an iteration. The ASCII-files belonging to these tasks, e.g. simulated curves, `J*.COL`-s, etc., will change during fitting.

Tasks within a frame like this are the components of a model shift. The ASCII-files belonging to these tasks, e.g. `*.ODE`-s, `*.TSK`-s, etc., will change in a model shift.

The files framed in this way take part in the iteration and the preparation. The files occurring here are created by the user before fitting but their contents will be changed by the programs in each iteration or a model shift.



At points marked with a frame like this, fitting can go in different directions depending on the temporary results and the user's aims.

The scheme requires some further explanation:

- The drawing program can be used anywhere during an iteration, so its usage is detailed only once in *Figure 4.1***a**.
- If you use `TASK.EXE` instead of `*.BAT` programs, the subprocesses rounded with dotted lines should be replaced by *Figure 4.1***b**.

4.3 Some Advice for Studying the Reference Guide

The first chapters of the Reference Guide and *Appendix C* give general information concerning the whole program package. It is very important to read and understand them because information presented in them is used repeatedly in this book. These parts of this book assume that the user has some knowledge of DOS. If you do not understand something, study [2] or other books!

The other chapters contain the detailed description of the programs and data files. Almost all the possibilities are illustrated through some examples. ZITA contains ten examples for this purpose. Nine examples are based on generated 'experimental data' but one of them uses real experiments[1].

In addition, there are five tests detailed in [13]. The simulation program (`ZITA.EXE`) was tested with them.

The necessary ASCII-files for the examples and tests can be generated in the way described in *Chapter 7*. This chapter analyzes the chemical background of the examples.

You should put away the Reference Guide only in when you understand the chemistry of the examples. If you understand how the examples and tests work, you can begin to solve own problems.

Finally, I wish you complete success!

[1] The chemical background of this example and the process of the calculations are detailed in [14]. It is that first reaction the mechanism of which was determined by the help of an earlier version of ZITA.

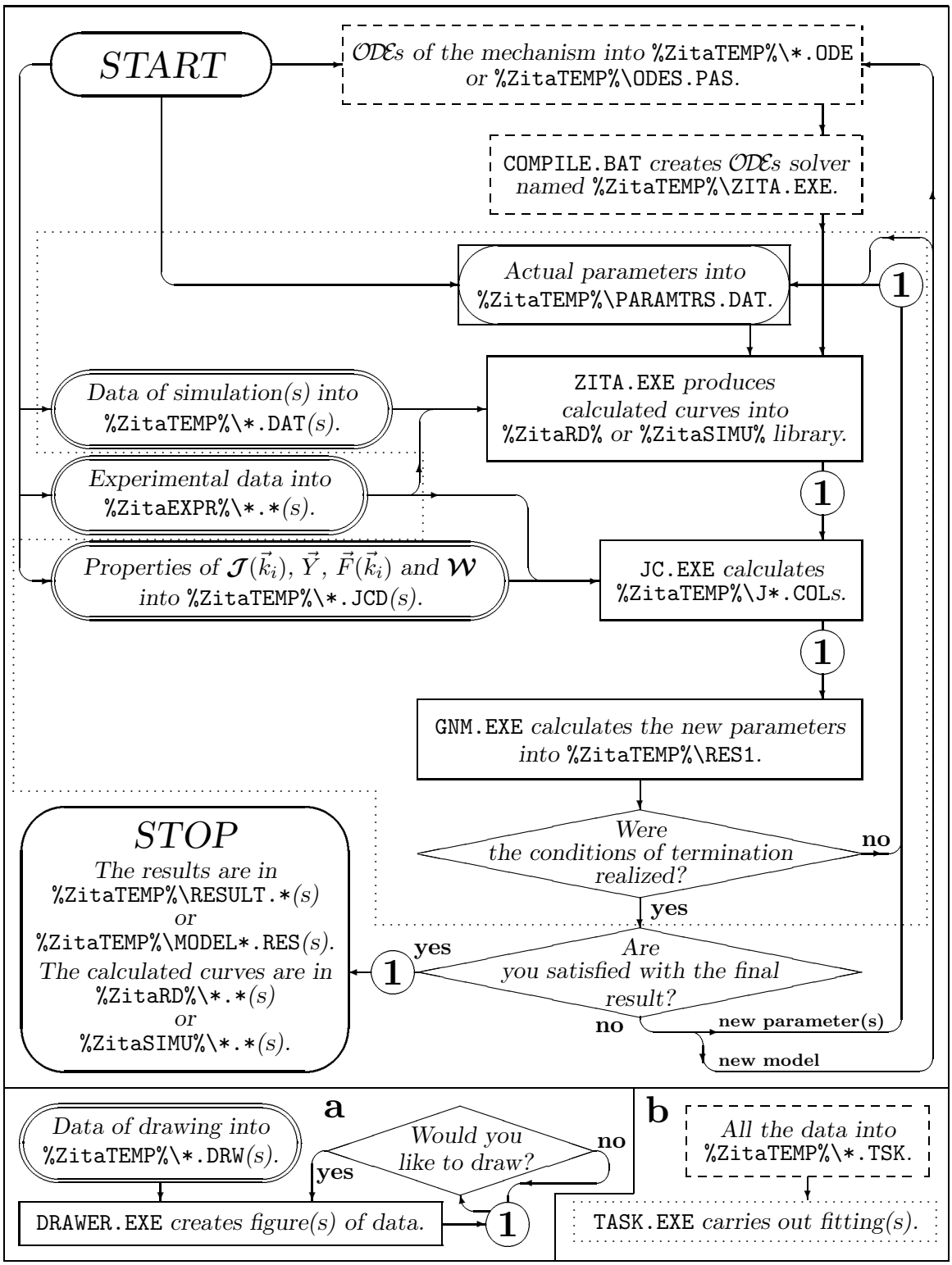


Figure 4.1: The block scheme of fitting (see remarks in text).

Part II

Reference Guide

"A computer program does what you tell it to do, no what you want it to do."

Greer's Third Law [17]

Chapter 5

Structure of ZITA and the Role of Files.

The program package consists of several libraries. This chapter explains the roles of them and the files in these libraries.

One reason for establishing sublibraries is that feature of DOS that the maximum number of the entries is usually limited depending on the type of the drive from 64 to 1024. Therefore the facilities of the ZITA would be restricted unnecessarily if a library of the ZITA were the same as the root directory of a drive. This kind of restriction is not valid for the sublibraries.

Another reason is that requirement that the well-organized files can be used more efficiently.

5.1 Common Library

This library contains all the executable programs (expect for the actual *ODEs*-solver and the compiler) which can be used by every user. The name of this library was chosen during the first installation process. The files in this library are listed in *Table 1.2*. The followings describe the main roles of them:

`GOZITA.EXE` helps to start ZITA. This program is detailed in *Section 1.5*.

`NEWEXE.BAT` creates the executable programs (except for *ODEs*-solver) by the help of the source texts, according to the computer configuration. Its usage is explained in *Section 6.2*.

`COMPILE.BAT` creates *ODEs*-solver named `ZITA.EXE` of a given model in the user's library. This batch program requires the PASCAL source text of the solving algorithm for *ODEs* and the model of the concrete task. The first one (`ZITA.PAS`) can be found in `PROGRAMS` sublibrary connected to the common library. The latter one can be given two ways: its source text can be written into the `ODES.PAS` or it can be described by the matrix-formalism of an `*.ODE` file in the user's library. `COMPILE.BAT` is detailed in *Section 9.2*.

`FINDER.BAT` looks for the place of a given runtime error in the source texts. If a program does not run well, it writes out the memory address of the runtime error and terminates

the run. FINDER.BAT uses this message to find the place of the error in the source text. It is easier to find out the reason for an error if the user is acquainted with the PASCAL programming language. This batch program is described in *Section 6.3*.

MAKEODE.EXE makes those procedures written in PASCAL which contain the source text of *ODEs*, the mathematical relationships between measured characteristics and concentrations and the other parameters of *ODEs*. The input data of this program are stored in a file with the extension ODE. The final result of MAKEODE.EXE is a file named ODES.PAS in the user's library. This is a PASCAL source file and it can also be created or revised manually. MAKEODE.EXE is not used directly but it is called only by COMPILE.BAT. The structure of ODES.PAS and *.ODE files are detailed in *Section 9.1*.

JC.EXE calculates the columns of $\mathcal{J}^T(\vec{k}_i)$ matrix and the $\left(\vec{Y} - \vec{F}(\vec{k}_i)\right)$ vector on the basis of the experimental and calculated data. The elements of the i^{th} column of the matrix are stored in a file named Ji.COL, e.g. the 5^{th} column is in J5.COL file and the elements of the vector are stored in J.COL file. The features of the calculation methods should be given as DOS parameters. The filenames of experimental and calculated curves to be taken into consideration are given in a file with the extension JCD. Its usage is described in *Section 10.1*.

GNM.EXE performs an iteration of fitting based on *.COL files. The further necessary data for an iteration are in PARAMTRS.DAT file. The result of i^{th} iteration is in a file named RESULT.i, e.g. the result of 5^{th} iteration is in RESULT.5 file. RES1 file contains the new parameter vector. If RES1 is renamed to PARAMTRS.DAT, the new parameter vector can be used for the new iteration. This program is detailed in *Section 10.2*.

TASK.EXE executes one or more fittings on the basis of the actual *ODEs* and experimental curves. Its input data are given in a file with the extension TSK. The usage of this program is explained in *Chapter 12*.

DRAWER.EXE is able to draw two dimensional figures on the basis of the experimental and calculated curves. Its input data are given in a file with the extension DRW or DRA. This program is detailed in *Chapter 11*.

EXAMPLES.EXE contains all the necessary files for the tutorials, examples and tests. *Chapter 7* explains the creation of these files by the help of this program.

5.2 PROGRAMS Library Connected to the Common Library

The files of this library are important (as they are working in the background) but the user never calls or edits them directly with the exception of ODES.SPL. The place of this library cannot be changed! The content of this library is given in *Table 1.2*.

TPC.EXE and TURBO.TPL are the parts of TP compiler. They are essential in making *.EXE files. The user does not meet them directly but NEWEXE.BAT, COMPILE.BAT and FINDER.BAT use them.

`ZITA4ALL.PAS` includes the PASCAL source text of error handling (detailed in *Section 6.3*). It is necessary for making up all the executable programs.

`ZITA.PAS` contains the source text of the *ODEs*-solver and its organizer program. The role of this file is evident in making up the executable simulation program.

`MAKEODE.PAS`, `JC.PAS`, `GNM.PAS` and `TASK.PAS` are the source texts of the executable programs. They are used by `NEWEXE.BAT` and `FINDER.BAT`.

`ODES.SPL` is a frame file for the user's *ODEs*. Its usage is explained in *Subsection 9.1.1*.

5.3 User's Library

The libraries explained in the previous sections can be used by any user. Besides, every user has an independent library for own work. It is called user's library. It contains the user's files and several necessary files created by the installation program. The latter ones are presented in *Table 1.2*.

`ZITA4#1.BAT`, `CONFZITA.4#1` and `AUTOZITA.4#1` helps to start ZITA and to set up the computer configuration desired by the user. These are detailed in *Section 1.5*.

`NEWEXE.BAT`, `COMPILE.BAT`, `FINDER.BAT`, `TASK.BAT`, `DRAWER.BAT` and `EXAMPLE.BAT` files make connection with the common library. The user can use the programs of common and `PROGRAMS` library through these batch files.

Two further files are in this library in every case which have preserved filename. These are `ODES.PAS` and `ZITA.EXE`. Since their contents depend on the concrete task, the user creates them but their names cannot be changed!

`ODES.PAS` contains the PASCAL source text of the *ODEs* of the actual task. *Section 9.1* details how the user can create it.

`ZITA.EXE` is the executable program of a given *ODEs* will be created by `COMPILE.BAT` DOS command file. The features of numerical integration are in a file with the extension `DAT`. It cannot be called `PARAMTRS.DAT` because this filename is preserved for the file containing the actual values of the parameters. Every calculated curve is stored in separate files either in `SIMU` library or on RAM disk. The syntax of `*.DAT` files will be detailed in *Section 9.3*.

5.4 EXPR Library

The files containing experimental curves should be stored here. It is compulsory by all means! There is no limitation in the filenames to be given.

The name of this library is stored in DOS environment variable called `ZitaEXPR` (see in *Section 6.1*). The location of this library can be changed by the help of the value of this variable. Initially the library called `EXPR` and connected to the user's library is supposed to be for this aim.

5.5 SIMU Library

If a calculated curve is intended to be kept after switching off the computer, the necessary files should be set in this library. It can be possible by the *ODEs* solver, too. If somebody works only with a RAM disk, every important file has to be saved from the memory disk before switching off the computer.

The name of this library is stored in DOS environment variable called `ZitaSIMU` (see in *Section 6.1*). The location of this library can be changed by the help of the value of this variable. Initially the library called `SIMU` and connected to the user's library is supposed to be for this aim.

There is another way to store the calculated curves. It is detailed in *Section 6.1* when `ZitaRD` DOS environment variable is discussed.

Chapter 6

General Features of the Program Package

This chapter contains some information concerning each program except the installation program.

6.1 How Does ZITA Use DOS Environment Variables?

ZITA stores the most important information in DOS environment variables [2]. The variable name between two percent signs means the value of the variable in question. ZITA gives an initial value for these variables. They can be changed by 'SET' DOS command directly from the DOS cursor[1]. If the user wants to make a change to be permanent, the appropriate 'SET' command has to be written into AUTOZITA.4#1 and/or ZITA4#1.BAT files (see this process in *Section 1.5*). Some of these environment variables have reserved content, so these values must not be changed! If this is so, it is indicated at the description of the variable in question.

ZITA also makes possible the change of the libraries by revising the value of their DOS environment variables. It is worth taking advantage of this possibility for several reasons. Partly because the user's RAM disk sometimes turns out to be too small to store the whole program package but some parts of it can be stored on the RAM disk, e.g. the EXPR library can be stored there in order to find the experimental files in a shorter period of time. On the other hand, this way the separation of the files of several users can be achieved.

There is a very important fact concerning that environment variables which contain the name of a library: these names cannot contain a closing backslash!

ZITA uses the following DOS environment variables:

%ZitaBASE% is the DOS name of the actual common library detailed in *Section 5.1*. It is chosen during the installation process and cannot be changed later!

%ZitaPROG% is the DOS name of PROGRAMS library connected to the common library. In other words, %ZitaPROG% equals to %ZitaBASE%\PROGRAMS. It cannot be changed after the installation!

[1]The user has to know that the 'SET' DOS command cannot be used correctly from a utility, e.g. Norton Commander, PC Shell, etc.! This command works fine only from the DOS cursor.

Type	*Range*	*Significant digits*	*Size in bytes*	*Possible values of* %ZitaREAL%
real	$\pm(2.9 \cdot 10^{-0039} - 1.7 \cdot 10^{+0038})$	11 – 12	6	`r`, `real`, `R`, `REAL`
single	$\pm(1.5 \cdot 10^{-0045} - 3.4 \cdot 10^{+0038})$	7 – 8	4	`s`, `single`, `S`, `SINGLE`
double	$\pm(5.0 \cdot 10^{-0324} - 1.7 \cdot 10^{+0308})$	15 – 16	8	`d`, `double`, `D`, `DOUBLE`
extended	$\pm(3.4 \cdot 10^{-4932} - 1.1 \cdot 10^{+4932})$	19 – 20	10	`e`, `extended`, `E`, `EXTENDED`

Table 6.1: Real data types handled by TP.

%ZitaTEMP% is the DOS name of the actual user's library. Since more users can use the same common library, the value of this variable depends on the user. The user ought not to change this library name if he wants to use a new library. Better method is to use the installation program to define a new user's library.

%ZitaEXPR% is the DOS name of that library which contains the files storing experimental curves. The initial value of it is '`.\EXPR`'.

%ZitaSIMU% is the DOS name of that library which contains the files storing calculated curves. The initial value of it is '`.\SIMU`'.

%ZitaRD% is almost the same as **%ZitaSIMU%**. The purpose of these two libraries is the same but only **%ZitaSIMU%** can be used in some cases. These exceptions are detailed in the further chapters.

What are the differences between these libraries? The user can choose the library to be used during the calculations. The programs of ZITA suppose that **%ZitaRD%** is on a RAM disk and **%ZitaSIMU%** is on a physically existing disk. They try to store the temporary results in **%ZitaRD%** and the most important results in **%ZitaSIMU%**. If an error occurs during a simulation, e.g. the RAM disk becomes full, the content of **%ZitaRD%** is moved into **%ZitaSIMU%** before the termination. It does not work inversely.

The initial value of **%ZitaRD%** depends on the installation of the user's library. If the user has declared that he wants to use a RAM disk, **%ZitaRD%** contains the drive letter of the RAM disk to be used and a colon. Otherwise, **%ZitaRD%** equals to **%ZitaSIMU%**. It also means that **%ZitaRD%** can indicate a directory on a hard disk or on a floppy diskette though it is supposed to be on a RAM disk in this manual.

%ZitaREAL% is used by `NEWEXE.BAT`, `COMPILE.BAT` and `FINDER.BAT`. This variable determines which kind of real data types is used by the programs to be created. The possible real data types are listed in *Table 6.1*. This table also contains all the possible values of this environment variable and their meaning. If this variable has no value, it is considered to be '`EXTENDED`'.

%ZitaDISPLAY% determines the screen type and the graphic mode for `DRAWER.EXE`. They had to be chosen when ZITA was installed. This choice can be modified through this

%ZitaDISPLAY%	*Type of screen*	*Resolution*
1 0	CGA	320 × 200
1 1	CGA	320 × 200
1 2	CGA	320 × 200
1 3	CGA	320 × 200
1 4	CGA	640 × 200
2 0	MCGA	320 × 200
2 1	MCGA	320 × 200
2 2	MCGA	320 × 200
2 3	MCGA	320 × 200
2 4	MCGA	640 × 200
2 5	MCGA	640 × 480
3 0	EGA	640 × 200
3 1	EGA	640 × 350
4 0	EGA64	640 × 200
4 1	EGA64	640 × 350
5 3	EGAMono	640 × 350
7 0	HercMono	720 × 348
8 0	ATT400	320 × 200
8 1	ATT400	320 × 200
8 2	ATT400	320 × 200
8 3	ATT400	320 × 200
8 4	ATT400	640 × 200
8 5	ATT400	640 × 400
9 0	VGA	640 × 200
9 1	VGA	640 × 350
9 2	VGA	640 × 480
10 1	PC3270	720 × 350
16 0	VESA16	800 × 600

Table 6.2: Screen types and graphic modes.

Color	*Color constant*
black	0
blue	1
green	2
cyan	3
red	4
magenta	5
brown	6
light gray	7
dark gray	8
light blue	9
light green	10
light cyan	11
light red	12
light magenta	13
yellow	14
white	15

Table 6.3: Color numbers for environment variables of `DRAWER.EXE`.

variable so ZIA does not need to be installed again if another type of graphic card is used. This environment variable contains two integer numbers separated by one or more space(s). The first one determines the screen type while the second one the graphic mode. *Table 6.2* presents all the possibilities. The actual values can be changed by modifying '`SET ZitaDISPLAY= ...`' command in `AUTOZITA.4#1` and/or `ZITA4#1.BAT`.

The values of `ZitaDISPLAY` can also be '`no`' or '`NO`'. If it is so, `DRAWER.EXE` determines the screen type automatically and chooses the graphic mode with the highest resolution. If this variable has no value, its value is considered to be '`NO`'.

The VESA16 card is an exception from the general rule. Although, `DRAWER.EXE` supports the standard 16-color VESA high resolution 800x600 mode, but without

autodetection. This program is not able to recognize the existing VESA card, so, this card can only be used after the explicit 'Set ZitaDISPLAY=16 0' DOS command.

%ZitaCOLOR% contains an integer number which determines the drawing color. All the possible colors and their numbers are listed in *Table 6.3*. If this variable has no value at all or has a wrong value, it considered to be white color.

%ZitaBKCOLOR% has the same meaning as %ZitaCOLOR% but it concerns the background color and its default value is black.

%ZitaDISPLAYXMAX% contains an integer number which must be less than the actual resolution of X-axis. If this environmental variable has a valid value, it will be the maximal number of the points along the X-axis. In this way, the X-resolution can be adjusted through the smallest steps. This possibility can be useful if the user wants to coordinate the resolution of the graphics screen and the printer.

%ZitaDISPLAYYMAX% works similarly as %ZitaDISPLAYXMAX%, but it regards the Y-axis.

6.2 Usage of Real Numbers

The TP is able to handle four types of real numbers. These types are given in *Table 6.1*.

The real type is handled by the software floating-point library and all the other ones are handled by a math coprocessor. If the user's computer does not have a math coprocessor, it is emulated during compilation thus all the types of real numbers can be treated by any computer. Only running times can be different which are much longer without a math coprocessor. Each program recognizes whether there is a math coprocessor or not.

There are three *.BAT programs in ZitA which carry out compilation as well, so the required type of real numbers should be given to them. COMPILE.BAT is detailed in *Section 9.2* while FINDER.BAT in *Section 6.3*. The following paragraphs are going to describe NEWEXE.BAT.

NEWEXE.BAT prepares MAKEODE.EXE, JC.EXE, GNM.EXE and TASK.EXE programs. Two DOS parameters should be given for the recompilation. These DOS parameters must not be combined small and capital letters, otherwise, NEWEXE.BAT terminates with an error message. The first DOS parameter is the filename without the extension of the executable program to be made. If it does not correspond to any of the four names mentioned above, NEWEXE.BAT terminates with an error message.

The real type used by the executable program is defined by the second DOS parameter, with the exception of TASK.EXE. The values of it are listed in the the last column of *Table 6.1*. NEWEXE.BAT writes the value of the second parameter into ZitaREAL environment variable and the compiler reads it before the compilation. There are the following possibilities:

- If the first DOS parameter is 'MAKEODE' , the second one is not essential, it is omitted. Consequently, MAKEODE.EXE runs in the same way on any computer so it should be recompiled only if there is an error in the original program.

- If the first DOS parameter is 'JC' or 'GNM' then NEWEXE.BAT controls the second DOS parameter. If it does not correspond to one of the strings listed in the last column of *Table 6.1*, NEWEXE.BAT terminates with an error message. If the second DOS parameter is not indicated, it is supposed to be 'EXTENDED' by NEWEXE.BAT. For example: the 'newexe jc single' or the 'newexe jc s' commands make %ZitaTEMP\JC.EXE executable program which uses single real numbers.
- If the first DOS parameter is 'TASK', the second DOS parameter has another meaning. TASK.EXE uses dynamic variables in the heap [1]. The second DOS parameter is the size of the heap in bytes. This value must be in the range 0 to 655360. The default value for it is 32768. If the user increases this value, more models can be given for TASK.EXE but the called programs can cause a memory overflow more easily.

If INSTALL.EXE has not created the executable programs during the installation, the user has to create them before any other work!

6.3 Error Handling

It is not satisfactory only to send out an error message onto the screen as fitting can be a really time-consuming process and some other information are indicated there, too. That is why a special error handling system is added to ZITA. ZITA4ALL.PAS contains the source text of this system and five executable programs (MAKEODE.EXE, ZITA.EXE, JC.EXE, GNM.EXE and DRAWER.EXE) work with it in the same way.

If an error occurs during running then the actual program sends an error message onto the screen and this message will be added into the file named %ZitaTEMP%\ERRORS. If this file has not existed yet, the actual program creates it[2]. The error message gives information about the exact date and time of the computer, the ordinal number of the error, the error itself and the name of the program where the error occurred. At last the actual program is terminated.

TASK.EXE handles own errors in another ways. It writes the error messages only onto the screen, and they are not stored in %ZitaTEMP%\ERRORS file. This is because TASK.EXE does not carry out calculations but calls all the other executable programs. In this way, the called program writes own error message into %ZitaTEMP%\ERRORS file, if TASK.EXE is working and an error occurs. That is why, *Appendix B* does not contain the error messages of TASK.EXE.

The error messages belong to two groups. The first one contains the runtime errors handled by TP. In this case, the error message stored in ERRORS file includes the segment and offset memory addresses of the error, too. They are two hexadecimal numbers divided by a colon. The following two lines show an example:

[2]If DRAWER.EXE is used independently from the program package (namely, the environmental variables called ZitaTEMP and ZitaBASE must not exist), the ERRORS file will not be created or the existing one will not include the error messages.

```
Date: 23.07.94. Time: 15h 53m 57s Runtime error 002 at 0000:07F6 in JC:
 File not found.
```

Those error messages belong to the second group which are handled by the programs themselves. The following two lines present an example:

```
Date: 23.07.90. Time: 15h 53m 10s Handled error 96 in MAKEODE. Message:
 Number of species <= 0!
```

All the possible error messages and their exit codes are listed in *Appendix B*. These codes will gain importance in the future because they can be treated in DOS command files (see in detail later).

The running programs can be terminated by pressing the **CTRL** and **BREAK** keys together. It is also treated as an error by the programs and this termination has the largest DOS exit code (255). This exit code and message is an exception because it is not written into `%ZitaTEMP%\ERRORS` file. However, the user can also use this exit code in DOS command files. Example 6 (see later) illustrates how this exit code can be used.

If a runtime error occurs, the exit code coincides with the number given in *Appendix B*. It is always useful to know when an error occurred because it can help the user to find out the reason for the error. If a handled error occurs, this code will be 13 independently of reason of the error. `ERRORS` file is sometimes recommended to be erased as it piles up only redundant information about the errors occurred earlier.

If the user understands the PASCAL programming language, the place of the errors can be found in the given source text. `FINDER.BAT` program gives help in finding this place.

This program works with three DOS parameters in which small and capital letters must not be combined. The first DOS parameter is the name of that program in which the actual error occurred. Its values can be the followings: '`MAKEODE`', '`ZITA`', '`JC`', '`GNM`', and '`TASK`'. The second DOS parameter is the memory address of the error. This address can be found in `ERRORS` file. The third DOS parameter corresponds to the second DOS parameter of the `NEWEXE.BAT`. It is detailed in *Section 6.2*. The last DOS parameter is redundant if the error occurred in `MAKEODE.EXE` or `TASK.EXE`. In the other cases that type of the handled real number should be given by the help of which the executable program was created otherwise `FINDER.BAT` cannot find the statement where the error occurred. For example, '`finder zita 0000:7E3A extended`', or '`finder zita 0000:7E3A e`' or '`finder zita 0000:7E3A`' commands explore the statement in the source text which coincides with 0000:7E3A memory address of the executable program.

It should be mentioned that `FINDER.BAT` sometimes makes a mistake because the exact memory address corresponding to the statement to be searched for is less by one than the memory address stored in `ERRORS` file. For example, if the displayed line cannot correspond to the error after '`finder jc 0000:321F`' command, it is suggested to try '`finder jc 0000:321E`' command as well. If the latter command displays a different line then the error is in the line displayed for the second time.

6.4 Important Restrictions Concerning the Program Package

FORTRAN users should pay attention to the fact that it is obligatory to indicate zero followed by a period and the fraction of real numbers! 

The letters in the data files can be embodied either by capitals or lower cases which does not influence the running of the programs.

The ZITA uses some filenames for the temporary storing of data. They are invisible for the user, however, these *filenames are not allowed to be used for the user's own purposes*. The following filenames are reserved: `OUT`, `V`, `D` and `*.CNC` either on the RAM disk or in the `%ZitaSIMU%`, `TDRW*.FIG` in the user's library and `TMP*.` either in the user's library or on the RAM disk. There are more reserved filenames (e.g. for storing the final results), but these files are not temporary ones, so the user can avoid the conflicts easily. 

Chapter 7

Tests and Examples

ZTA contains five *tests*, one *base example* and ten *examples* which present the capabilities of ZTA and they help to understand its usage. The tests are marked from A to E and the examples are enumerated from 1 to 10. `EXAMPLES.EXE` contains all the necessary files which can solve them. The names of these files are listed in *Tables 7.1-7.2*. This executable program is in the common library and it can be used through `EXAMPLE.BAT` batch program placed into user's library.

This program requires at least one DOS parameter. It can be a letter (either upper or lower cases) between `A` and `E`, a number between `1` and `10` or `T`. If the DOS parameter is a letter between `A` and `E`, the files of the appropriate test are created in the user's library. If the DOS parameter is a number, the files of the appropriate example are created in the EXPR library and the user's library. If the DOS parameter is `T`, the files of *Chapter 3* are created in the user's library.

`EXAMPLE.BAT` examines the second DOS parameter as well. If it is '`N`' or '`n`', the files to be created contain not only the necessary data but they contain their exact filename and every line contains a leading serial number, too. It is a useful thing, since the Reference Guide often refers the contents of these files through these names and serial numbers. For example, if the user executes '`EXAMPLE C`' command from his user's library, the files of the third test are created. The simulation and drawing can be carried out by the help of these files. However, if '`EXAMPLE C N`' command is executed instead of the previous one, the created files cannot be used by the programs because the files contain own name and serial numbers, too. The latter command is useful when the user studies the Reference Guide and it refers a concrete line of a file. The best method to study these files is to print them with the name and serial numbers and to use them without the name and serial numbers.

`EXAMPLE.BAT` can also be started without any DOS parameter. In this case, the files for a base example are generated. Probably, the base example is the simplest one and its `*.TSK` file (see *Chapter 12*) includes every *keystring* in detail. If the user wants to create files for her/his problem, probably the most suitable way is to generate the files of the base example and to change them.

Test A	*Test B*	*Test C*
%ZitaTEMP%\Testa.ode %ZitaTEMP%\Testa.dat %ZitaTEMP%\Testa.drw	%ZitaTEMP%\Odes.pas %ZitaTEMP%\Testb.dat %ZitaTEMP%\Testb.drw	%ZitaTEMP%\Odes.pas %ZitaTEMP%\Testc.dat %ZitaTEMP%\Testc.drw
Test D	*Test E*	*Base Example*
%ZitaTEMP%\Odes.pas %ZitaTEMP%\Testd.dat %ZitaTEMP%\Testd.drw	%ZitaTEMP%\Odes.pas %ZitaTEMP%\Teste.dat %ZitaTEMP%\Teste.drw	%ZitaEXPR%\Base0001.exp %ZitaTEMP%\Base.ode %ZitaTEMP%\Base.tsk %ZitaTEMP%\Base.tsk
Example 1	*Example 2*	*Example 3*
%ZitaEXPR%\Exlv1.exp %ZitaEXPR%\Exlv2.exp %ZitaTEMP%\Paramtrs.dat %ZitaTEMP%\Exlv.ode %ZitaTEMP%\Exlv.dat %ZitaTEMP%\Exlv1.dat %ZitaTEMP%\Exlv2.dat %ZitaTEMP%\Exlv3.dat %ZitaTEMP%\Exlv.jcd %ZitaTEMP%\Exlv1.jcd %ZitaTEMP%\Exlv2.jcd %ZitaTEMP%\Exlv3.jcd %ZitaTEMP%\Exlv.drw %ZitaTEMP%\Exlv.bat	%ZitaEXPR%\Cstr.exp %ZitaTEMP%\Paramtrs.dat %ZitaTEMP%\Cstr.ode %ZitaTEMP%\Cstr.dat %ZitaTEMP%\Cstr.jcd %ZitaTEMP%\Cstr1.jcd %ZitaTEMP%\Cstr2.jcd %ZitaTEMP%\Cstr3.jcd %ZitaTEMP%\Cstr4.jcd %ZitaTEMP%\Cstr5.jcd %ZitaTEMP%\Cstr6.jcd %ZitaTEMP%\Cstr.drw %ZitaTEMP%\Cstrfigs.dra %ZitaTEMP%\Cstr.bat	%ZitaEXPR%\Lvrel.1 %ZitaEXPR%\Lvrel.2 %ZitaTEMP%\Paramtrs.dat %ZitaTEMP%\Lvrel.ode %ZitaTEMP%\Lvrel.dat %ZitaTEMP%\Lvrel1.dat %ZitaTEMP%\Lvrel2.dat %ZitaTEMP%\Lvrel3.dat %ZitaTEMP%\Lvrel.jcd %ZitaTEMP%\Lvrel1.jcd %ZitaTEMP%\Lvrel2.jcd %ZitaTEMP%\Lvrel3.jcd %ZitaTEMP%\Lvrel.drw %ZitaTEMP%\Lvrel.bat
Example 4	*Example 5*	*Example 6*
%ZitaEXPR%\Lvabs.1 %ZitaEXPR%\Lvabs.2 %ZitaTEMP%\Paramtrs.dat %ZitaTEMP%\Lvabs.ode %ZitaTEMP%\Lvabs.dat %ZitaTEMP%\Lvabs1.dat %ZitaTEMP%\Lvabs2.dat %ZitaTEMP%\Lvabs3.dat %ZitaTEMP%\Lvabs.jcd %ZitaTEMP%\Lvabs1.jcd %ZitaTEMP%\Lvabs2.jcd %ZitaTEMP%\Lvabs3.jcd %ZitaTEMP%\Lvabs.drw %ZitaTEMP%\Lvabs.bat	%ZitaEXPR%\Tem.295 %ZitaEXPR%\Tem.315 %ZitaEXPR%\Tem.335 %ZitaEXPR%\Tem.355 %ZitaTEMP%\Paramtrs.dat %ZitaTEMP%\Tem.ode %ZitaTEMP%\Tem.dat %ZitaTEMP%\Tem1.dat %ZitaTEMP%\Tem2.dat %ZitaTEMP%\Tem4.dat %ZitaTEMP%\Tem.jcd %ZitaTEMP%\Tem1.jcd %ZitaTEMP%\Tem2.jcd %ZitaTEMP%\Tem4.jcd %ZitaTEMP%\Tem.drw %ZitaTEMP%\Tem.bat	%ZitaEXPR%\Exlv1.exp %ZitaEXPR%\Exlv2.exp %ZitaTEMP%\Param1.dat %ZitaTEMP%\Param2.dat %ZitaTEMP%\Param3.dat %ZitaTEMP%\Model.ode %ZitaTEMP%\Model.dat %ZitaTEMP%\Modelk1.dat %ZitaTEMP%\Modelk2.dat %ZitaTEMP%\Modelk3.dat %ZitaTEMP%\Modelk4.dat %ZitaTEMP%\Modelk5.dat %ZitaTEMP%\Model.jcd %ZitaTEMP%\Model.drw %ZitaTEMP%\Model1.bat %ZitaTEMP%\Model2.bat %ZitaTEMP%\Model3.bat

Table 7.1: Files created by `EXAMPLE.BAT`. PART 1/2.

Example 7		
%ZitaEXPR%\Hocl0001.exp	%ZitaEXPR%\Hocl0032.exp	%ZitaEXPR%\Hocl0063.exp
%ZitaEXPR%\Hocl0002.exp	%ZitaEXPR%\Hocl0033.exp	%ZitaEXPR%\Hocl0064.exp
%ZitaEXPR%\Hocl0003.exp	%ZitaEXPR%\Hocl0034.exp	%ZitaEXPR%\Hocl0065.exp
%ZitaEXPR%\Hocl0004.exp	%ZitaEXPR%\Hocl0035.exp	%ZitaEXPR%\Hocl0066.exp
%ZitaEXPR%\Hocl0005.exp	%ZitaEXPR%\Hocl0036.exp	%ZitaEXPR%\Hocl0067.exp
%ZitaEXPR%\Hocl0006.exp	%ZitaEXPR%\Hocl0037.exp	%ZitaEXPR%\Hocl0068.exp
%ZitaEXPR%\Hocl0007.exp	%ZitaEXPR%\Hocl0038.exp	%ZitaEXPR%\Hocl0069.exp
%ZitaEXPR%\Hocl0008.exp	%ZitaEXPR%\Hocl0039.exp	%ZitaEXPR%\Hocl0070.exp
%ZitaEXPR%\Hocl0009.exp	%ZitaEXPR%\Hocl0040.exp	%ZitaEXPR%\Hocl0071.exp
%ZitaEXPR%\Hocl0010.exp	%ZitaEXPR%\Hocl0041.exp	%ZitaEXPR%\Hocl0072.exp
%ZitaEXPR%\Hocl0011.exp	%ZitaEXPR%\Hocl0042.exp	%ZitaEXPR%\Hocl0073.exp
%ZitaEXPR%\Hocl0012.exp	%ZitaEXPR%\Hocl0043.exp	%ZitaEXPR%\Hocl0074.exp
%ZitaEXPR%\Hocl0013.exp	%ZitaEXPR%\Hocl0044.exp	%ZitaEXPR%\Hocl0075.exp
%ZitaEXPR%\Hocl0014.exp	%ZitaEXPR%\Hocl0045.exp	%ZitaEXPR%\Hocl0076.exp
%ZitaEXPR%\Hocl0015.exp	%ZitaEXPR%\Hocl0046.exp	%ZitaEXPR%\Hocl0077.exp
%ZitaEXPR%\Hocl0016.exp	%ZitaEXPR%\Hocl0047.exp	%ZitaEXPR%\Hocl0078.exp
%ZitaEXPR%\Hocl0017.exp	%ZitaEXPR%\Hocl0048.exp	%ZitaEXPR%\Hocl0079.exp
%ZitaEXPR%\Hocl0018.exp	%ZitaEXPR%\Hocl0049.exp	%ZitaEXPR%\Hocl0080.exp
%ZitaEXPR%\Hocl0019.exp	%ZitaEXPR%\Hocl0050.exp	%ZitaEXPR%\Hocl0081.exp
%ZitaEXPR%\Hocl0020.exp	%ZitaEXPR%\Hocl0051.exp	%ZitaEXPR%\Hocl0082.exp
%ZitaEXPR%\Hocl0021.exp	%ZitaEXPR%\Hocl0052.exp	%ZitaEXPR%\Hocl0083.exp
%ZitaEXPR%\Hocl0022.exp	%ZitaEXPR%\Hocl0053.exp	%ZitaEXPR%\Hocl0084.exp
%ZitaEXPR%\Hocl0023.exp	%ZitaEXPR%\Hocl0054.exp	%ZitaEXPR%\Hocl0085.exp
%ZitaEXPR%\Hocl0024.exp	%ZitaEXPR%\Hocl0055.exp	%ZitaEXPR%\Hocl0086.exp
%ZitaEXPR%\Hocl0025.exp	%ZitaEXPR%\Hocl0056.exp	%ZitaEXPR%\Hocl0087.exp
%ZitaEXPR%\Hocl0026.exp	%ZitaEXPR%\Hocl0057.exp	%ZitaTEMP%\Hocl.ode
%ZitaEXPR%\Hocl0027.exp	%ZitaEXPR%\Hocl0058.exp	%ZitaTEMP%\Hocl.drw
%ZitaEXPR%\Hocl0028.exp	%ZitaEXPR%\Hocl0059.exp	%ZitaTEMP%\Hocl.tsk
%ZitaEXPR%\Hocl0029.exp	%ZitaEXPR%\Hocl0060.exp	%ZitaTEMP%\Hocl.bat
%ZitaEXPR%\Hocl0030.exp	%ZitaEXPR%\Hocl0061.exp	
%ZitaEXPR%\Hocl0031.exp	%ZitaEXPR%\Hocl0062.exp	

Example 8		
%ZitaEXPR%\Rcon0001.exp	%ZitaEXPR%\Pot0002.exp	%ZitaTEMP%\Rcon.tsk
%ZitaEXPR%\Rcon0002.exp	%ZitaEXPR%\Abs0001.exp	%ZitaTEMP%\Lcon.tsk
%ZitaEXPR%\Lcon0001.exp	%ZitaEXPR%\Abs0002.exp	%ZitaTEMP%\Pot.tsk
%ZitaEXPR%\Lcon0002.exp	%ZitaTEMP%\Rcon.ode	%ZitaTEMP%\Abs.tsk
%ZitaEXPR%\Pot0001.exp	%ZitaTEMP%\Rcon.drw	

Example 9		*Example 10*
%ZitaEXPR%\Ex90001.exp	%ZitaEXPR%\Ex90005.exp	%ZitaEXPR%\Ex100001.exp
%ZitaEXPR%\Ex90002.exp	%ZitaEXPR%\Ex90006.exp	%ZitaTEMP%\Ex10.ode
%ZitaEXPR%\Ex90003.exp	%ZitaEXPR%\Ex9.ode	%ZitaTEMP%\Ex10.drw
%ZitaEXPR%\Ex90004.exp	%ZitaTEMP%\Ex9.drw	%ZitaTEMP%\Ex10.tsk
	%ZitaTEMP%\Ex9.tsk	

Table 7.2: Files created by `EXAMPLE.BAT`. PART 2/2.

Example	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
time (min)	6	6	5	5	7	9	250	$75^r/27^o$	12	2
disk space (Kbyte)	200	500	200	200	200	250	1800	$1750^r/1000^o$	800	150

Table 7.3: Time and required disk space needed for the examples (r means relative and o means orthogonal fitting).

7.1 Tests

The purpose of the tests is to illustrate the facilities of *ODEs*-solver and `DRAWER.EXE`. The structures of the files belonging to these tests are not detailed independently here but the appropriate parts of the Reference Guide refer and explain them. Five problems belong to this group described in [13, pages 29–32] and the original solutions are in [13, pages 41–47]. *ODEs*-solver of ZITA underwent the test of these tasks. The final solutions and the running circumstances can be found in *Appendix D*. An AT 386-DX/40 MHz IBM compatible computer with 40 Mbyte hard disk was used for every computation which had a math coprocessor.

7.2 Examples

ZITA includes ten examples which show the usage and the faculties of ZITA. As the usage of ZITA can be acquired more easily by the help of these examples, therefore, this description often refers to them as it was mentioned earlier. Some final results are detailed in *Appendix E*. The examples require different running time and disk space during the run. *Table 7.3* shows the estimated requirements of them. These values were measured with the computer described in the previous section.

In order to run these examples the appropriate `ZITA.EXE` file should be created and the computation can be started by the appropriate `*.BAT` or `*.TSK` file.

Two experimental methods came into general use to study the kinetics of chemical reactions in solutions. In the first case transfer of substance does not take place between the reaction system and its environment. In the other case the reaction goes on in a continuous fed and stirred tank reactor (CSTR) and continuous inflow and outflow of substance take place. In the future the first type will be called *batch reaction* while the second one will be called *CSTR reaction*.

Several other reactor types are possible in practice. *Appendix A* details those reactor types which are handled automatically by the program package. The programmable feature of ZITA makes possible the usage of other types as well.

With the exceptions of examples 5, 7, 9 and 10 all the other ones are based on a Lotka-Volterra type of *ODEs*. The model is the following:

$$\mathbf{G} + \mathbf{R} \xrightarrow{k_1} 5\,\mathbf{R} \tag{7.1}$$

$$5\,\mathbf{R} + \mathbf{L} \xrightarrow{k_2} 2\,\mathbf{L} \tag{7.2}$$

$$\mathbf{L} \xrightarrow{k_3} \mathbf{N} \tag{7.3}$$

This model can be demonstrated very well by the classical interpretation of *rabbits and lions*. This example shows the population changes concerning two animals as a function of time under the following simple conditions that the grass, rabbits and lions belong to the same closed system. In *Eqs. 7.1-7.3*, **G**, **R**, **L** and **N** correspond to the populations of the grass, rabbits, lions and dead lions.

Integrating this *ODEs* the populations as a function of time show oscillation under certain conditions. This is one of the reasons why this model was chosen because the integration of such an *ODEs* (the solutions of which are oscillating functions) is rather difficult. The other reason is that this model is quite simple to carry out the fitting within an acceptable period of time in spite of the difficulties of the example.

The solution of *ODEs* given in *Eqs. 7.1-7.3* is an oscillating function only under certain initial conditions. The initial conditions are the concrete values of the populations (**G**, **R**, **L** and **N**) and the 'rate constants' (k_1, k_2 and k_3) in these examples. When deciding on these values the faculty of the graphic representation of the integrated curves was also taken into considerations. This means that the minimum and maximum values of the animal populations should not differ from each other with many orders of magnitude and the periodic time of oscillation should not be too long in comparison with the time of the growth and decline of the populations. It is very difficult to find such initial conditions among chemical mechanisms but they can be discovered in Lotka-Volterra model relatively easily. On the basis of these considerations, a fraction (less than 1.0) was chosen for the initial values of the populations which obviously cannot be the number of an animal population. Later on, these values will be treated as dimensionless quantities with regard to a unit and the units of measurement are not indicated in the figures. The same can be said about time so the parameters are also dimensionless.

7.2.1 Base Example

The base example shows the possibilities of the program package through a very simple model. This is the shortest consecutive reaction mechanism: $\mathbf{A} \xrightarrow{k_1} \mathbf{B} \xrightarrow{k_2} \mathbf{C}$. In this example, $\mathbf{A}_0 = 0.3\,\mathrm{M}$, $\mathbf{B}_0 = \mathbf{C}_0 = 0.0\,\mathrm{M}$. The initial rate constants are $0.1\,\mathrm{M}^{-1}$ and $1.0\,\mathrm{M}^{-1}$, respectively. The experimental file consists of four columns: t, $\mathbf{A}_t$, $\mathbf{B}_t$ and $\mathbf{C}_t$. Each concentration curve is taken into consideration during the fitting process. The parameters to be fitted are k_1, k_2 and $\mathbf{A}_0$. The final fitted values are $k_1 = 0.020048\,\mathrm{M}^{-1}$, $k_2 = 0.050249\,\mathrm{M}^{-1}$ and $\mathbf{A}_0 = 0.29889\,\mathrm{M}$. In the example, the concentration *vs.* time curves are calculated by the analytical formulas, too. In this way, it is possible to compare the results from the simulation and the expression evaluation.

Even though, this example is simple, it shows how the user can apply the advanced techniques of ZITA, like transformation of calculated concentrations, usage of `*.TSK` file and fitting of initial concentration.

7.2.2 Example 1

The first example is analogous to a batch reaction to which two 'experimental' curves were generated. In these cases the 'measured' characteristics are always the populations as a function of time.

- The exact parameters to be fitted later: $k_1 = 0.02$, $k_2 = 2.0$ and $k_3 = 0.2$.
- The initial values of the parameters are 0.03, 3.0 and 0.1 in the same sequence.
- The initial populations are: $\mathbf{G}_0 = 1.0$, $\mathbf{R}_0 = 0.1$, $\mathbf{L}_0 = 0.0001$ and $\mathbf{N}_0 = 0.0$ in the case of the first 'experimental' curve. The initial populations of the second curve are 1.0, 0.01, 0.0002 and 0.0 in the same sequence.

The value of $\mathbf{G(t)}$ is constant during the calculations. In chemistry it is analogous with a reaction carried out in large excess of one of the reactants. The fitting uses only the 'experimental' points of $\mathbf{R(t)}$ and $\mathbf{L(t)}$ even though `EXLV*.EXP` files include all the populations. The curves to be fitted are shown on *Figure E.1.*

7.2.3 Example 2

Only one 'experimental' curve was generated to the second example which shows two significant differences in comparison with the first example. The first difference is that this model is analogous with a CSTR reaction. The second difference is that the primary 'measured' characteristics are not the populations (or the concentrations) but they are the well-known functions of the populations. Two 'measured' characteristics occur in this example ($\mathbf{M}_1$ and $\mathbf{M}_2$) which can be expressed by the populations in the following way:

$$\mathbf{M}_{1,i} = k_4 + \log\left(\frac{\mathbf{R}_i}{\mathbf{L}_i}\right) \tag{7.4}$$

$$\mathbf{M}_{2,i} = k_5 \cdot \mathbf{G}_i + k_6 \cdot \mathbf{N}_i \tag{7.5}$$

where $\mathbf{G}_i$, $\mathbf{R}_i$, $\mathbf{L}_i$ and $\mathbf{N}_i$ are the values of populations in the i^{th} 'experimental' point. The k_4, k_5 and k_6 are parameters to be fitted together with k_1, k_2 and k_3 'rate constants'. At last, $\mathbf{M}_{1,i}$ and $\mathbf{M}_{2,i}$ are the primary 'experimental' data in the i^{th} 'measured' point. $\mathbf{M}_1\mathbf{(t)}$ is analogous with a 'measured' electrode potential where k_4 is equivalent to a standard electrode potential. $\mathbf{M}_2\mathbf{(t)}$ is analogous with an absorption measured photometrically where k_5 and k_6 are equivalents to a molar absorption in chemistry.

- The exact parameters to be fitted are the following: $k_1 = 0.02$, $k_2 = 2.0$, $k_3 = 0.1$, $k_4 = 0.5$, $k_5 = 10.0$ and $k_6 = 40.0$.

- The initial values of the parameters are 0.025, 2.5, 0.05, 0.2, 15.0, 20.0 in the same sequence.
- The initial populations in the reactor are the following: $\mathbf{G}_0 = 0.0$, $\mathbf{R}_0 = 0.0001$, $\mathbf{L}_0 = 0.02$ and $\mathbf{N}_0 = 0.0$. The same values in the entering substance are 0.2, 0.0999, 0.0 and 0.0 in the same sequence.
- The reciprocal of residence time (k_0) equals to 0.005. The 'volume' is constant in the 'reaction'.

In this example (as a result of the continuous inflow and outflow) there is not any population which is independent of time so **G** also depends on time in this example. The fitting takes places on the basis of 'experimental' curves $\mathbf{M_1(t)}$ and $\mathbf{M_2(t)}$. `CSTR.EXP` contains these values.

7.2.4 Examples 3 and 4

The third and the fourth examples are different from the first one only in their 'experimental' data. All the data contain a random generated error in the 3rd and the 4th examples.

In the 3rd example, every point has a relative error (its maximum value can be $\pm 20\,\%$) in regular distribution. These 'experimental' curves presented in `LVREL.?` files.

In the 4th example every point has an absolute error in regular distribution. The maximum value of the error was calculated in the following way: the absolute value of the difference between the maximum and the minimum of the 'experimental' data was calculated and 5 % of this value could be the highest value of the error. `LVABS.?` files contain these points.

7.2.5 Example 5

Example 5 gives an example for the evaluation of 'experimental' curves measured at different temperatures. The model is the following:

$$3\,SO_3^{2-} + IO_3^- \xrightarrow{k_1} 3\,SO_4^{2-} + I^- \tag{7.6}$$

$$IO_3^- + 5\,I^- + 6\,H^+ \xrightarrow{k_2} 3\,I_2 + 3\,H_2O \tag{7.7}$$

$$SO_3^{2-} + I_2 + H_2O \xrightarrow{k_3} SO_4^{2-} + 2\,I^- + 2\,H^+ \tag{7.8}$$

A high value is chosen for k_3, k_2 is supposed to be independent of temperature and k_1 depends on temperature in the following way:

$$k_1 = A_1 \cdot p_1^{(1000.0/T)}, p_1 = e^{-\Delta H_1/R/1000.0} \tag{7.9}$$

where A_1 and p_1 are the parameters to be fitted, T is the value of temperature, R is the Molar Gas Constant and ΔH_1 is the activation entalphy of the reaction given in *Eq. (7.6)*.

Using p_1 instead of ΔH_1 as a fittable parameter is justified because of numerical difficulties. There is one 'measured' characteristic (**M**) the value of which is the following in the i^{th} 'measured' point:

$$\mathbf{M}_i = 0.534 + 4.309 \cdot 10^{-5} \cdot T \cdot \ln([I_2]) \tag{7.10}$$

- The value of k_3 is constant: $1.0 \cdot 10^{+5}\,M^{-1}\,s^{-1}$.
- The exact parameter values to be fitted are the following: $A_1 = 1.3 \cdot 10^{+6}\,M^{-1}\,s^{-1}$, $p_1 = 0.008$ and $k_2 = 80.0\,M^{-1}\,s^{-1}$.
- The initial parameters are $1.0 \cdot 10^{+6}$, 0.01 and 40.0 in the same sequence.
- There are four 'experimental' curves altogether at four degrees of temperature, the values of which are 295.0, 315.0, 335.0 and 355.0 K. `TEM.??5` files contain these data.
- The initial concentrations are the same for all the four curves: $[SO_3^{2-}]_0 = 2.0 \cdot 10^{-5}\,M$, $[IO_3^-]_0 = 1.0 \cdot 10^{-3}\,M$, $[I^-]_0 = 0.0\,M$ and $[I_2]_0 = 1.0 \cdot 10^{-11}\,M$.

7.2.6 Example 6

Example 6 is also based on the Lotka-Volterra model. Two new elements appear here. The first one is that three models are given in `MODEL.ODE` file. The first and the third equations of all the three models correspond to *Eq. (7.1)* and *Eq. (7.3)*, however, the second equation of models 1 and 2 look like this:

$$\mathbf{2\,R} + \mathbf{L} \xrightarrow{k_2} \mathbf{2\,L} \qquad \text{(for model 1)} \tag{7.11}$$

$$\mathbf{9\,R} + \mathbf{L} \xrightarrow{k_2} \mathbf{2\,L} \qquad \text{(for model 2)} \tag{7.12}$$

The second equation of model 3 corresponds to *Eq. (7.2)*. The 'reaction' orders of the populations are the same in every equation of this *ODEs* in all the three models. The only difference is in the stoichiometric coefficients of all the second equations. This example shows how to decide between different models. The other new element demonstrates the automatic and simultaneous fitting in several models if `TASK.EXE` is not used.

The fitting is organized in `MODEL?.BAT` files in such a way that the interruptions of runs would be possible. At restarting, the calculations go on from the point of the previous interruption. It will be detailed in *Section 10.3*.

The 'experimental' data of this example coincide with those of the first example. The initial and exact parameter values can be found in *Table 7.4*.

7.2.7 Example 7

The seventh example shows that reaction the mechanism of which was determined by an earlier version of ZiTA. This is the reaction between HOCl and ClO_2^- in buffered solution.

Model	*Initial parameters*					*Exact parameters*				
	k_1	k_2	k_3	k_4	k_5	k_1	k_2	k_3	k_4	k_5
1.	0.03	3.0	0.0	0.0	0.1	0.01822	3.133	0.0	0.0	0.2674
2.	0.03	0.0	3.0	0.0	0.1	0.02057	0.0	1.67	0.0	0.1703
3.	0.03	0.0	0.0	3.0	0.1	0.02	0.0	0.0	2.0	0.2

Table 7.4: Values of the parameters in Example 6.

The following mechanism was accepted among a lot of possible ones:

$$\mathrm{ClO_2^-} + \mathrm{HOCl} + \mathrm{H^+} \xrightarrow{k_1} \mathrm{Cl_2O_2} + \mathrm{H_2O} \tag{7.13}$$

$$\mathrm{Cl_2O_2} + \mathrm{ClO_2^-} \xrightarrow{k_2} 2\cdot\mathrm{ClO_2} + \mathrm{Cl^-} \tag{7.14}$$

$$2\,\mathrm{HOCl} + \mathrm{ClO_2^-} \xrightarrow{k_3} \mathrm{ClO_3^-} + \mathrm{Cl_2} + \mathrm{H_2O} \tag{7.15}$$

$$\mathrm{ClO_2^-} + \mathrm{Cl_2} \xrightarrow{k_4} \mathrm{Cl_2O_2} + \mathrm{Cl^-} \tag{7.16}$$

$$\mathrm{Cl_2O_2} + \mathrm{H_2O} \xrightarrow{k_5} \mathrm{ClO_3^-} + \mathrm{Cl^-} + 2\,\mathrm{H^+} \tag{7.17}$$

$$\mathrm{Cl_2} + \mathrm{H_2O} \underset{k_7}{\overset{k_6}{\rightleftharpoons}} \mathrm{HOCl} + \mathrm{Cl^-} + \mathrm{H^+} \tag{7.18}$$

The chemical background and the experimental work are detailed in [14][1]. Since the mechanism depends on *p*H as well, three initial concentrations belong to one experimental curve: $[\mathrm{HOCl}]_0$, $[\mathrm{ClO_2^-}]_0$ and $[\mathrm{H^+}]_0$. 87 experimental files contain the initial concentrations in this sequence in their first line. Since the reactions were carried out in buffered solution, $[\mathrm{H^+}]_0 = [\mathrm{H^+}]_t$. The measured characteristic is $[\cdot\mathrm{ClO_2}]_t$. The chlorine-dioxide is a product of this reaction and its yellow color can be registered well by a spectrophotometer. *Figures E.4-E.5* show the initial concentrations, the experimental and calculated curves. The initial and exact values of the parameters are indicated in *Table 7.5*.

The purpose of this example is to show how can a huge data set be handled efficiently. Besides, this example uses the possibilities of `TASK.EXE` program.

`HOCL.BAT` is also generated, if this example is studied. This batch file is inessential because it works only after '`task hocl c`' command (see in *Chapter 12*) and solves the same task which can be solved by the help of `TASK.EXE` and `HOCL.TSK`. However, if the user is interested in the structure of batch files, he can study this very complicated file. It shows how a batch file can be organized in order to fit based on several models automatically.

[1] An improved model has given a better solution since 1992.

[†] This number is taken from [15].

	First task		*Second task*	
	Initial values	*Exact values*	*Initial values*	*Exact values*
k_1	$1 \cdot 10^5$	$1.12 \cdot 10^6$	$1 \cdot 10^6$	$1.09 \cdot 10^6$
k_2	$2 \cdot 10^6$	$5.62 \cdot 10^5$	$5 \cdot 10^5$	$6.08 \cdot 10^5$
k_3	$2 \cdot 10^2$	$2.11 \cdot 10^3$	$2 \cdot 10^3$	$2.25 \cdot 10^3$
k_4	$2 \cdot 10^2$	$4.20 \cdot 10^4$	$4 \cdot 10^4$	$3.77 \cdot 10^4$
k_5	$1 \cdot 10^1$	$1 \cdot 10^1$	$1 \cdot 10^1$	$1 \cdot 10^1$
k_6	$1.11 \cdot 10^{1\dagger}$	$1.11 \cdot 10^1$	$1.11 \cdot 10^{1\dagger}$	$1.11 \cdot 10^1$
k_7	$1.8 \cdot 10^{4\dagger}$	$1.8 \cdot 10^4$	$1.8 \cdot 10^{4\dagger}$	$3.03 \cdot 10^5$

Table 7.5: Values of the parameters in Example 7.

7.2.8 Example 8

This example is also based on the Lotka-Volterra model. `RCON000?.EXP` and `LCON000?.EXP` files have the same initial concentrations as `EXLV?.EXP` but `RCON*.EXP` files contain only the population of rabbits and `LCON*.EXP` contain only the population of lions. `POT0001.EXP` and `ABS0001.EXP` have the same initial conditions as `CSTR.EXP` but the first one contains only the 'measured' data of $\mathbf{M}_{1,t}$ (defined in *Eq. (7.4)*) and the second one contains only the 'measured' data of $\mathbf{M}_{2,t}$ (defined in *Eq. (7.5)*). `POT0002.EXP` and `ABS0002.EXP` also present a CSTR reaction with other initial concentrations.

- The exact parameters to be fitted are the following: $k_1 = 0.02$, $k_2 = 2.0$, $k_3 = 0.2$, $k_4 = 0.5$, $k_5 = 10.0$ and $k_6 = 40.0$.
- The initial values of the parameters are 0.025, 2.5, 0.1, 0.2, 15.0, 20.0 in the same sequence.
- The initial populations in the reactor are the following: $\mathbf{G}_0 = 0.0$, $\mathbf{R}_0 = 0.0001$, $\mathbf{L}_0 = 0.01$ and $\mathbf{N}_0 = 0.0$. The same values in the entering substance are 0.2, 0.0499, 0.0 and 0.0 in the same sequence.
- The reciprocal of residence time (k_0) equals to 0.005.

This example shows how a fitting can be carried out based on more experimental files having different structure and experimental conditions. It also illustrates how `TASK.EXE` can be used with several `*.TSK` files. Moreover, the result presented in *Section E.8* show the difference between relative and orthogonal fitting.

7.2.9 Example 9

This example illustrates a solution of the following problems:

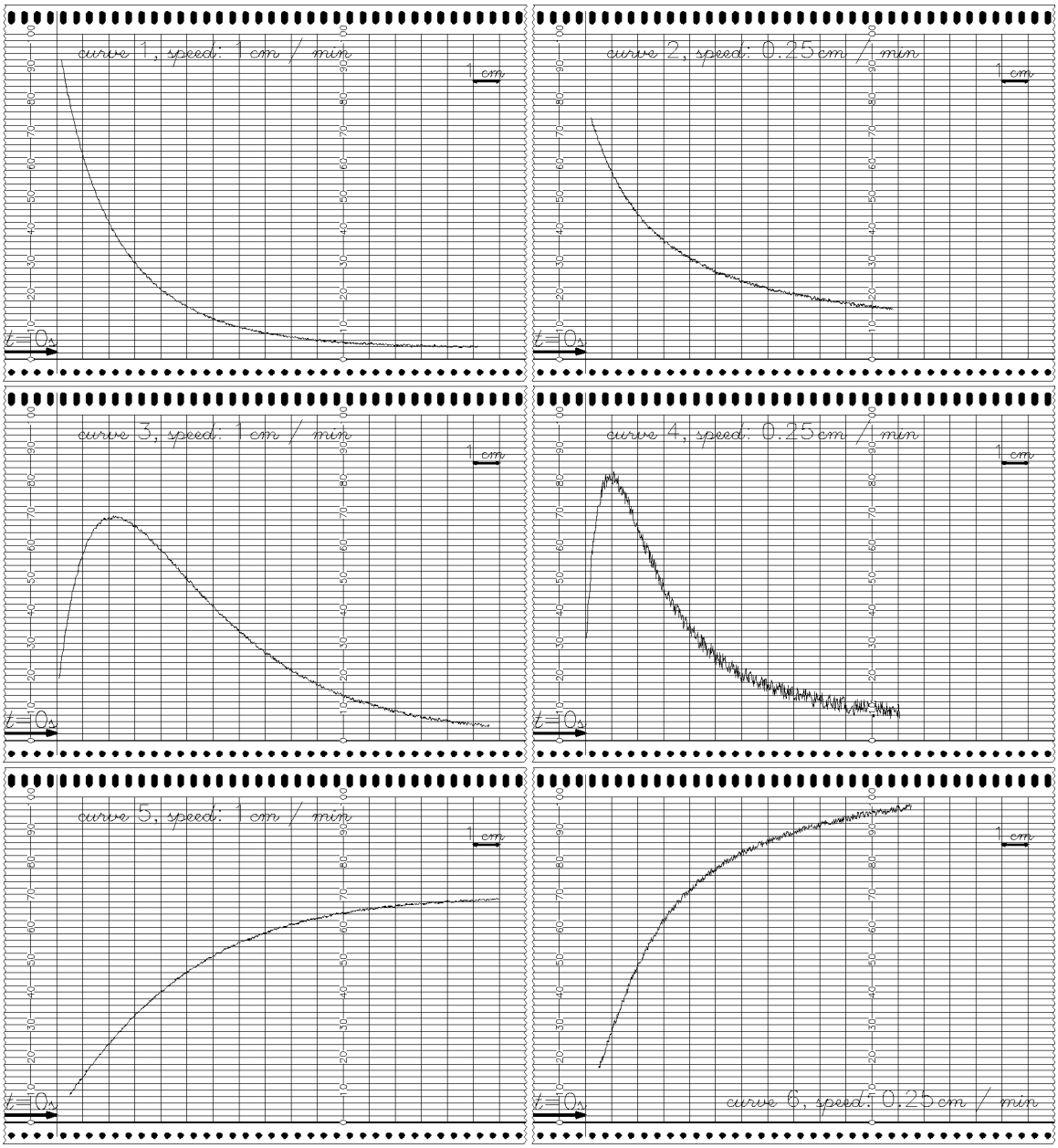


Figure 7.1: The 'experimental curves' on recording charts in Example 9.

- How can those known parameters be given which characterize one or only few experimental curves?
- How can those unknown parameters be fitted which characterize one or only few experimental curves?
- How can the relations be given between the experiments and concentrations if the primary experimental data are not concentrations and they depend on the concentration

	Curve 1	*Curve 2*	*Curve 3*	*Curve 4*	*Curve 5*	*Curve 6*
$[\mathbf{A}]_0$	0.000803	0.0004011	0.000788	0.000413	0.0008012	0.000402
$[\mathbf{B}]_0$	0.002170	0.0003834	0.002010	0.000392	0.0018700	0.000397
$[\mathbf{I}]_0$ and $[\mathbf{P}]_0$	0.0					
range (A/100 div)	0.0–0.8	0.0–0.5	0.0–0.5	0.0–0.1	0.0–0.8	0.0–0.25
baseline (div)	1.5	0.3	0.2	-2.9	-0.4	1.1
wavelength	λ_1	λ_1	λ_2	λ_2	λ_3	λ_3

Table 7.6: Circumstances for Example 9.

parameter	k_1	k_2	$\varepsilon_{\mathbf{A}}^{\lambda_1}$	$\varepsilon_{\mathbf{I}}^{\lambda_1}$	$\varepsilon_{\mathbf{P}}^{\lambda_1}$	$\varepsilon_{\mathbf{A}}^{\lambda_2}$	$\varepsilon_{\mathbf{I}}^{\lambda_2}$	$\varepsilon_{\mathbf{P}}^{\lambda_2}$	$\varepsilon_{\mathbf{A}}^{\lambda_3}$	$\varepsilon_{\mathbf{I}}^{\lambda_3}$	$\varepsilon_{\mathbf{P}}^{\lambda_3}$
initial values	10.0	0.01	1000.0	10.0	10.0	10.0	1000.0	10.0	20.0	1000.0	10.0
exact values	5.0	0.005	1000.0	150.0	20.0	70.0	900.0	10.0	30.0	250.0	700.0

Table 7.7: Parameter values in Example 9.

in a difficult way?

The supposed mechanism for this example is

$$\mathbf{A} + \mathbf{B} \xrightarrow{k_1} \mathbf{I}, \qquad \mathbf{I} \xrightarrow{k_2} \mathbf{P}$$

and the circumstances are the following:

- The reactions are followed by a spectrophotometer at three different wavelengths.
- **A**, **I** and **P** species are supposed to be colorful species and **B** is a colorless species.
- There are six 'experimental curves' created by simulation. These are the values of divisions on the recording charts as a function of time. The next equation gives the relation between the concentrations and measured characteristics:

$$\mathbf{D}_t^i = 100 \cdot (\varepsilon_{\mathbf{A}}^i [\mathbf{A}]_t + \varepsilon_{\mathbf{I}}^i [\mathbf{I}]_t + \varepsilon_{\mathbf{P}}^i [\mathbf{P}]_t) / \mathbf{R}^i + \mathbf{D}_0^i$$

where

i is the serial number of the actual curve ($i = 1 \ldots 6$).

$\varepsilon_{\mathbf{X}}^i$ is the molar absorbance of **X** species at the actual wavelength. The wavelength is λ_1 if $i \in (1, 2)$, λ_2 if $i \in (3, 4)$ and λ_3 if $i \in (5, 6)$.

$\mathbf{R}^i$ is the absorbance range on the recording chart between lines 0 % and 100 % concerning the i^{th} curve.

$\mathbf{D}_0^i$ is the value of the base line in the i^{th} curve.
$\mathbf{D}_t^i$ is the value of the measured characteristic.

The circumstances of the 'experimental curves' are in *Table 7.6*. The places of the base line and the ranges are individual fixed parameters while the rate constants and molar absorbances are fitted parameters. *Table 7.7* details the values of the fitted parameters.

7.2.10 Example 10

This example demonstrates the way of fitting initial concentrations and evaluating the experimental data measured under constant pressure.

The assumed model is the simple $\mathbf{A} \xrightarrow{k} 2\,\mathbf{B}$ gas reaction. The measured characteristic is the volume of the reaction mixture as a function of time. The relation between the volume and the concentrations can be expressed by the *Eq. (7.19)*. There is only one 'experimental curve' calculated from *Eq. (7.20)* (this is the analytical solution of the appropriate *ODEs*).

$$V_t = V_0 \frac{2\,[\mathbf{A}]_0 + [\mathbf{B}]_0}{2\,[\mathbf{A}]_t + [\mathbf{B}]_t} \qquad (7.19) \qquad V_t = V_0 \frac{2\,[\mathbf{A}]_0 + [\mathbf{B}]_0 - [\mathbf{A}]_0 e^{-kt}}{[\mathbf{A}]_0 + [\mathbf{B}]_0} \qquad (7.20)$$

- The initial volume is an individual fixed parameter and its value is 1.0.
- The parameters to be fitted are k, $[\mathbf{A}]_0$ and $[\mathbf{B}]_0$ (the latter two depend on each other). Their initial values are 0.02, 0.02 and 0.02.
- The exact values of the parameters are 0.005, 0.03474 and 0.00613.

Chapter 8

Input of Experimental Data

This chapter is going to discuss the structure of experimental files (see *Section 5.4*). These files consist of two parts. The first one contains the conditions, e.g. the value of temperature, and initial values of concentrations of an experimental curve, etc. One file includes only the data of those experimental curves which have the same initial conditions. . The second part contains the measured data of the experimental curves.

8.1 Initial Conditions for Experimental Curves

The first part can be divided into seven units. This can be described in the following way by the symbol system detailed in *Appendix C*:

Unit 1: [C_1 $I1_{1a}$ $I1_{1b}$ $I1_{2a}$ $I1_{2b}$ ... $I1_{ya}$ $I1_{yb}$ $\{1 \leq y \leq ic\}$|C_1 $I1_b$ *]
Unit 2: C_2 $I2_1$ $R1_1$ $I2_2$ $R1_2$ ... $I2_z$ $R1_z$ $\{1 \leq z \leq ic\}$
Unit 3: C_3
Unit 4: R2 *
Unit 5: [$R3_1$|:$I3_1$] [$R3_2$|:$I3_2$] ... [$R3_n$|:$I3_n$]
Unit 6: [$R4_1$|:$I4_1$] [$R4_2$|:$I4_2$] ... [$R4_m$|:$I4_m$]
Unit 7: [R5|:I5] *
or [R5|:I5] [R6|:I6] *

where n and m are the number of the indicated initial concentrations. Only those concentrations should be given which are different from 0.0, however, the zeroes can also be given. *ic* is the dimension of ip and cp arrays (see below in detail).

Units 1 and **2** are probably the most difficult parts regarding the input experimental data files so they must have a little bit deeper explanation. Many times, it is worth carrying the fitting out on the primary measurements (e.g. volume, absorbance, etc.), so the final result requires well-determined transformations from the simulated concentrations to the calculated analogous of the primary measurements. These transformations can contain such constants that are concerned by *only* some of the experimental curves (e.g. measurements on different wavelengths should imply different molar absorbances of a compound). For these constants, the program package offers the elements of two floating-point arrays with

the names of cp[1]...cp[*ic*] and ip[1]...ip[*ic*]. cp can be used in **Unit 1** while **Unit 2** uses ip. The dimensions of these arrays are the same. It can be defined in *.ODE files or its default value is 50 (see *Subsection 9.1.2* in detail).

With the help of these variables the user is able to do the transformations easily, moreover, the constants defined in **Unit 1** become also fittable together with the parameters of the actual mechanism! The only difference between these two units is that the constants belonging to **Unit 2** can only be fixed ones! Since each parameter to be fitted must be included in PARAMTRS.DAT file, **Unit 1** has a connection to this file while **Unit 2** has not.

To understand the structure of these two units it is essential to study examples 9 and 10 carefully! Solving the problems outlined there needs these units.

Unit 1: C_1 can only be 'C' or 'T'. In the case of 'C', this row instructs the simulation program to assign the value of the $(I1_{yb}+ez+tk)^{th}$ parameter of PARAMTRS.DAT to $cp[I1_{ya}]$ variable before the simulations. The ez is the number of the equations and tk is the number of the temperature dependent rate constants in the actual model. In this way the individual parameters of the experimental curves become fittable. *Subsection 9.1.1* details the meaning and usage of ez, tk and cp[i] ($1 \leq i \leq ic$) variables. *Section 9.3* and *Subsection 10.2.1* detail the structure of the PARAMTRS.DAT file. There are a few restrictions concerning the sequence of parameters in PARAMTRS.DAT file. They are detailed in *Subsection 9.1.2*. If there is not such a parameter, **Unit 1** is to be omitted.

If C_1 is 'T', the value of the $(I1_b+ez+tk)^{th}$ parameter of PARAMTRS.DAT will be the initial value of time for the actual data. In this way, the dead time and/or the mixing time can also be fitted or fixed parameter *independently* for every experimental curve in stopped flow measurements. This setting supersedes the setting of 'EXPERIMENTAL␣CURVES' keystring in *.TSK files (see *Chapter 12*)!

Unit 2: C_2 can only be 'I'. This row instructs the simulation program to assign $R1_z$ to $ip[I2_z]$ variable before the simulations. *Subsection 9.1.1* details the usage of ip[i] ($1 \leq i \leq ic$) variables. The value of the fixed individual parameters can be given in this way. If there is not such a parameter, **Unit 2** is to be omitted.

Unit 3: C_3 can be 'P' or 'S'. If this unit is indicated in an experimental file, the simulation program supposes that the experiments describe a gas reaction carried out under constant pressure. **Unit 3** must be omitted in any other case.

'P' and 'S' have essentially the same meaning. They should be distinguished only for that species, the concentration of which is kept at a constant value (see the description of the vector between the order and formation matrix in *Subsection 9.1.2*). In the case of 'P', the volume correction is not carried out for this kind of species. Its concentration remains the initial one. For instance, it makes possible the modeling of the catalyst in a heterogeneous catalytic reaction. In the case of 'S', the volume correction is done for all species without any exception.

The sequence of the first three units is arbitrary but it is not true for the next units.

Unit 4: R2 is the value of temperature belonging to the given experimental curve. If temperature dependent measurements have not been performed, **Unit 4** is to be omitted.

Unit 5: The initial concentrations in the reaction vessel have to be given in **Unit 5** where $\texttt{R3}_i$ (or $\texttt{:I3}_i$) means the i^{th} indicated concentration. These data can stand in more than one line but nothing can follow $\texttt{R3}_\texttt{n}$. The sequence of $\texttt{R3}_i$-s is not fixed but the concrete sequence has to be known because *ODEs*-solver demands it. This way it is not necessary to change the experimental files in the case of a model shift (detailed later on).

Unit 6: $\texttt{R4}_i$ (or $\texttt{I4}_i$) means the i^{th} indicated concentration of the entering substance. Consequently, **Unit 6** is to be omitted in the case of batch reactions. What was told about **Unit 5** can also be told about **Unit 6**.

If the substance flows into the reactor through several tubes, $\texttt{R4}_i$ is not the i^{th} real concentration but it must be given by the following formula:

$$\texttt{R4}_i = C_i \cdot \frac{v_i}{V} \tag{8.1}$$

where C_i is the real concentration of the i^{th} species, v_i is the flow rate (its dimension is [volume/time]) of the i^{th} entering species and V is the total flow rate of the entering solutions.

Unit 7: This unit has to be given in the case and/of inflow or outflow.

- R5 (or :I5) is the reciprocal of the residence time in the case of CSTR. In this case the inflow rate and the outflow rate must be the same during a reaction. *Appendix A* defines this value exactly.

  This syntax is used for semibatch reactor as well. R5 means k^{in} defined in *Appendix A* in this case.

- R5 and R6 (or :I5 and :I6) must be given if the inflow rate and the outflow rate do not equal to each other. The definition of R5 equals to the k^{in} defined in *Appendix A* and the definition of R6 equals to k^{out} defined in *Appendix A*.

There are some illustrations in the examples.

- The data of EXLV?.EXP, LVREL.? and LVABS.? files were 'measured' in a batch vessel, they are independent of temperature and they do not have individual parameters so the first part of these files contain only **Unit 5**. This unit includes the values of $\mathbf{G}_0$, $\mathbf{R}_0$ and $\mathbf{L}_0$. The value of $\mathbf{N}_0$ is not given because it is 0.0.

- The first part of CSTR.EXP consists of **Units 5**, **6** and **7**. The numbers indicated here mean the $\mathbf{R}_0$ and $\mathbf{L}_0$ in the reactor, $\mathbf{G}_0$ and $\mathbf{R}_0$ in the entering solution and k_0. All the other initial data are 0.0.

- The first part of `TEM.*` files consists of **Units 4** and **5** because the reaction was carried out in a batch vessel, the temperature was changed and there is not any individual parameter. This part contains the value of temperature (in Kelvin), $[SO_3^{2-}]_0$, $[IO_3^-]_0$ and $[I_2]_0$ (in mole/dm^3).

- The first part of the experimental files of example 7 contain only **Unit 5** because the reaction was carried out in a batch vessel, the temperature was not changed and there is not any individual parameter. This part contains $[HOCl]_0$, $[ClO_2^-]_0$ and $[H^+]_0$ (in mole/dm^3).

- The experimental files of example 8 have that structure which can be found in `EXLV?.EXP` or `CSTR.EXP` files.

- Example 9 has the `EX9000?.EXP` files for storing experimental data. The first part of them consists of **Units 1**, **2** and **5**.

  The first line of these files assigns values to `cp[1]`, `cp[2]` and `cp[3]` variables. These variables contain individual parameters to be fitted (see *Subsection 9.1.1* in detail). Since `ez`=2 and `tk`=0, the first and second experimental files assign the values of lines 3–5 of `PARAMTRS.DAT` to these variables. The third and fourth experimental files assign the values of lines 6–8 of `PARAMTRS.DAT` to them. Finally, the fifth and sixth experimental files assign the values of lines 9–11 of `PARAMTRS.DAT` to them.

  Every second line of the experimental files contains the value of the absorbance range and the base line. They are assigned to `ip[1]` and `ip[2]` variables before the simulation. The third lines contain the values of $[\mathbf{A}]_0$ and $[\mathbf{B}]_0$.

- The `EX100001.EXP` file belonging to example 10 contains **Units 3**, **2** and **5**. The first line of this file means that the data were measured under constant pressure. The second line assigns 1.0 to `ip[1]`. The third line shows that $[\mathbf{A}]_0$ and $[\mathbf{B}]_0$ are fitted parameters[1] and their values are in the second and third lines of `PARAMTRS.DAT` file.

At last, it should be emphasized that the first part always contains only concentration independently from the measured characteristics.

8.2 Points on Measured Curves

The second part of the experimental files contains the points of measured curves. One line consists of points measured at the same time. At first, the experimental time is given then the values of measured characteristics or concentrations follow it in each line. The sequence of the measured characteristics must be the same in each line but the sequence is determined by the user.

If there are several characteristics measured at the same initial conditions but these characteristics were measured at different points of time, the values of different characteristics

[1] Naturally, these two parameters depend on each other because $[\mathbf{A}]_0 + [\mathbf{B}]_0 =$ constant, therefore, only one of them can be fitted. *Subsection 9.1.2* explains how the user can solve this problem.

can be written into the same file. In this case the first column must contain all those time points in ascending sequence at which one or more characteristics were measured. Since all the columns with the exception of the first one can have missing values, which means that the user must fill these places with letters 'M'.

There are some illustrations in the examples.

- The 'experimental' curves belonging to the 1st, the 3rd, the 4th and the 6th examples contain the values of time, $\mathbf{G(t)}$, $\mathbf{R(t)}$, $\mathbf{L(t)}$ and $\mathbf{N(t)}$ (defined in *Eqs. 7.1-7.3*) in one line.
- Each line of `CSTR.EXP` gives three numbers which are the time, $\mathbf{M_1(t)}$ and $\mathbf{M_2(t)}$ (defined in *Eqs. 7.4-7.5*).
- `TEM.*` files belonging to the 5th example consist of two columns which are the values of time and $\mathbf{M(t)}$ (defined in *Eq. (7.10)*).
- Every line of `HOCL*.EXP` files in the 7th example consists of the values of time and $[\cdot ClO_2]_0$.
- The experimental files of example 8 have different structure. Every file consists of two columns. The first one contains the values of the time in each case. The second one contains $\mathbf{R(t)}$ in `RCON*.EXP`, $\mathbf{L(t)}$ in `LCON*.EXP`, $\mathbf{M_1(t)}$ in `POT*.EXP` and $\mathbf{M_2(t)}$ in `ABS*.EXP` files.
- The experimental files of example 9 contain the time points and the divisions on the recording charts.
- The experimental file of example 10 contains the values of time and volume.

Chapter 9

Numerical Integration

This chapter gives a detailed description of the simulation. The *ODEs*-solver itself is a completely independent program so this part of ZeTA can also be used even though the user does not intend to fit only to simulate.

9.1 Input of the *ODEs* and Relationships between Concentrations and Measured Data

The *ODEs* can be given in two ways. The user either can write a PASCAL source text manually or can work with the matrix-formalism detailed in *Subsection 9.1.2*. In the second case the source text is generated by `MAKEODE.EXE` program. In the beginning the first way is described. The basic notions, variables and symbols will be introduced here as well. These are also used by the matrix-formalism.

The *ODEs*-solver always calculates concentrations so the dependent variables of the *ODEs* must be concentrations or its analogous quantities in the case of non-chemical problems. If the data to be calculated are not concentrations, their mathematical relationships must be given in the same file that contains the source text or the matrix-formalism of the actual *ODEs*.

The reference of TP can be found in [1]. This book describes the syntax of the Turbo Pascal programming language.

The following theoretical mechanism illustrates the possibilities in this section:

$$
\begin{array}{lll}
Y_1 \xrightarrow{k_1} 2\,Y_3, & r_1 = k_1 \cdot Y_1, & k_1 = p_1 \cdot p_5^{1000/T} \\
Y_1 + Y_3 \xrightarrow{k_2} Y_2, & r_2 = k_2 \cdot Y_1^2 \cdot Y_3, & k_2 = p_2 \\
Y_2 \xrightarrow{k_3} Y_1 + Y_3, & r_3 = k_3 \cdot Y_2, & k_3 = p_3 \cdot p_6^{1000/T} = p_1 \cdot (2 \cdot p_5)^{1000/T} \\
Y_2 + Y_3 \xrightarrow{k_4} 2 \cdot Y_1, & r_4 = k_4 \cdot Y_2 \cdot \sqrt{Y_3}, & k_4 = p_4 = 150.0 \cdot p_2
\end{array}
\tag{9.1}
$$

$$
\begin{aligned}
M_1 &= p_7 - \log Y_1 \\
M_2 &= p_8 \cdot Y_2
\end{aligned}
$$

where Y_i is the i^{th} species, k_i is the rate constant of the i^{th} step, r_i is the rate of the i^{th} step, p_i is the i^{th} parameter, T is the temperature in Kelvin and M_i is the i^{th} characteristic to be calculated. Besides them, it is supposed that the reaction is carried out in CSTR and Y_1 flows into reactor.

This theoretical mechanism is intended to illustrate almost all the possibilities. For studying this chapter for the first time, it is advised to print out `*.ODE` and `ODES.PAS` files of the tests and examples. This chapter can be understood more easily by the help of them.

9.1.1 Direct Input of the PASCAL Source Text

If the user writes the source text of *ODEs* manually, `%ZitaTEMP%\ODES.PAS` file has to contain it. This file is a separate unit of the source text for the *ODEs*-solver. Besides, the type and variable declarations and the compiler directives can be found in `ODES.PAS`. They are independent of the actual *ODEs*. ZITA includes the permanent parts of `ODES.PAS` under the name of `%ZitaPROG%\ODES.SPL`[1]. At first, `%ZitaPROG%\ODES.SPL` must be copied into `%ZitaTEMP%\ODES.PAS` if the source text is written manually. The features of the actual *ODEs* must be declared in lines 13–31 by replacing the question marks with the actual values.

Setting up the correct values of the next variables is extremely important! These values is used for reserving memory spaces for arrays and vectors. The simulation program does not check the actual indices of these arrays and vectors during the calculation because it would take a lot of time. Therefore, the user can overwrite some reserved spaces if somewhat of these variables are incorrect. Usually this can be the reason if the user notices an extremely strange error during the calculations.

- The value of `n` is the number of species defined in *Eq. (2.1)*. It equals to 3 in the theoretical mechanism.
- The value of `ez` is the number of equations of the chemical model. It equals to 4 in the theoretical mechanism.
- The value of `p` is the number of parameters. It is the same as `ez` if the parameters are only rate constants, otherwise, `p` is greater than `ez`. This value equals to 8 in the theoretical mechanism.
- The value of `tk` is the number of temperature dependent rate constants. This value equals to 2 in the theoretical mechanism.
- The value of `cs` determines the type of the reactor(s) used during the experimental works.
    - This value equals to 0 in the case of a batch reactor with constant volume.
    - In the case of a CSTR this value equals to 1.
    - In the case of a semibatch reactor [18, 19] this value equals to 2.

[1] The user should print out this file with line numbers now.

- If the inflow rate does not equal the outflow rate but they do not depend on time, `cs` equals to 3.
- In the case of a batch reactor *under constant pressure* this value equals to 4.

The value of `cs` determines the structure of the experimental files as well. Two difficulties may come up:

- The user wants to fit such experimental files which have different structures. In this case the more complex structure must be declared. For example, there are four batch experiments and four CSTR experiments in example 8 so `cs`=1, however, the concentrations in the entering substance and the values of the reciprocal of the residence time are zero in the batch experiments.
- The used reactor type is not listed above. In this case the user must write the own *ODEs* manually and the experimental files can give the approximate values for the simulation program through `cp[i]` and `ip[i]` ($1 \leq$ `i` ≤ 50) variables. This method is detailed later in this subsection.

The values of `cs` equals to 1 in the theoretical mechanism.

- If the experimental data are concentrations or its analogous quantities, `trn` equals to 0, otherwise, it is 1. If `trn` is 1, `cn` is the number of the measured characteristics. In the theoretical mechanism `trn` and `cn` equal to 1 and 2, respectively.
- If temperature dependent data are to be simulated, `tc` equals to '`T`' otherwise it is '`N`'. It equals to '`T`' in the theoretical mechanism.
- The `tmax` means the maximum number of the experimental points in one curve. The value of `tmax` is not important if the calculated data of the simulation must be stored at equal intervals. In this case its value must be chosen to be 1. This value equals to 20 in the theoretical mechanism. If the user wants to use more time points than the value of `tmax`, the simulation program cuts the overloaded ones.
- The user can define mathematical relations between the parameters. The value of `rels` is the number of such relations. This variable must be zero if there is no any relation. The first element of `rks` is always zero. The further elements of this vector are the serial numbers of those parameters which depend on the other ones. These elements are separated by commas and rounded by parentheses. `rels` equals to 3 in the theoretical mechanism. These parameters are p_3, p_4 and p_6 so `rks` equals to `(0,3,4,6)` in the theoretical mechanism.
- If `ODES.PAS` is written manually then `ap` is not intended for the user's own purpose so line 31 must not be changed.

Lines 63–64, 70–71, 77–81, 87–88 and 98–113 of `ODES.PAS` created from `ODES.SPL` should be replaced by the user's own model. It is evident from the source text that the user has to write the statement part of five PASCAL procedures. Their detailed syntax is to be discussed in the following paragraphs.

```
      Column
Line  5   10   15   20   25   30   35
 1:  kt[1]:= ks[5]*Exp(tpert*Ln(ks[1]));
 2:  kt[3]:= ks[6]*Exp(tpert*Ln(ks[3]))
```

Text 9.1: Contents of **Temperature** procedure of the theoretical mechanism.

```
      Column
Line  5   10   15
 1:  ks[3]:=ks[1];
 2:  ks[4]:=150.0*ks[2];
 3:  ks[6]:=2.0*ks[5]
```

Text 9.2: Contents of **Krelations** procedure of the theoretical mechanism.

```
      Column
Line  5   10   15   20   25   30   35   40   45   50
 1:  Var lny3,dummy:Real;
 2:  Begin
 3:    lny3:=Ln(y[3]);
 4:    v[1]:= kt[1]*y[1];
 5:    v[2]:= ks[2]*y[1]*y[1]*y[3];
 6:    v[3]:= kt[3]*y[2];
 7:    v[4]:= ks[4]*y[2]*Exp(0.5*lny3);
 8:    d[1]:= -v[1]-v[2]+v[3]+2*v[4]+k0*(cstr[1]-y[1]);
 9:    d[2]:= +v[2]-v[3]-v[4]-k0*y[2];
10:    d[3]:= +2*v[1]-v[2]+v[3]-v[4]-k0*y[3]
```

Text 9.3: Contents of **Diffun** procedure of the theoretical mechanism.

Temperature procedure are carried out at the same temperature. If it is not so, the method of calculating the rate constants from the parameters is to be given in this procedure. The following variables can be used: `ks[i]` is the value of the $\mathtt{i}^{\text{th}}$ parameter in `PARAMTRS.DAT` file, `temp` is the temperature value in Kelvin and `tpert` equals to 1000.0/`temp`. The assignment statements should be written in such a form that finishing this procedure the value of the $\mathtt{j}^{\text{th}}$ temperature dependent rate constant must be stored in `kt[j]` variable. This procedure is called before integrating the new curves. *Text 9.1* presents this procedure in the theoretical mechanism. It can be seen that p_1 and p_3 are temperature dependent rate constants and the relation between the rate constants and temperature can be described by the Arrhenius-equation defined in *Eq. (7.9)*.

Krelations procedure is empty if there is not any mathematical relation between the parameters. If it is not so, these relations must be written into this procedure. Only `ks[i]`, $\mathtt{i} = 1 \ldots \mathtt{p}$ (detailed in the previous paragraph) can take a new value within this procedure and only the values of `ks[i]` can be used in the right side of an assignment statement! The assignment statements should be written in such a form that finishing this procedure the value of the $\mathtt{j}^{\text{th}}$ parameter must be stored in `ks[j]` variable. This procedure is called only before integrating the first curve. *Text 9.2* presents this procedure in the concrete mechanism. This text contains the relations indicated in *Eq. (9.1)*.

There is something else which is important concerning this procedure. If a parameter is a function of another ones, the value of this parameter stored in `PARAMTRS.DAT` file is not used. However, a value must be assigned to this parameter because the *ODEs*-solver reads

```
Line  Column 5 10 15 20 25 30 35 40
 1: Var lny3,v1d1,v2d1,v3d2,v4d2,
 2:     v2d3,v4d3,dummy:Real;
 3: Begin
 4:   lny3:=Ln(y[3]);
 5:   v1d1:= +kt[1];
 6:   v2d1:= +2*ks[2]*y[1]*y[3];
 7:   v3d2:= +kt[3];
 8:   v4d2:= +ks[4]*Exp(0.5*lny3);
 9:   v2d3:= +ks[2]*y[1]*y[1];
10:   v4d3:= +0.5*ks[4]*y[2]*Exp(-0.5*lny3);
11:   pd[1,1]:= -v1d1-v2d1-k0;
12:   pd[1,2]:= +v3d2+2*v4d2;
13:   pd[1,3]:= -v2d3+2*v4d3;
14:   pd[2,1]:= +v2d1;
15:   pd[2,2]:= -v3d2-v4d2-k0;
16:   pd[2,3]:= +v2d3-v4d3;
17:   pd[3,1]:= +2*v1d1-v2d1;
18:   pd[3,2]:= +v3d2-v4d2;
19:   pd[3,3]:= -v2d3-v4d3-k0
```

Text 9.4: Contents of **Pederv** procedure of the theoretical mechanism.

```
Line  Column 5 10 15 20 25 30
 1: 4 3 20 8 1 3
 2: Y1 Y2 Y3
 3:  1  0  0   r1
 4:  2  0  1   r2
 5:  0  1  0   r3
 6:  0  1  0.5 r4
 7: 3
 8: 3 ks[1]
 9: 4 150.0*ks[2]
10: 6 2.0*ks[5]
11:
12:  1  1  1
13:
14: -1  0  2    r1
15: -1  1 -1    r2
16:  1 -1  1    r3
17:  2 -1 -1    r4
18: 1 1 2
19:  1  0  0
20: tr[1]:=ks[7]-Ln(y[1])/Ln(10.0);
21: tr[2]:=ks[8]*y[2];
```

Text 9.5: Contents of `*.ODE` file of the theoretical mechanism.

```
Line  Column 5 10 15 20 25 30 35 40 45 50 55 60 65 70
 1: Assign(ddd,ZitaRd+'\tmp');
 2: {$I-} Reset(ddd); {$I+}
 3: If IOResult = 0 Then Erase(ddd);
 4: Assign(ddd,ZitaRd+'\out');Rename(ddd,ZitaRd+'\tmp');Reset(ddd);
 5: If w = 1 Then Assign(out,ZitaRd+'\out') Else Assign(out,ZitaSimu+'\out');
 6: Rewrite(out);
 7: Repeat
 8:   Read(ddd,t);Write(out,Swr(t,cor,less),' ');
 9:   For i:=1 To u Do Read(ddd,y[i]);Readln(ddd);
10:   tr[1]:=ks[7]-Ln(y[1])/Ln(10.0);
11:   tr[2]:=ks[8]*y[2];
12:   For i:=1 To cn Do Write(out,Swr(tr[i],cor,less),' ');Writeln(out)
13: Until SeekEof(ddd);
14: Close(ddd);Close(out)
```

Text 9.6: Contents of **Transform** procedure of the theoretical mechanism.

the values of `PARAMTRS.DAT` firstly and after this calls **Krelations** procedure.

Diffun procedure includes the rate equations optionally and the formulae of the concentrations' difference quotients with respect to time. Entering this procedure `y[k]` variable comprises the $\texttt{k}^{\text{th}}$ concentration in the reactor while `ks[m]` contains the $\texttt{m}^{\text{th}}$ parameter (it is usually a rate constant). If there are inflow and/or outflow, `cstr[i]` contains the concentration of the $\texttt{i}^{\text{th}}$ species in the entering substance, `k0` contains the value of k or k^{in} defined in *Appendix A* and `ko` can contain the value of k^{out} defined in *Appendix A*. In the case of temperature dependent measurements `kt[j]` contains the $\texttt{j}^{\text{th}}$ temperature dependent rate constant. The value of `j` is determined by the user in **Temperature** procedure but this value cannot be greater than the values of `p` declared in the 15^{th} line of `ODES.SPL`. The value

of actual time is stored in `t` variable. Finishing this procedure, `v[i]` can contain the rate of the $\mathtt{i}^{\text{th}}$ reaction step while `d[h]` must contain the value of the $\mathtt{h}^{\text{th}}$ difference quotient. Other values can be stored only in variables declared within this procedure. *Text 9.3* contains the contents of this procedure concerning the concrete mechanism.

The analytical expression of the derivatives of difference quotients with respect to a given concentration can also be supplied. They are used by some integrating method and their application usually results in a bit faster execution. These expressions are stored in **Pederv** procedure. Entering this procedure, the same variables can be used as in Diffun procedure. Finishing this procedure, `pd[i,j]` variable must contain the derivative of the $\mathtt{i}^{\text{th}}$ difference quotient with respect to the $\mathtt{j}^{\text{th}}$ concentration. The formula shows it in the following way:

$$\mathtt{pd[i,j]} := \frac{\partial\left(\frac{\Delta y_{\mathtt{i}}(t)}{\Delta t}\right)}{\partial y_{\mathtt{j}}(t)} \tag{9.2}$$

Pederv procedure is empty if numerical differentiation is used instead of the analytical one for the simulation. *Text 9.4* contains the contents of this procedure concerning the concrete mechanism.

Transform procedure makes possible the fact that not only the concentration vs. time curves can be the final results of the numerical integration. If the concentration vs. time curves are wanted, this procedure is empty. If the user wants to write this procedure manually, the 113^{th} line and the part '`(*`' of the 98^{th} line of `ODES.SPL` have to be deleted and the 108^{th} line of `ODES.SPL` have to be replaced by the actual relationship between concentrations and measured characteristics. The right side of the assignment statements can comprise constants and variable names of any concentration or any parameter. These statements can also consist of several lines. One line can contain maximum 126 characters. The only requirement is that finishing this procedure the $\mathtt{i}^{\text{th}}$ measured characteristics must get into the `tr[i]` variable. *Text 9.6* contains the contents of this procedure concerning the concrete mechanism.

There are two more vectors which can contain real numbers. These are called `ip[i]` and `cp[i]` ($1 \leq \mathtt{i} \leq 50$). These variables can get their values from the experimental files or `PARAMTRS.DAT`. The process of the assignment is detailed in *Section 8.1.* The values of these variables can be used in all the five procedures detailed above if the *ODEs* is written manually. If the user uses the matrix-formalism (*Subsection 9.1.2*), these variables can be used only in **Transform** procedure.

The user is supposed to know the PASCAL programming language if s/he wants to write the source text of *ODEs* manually. In all the other cases it is not necessary (with only two exceptions explained later).

The `ODES.PAS` files of the tests show some examples how the special problems can be handled. *Tests B–E* show how **Diffun** and **Pederv** procedure can be used. *Tests C–E* use **Transform** procedure, too. **Transform** procedure of *Tests C–E* and **Diffun** procedure of *Test D* give examples how a new variable can be declared and used[2]. Lines 22, 31–42, 48

[2] In these examples the variable `l` equals to the natural logarithm of ten.

and 53 of ODES.PAS in *Test C* show an example how a new procedure can be defined and used.

The user can get more examples if ODES.PAS file of the examples is created by the help of *.ODE files and COMPILE.BAT program. These files are detailed in *Subsection 9.1.2* and *Section 9.2*.

9.1.2 Matrix-Formalism for Generating the Source Text of the *ODEs*

The method described in *Subsection 9.1.1* sometimes essential though it seems complicated. Therefore, ZIA contains a program named MAKEODE.EXE which helps to write ODES.PAS. In this case the data of *ODEs* must be given in an *.ODE file situated in the user's library. This file is described in the following paragraphs. This file of the theoretical mechanism can be found in *Text 9.5*.

There are three possibilities for the syntax of the first line in *.ODE files (the notation system is described in *Appendix C*):

Syntax A: I1 I2 I3
Syntax B: I1 I2 I3 I4
Syntax C: I1 I2 I3 I4 $I5_1$ $I5_2$ ... $I5_k$ $\{1 \le k \le I1\}$

The meanings of I1, I2, I3 and I4 correspond to the meanings of ez, n, tmax and p variables in ODES.SPL in the same sequence. If I4 is not indicated, it automatically equals to the value of I1.

Syntax C is needed if temperature dependent measurements are carried out. In this case MAKEODE.EXE presupposes that the Arrhenius equation (defined in *Eq. (7.9)*) is valid for each rate constant which has temperature dependence. Thus k is the number of temperature dependent rate constants and the value of $I5_j$ corresponds to the ordinal number of the j^{th} temperature dependent rate constant among all the parameters. For example: the '30 5 45 33 3 17 6' line means that there are 30 chemical equations (and rate constants), 5 species and maximum 45 experimental points in each curve. The 3rd, the 6th and the 17th rate constants depend on temperature so their values are determined by two parameters at a given temperature thus there are 33 parameters altogether. The parameters of the temperature dependent rate constant are p and A defined in *Eq. (7.9)*.

Some conditions must be fulfilled in the sequence of the parameters in order to make their meanings unambiguous. If a chemical model consists of e equations, the values of the first e parameters must be the rate constants. If the i^{th} rate constant ($1 \le i \le e$) depends on temperature then the i^{th} parameter is not the rate constant itself but its p_i part (defined in *Eq. (7.9)*). If a model contains some other parameters, e.g. molar absorption as well, they could only follow the rate constants, p_i-s and A_i-s. In this way, the $(e+1)^{th}$ parameter is part A (defined in *Eq. (7.9)*) of the $(I5_1)^{th}$ temperature dependent rate constant, the $(e+2)^{th}$ parameter is part A of the $(I5_2)^{th}$ temperature dependent rate constant, etc. It means that the 3rd, the 6th, the 17th, the 31th, the 32th and the 33th parameters correspond

to the values of p_3, p_6, p_{17}, A_3, A_{17} and A_6 in the same sequence in the example outlined above. The other parameters are simply rate constants.

The theoretical mechanism contains 8 parameters. The indices of them correspond to the conditions described now as the user can see it in *Eq. (9.1)*. The first line of *Text 9.5* can be interpreted in the following way:

- The model consists of four equations.
- k_2 and k_4 equal to p_2 and p_4. Since k_1 and k_3 are temperature dependent rate constants, k_1 is determined by p_1 and p_5. p_3 and p_6 determine the value of k_3.
- Those parameters which do not depend on the rate constants (p_7 and p_8) follow all the others.

The second line of `*.ODE` either contains remarks or it has to be an empty one.

After this the order matrix follows the meaning of which can be explained in the following way: the j^{th} element in the i^{th} line ($\mathtt{O_{i,j}}$) of the matrix gives the order of the j^{th} species in the i^{th} rate equation. It can be either an integer number or such real one which corrects to maximum two places of decimals. If its value is 0, the concentration of the j^{th} species is not contained in the i^{th} rate equation, otherwise, the power of this concentration equals to the element in the j^{th} column and in the i^{th} row of the matrix. The power can also be less than 0, for example the inhibition can be taken into consideration in such a way. One line can contain only the data of one rate equation. The order matrix of the theoretical mechanism is given in lines 3–6 of *Text 9.5*. The syntax of this matrix can be described in the next way (C is defined later in this subsection):

$$\begin{array}{cccc} \mathtt{O/C_{1,1}} & \mathtt{O/C_{1,2}} & \cdots & \mathtt{O/C_{1,n}} \quad * \\ \mathtt{O/C_{2,1}} & \mathtt{O/C_{2,2}} & \cdots & \mathtt{O/C_{2,n}} \quad * \\ \vdots & \vdots & & \vdots \\ \mathtt{O/C_{ez,1}} & \mathtt{O/C_{ez,2}} & \cdots & \mathtt{O/C_{ez,n}} \quad * \end{array}$$

The next part of an `*.ODE` file is necessary only in that case if there is at least one such kind of parameter which is expressed by the others. In this case the relations must be given directly after the order matrix. In other words, either an empty or a remark line cannot be placed between the order matrix and this part. If there is not any relation, this part must be omitted. The syntax of this part is the following:

$$\begin{array}{ll} \mathtt{I1} & * \\ \mathtt{I2_1} & \mathtt{S_1(115)} \\ \mathtt{I2_2} & \mathtt{S_2(115)} \\ \vdots & \vdots \\ \mathtt{I2_{I1}} & \mathtt{S_{I1}(115)} \end{array}$$

where

`I1` is the number of those parameters which are expressed by the others.

$\texttt{I2}_\texttt{i}$ is the ordinal number of the $\texttt{i}^{\text{th}}$ parameter to be expressed by the others if $\texttt{I2}_\texttt{i}$ is greater than 0. If $\texttt{I2}_\texttt{i}$ is a negative number the $\texttt{(ez+tk+Abs(I2}_\texttt{i}\texttt{))}^{\text{th}}$ parameter value will be changed according to the assignment statement.

$\texttt{S}_\texttt{i}$ is the right side of that assignment statement which gives the actual value for the $\texttt{i}^{\text{th}}$ parameter expressed by the others.

In the theoretical mechanism there are two rate constants (k_3 and k_4) which depend on the others. Since k_1 is a temperature dependent rate constant, its value depends on two parameters. Therefore, there are three parameters expressed by the others. They are p_3, p_4 and p_6. This part of the theoretical mechanism can be found in Lines 7–10 of *Text 9.5*. Another example can be found in example 10, where $[\mathbf{A}]_0$ and $[\mathbf{B}]_0$ are parameters to be fitted but $[\mathbf{A}]_0 + [\mathbf{B}]_0$ must be constant.

After this the following line is an empty one then the following line includes a vector all the elements of which can be 0 or 1 and its length equals to the number of species. If the j^{th} element of the vector is 0, the concentration of the j^{th} species will be kept to be constant during the simulation (it may be necessary in a buffer solution or in a catalytic reaction). If the j^{th} element is 1, the concentration of the j^{th} species will change according to the *ODEs*. This line is also followed by an empty line. The line belonging to the vector can contain any kind of remarks standing behind the data. Line 12 of *Text 9.5* shows that all the concentrations change during the simulation in the theoretical mechanism.

Now the stoichiometric matrix has to be given. It also consists of either integer numbers or such real ones which corrects to maximum two places of decimals. Its the j^{th} element in the i^{th} line ($\texttt{C}_{\texttt{i,j}}$) equals to the difference between the stoichiometric coefficients of the right and the left side in the i^{th} equation regarding to the j^{th} species. The coefficient matrix of the theoretical mechanism is given in Lines 14–17 of *Text 9.5*. The syntax of coefficient and order matrix is the same.

The following line contains up to four integer numbers. The first two ones are required, the rest is optional. The meaning of the first number is the same as the meaning of `cs` detailed in *Subsection 9.1.1*. The second number determines what kind of the final results will be. If its value is 0, the final results will be concentrations. If its value is 1, the final results will be analogous to the measured characteristics. In this case the relationships between concentrations and measured characteristics should be given in `*.ODE` files, too. The third datum gives the number of measured characteristics. This datum is necessary only if the second datum of this line equals to 1 or the fourth number should also be given. The last data is the dimension of `ip` and `cp` arrays. If it is not given then the default value is 50.

Depending on the first two data of this line the last part of `*.ODE` file has several different forms:

- In the case of '`0 0`' these two numbers are the last data of `*.ODE` file.
- In the case of '`1 0`', '`2 0`' and '`3 0`' the following line consists of a vector. The element of it can be 0 or 1 and the length of it is the number of species. The value of the j^{th}

element is 1 if the j^{th} species flows into the reactor, otherwise, this value is 0. The role of this vector can be understood by comparing **Diffun** procedure of *Text 9.3* and line 19 of *Text 9.5*. Only if 1 belongs to the j^{th} species in the 19^{th} line of *Text 9.5*, the difference quotient of this species contains the increase in concentration deriving from the flow. Any remark can be placed into this line after the vector.

- In the case of '`4 0`' the `MAKEODE.EXE` supposes that some of the experimental files are measured under constant pressure and also generates the appropriate part of the source text of the *ODEs*. This option cannot be combined with inflow and/or outflow so these two numbers are the last data of `*.ODE` file.
- In the case of '`0 1`' the relationships between concentrations and measured characteristics should be given according to the detailed description at the end of *Subsection 9.1.1*. Only the assignment statements of the relationships have to be written into `*.ODE` file, all the other parts of **Transform** procedure will be generated by `MAKEODE.EXE`[3]. In this case `*.ODE` file must not contain remarks standing behind the assignment statements. These remarks would be interpreted as a PASCAL source text and it would cause an error during the compilation. *Text 9.6* and lines 20–21 of *Text 9.5* shows an example concerning this possibility.
- If the options detailed above are combined, the vector (mentioned in the case of '`1 0`') should be given first and then the assignment statements follow it.

`ODES.PAS` can be generated by `MAKEODE.EXE` on the basis of an `*.ODE` file. This method can be applied only in that case if the user's model contains only elementary or mechanistic steps and miv`n*ez` is less than 65000. Since `MAKEODE.EXE` uses dynamic variables, a smaller model can also cause an error message, if the user's computer has not enough memory. If these conditions has not been fulfilled, the `ODES.PAS` can be created only by the method detailed in *Subsection 9.1.1*.

9.2 Creating the *ODEs*-Solver

`COMPILE.BAT` creates the concrete *ODEs*-solver. It requires either `%ZitaTEMP%\ODES.PAS` file written manually or a `%ZitaTEMP%\*.ODE` file. This DOS command file can be used in two ways.

- If `ODES.PAS` file is ready (it has been written manually) then `COMPILE.BAT` carries out compilation and linking either by the help of one DOS parameter or without it. The role of the DOS parameter is the same as that of the second DOS parameter of `NEWEXE.BAT` detailed in *Section 6.2*. If the DOS parameter is missing, the type of the real numbers will be an extended one.
- If the data of the actual *ODEs* are stored in an `*.ODE` file, two or three DOS parameters are required. The first one is the filename of an `*.ODE` file either without the extension

[3] However, the user must use semicolons corresponding to the syntax of PASCAL!

or with it. The second one can be 1 or 2. If it is 1, **Pederv** procedure will be empty and only such a simulation method can be chosen that applies numerical differentiation. If the second DOS parameter is 2, the analytical derivatives will be generated, too. The third DOS parameter is the same as that of the second DOS parameter of `NEWEXE.BAT`.

After creating `ODES.PAS`, `COMPILE.BAT` compiles and links the source texts by the help of `TPC.EXE`, and `TURBO.TPL`. The final result is `ZITA.EXE` executable program which is situated in the user's library. `ZITA.EXE` runs with one DOS parameter. It is the filename of the actual `*.DAT` file (the extension can be omitted if it is `DAT`). It includes the data of numerical integration. If the DOS parameter has not been given, `ZITA.EXE` asks for it interactively. `ZITA.EXE` can interpret more DOS parameters as well. This possibility is described in *Section 9.4*.

Using `COMPILE.BAT` in practice can sometimes be risky. If the user fails to give the wanted DOS parameters and there is an old `ODES.PAS` file in existence on the disk, the batch file named `COMPILE.BAT` works without an error message but not according to `*.ODE` file of the expected *ODEs*.

Finally, there are some limitations concerning the size of the model. All the following three ones must be fulfilled in order to create the executable *ODEs*-solver:

- The size of `ODES.PAS` must be less than 64613 bytes.
- The data memory size in `ODES.PAS` must be less than 65520 bytes.
  This value equals to $552 + 2 \cdot \texttt{rels} + S \cdot (\texttt{n}^2 + 3 \cdot \texttt{n} + \texttt{ez} + 2 \cdot \texttt{p} + 108)$.
- The data memory size in `ZITA.PAS` must be less than 65520 bytes.
  This value equals to $22152 + 2 \cdot \texttt{rels} + 36 \cdot \texttt{n} + S \cdot (\texttt{n}^2 + 23 \cdot \texttt{n} + \texttt{tmax} + \texttt{ez} + 2 \cdot \texttt{p} + 261)$.

where S is the size of a real number in bytes (see in *Table 6.1*) and `n`, `ez`, `p`, `rels` and `tmax` are the values of the variables named `n`, `ez`, `p`, `rels` and `tmax` in `ODES.SPL`.

9.3 Constructing the Data File for Simulation

The input data of *ODEs*-solver (`ZITA.EXE`) are stored in two files: in an `*.DAT` (its name is the first DOS parameter) and in `PARAMTRS.DAT`. These files have to be placed into user's library.

If that file the contents of which is to be used has another extension, it must be indicated in the first DOS parameter, otherwise, the extension can be omitted. `PARAMTRS.DAT` includes the values of the parameters one by one (the i^{th} value is stored in the i^{th} line). This filename is reserved so the user cannot use it for his own purposes!

The structure of `PARAMTRS.DAT` is more complicated as it is outlined above. This is because `GNM.EXE` also uses this file and `PARAMTRS.DAT` can also contain the parameters of `GNM.EXE`. These data do not affect the simulation so the exact structure of `PARAMTRS.DAT` is detailed only in *Subsection 10.2.1*.

The simulation takes places in the following way: `ZITA.EXE` gets those input data from `*.DAT` file which determine the method of calculation. The calculated curves can be found in files which are stored either in `%ZitaSIMU%` or `%ZitaRD%` library. The data files can be constructed in such a way that any number of curves can be calculated by a single run of `ZITA.EXE`. A plenty of examples can be found in the tests and examples.

`*.DAT` files can be divided into three parts. The first one consists of only titles or remarks. The second one contains those information which refer to all the calculated curves, e.g. the method flag, the relative error bound, etc., until this part is not repeated again. The third part carries those special information which always concern only the curve under calculation. For example: lines 1–3 belong to the first part, lines 4–10 belong to the second part and lines 11–66 belong to the third part in `CSTR.DAT` file.

Later on a number and a capital letter separated by a period will indicate a logical unit of `*.DAT`. The number shows to which group the given unit belongs to and the letter expresses the sequence of the logical units.

UNIT 1.A

Syntax:
```
S1(100)
S2(100)
S3(100)
```

This unit can contain titles and remarks. The first three lines of the data file belong to here. If the user does not want to use titles or remarks, these lines must be left empty. These lines is used by `ZITA.EXE` only if formatted results are expected (see below in detail). Each example file contains remarks.

UNIT 2.A

Syntax : `I1 I2 I3 [ |I4  *]`

Information concerning the general features of the result files must be given in this logical unit.

`I1` is the number of the significant digits of a real number stored in the result files.

`I2` is the maximum number of the characters in one line. If the user would like to print the result files, `I2` should be chosen in such a way that one line of a result file would not be longer than the length of a printed line.

`I3` equals to 0 if the result files are stored in `%ZitaSIMU%` library. If it equals to 1, the files are stored in `%ZitaRD%` library.

The result files are always stored by the *ODEs*-solver in `%ZitaRD%` library at first and these files are moved into `%ZitaSIMU%` library if one of the following conditions is fulfilled:

- I3 equals to zero.
- If the user likes to have formatted results.
- The free space on RAM disk is less then 4096 bytes after storing the data of the last calculated curve. In this case moving takes place by all means in order to keep the result calculated until the actual time. It also means that the program can sometimes make a mistake if the next result file is greater than 4096 bytes. This problem can be solved by using a larger RAM disk or cutting the actual *.DAT file into pieces.

  If the user is not satisfied with the 4096 bytes limit, this value can be changed if the number 4096 in the 1455th line of ZITA.PAS is replaced by the appropriate value. The meaning of this value is the amount of the minimum free space in bytes on RAM disk before simulation. If ZITA.PAS is changed, the *ODEs* solver has to be recompiled!

If the results are saved into %ZitaSIMU% library, all the space in %ZitaRD% library will be set free.

I4 equals to 0 if the user does not want to get technical information about the runs, otherwise, I4 equals to 1. If another value is given to I4, it is considered to be 0 by the *ODEs*-solver. The pieces of information about runs are added to the file named MESSAGES in the user's library. If this file has not existed, the program creates it. The form of the information is the following (using the symbol system defined in *Appendix C*):

```
The N1 was calculated within R1s.
The step size last successfully used:          R2
The order last successfully used:              I1
The cumulative number of steps taken:          I2
The cumulative number of Diffun calls:         I3
The cumulative number of Jacobian evaluations: I4
```

All the tests write technical information into MESSAGES file.

UNIT 2.B

Syntax: `I1 R1 R2 *`

I1 defines the integration method to be applied. Its value can be the following:

- **10** means the Adams method with functional iteration.
- **11** means the Adams method using the analytical derivatives.
- **12** means the Adams method using the numerical derivatives.
- **13** means the Adams method using the diagonal approximation based on a directional derivative.

20, **21**, **22** and **23** mean the same as the values less by **10** but the Gear stiff integrator of is used instead of the Adams method.

31 and **32** mean the same as the values less by 20 but ROW4 is used instead of the Adams method.

The theoretical background is described in [3, 4, 5, 6].

`R1` is the maximum value of the relative error bound in two successive concentration calculations. This value is detailed in [3].

`R2` is the initial step size in time. This value is detailed in [3].

If `ZITA.EXE` is terminated with an overflow or underflow error, the reason for which in most cases is that the values of `R1` and `R2`, defined in this unit, were not chosen correctly. By changing these values the problem in question can usually be solved.

UNIT 2.C

Syntax:

`I1`	`*`
$\texttt{I2}_1$	$\texttt{S}_1\texttt{(m)}$
$\texttt{I2}_2$	$\texttt{S}_2\texttt{(m)}$
$\vdots$	$\vdots$
$\texttt{I2}_{\texttt{I1}}$	$\texttt{S}_{\texttt{I1}}\texttt{(m)}$

This logical unit determines the contents of the result files.

`I1` is the number of concentrations to be stored.

$\texttt{I2}_\texttt{k}$ expresses which calculated concentration is stored in the $(\texttt{k}+1)^{\text{th}}$ column of the result file (the first column always contains the values of time).

$\texttt{S}_\texttt{k}\texttt{(m)}$ is the title of the $\texttt{k}^{\text{th}}$ column. The maximum value of `m` can be greater by 2 than `I1` defined in UNIT 2.A. $\texttt{S}_\texttt{k}\texttt{(m)}$ ought to be given only then if the user prefers to get formatted result files.

If the result file contains measured characteristics and not concentrations, this logical unit has a different meaning. In such cases the titles of the columns are not important because the name of the i^{th} calculated characteristic will be '`res`i' by all means.

The procedure of making up the calculated characteristics is the following: at first, the *ODEs*-solver calculates the concentrations of the $\texttt{I2}_1$–$\texttt{I2}_{\texttt{I1}}$ species at given points of time and these values are stored in the file named `OUT` in `%ZitaRD%` library. The result file is created by the help of `OUT` file. Consequently, `y[i]` variable used in **Transform** procedure does not always correspond to the $\texttt{i}^{\text{th}}$ species but it is the same as that species the values of which stand in the $(\texttt{i}+1)^{\text{th}}$ column of `OUT` file. The user should pay attention to this when he is working on an `*.ODE` file!

It would be useful if `y[i]` in **Transform** procedure corresponded to the `i`$^{\text{th}}$ species so the possibility of a mistake decreases. It can be attained if `I1` is the number of species altogether and `I2`$_{\texttt{k}}$ is the ordinal number of the `k`$^{\text{th}}$ species.

UNIT 3.A

Syntax A: `I1` $\{=0\}$ `[|P]` `*`

Syntax B: `I1` $\{>0\}$ `R1 C1 [|P]` `*`

Syntax C:

`I1` $\{<0\}$ `*`

`I2`$_1$	`R2`$_1$ `*`
`I2`$_2$	`R2`$_2$ `*`
$\vdots$	$\vdots$
`I2`$_{\text{abs(I1)}}$	`R2`$_{\text{abs(I1)}}$ `*`

This logical unit determines the values of parameters used for the calculation of the given curve.

- If `I1` equals to 0, the calculation is carried out with those values which are stored in `PARAMTRS.DAT` without changes.
- If `I1` is greater than 0, `I1` means the ordinal number of a parameter, `R1` is a real number and `C1` can be '`+`' or '`-`'. The original value of the `I1`$^{\text{th}}$ parameter is changed by `R1` percent. If `C1` is '`+`', the value of this parameter becomes larger, otherwise, this values becomes smaller. Finishing the calculation of the actual curve the original value stored in `PARAMTRS.DAT` are valid again. For example: the '`3 1.0 +`' line expresses that during the next calculation the third parameter will be larger by 1.0 percent but after this calculation the original value of this parameter will be valid again. This possibility helps to calculate the numerical derivatives in a simple way. They are necessary for fitting.
- If `I1` is less than zero, the values of the parameter loaded from `PARAMTRS.DAT` file can be overwritten[4]. This method makes possible to work without `PARAMTRS.DAT` file. There are several illustrations concerning this possibility in the tests. The meaning of the symbols are the followings:

  abs(`I1`) is the number of the parameters to be revised.
  `I2`$_{\texttt{k}}$ $(1 \leq \texttt{k} \leq \text{abs}(\texttt{I1}))$ is the ordinal number of a parameter.
  `R2`$_{\texttt{k}}$ $(1 \leq \texttt{k} \leq \text{abs}(\texttt{I1}))$ is the new value of the `I2`$_{\texttt{k}}$$^{\text{th}}$ parameter.

- If `P` is given, the actual curve is calculated only if the serial number of the changed parameter (see the second item!) is *not* in the parameter list defined by `P`. This possibility helps to avoid the unnecessary calculations.

  Sometimes it is possible that a parameter does not have influence on the curve to be calculated. In this case, we would get the same result at arbitrary parameter values,

[4] If `PARAMTRS.DAT file` does not exist, `ZITA.EXE` sends a message onto the screen and makes equal the parameter values with zero.

so the recalculation of the calculated curve is not necessary. The user can define the omittable parameters for the curve to be calculated through P.

If I1=0 then P is considered only if the parameter to be changed are defined through the DOS parameters (see *Section 9.4*). If it is not so, the calculated curve will be integrated.

If somebody wants to carry out only simulations, this possibility is perhaps totally unnecessary. It plays role during only fitting. For a more detailed explanation, see the keystring EXPERIMENTAL␣CURVES in *Section 12.1*.

The parameter values can be changed by the DOS parameters, too. This method is described in *Section 9.4*.

UNIT 3.B

Syntax A:

I0 $\{=1\}$ *
[|C_1 $I1_{1a}$ $I1_{1b}$ $I1_{2a}$ $I1_{2b}$ ... $I1_{ya}$ $I1_{yb}$] $\{1 \le y \le 50\}$
[|C_2 $I2_1$ $R1_1$ $I2_2$ $R1_2$ ... $I2_z$ $R1_z$] $\{1 \le z \le 50\}$
[|C_3]
[|R2 *]
$R3_1$ [|$R4_1$] *
$R3_2$ [|$R4_2$] *
⋮
$R3_n$ [|$R4_n$] *
[|R5] [|R6] *

Syntax B:

I0 $\{=2\}$ N1 I3 $I4_1$ $I4_2$ ... $I4_{I3}$ *

Syntax C:

I0 $\{=2\}$ N1 I3 $I4_1$ $I4_2$ ... $I4_{I3}$ *
I5 $I6_1$ $I6_2$ ... $I6_{I5}$ *

The initial values of the concentrations are determined in this logical unit. These values can be given in several ways.

- If I0 equals to 1, the initial values should be given directly in the actual *.DAT file. In this case all the concentration values have to be defined even though they equal to 0.0. By the help of this option the *ODEs*-solver can be used independently of the other parts of ZIVA.

  n has the same meaning as the variable named 'n' in ODES.SPL.

  C_1, $I1_{ya}$, $I1_{yb}$, C_2, $I2_z$, $R1_z$ and C_3 have already been defined in *Section 8.1*. Their meanings are exactly the same here. If there is not such a parameter, the appropriate line(s) must be omitted from the data file.

`R2` is the temperature value. If it is not necessary (because there is not any temperature dependent rate constant), this line must be omitted from the data file.

$\texttt{R3}_\texttt{i}$ is the initial value of the $\texttt{i}^{\text{th}}$ concentration in the reactor.

$\texttt{R4}_\texttt{i}$ is the initial value of the $\texttt{i}^{\text{th}}$ concentration in the entering substance ($\texttt{R4}_\texttt{i}$ equals to $\texttt{R4}_i$ defined in *Eq. (8.1)*). These values can be omitted if the value of `cs` (defined in *Subsection 9.1.1*) equals to 0 or 4.

`R5` is the reciprocal of residence time (k_0) in CSTR, otherwise, this is k^{in} defined in *Appendix A*. If `cs` is 0 or 4, this line must be omitted from the data file.

`R6` is k^{out} defined in *Appendix A*. If `cs` is not 3, this number must be omitted from the data file.

`*.DAT` files of the tests give some illustrations.

- If `I0` equals to 2, the initial values are stored in an experimental file of `%ZitaEXPR%` library. **Syntax B** is valid if `cs` $\in (0, 4)$, otherwise, **Syntax C** must be used.

  `N1` is the DOS name of the experimental file.

  `I3` is the total number of those initial concentrations in the reactor, which are given in `N1` file. The other concentration values are 0.0 automatically.

  $\texttt{I4}_\texttt{i}$ means that the $\texttt{i}^{\text{th}}$ initial concentration in `N1` file corresponds to the initial value of the $\texttt{I4}_\texttt{i}{}^{\text{th}}$ concentration in the reactor.

  `I5` is analogous with `I3` concerning not the reactor but the entering substance. Either `I3` or `I5` can equal to zero. In this case all the initial concentration values are 0.0 in the reactor and the entering substance.

  $\texttt{I6}_\texttt{i}$ is analogous with $\texttt{I4}_\texttt{i}$ concerning not the reactor but the entering substance.

  There is an example for **Syntax B** in the 12^{th} line of `EXLV.DAT` file. This line means that the first line of `EXLV1.EXP` contains the initial concentrations of the 1^{st}, the 2^{nd} and the 3^{rd} species.

  Another example demonstrates **Syntax C** in the 12–13^{th} lines of `CSTR.DAT`. These lines mean that the initial values stored in `CSTR.EXP` file are the 2^{nd} and the 3^{rd} initial concentrations in the reactor, the 1^{st} and the 2^{nd} initial concentrations in the entering substance and k_0.

UNIT 3.C

Syntax A: `I1` $\{= 1$ or $7\}$ `R1 R2 R3 [ |I2` $\texttt{I3}_1\ \texttt{I3}_2$ `...` $\texttt{I3}_{\texttt{I2}}$ `C1 R4 *]`

Syntax B: `I1` $\{= 2\}$ `R1 I4 *`
$\texttt{R5}_1$ `*`
$\texttt{R5}_2$ `*`
⋮

R5$_{I4}$ *

Syntax C: I1 {= 3} R1 N1 I5 *

Syntax D: I1 {= 3} R1 *
{only if **Syntax B** or **C** is chosen in UNIT 3.B}

Syntax E: I1 {= 4 or 10} R1 R6 I6 R3 [|I2 I3$_1$ I3$_2$ ... I3$_{I2}$ C1 R4 *]

Syntax F: I1 {= 5 or 8} I7 R1 I4 *
R5$_1$ *
R5$_2$ *
⋮
R5$_{I4}$ *

Syntax G: I1 {= 6 or 9} I7 R1 N1 I5 *

Syntax H: I1 {= 6 or 9} I7 R1 *
{only if **Syntax B** or **C** is chosen in UNIT 3.B}

This logical unit determines those time values where the calculated data should be saved. This is the most complicated unit so the user should read it carefully.

- R1 is always the initial value of time. The initial concentrations are given at this time point.
- If I1 equals to 1 then the difference between two successive time points is constant.
  - R2 is the length of the interval between two successive points of time to be stored. The first time point is R1.
  - R3 is the last point of time where the user wants to store the calculated values.
  - If I2 equals to 0, the data standing behind it are redundant and the storing of the calculated data takes places at each point of time.
  - If I2 does not equal to 0, there are some restrictions concerning the concentration to be saved. Namely, storing will be carried out only in that case if the average of the relative or absolute change is larger than a given number (R4) in comparison with the previously stored concentrations. I2 means how many species altogether are to be taken into account with respect to the conditions mentioned above. I3$_i$ is the ordinal number of the i^{th} species taken into account. C1 can get the following values: 'r', 'R', 'a' and 'A'. In the first two cases the comparison is carried out on the basis of relative values while in the last two cases absolute values are used. The average change must be greater than R4. For example: the '1 0.0 0.5 1000.0 3 1 3 15 a 1.5' line means that storing has to take place

at the actual point of time if the average of absolute concentration changes with respect to the 1st, the 3rd and the 15th species is greater or equals to 1.5. The '`1 0.0 0.5 1000.0 4 1 2 3 5 R 0.01`' line means that storing has to take place at the actual point of time if the average of relative concentration changes with respect to the 1st, the 2nd, the 3rd and the 5th species is greater or equals to 1.0 percent. If any of the indicated concentrations was 0.0 in the previous storing, the relative condition does not make sense so storing takes place by all means.

For example, line 13 of `EXLV.BAT` means that the calculated concentrations are stored in that cases if the value of time is 0.0, 0.5, 1.0, ... Another example can be found in line 33 of `TESTE.DAT`. It means that the calculated concentrations can be stored if the value of time is 0.0, 100.0, 200.0, ... but they are stored only if the average of relative concentration changes with respect to the 2nd species is greater or equals to 5.0 percent. The meaning of `I1` = 7 is described later in this section.

- If `I1` equals to 2, those values of time, where the calculated data should be stored, are given in the actual `*.DAT` file.
  - `I4` is the total number of time values. `I4` must be smaller or must equal to the value of `tmax` defined in `ODES.SPL` file.
  - $\texttt{R5}_{\texttt{i}}$ is that the $\texttt{i}_{\texttt{th}}$ time value, where the calculated data should be stored. Each line is allowed to contain only one time value.
- If `I1` equals to 3, the values of time are stored in an experimental file which is always in `%ZitaEXPR%` library.
  - `N1` means the DOS name (without path) of the experimental data file.
  - `I5` is the number of those lines which must be skipped at the beginning of the experimental data file before the program begins to read the values of time. These lines contain the initial concentrations which can be avoided in this way.

  There is a restriction in this unit but it makes the work easier for the user. If **Syntax B** or **C** was chosen in UNIT 3.B then `N1` and `I5` should not be given. Even though they are defined, they will be ignored. In this case the program reads time values from that experimental file which is given in **Syntax B** or **C** of UNIT 3.B. Therefore, it is not possible to store the initial conditions and the measured data of a given reaction in separate files. The confusion of the experimental data becomes less probable in this way.

  For example, lines 19–20 of `EXLV1.DAT` mean that the time values are stored in the file named `EXLV2.EXP`.
- If `I1` equals to 4, the difference between *the logarithm of two successive time points* is constant instead of the difference between the time points.
  - `R6` means the first time point to be stored. Since `R1` can be less than zero or equal to it, the user must give the first time value explicitly. `R6` must be greater than zero.

1 1.0 0.1 0.0001 0.0	1 0.0 0.2 0.0001 0.0999 0.02 0.0 0.0 0.0 0.005	1 295.0 0.00002 0.001 0.0 1.0E-11	1 0.0 2.0 78.0 *List 4* 9 3 0.0 cstr.exp 3 *or* 6 3 0.0	2 0.0 30 2.0 4.0 ⋮ 74.0 78.0	8 3 0.0 40 10.0 20.0 ⋮ 390.0 400.0
List 1	*List 2*	*List 3*	*List 6*	*List 5*	*List 7*

Table 9.1: Some examples concerning UNITS 3.B and 3.C.

- ◦ I6 is the number of the calculated points to be stored between R6 and R3.
- ◦ The meaning of the other symbols are the same as if I1 would be 1.

For example, line 18 of TESTA.DAT means that the first and last time points to be saved are 0.001 and $4.0 \cdot 10^7$ and 98 additional points are stored between the first and last ones. The logarithm of the successive time values increases linearly.

- If I1 equals to 5, 6, 8 or 9, the calculated concentrations are stored not only in that points which are indicated in the actual *.DAT or experimental file but storing is also carried out I7 times linearly within every interval. Otherwise, cases 5,8 are the same as case 2 and cases 6,9 are also the same as case 3.

  For example, lines 10–11 of MODEL.DAT mean that the calculated concentrations must be stored at the time points given in EXLV1.EXP and three further points are stored in every interval. In this way the time values to be stored are 0.0, 0.5, 1.0 ..., not only 0.0, 2.0, 4.0 ... .

- If I1 is greater than 6, the accomplished range is indicated on the screen during the simulation. This possibility requires a bit more running time but the user can follow the process of the simulation.

Table 9.1 demonstrates some examples concerning UNIT 3.B and 3.C. The example files named *.DAT do not show all the possibilities so they are supplemented by these lists. The 12^{th} line of EXLV.DAT can be replaced by the contents of *List 1*. The same things can be told about the $12–13^{\text{th}}$ lines of CSTR.DAT and *List 2* as well as about the 9^{th} line of TEM.DAT and *List 3*. The influence of *List 4* or *List 5* is the same as that of the 13^{th} line of EXLV1.DAT though the result file calculated on the basis of *List 4* will consist of more points than the original one. The role of *List 6* or *List 7* is the same as that of the 14^{th} line of CSTR.DAT.

UNIT 3.D

Syntax:
```
C1 [ |N1]
C2 [ |N2]
C3 [ |N3]
```

```
0.00000E+0 1.00000E+0 1.00000E-1 1.00000E-4 0.00000E+0
5.00000E-1 1.00000E+0 1.06100E-1 1.11019E-4 5.26666E-6
.          .          .          .          .
.          .          .          .          .
.          .          .          .          .
7.90000E+1 1.00000E+0 4.48152E-3 1.22448E-4 8.39014E-2
7.95000E+1 1.00000E+0 4.75437E-3 1.17286E-4 8.39074E-2
```

Text 9.7: Segments of an unformatted result file.

This logical unit determines what kind of the result files will be created. Three kinds are possible: the first one contains the data of the calculated curve, the second one the rate values for each reaction step and the third one the derivatives of the concentrations with respect to time. The three lines of this unit contain the general data of these files in the same sequence. Each file belongs to only one line.

`N1`, `N2` and `N3` are the DOS names of the result files.

`C1`, `C2` and `C3` are printable characters. If the character is not '`P`' or '`R`', the result file in question is not stored so the DOS filename is not necessary to be given. If this character equals to '`P`', the result file contains only the calculated data. These sorts of files can be used directly by the other programs for fitting and drawing. If the character corresponds to '`R`', the result file contains other important pieces of information (e.g. parameter values, column names, etc.) besides the calculated data and the contents of this file is formatted. This possibility is worth using if a result file is the final result and the user does not want to go on calculating with them. The files containing formatted results are stored in `%ZitaSIMU%` library independently of `I3` defined in UNIT 2.A.

Text 9.7 shows the segments of that result file named `EXLV1.OUT` which is calculated by the help of `EXLV.BAT` during the first iteration. If the 4^{th} and the $14–16^{th}$ lines of `EXLV.DAT` are replaced by '`6 60 1`', '`r exlv1.out`', '`r exlv1.v`' and '`r exlv1.d`' respectively, the user can get that result file the segments of which are in *Text 9.8*. It can be seen from *Text 9.8* that if formatted files are wanted, all the calculated data are stored only in one file. That filename is chosen from `N1`, `N2` and `N3` in front of which letter '`r`' occurs for the first time. The other names are ignored by the program. However, if a '`p`' is in front of a filename, the file in question is always created separately since this file is supposed to be used by further calculations.

```
This is an *.dat file for solving the Lotka-Volterra model.
G: grass, R: rabbit, L: lion N: dead lion
We try to fit the calculated data on basis of Gear method.
parameter( 1)= 3.00000E-0002
parameter( 2)= 3.00000E+0000
parameter( 3)= 1.00000E-0001

********** VALUES OF Y-s: **********

 t              G              R              L
 0.00000E+0000  1.00000E+0000  1.00000E-0001  1.00000E-0004
 5.00000E-0001  1.00000E+0000  1.06100E-0001  1.11019E-0004
 .              .              .              .
 .              .              .              .
 .              .              .              .
 7.90000E+0001  1.00000E+0000  4.48152E-0003  1.22448E-0004
 7.95000E+0001  1.00000E+0000  4.75437E-0003  1.17286E-0004

 t              N
 0.00000E+0000  0.00000E+0000
 5.00000E-0001  5.26666E-0006
 .              .
 .              .
 .              .
 7.90000E+0001  8.39014E-0002
 7.95000E+0001  8.39074E-0002

********** VALUES OF V-s: **********

 t              v(1)           v(2)           v(3)
 0.00000E+0000  3.00000E-0003  3.00000E-0005  1.00000E-0005
 5.00000E-0001  3.18299E-0003  3.53373E-0005  1.11019E-0005
 .              .              .              .
 .              .              .              .
 .              .              .              .
 7.90000E+0001  1.34446E-0004  1.64626E-0006  1.22448E-0005
 7.95000E+0001  1.42631E-0004  1.67286E-0006  1.17286E-0005

********** VALUES OF dY/dT-s: **********

 t              dY(1)/dt       dY(2)/dt       dY(3)/dt
 0.00000E+0000  0.00000E+0000  1.18500E-0002  2.00000E-0005
 5.00000E-0001  0.00000E+0000  1.25553E-0002  2.42354E-0005
 .              .              .              .
 .              .              .              .
 .              .              .              .
 7.90000E+0001  0.00000E+0000  5.29551E-0004 -1.05986E-0005
 7.95000E+0001  0.00000E+0000  5.62160E-0004 -1.00557E-0005

 t              dY(4)/dt
 0.00000E+0000  1.00000E-0005
 5.00000E-0001  1.11019E-0005
 .              .
 .              .
 .              .
 7.90000E+0001  1.22448E-0005
 7.95000E+0001  1.17286E-0005
```

Text 9.8: Segments of a formatted result file.

UNIT 3.E

Syntax: `C1`

- If `C1` is '`Y`', the actual `*.DAT` file contains some more data for calculating a new curve. In this case the third part is completely repeated after this `C1`.
- If `C1` equals to '`F`', the second and third part is completely repeated after this `C1`. Further curves to be simulated use the values of the changed second part.
- If `C1` equals to '`N`', there are not further curve to be simulated. After `C1` containing '`N`' anything included in `*.DAT` file is ignored.

9.4 Further Methods for Changing the Parameter Values

There are three further methods by the help of which the values of parameters can be changed:

- The user also can use DOS parameters for changing the parameter values. If the 2nd, the 3rd and the 4th DOS parameters exist, UNIT 3.A of the actual `*.DAT` file is disregarded and `I1`, `R1` and `C1` defined in UNIT 3.A are replaced by the 2nd, the 3rd and the 4th DOS parameters. It concerns all the curves to be stored.

  This possibility is very important because the user can avoid creating many `*.DAT` files by the help of it. For example, if the user wants to compute all the curves of example 5 which are needed for an iteration, four files are needed: `TEM.DAT`, `TEM1.DAT`, `TEM2.DAT` and `TEM4.DAT`. The DOS commands in *Text 9.9* calculate the necessary curves. However, this calculation can be carried out by the help of only `TEM.DAT` file, if DOS parameters are applied. *Text 9.10* contains the appropriate commands. This method makes possible to work with only few files. It is very important if `TASK.EXE` is used (see *Chapter 12*).
- If the user carries out transformations on the calculated concentrations, only the transformed values are stored in the simulated data files. However, if the second DOS parameter is '`OPT`', files containing the calculated concentrations are also created. The names of them are the same as those of the simulated data files but their extension is `CNC`.

  There is one more possibility for creating transformed values. The previous item detailed the common usage of the 2-4th DOS parameters. If the fifth DOS parameter is '`OPT`' in that case and the model parameter numbered by the second DOS parameter does not take part in the *ODEs*, the simulation is not carried out, only the transformations are done. `ZITA.EXE` supposes that there was a simulation earlier and the simulated concentrations are in the appropriate `*.CNC` file. The simulated concentrations are used from this file.

  Why can this possibility be useful? It is possible that a parameter has role only in the transformations and it does not take part in the *ODEs*. If the user wants to calculate

```
Zita Tem
Zita Tem1
Zita Tem2
Zita Tem4
```

Text 9.9: Necessary DOS commands with more `TEM?.DAT` files.

```
Zita Tem 4 2.0 -
Rename %ZitaRD%\Tem??5.out Tem??5l4.out
Zita Tem 2 2.0 -
Rename %ZitaRD%\Tem??5.out Tem??5l2.out
Zita Tem 1 2.0 -
Rename %ZitaRD%\Tem??5.out Tem??5l1.out
Zita Tem
```

Text 9.10: Necessary DOS commands with only `TEM.DAT` file.

several curves with different values of this parameter, only one numerical simulation is required for calculating several transformed curves in this way.

- If **Krelations** procedure of `ODES.PAS` file is not empty and the parameter values are placed into `PARAMTRS.DAT`, `ZITA.EXE` rewrites this file before the simulation. At first, **Krelations** procedure calculates the values of those parameters which depend on the others. After this, `ZITA.EXE` writes back the actual values into `PARAMTRS.DAT`. If a parameter depending on the others was marked as parameter to be fitted[5], this marking is erased further on, so this kind of parameters cannot be fitted but it can be only fixed parameter!

[5] See the structure of `PARAMTRS.DAT` in *Subsection 10.2.1.*

Chapter 10

Parameter Estimation

The user is able to calculate the analogous form of any kind of experimental curve by the help of optional parameters on the basis of the previous chapter. This chapter presents how to use the calculated and experimental data for fitting.

It is necessary to calculate the elements of $\boldsymbol{\mathcal{J}}(\vec{k}_i)$ matrix and $\left(\vec{Y} - \vec{F}(\vec{k}_i)\right)$ vector (defined in *Eq. (2.5)*) for the application of Gauss-Newton-Marquardt method [7, 8, 9]. Since the analytical forms of $\vec{y}(t)$ (defined in *Eq. (2.1)*) are usually unknown, numerical differentiation is needed. This can be done in the following way: the user calculates several curves by decreasing or increasing (or both) the parameter values[1] to be fitted (0.5–5.0 % is recommended). These curves contain the calculated data at the same points of time. The difference quotients with respect to a given parameter can be calculated on the basis of these curves. The elements of the matrix and the vector are calculated by `JC.EXE` and stored in files with the extension `COL`. `GNM.EXE` calculates the new parameter values, their standard deviations, the correlation matrix, the Goodness-of-Fit Statistics and the Analysis of Residuals [9, 20].

10.1 Usage of `JC.EXE`

This program presupposes that all the experimental and calculated files are ready.

10.1.1 DOS Parameters of `JC.EXE`

The program requires three or five DOS parameters. If they are not indicated, they must be given interactively.

- The first DOS parameter is the name of an `*.JCD` file (the extension can be omitted). It contains the name of the files including the experimental and calculated data and some other values about fitting. A detailed description about the structure of this file is given in the next subsection.

[1] *Section 9.4* details the most effective method to do it.

- The second DOS parameter determines the method of the calculation of the differences between the measured and calculated data. This parameter can be '`a`', '`r`' or '`o`'. They mean absolute, relative or orthogonal fitting, respectively.

  - In the case of absolute fitting, the measured and the calculated data are used directly in *Eq. (2.3)*. The role of the weighting factors will be detailed later.
  - In the case of relative fitting, the program calculates the difference (DIF) between the maximum and minimum of the experimental data for all the curves. The differences between the measured and the calculated data at the same points are divided with the values of DIF. $S(\vec{k})$ (defined in *Eq. (2.3)*) will be calculated on the basis of these values. In this way all the measured curves carry experimental information equally.
  - The orthogonal fitting takes into consideration the experimental error of both the independent and dependent variables. The precise definition of the sum of squares of residuals is given in *Eq. (A.8)* in this case. *Appendix A* details the mathematical background of this method. *The orthogonal fitting is not recommended in the case of relatively large experimental errors!*.

- If the third DOS parameter equals to 0, the program calculates the elements of $\left(\vec{Y} - \vec{F}(\vec{k}_i)\right)$ vector. In this case, this is the last DOS parameter and the result gets into `J.COL` file. If the i^{th} column of $\boldsymbol{\mathcal{J}}(\vec{k}_i)$ matrix should be generated, this DOS parameter corresponds to the ordinal number of the i^{th} parameter and the result is in `J`i`.COL` file, e.g. `J5.COL` contains the data for the fitting of the 5^{th} parameter.
- The fourth DOS parameter can be '`+`', '`-`' or '`p`'. It determines the method of numerical differentiation. If it is '`+`', the derivative with respect to the parameter (determined by the 3^{rd} DOS parameter) is approached by the slope of a secant. This slope is calculated on the basis of those curves which are simulated by the original and the increased parameter values. The same facts can be told about '`-`' but the decreased parameter value is applied instead of the increased one. At last, if this DOS parameter is '`p`', the program calculates the tangent of a parabola from three points and the derivative is approached by the slope of this tangent.
- The fifth DOS parameter is a real number. It shows by how many percent the actual parameter value was modified during the simulation. Consequently, the percent value of the modification must be the same either increasing or decreasing the actual parameter value.

There are some examples for calling `JC.EXE` in `*.BAT` files of the examples.

10.1.2 Structure of `*.JCD` Files

The first line of an `*.JCD` file contains two integer numbers. All the other lines contain the necessary data concerning a concrete curve. The first number in the first line indicates where the curves, calculated from the original parameters, can be found. If it is 1, these

curves are stored in %ZitaRD% library, if it is 0, they are situated in %ZitaSIMU% library. The second integer number means the same as the first one but with regard to the curves calculated from the modified parameters.

All the other lines can be described by a syntax from the followings:

Syntax A: N1 [I1 {≥ 0} | I1 {< 0} P] I2 N2 I3 R1 [*|R2 *]
{if the 3rd DOS parameter is 0}

Syntax B: N1 [I1 {≥ 0} | I1 {< 0} P] I2 N2 I3 R1 N3 [*|R2 *]
{if the 4th DOS parameter is '+'}

Syntax C: N1 [I1 {≥ 0} | I1 {< 0} P] I2 N2 I3 R1 N4 [*|R2 *]
{if the 4th DOS parameter is '-'}

Syntax D: N1 [I1 {≥ 0} | I1 {< 0} P] I2 N2 I3 R1 N3 N4 [*|R2 *]
{if the 4th DOS parameter is 'p'}

N1 is the DOS name of an experimental file which is supposed to be in %ZitaEXPR% library. The number of the experimental time values is limited up to 15400 within an experimental data file. Since JC.EXE uses dynamic variables, this maximum can be less if the user's computer has not got enough memory. JC.EXE always writes the actual value of this limit to the screen at the beginning of a run. Using extended type for real numbers, several thousand (let me say ~8000) data pairs can belong to an experimental curve if the computer has a usual configuration. Changing the type of real numbers to less accuracy (double, real or single) allows more data pairs within a curve on the same computer.

I1 is the ordinal number of that column in N1 file, the experimental data of which should be used.

I2 is the number of those lines at the beginning of the experimental file which must be skipped (these lines contain the initial conditions). If I2 is negative, its absolute value is used. I2 cannot be lower than -30000!

N2 is the DOS name of that file which contains the data calculated with the original parameter values.

I3 is the ordinal number of that column in N2 file, the calculated data of which should be used.

R1 is a weighting factor which is an element of $\mathcal{W}$ matrix (defined in *Eq. (2.5)*). Consequently, weighting is possible only curve by curve so the same weighting factor has to belong to each point of a given curve.

N3 is the DOS name of that file, data of which have been calculated by increasing the actual parameter value to be fitted.

N4 is the same as N3 but the actual parameter value was decreased during the simulation.

R2 is the ratio of errors of experimental time and measured characteristic. It is taken into consideration only in the case of orthogonal fitting. If it is not indicated in the file, its value is supposed to be 1.0. The exact definition of R2 can be found in *Appendix A*.

P includes the parameter numbers for which the calculated analogous curves of N1 have not been simulated with decreased and/or increased parameter values. If the 3rd DOS parameter is in the list defined by P, the actual parameter does not have influence on the actual curve so there are not files for calculating the element of J*.COL. Therefore, all the required elements in J*.COL file will be zeroes without any further consideration. For a more detailed explanation, see the keystring EXPERIMENTAL␣CURVES in *Section 12.1.*

The number and the sequence of the columns and the experimental points must be the same in all the three kinds of calculated curves.

If an experimental curve contains several measured characteristics to be used, the data of the appropriate files should be given in more than one line. Only the value of I1 and I3 will be the difference in these lines.

JC.EXE writes the results into J*.COL files. *Text 10.1* shows an example for the structure of *.COL files. It includes the contents of J.COL and J1.COL files which were calculated in the first iteration of example 1.

J.COL stores the elements of $\left(\vec{Y} - \vec{F}(\vec{k}_i)\right)$ vector. In this file there are five important data preceding the elements of every curve:

- The first one is the ordinal number of the actual curve in *.JCD file.
- The second one is the number of the points in the actual curve.
- The third one is the square root of $S(\vec{k})$ only regarding to the actual curve without the weighting factor.
- The fourth one is the weighting factor of this curve.
- The fifth data is the filename of the used experimental file. This is only a useful remark for the user.

There are some more lines at the end of the file. The first of them includes the number of points and the square root of $S(\vec{k})$ regarding to all the curves without the weighting factor. The rest of these lines contain some statistical data and the *.COL files pass them on. They were calculated by JC.EXE but GNM.EXE uses and interprets them. Their explanations are in the next section.

The Jm.COL file stores information for that column of $\boldsymbol{\mathcal{J}}(\vec{k}_i)$ matrix which belongs to the m^{th} parameter. The value of m is not necessarily the ordinal number of the column in question, because there can also be fixed parameters. In the followings the m^{th} parameter is supposed to be identical with the p^{th} fitted parameter. The j^{th} numerical value in the Jm.COL file equals to the following approximation:

No.	J.COL	J1.COL
1:	1 31 3.78038335737347199E-1 1.00000E+0 exlv1.exp	
2:	0.00000000000000000E+0	0.00000000000000000E+0
3:	-2.69489415117479570E-2	8.59651439619242269E-2
.	.	.
.	.	.
.	.	.
31:	3.66985991357675231E-1	-6.14622501319876017E-3
32:	5.01020129134350795E-1	-4.43540553632493918E-3
33:	2 31 3.68373564889013006E-1 1.00000E+0 exlv1.exp	
34:	0.00000000000000000E+0	0.00000000000000000E+0
35:	-1.16280990736524847E-3	2.73995785300126119E-4
.	.	.
.	.	.
.	.	.
63:	-1.61921435737914973E-3	8.56236829062894055E-5
64:	8.09355344328085777E-4	-4.02934978382538018E-6
65:	3 37 3.65191181602085633E-1 1.00000E+0 exlv2.exp	
66:	0.00000000000000000E+0	0.00000000000000000E+0
67:	-1.88092024420773634E-3	6.03700607145778328E-3
.	.	.
.	.	.
.	.	.
101:	9.83644109521116433E-3	7.55670051580498136E-2
102:	7.23605294456313797E-3	1.42739886552820601E-1
103:	4 37 3.27124645015233925E-1 1.00000E+0 exlv1.exp	
104:	0.00000000000000000E+0	0.00000000000000000E+0
105:	-4.79072306813839160E-4	2.04126934501976717E-5
.	.	.
.	.	.
.	.	.
139:	-1.58699709990168790E-3	-2.62439195458041364E-3
140:	-1.51676110427824776E-3	-9.77087593149461753E-4
141:	0	
142:	Number of points = 136 Fitting = 3.59031109451128673E-1	
143:	Unweighted Fitting = 3.59031109451128673E-1	
144:	Correlation = 1.44345838962870225E-1	
145:	Coeff. of det. = -4.60279583875257259E-1	
146:	R-squared = 3.01321361089838711E-2	
147:	First term of MSC = -3.78627913192234557E-1	
148:	Serial Correlation = 1.04588358767441483E+1	
149:	Heteroscedactivity = 8.98770407162066702E-1	
150:	Skewness = -3.86426595284982297E+0	
151:	Kurtosis = 4.78309740928028529E+0	

Text 10.1: Main parts of J*.COL files in the first iteration of Example 1.

$$k_{i,p}\frac{\Delta\vec{F}_j(\vec{k}_i)}{\Delta k_{i,p}} \qquad \left(\approx k_{i,p}\frac{\partial\vec{F}_j(\vec{k}_i)}{\partial k_{i,p}}\right) \tag{10.1}$$

where

$\vec{F}_j(\vec{k}_i)$ is the j^{th} number in the $\vec{F}(\vec{k}_i)$ vector.

$k_{i,p}$ is the value of the p^{th} fitted parameter in the i^{th} iteration.

In Jm.COL files there is an empty line between the data of the different curves. It only makes easier studying these data.

10.2 Usage of GNM.EXE

This program can be used if all the components in the right side of *Eq. (2.5)* (page 20) have already been calculated: $\vec{k}_i$ is in PARAMTRS.DAT, $\boldsymbol{\mathcal{W}}$ and $\left(\vec{Y}-\vec{F}(\vec{k}_i)\right)$ are in J.COL and $\boldsymbol{\mathcal{J}}(\vec{k}_i)$ is in J*.COL files. GNM.EXE applies *Eq. (2.5)* to calculate the new parameter vector.

Besides them, GNM.EXE requires %ZitaTEMP%\ODES.PAS or the appropriate *.ODE file of the actual ZITA.EXE. This is because the features of the actual model are stored in these

files so GNM.EXE can read them from the files in question. That is why these files must not be erased after the compilation!

10.2.1 Structure of PARAMTRS.DAT file

PARAMTRS.DAT contains the parameter values of the chemical model and GNM.EXE. The structure of this file is the following if the actual model contains p parameters:

```
R1_1  [ |C_1]  *
R1_2  [ |C_2]  *
 ⋮     ⋮
R1_p  [ |C_p]  *
R2    R3       *
R4    R5       R6  *
I   *
```

where

$R1_i$ is the value of the i^{th} parameter.

The parameter values can be either positive or negative numbers. *The change of their signs are not allowed* during the calculations since GNM.EXE calculates the better approximations from relative changes. In this way, the rate constants are always positive numbers, however, the sign of other fitted parameters may be any of the two ones. If the sign of a parameter can be reversed during fitting, **Transform** procedure must be used in order to change the parameter value into positive or negative. In this case, the statistical description regards to the fitted value and not to the original one! The standard deviation must be recalculated manually for the original parameter as well. To take the error propagation into consideration, the user must apply the following general formula [22]0:

$$S_Z = \sqrt{\sum_{i=1}^{k} \left(\frac{\partial Z}{\partial X_i}\right)^2 S_{X_i}^2 + 2\sum_{j=2}^{k}\sum_{i=1}^{j-1} \frac{\partial Z}{\partial X_i}\frac{\partial Z}{\partial X_j} V_{X_i X_j}}$$

where

$X_1, X_2, \ldots, X_k$ are random variables.

Z is an arbitrary function of k random variables: $Z = f(X_1, X_2, \ldots, X_k)$. So, Z is also a random variable.

S_Z and S_{X_i} are the standard deviation of Z and X_i random variables (see *Eq. (10.3)*).

$V_{X_iX_j}$ is the *sample covariance* defined by the following formula for n elements of both X_i and X_j:

$$V_{X_iX_j} = \frac{1}{n-1}\sum_{m=1}^{n}\left(X_i^m - \frac{1}{n}\sum_{q=1}^{n}X_i^q\right)\left(X_j^m - \frac{1}{n}\sum_{q=1}^{n}X_j^q\right)$$

This numerical value can be calculated easily from the last `RESULT.*` file, because this file includes the total correlation coefficient (S_{ij}) of X_i and X_j and *Subsection 10.2.3* gives the appropriate formula for this calculation.

For example, if a measured characteristic equals to $p_i + p_j \cdot \ln(y_1(t))$ where $-100 \leq p_i \leq -10$ and $-30 \leq p_j \leq 50$ from previous considerations, the following assignment statement should be given in `*.ODE` or `ODES.PAS` file:

```
tr[i]:=-p[i]+(p[j]-100)*Ln(y[1]);
```

This is because the parameter values to be fitted are within (10–100) in the case of p_i and (70–150) in the case of p_j.

`Ci` can be a printable character or can be omitted. If it is an '`*`', the `i`$^{\text{th}}$ parameter value is fitted, otherwise, the value of the parameter in question is fixed. *Any remark can follow* `R1` *or* `C`*, however, this remark cannot be interpreted as a number*. The number of parameters is limited by only the memory size because `GNM.EXE` uses dynamic variables. A fixed parameter requires only a few bytes in the memory. The number of fitted parameters can be ∼80 in practice but using single type for real numbers it can be ∼200 (however this setting is not recommended!).

`R2` is the value of Marquardt-parameter (λ) in the actual iteration. The choice of the initial value of λ is based on experience, does not influence the results and has a minor influence on running time. The initial value is recommended to be 0.01or less. The actual λ decreases in the ratio $\lambda_{i+1}/\lambda_i = 0.1$ between two successive iterations if the function value of $S(\vec{k})$ (defined in *Eq. (2.3)* on page 18) decreases. λ does not change if the function value of $S(\vec{k})$ increases. λ increases only in that case if there would be a negative element among the new predicted values of parameters. In this case λ is duplicated and the new parameters will be calculated again.

λ can also be a negative number. In this case, `GNM.EXE` uses its absolute value, however, the value of λ remains the original one through the iterations. This possibility can be used for very sensitive models or making the refinements by little steps and many iterations.

`R3` is a real number the value of which is greater than 0.0 but less than 1.0. It is called acceptable relative step. This number controls the changes of the parameter values. The next paragraph describes how to use it.

The program always examines the new parameter values. If a negative one is found, it can be reversed in two ways. One hand, the changes of the parameters can be decreased proportionally regarding to all the parameters. On the other hand, they

can be calculated again with a greater value of λ. The program works with the combination of these two ways. After calculating the new parameter values, the program determines that positive number (PN) by which the changes of all the parameter values can be multiplied in order to make all the new parameter values positive. If PN equals to or is greater than 1.0, the convergence is guaranteed. If PN is less than `R3`, the program duplicates the actual λ and recalculates the new parameter values. If PN is between `R3` and 1.0, the program calculates the new parameter values on the basis of the first way mentioned above. The recommended value of `R3` is 0.99 if the actual model can already describe the experimental curves approximately.

`R4` is a negative real number if the first iteration is to be carried out and it is the standard deviation of data (see below in detail) belonging to the previous iteration in all the other cases. The weighting factors contained in `J.COL` file are taken into account only in iteration formula *Eq. (2.5)* but not during the calculation of this values. It also means that the standard deviation of data can increase temporarily but it does not cause any mistake.

`R5` is the limit of the relative change of $S(\vec{k})$. If the relative change of $S(\vec{k})$ is smaller than `R5`, `GNM.EXE` does not create new parameter vector so the fitting is terminated.

`R6` is the largest norm of parameter values to be allowed for the termination. The norm of parameters is defined in the following way:

$$N_{\vec{k}} = \sqrt{\frac{1}{p} \cdot \sum_{i=1}^{p} \left(\frac{k_{i,j} - k_{i,j+1}}{k_{i,j}} \right)^2} \tag{10.2}$$

where

$N_{\vec{k}}$ is the norm of parameter values.
p is the number of parameter values to be fitted.
$k_{i,j}$ is the value of the i^{th} fitted parameter after the j^{th} iteration.

`I` is the maximum number of the iterations.

10.2.2 Calling `GNM.EXE`

`GNM.EXE` can be called either using DOS parameters or interactively. The following possibilities can be occurred:

- The first DOS parameter is an '`*`'. In this case `GNM.EXE` reads the number of parameters from `ODES.PAS` file. `TASK.EXE` (see *Chapter 12*) uses only this possibility.
- The first DOS parameter is the filename of an `*.ODE` file. In this case `GNM.EXE` reads the number of parameters from this file. This DOS parameter must contain the `ODE` extension because the program recognizes the structure of the file on the basis on it. Therefore, `ODE` extension is compulsory, not only recommended in this case!

- If the first DOS parameter is either an '*' or a filename with the extension ODE, the second DOS parameter can be given as well. If it does not exist, all pieces of information are read from PARAMTRS.DAT file. If the second DOS parameter exists, the program interprets it as the name of that DOS environment variable which contains the ordinal numbers of the parameters to be fitted. If this is not so, GNM.EXE terminates with an error message, otherwise, these ordinal numbers must be separated by space(s) in the environment variable. In this case only the parameter values of the chemical model are read from PARAMTRS.DAT file, C-s are not.

  For example, if the DOS environment variable named DOING contains the ' 3 1 7 4' string then 'Gnm Act.ode doing' command carries out an iteration. The parameter values of the model are read from PARAMTRS.DAT and the program calculates the new values of the 1^{st}, the 3^{rd}, the 4^{th} and the 7^{th} parameters.

- If the first DOS parameter does not exist or it contains neither '*' nor '.ODE' string, GNM.EXE asks the number of the parameters to be fitted, their ordinal numbers, the values of Marquardt-parameter and acceptable relative step interactively. In this case only the parameter values of the chemical model are read from PARAMTRS.DAT file. Since there cannot be an iteration loop in this case, the conditions of termination are not asked by the program.

GNM.EXE writes the new calculated values in %ZitaTEMP%\RES1 file if the conditions of termination are not valid. In this way the result of the last iteration can be found in RES1 file in the user's library. The structure of this file corresponds to that of PARAMTRS.DAT. New actual values are given to the fitted parameter values, R2 and R4 but the other values remain unchanged. R2 will be λ used for the last time so its value approaches 0.0 if the fit becomes better. R4 is read from the actual J.COL file. If input R4 is negative, all RESULT.*  (described in this section later) files in the user's library are erased before the calculation. These files may contain important results from the previous fitting. If the user wants to keep them, they should be renamed before the new fitting!

Fitting is finished if *the relative change of* $S(\vec{k})$ *is less than* R5 *or the change of* $N_{\vec{k}}$ *is less than* R6 in two successive iterations. In this case, the program erases RES1 file in the user's library if it exists. Consequently, if these files cannot be found in the user's library then fitting is finished and the last iteration gives the final results. The automatic fitting can be organized by the help of this feature of GNM.EXE because the 'EXIST' DOS condition (see it in [2]) is able to examine whether a file exists or not. The value of λ decreases then, and only then, if $N_{\vec{k}}$ is less than 0.1. This way, the extreme changes of the parameter values can be avoided.

10.2.3 Interpretation of RESULT.* files

The result of the i^{th} iteration is stored in RESULT.i file where i is an integer number between 1 and 999. If its value equals to 1, R4 is a negative number. In this case, the existing RESULT.* files are erased and the result will be stored in RESULT.1 file. If R4 is positive, the program searches for that RESULT.k file where k has the largest value among

the existing `RESULT.`i files. The new results are stored in `RESULT.`k+1 file. So, the results of the successive iterations can be attained in `RESULT.1`, `RESULT.2`, etc. files. Consequently, maximum 999 iterations are possible during a fitting[2].

A `RESULT.*` file contains the next approximation of the parameter values and their statistical interpretation after the actual iteration. This file gives every numerical value with the most possible digits, however, it is the user's responsibility to know how many digits have real meaning! Every `RESULT.*` file includes all statistical parameters, but some of them are allowed to be interpreted after the last iteration only. The user should also keep this fact in her/his mind!

Since these files are the most important ones during fitting, their structure is detailed very deeply through the next few paragraphs. To measure relatively huge sets of data and to calculate their statistical description besides the parameter estimation is simply essential for the most reaction models. Unfortunately, this fact is not generally accepted in practice! To emphasize, how much important it is, a paragraph from [20, Chapter 14: Modeling of Data] is quoted:

> 'The important message we want to deliver is that fitting of parameters is not the end-all of parameter estimation. To be genuinely useful, a fitting procedure should provide (i) parameters, (ii) error estimates on the parameters, and (iii) a statistical measure of goodness-of-fit. When the third item suggests that the model is an unlikely match to the data, then items (i) and (ii) are probably worthless. Unfortunately, many practitioners of parameter estimation never proceed beyond item (i)! They deem a fit acceptable if a graph of data and model "looks good." This approach is known as *chi-by-eye*. Luckily, its practitioners get what they deserve.'

The `RESULT.*` files consist of several groups. Each group is introduced with a leading '`********`' string and the name of the group follows this string. The only exception is the first row which summarizes the most important pieces of information belonging to the actual iteration. For describing the groups, the following abbreviations are introduced in the rest of this section:

- n is the number of the elements of $\vec{Y}$ or $\vec{F}(\vec{k_i})$ vectors defined in *Eq. (2.5)* on page 20. It also means the number of diagonal elements of $\boldsymbol{\mathcal{W}}$ matrix also defined in *Eq. (2.5)*. In simpler term, n is the total number of measured data pairs.
- p is the number of fitted parameters.
- m_i is the i^{th} element of $\vec{Y}$ vector, namely the i^{th} measured data.
- c_i is the i^{th} element of $F(\vec{k_i})$ vector, namely the i^{th} calculated data.
- w_i is the i^{th} element of $\boldsymbol{\mathcal{W}}$ matrix, namely the i^{th} weighting factor. The definition of weighting factor usually concerns the square of residuals. In this book w_i is interpreted in a slightly different way since it concerns the i^{th} residual itself. In principle, the

[2]The writer hopes, it will be enough for all the users!

user must have several repetitions of the same measurements in order to estimate the values for the individual weighting factors. This condition is fulfilled in kinetics very rarely. Therefore, the order of magnitude of the weighting factors comes from the measurements and the underlying chemical model. A value depending linearly on the experiments and model can be handled easily. This idea introduced the modified definition.

Beside the ordinal number of the iteration, the first line of a `RESULT.*` file includes the Weighted Standard Deviation of Data (`WSTD`), the Number of Measured Data (`NMD`) and the value of Marquardt-Parameter (`MP`), respectively. These pieces of information can also be found in one of the following groups.

The '**Initial Parameter Value(s)**' group consists of that parameter values by the help of which the calculations in the actual iteration were done. This group can only be found in `RESULT.1` file. The actual initial values for the fitted parameters and the results of the previous iteration are identical so these initial values for the $(j+1)^{\text{th}}$ iteration can be found in `RESULT.`j file. The values of the fixed parameters are constants during a run, so they can be found in `RESULT.1` file.

The '**Result(s)**' group gives the new estimation for the fitted parameter values, their absolute standard errors, their relative errors expressed in percent and the change of the parameter values between the previous and actual iteration.

The '**Goodness-of-fit Statistics**' group gives some useful statistical parameters. They help to assess whether the actually used model is appropriate or not. Nowadays, most of these parameters are becoming important goodness-of-fit indicators in the field of modeling data.

- The *Marquardt-Parameter* has already been detailed on page 125 where `R2` was defined.
- The actually used value for the *Relative Step* is also given in this group. The meaning of it is detailed on page 125 under the definition of `R3`.
- The *Number of Measured Data* always means the total number of the observations expressed in data pairs. For example, if a row in an experimental file contains a time value and three measured characteristics, they are considered as three data pairs. This parameter does not include the missing values (see *Section 8.2*). The value of this parameter is denoted by n in this section.
- The *Number of Fitted Parameters* has an obvious meaning so it does not require more explanation. The values of these parameters are followed by an '`*`' in the `PARAMTRS.DAT` file. The value of this parameter is denoted by p in this section.
- The *Weighted Standard Deviation of Data* is a well-known and probably the most useful statistical parameter. It is defined by the following formula:

$$\sqrt{\frac{\sum_{i=1}^{n}(w_i(m_i-c_i))^2}{n-p}} \tag{10.3}$$

The interpretation of this parameter depends on the method for calculating differences (see *Subsection 10.1.1*). It can be considered as the average of the residuals. This average can have a value carrying either absolute, relative or orthogonal meaning. The absolute and relative differences are explained in the previous section while the orthogonal one is detailed in *Appendix A*. If there are more measured characteristics to be fitted together and their weighted means are different from each other to a larger degree than an order of magnitude, the usage of absolute fitting is not recommended.

The *Unweighted Standard Deviation of Data* can be interpreted in the same way but all the w_i values are supposed to be 1.0.

- The *Correlation between Measured and Calculated Values* expresses, how much the changes regarding the measured data are related to the changes in their calculated analogous. The closer to unity the absolute value of this parameter is, the higher the correlation between the measured and calculated data is. This parameter is defined by the following formula:

$$\frac{\sum_{i=1}^{n}\left(\left(w_i m_i-\frac{1}{n}\sum_{j=1}^{n}w_j m_j\right)\left(w_i c_i-\frac{1}{n}\sum_{j=1}^{n}w_j c_j\right)\right)}{\sqrt{\sum_{i=1}^{n}\left(w_i m_i-\frac{1}{n}\sum_{j=1}^{n}w_j m_j\right)^2}\sqrt{\sum_{i=1}^{n}\left(w_i c_i-\frac{1}{n}\sum_{j=1}^{n}w_j c_j\right)^2}} \tag{10.4}$$

- The *Coefficient of Determination* (CoD) is a more appropriate measure of the goodness-of-fit than the correlation. It is defined by the following formula:

$$1.0-\frac{\sum_{i=1}^{n}\left(w_i\left(m_i-c_i\right)\right)^2}{\sum_{i=1}^{n}\left(w_i m_i-\frac{1}{n}\sum_{j=1}^{n}w_j m_j\right)^2} \tag{10.5}$$

This statistical parameter is a measure of that fraction of the total variance ($\sum_{i=1}^{n}(w_i m_i-(\sum_{j=1}^{n}w_j m_j)/n)^2$) the fitted model accounts for which. The largest numerical value for this parameter is 1.0. The closer to unity this value is, the larger the fraction of the total variance explained by the model is.

- The *R-squared* has a very similar meaning to that of the Coefficient of Determination. This parameter operates with the sum of the squares of the weighted measured data (SSWMD) instead of the total variance. The upper limit is 1.0

for the numerical value of this parameter. The closer to unity this value is, the larger the fraction of SSWMD described by the model is. The defining formula for R-squared is the following:

$$\left| 1.0 - \frac{\sum_{i=1}^{n} \left(w_i \left(m_i - c_i \right) \right)^2}{\sum_{i=1}^{n} \left(w_i m_i \right)^2} \right| \tag{10.6}$$

- The *Model Selection Criterion* (MSC) is a modification of the Akaike Information Criterion [21]. It attempts to represent the "information content" of the fitted parameter values by relating the CoD[3] to the ratio of p and n. The following formula defines this statistical parameter:

$$\ln \left(\frac{\sum_{i=1}^{n} \left(w_i m_i - \frac{1}{n} \sum_{j=1}^{n} w_j m_j \right)^2}{\sum_{i=1}^{n} \left(w_i \left(m_i - c_i \right) \right)^2} \right) - \frac{2p}{n} \tag{10.7}$$

  The MSC is useful for comparing more models with different numbers of fitted parameters. Models including more fitted parameters usually have a better Coefficient of Determination, however, they do not often give a better description of the observations. The MSC tries to quantify how much better the CoD must be for the really more appropriate model. MSC is supposed to be proportional to the "information content" of the fitted parameter values so the largest MSC value should belong to the most acceptable model. Since the MSC has normalized numerical value and it is independent of the scaling of the data points, the values of MSC derived from different models are truly comparable. With experience, MSC seems to be the most useful goodness-of-fit parameter for evaluating the suitability of different models.

The '**Residual Analysis**' group includes such kind of statistical parameters that measure the deviations of residuals from the Gaussian (normal) distribution. Their numerical values are meaningful only for huge sets of observations (let's say, more than 1000 data pairs). For a smaller set, they must be used with caution or not at all!

All the following statistical parameters are normalized and their expected value is zero. Their numerical values are given in units of standard deviation.

- The *Serial Correlation* indicates whether the residuals approximate to be distributed randomly around zero or they tend to form groups of positive and negative values. A large positive value of this statistical parameter warns the user for

[3] See the definition of the *Coefficient of Determination*.

the existence of systematic deviation from the normal distribution. It is defined by the following formula:

$$\sqrt{n-1}\frac{\sum_{i=1}^{n} w_i(m_i - c_i)w_{i-1}(m_{i-1} - c_{i-1})}{\sum_{i=1}^{n}(w_i(m_i - c_i))^2}, \qquad \begin{matrix} m_0 = m_n \\ c_0 = c_n \\ w_0 = w_n \end{matrix} \tag{10.8}$$

- The *Heteroscedactivity* indicates whether the magnitude of the residuals correlates with the magnitude of the measured values to be fitted. The larger the absolute value for this statistical parameter is, the smaller the correlation is. The following formula defines the heteroscedactivity:

$$\frac{(n-1)\left(\sum_{i=1}^{n} w_i m_i\right)\left(\sum_{i=1}^{n}(w_i(m_i - c_i))^2\left(w_i m_i - \frac{1}{n}\sum_{j=1}^{n} w_j m_j\right)\right)}{(n-2)\left(\sum_{i=1}^{n}(w_i(m_i - c_i))^2\right)\left(\sum_{i=1}^{n}\left(w_i m_i - \frac{1}{n}\sum_{j=1}^{n} w_j m_j\right)\right)} \tag{10.9}$$

- The *Skewness* is a measure of the asymmetry for the distribution of the residuals around their arithmetic mean. A positive value signifies that an asymmetric tail extends out towards more positive residual values. In the case of a negative value, this tail extends out towards more negative residuals. The following formula defines this normalized statistical parameter:

$$\frac{\sum_{i=1}^{n}(w_i(c_i - m_i))^3}{\sqrt{6n\left(\frac{1}{n-1}\sum_{i=1}^{n}(w_i(m_i - c_i))^2\right)^3}} \tag{10.10}$$

- The *Kurtosis* is a measure of the flatness for the distribution of the residuals. A positive value signifies that the peak of this distribution is sharper than that of the normal distribution. A negative value indicates that the distribution is more flat than the Gaussian one. The definition for this normalized statistical parameter is

$$\sqrt{\frac{n}{24}}\left(\frac{\sum_{i=1}^{n}(w_i(m_i - c_i))^4}{n\left(\frac{1}{n-1}\sum_{i=1}^{n}(w_i(m_i - c_i))^2\right)^2} - 3\right). \tag{10.11}$$

The '**Correlation Matrix (rij\Ri\Sij)**' group gives the correlation coefficients arranged in matrix form. This matrix provides information for the extent of the mutual dependence on parameters. The diagonal elements contain the multiple correlation coefficients (R_i), the lower triangle contains the partial ones (r_{ij}), while the upper triangle contains the total correlation coefficients (S_{ij}). The values of these coefficients are the normalized values of the variance matrix [9, 20], and they can be between -1.0 and 1.0. The larger the absolute value of a coefficient is, the larger the interdependence between the parameters is. A high absolute value (>0.95) indicates that the measured data includes insufficient information for determining both parameter values accurately. A zero value indicates total independence.

- The *partial correlation coefficient* gives the measure of interdependence between two parameters assuming that the other parameters have fixed values.
- The *total correlation coefficient* gives the measure of interdependence between two parameters assuming that the other parameters have fitted values.
- The *multiple correlation coefficient* gives the measure of interdependence of a given parameter from that of all the others.

The partial, total and multiple correlation coefficients are calculated from the symmetrical Variance-Covariance Matrix ($V = \boldsymbol{\mathcal{J}}^T(\vec{k}_i) \cdot \boldsymbol{\mathcal{W}}^2 \cdot \boldsymbol{\mathcal{J}}(\vec{k}_i)$) by the $-V_{ij}/\sqrt{V_{ii}V_{jj}}$, $V_{ij}^{-1}/\sqrt{V_{ii}^{-1}V_{jj}^{-1}}$ and $(1 - 1/(V_{ii}V_{ii}^{-1}))$ formulae, respectively. This matrix is not indicated in the `RESULT.*` files, because the ordering of this matrix is identical to that of the correlation matrix, but the values of the Variance-Covariance Matrix are not normalized. However, the user can calculate the elements of this matrix by the following formula:

$$V_{ij} = \frac{S_{ij}}{\texttt{WSTD}^2} S_{X_i} S_{X_j}$$

where

`WSTD` was defined earlier in this subsection.
S_{ij} is the appropriate total correlation coefficient from the last `RESULT.*` file.
S_{X_q} is the absolute standard deviation of the q^{th} fitted parameter.

This calculation can only be necessary if a transformed parameter value is fitted (see the definition of `R1` in *Subsection 10.2.1*).

The '**95% Univariate** and **Supporting Plane Confidence Interval(s)**' groups give that boundaries of regions in which the true value for the estimated parameter can be found with 95% probability. The calculation of both confidence intervals assumes that the parameters behave linearly near the minimum of $S(\vec{k})$ defined in *Eq. (2.3)* on page 18. The difference between these two confidence regions is their underlying assumption. For the Univariate Confidence Interval, there is no correlation supposed between the parameter in question and the others. For the Supporting Plane Confidence Interval, the boundaries are calculated for a joined confidence region because

the correlation between the parameters are taken into consideration. The algorithm of their calculation can be found in [20, Chapter 14]. The following expression is used for the calculation of confidence boundaries of a fitted parameter denoted by P:

$$P \pm s_P \cdot \sqrt{\frac{1 - I_{0.95}\left(\frac{n-p}{2}, \frac{N_{\text{fit}}}{2}\right)}{I_{0.95}\left(\frac{n-p}{2}, \frac{N_{\text{fit}}}{2}\right)} \cdot \frac{n-p}{N_{\text{fit}}}}$$

where

s_P is the standard error of P.

N_{fit} is 1 in the case of Univariate Intervals. For Supporting Plane Intervals, this is the number of fitted parameters (p).

$I_{0.95}$ is the incomplete β-function.

The last `RESULT.*` file can include an additional part. If at least one of the conditions for normal termination is fulfilled, the last file contains the reason of the termination.

If an iteration is ready, a new one can be started. To reach it, `RES1` file containing new parameters can simply be copied into `PARAMTRS.DAT`.

Appendix E contains the results of the main iterations of the examples.

10.3 Automatic Fitting

On the basis of the facts detailed until now the automatic fitting can be organized into a DOS command file [2]. *Chapter 12* details a more effective method for automatic fitting but that method requires keeping of some conventions and restrictions while a DOS command file can be used at any time for anything. Very special problems can also be handled by them. This section analyzes `MODEL?.BAT` files of the 6th example in detail because they, themselves are well-organized DOS command files. This example is described in *Chapter 7*. Now the following paragraphs describe (by analyzing `MODEL1.BAT` file) the technique how to carry out fitting.

`MODEL?.BAT` files carry out the fitting of three models one by one. The initial parameter values are stored in `PARAM?.DAT` files.

Typing `'Model1'` command, fitting starts in the user's library after creating *ODEs*-solver. The run can be interrupted by pressing **CTRL** and **BREAK** keys together at any time. The run can be continued by `'Model1'` command if the user has not modified the files created during fitting.

An iteration loop for fitting is carried out in the 5-37th lines of any `MODEL?.BAT` file.

Line 6 sends a message onto the screen to indicate the ordinal number of the fitted model.

Line 8 creates the curves calculated by the exact parameter values, lines 14, 20 and 26 create the curves calculated by the modified parameters.

Line 11 creates the $\left(\vec{Y} - \vec{F}(\vec{k}_i)\right)$ vector and lines 17, 23 and 29 create the $\boldsymbol{\mathcal{J}}(\vec{k}_i)$ matrix.

The new parameter values are calculated in line 32.

Lines 9, 12, 15, 18, 21, 24, 27, 30 and 33 handle errors. If an error occurs during the run of any executable program, the commands in these lines jump out from the iteration loop.

Lines 7, 10, 13, 16, 19, 22, 25, 28, and 31 send a message onto the screen in order to tell what program is working at present.

Line 34 examines whether `RES1` file exists or not. If it does not exist, fitting of the actual model is finished, so the iteration loop should be left.

The maximum number of the iterations is determined in line 35. The iteration loop is left if there are 20 files named `RESULT.*`.

Lines 5, 36 and 37 realize the iteration loop. If the conditions for leaving the loop are not fulfilled, line 36 copies the new parameter values into `PARAMTRS.DAT` and line 37 jumps to the label at the beginning of the loop.

After this, the role of lines 3–4 can be understood. These lines copy the initial values of the parameters into `PARAMTRS.DAT` if the first iteration is not ready. In all the other cases `PARAMTRS.DAT` contains the parameter values calculated for the last time.

Lines 38–43 examine the reason for jumping out of the iteration loop. Line 39 examines whether the user pressed the **CTRL** and **BREAK** keys together or not. If he did so, fitting is terminated because this line jumps to the end of the actual `MODEL?.BAT` file. If the keys were not pressed, the reason for the error is stored in `ERRORS` file and the next `MODEL?.BAT` file will fit the next model. Line 40 renames the result files in order to keep their contents. Line 41 saves the parameter values used for the last time. Lines 42 and 43 examine whether at least one successful iteration is ready or not. If there is not any `RESULT.*` file, an error message will be stored in that file which should contain the result of the first iteration. In this case this file is created by line 43. It is necessary for preventing the programs from trying to solve a wrong task again and again. Such a phenomenon could occur after interrupting and restarting fitting.

Lines 45 calls the next task for fitting but this line of `MODEL3.BAT` sends a message to the screen. It says that all the tasks are ready.

After this, the role of line 2 is understandable. This line examines if the given task is ready or not. If it is ready, the next task is called, otherwise, the calculation of the given task goes on.

Chapter 11

Graphic Representation of Data

DRAWER.EXE can draw two dimensional figures. It can display maximum 2000 curves in a figure (*but 60 of them may cause a total chaos!*). A curve to be plotted presents the data of a column in a file as a function of the other column in the same file. This program can be used independently from the other parts of ZiTA so the user has a fast, simple and well-used drawing program if he is familiar with handling of it. If the ZitaTEMP and ZitaBASE environmental variables (see *Section 6.1*) do not exist, the ERRORS file will not be created or the existing one will not include the error messages. They will be indicated only on the screen.

Section 6.1 details those DOS environment variables which are used by DRAWER.EXE. That section also enumerates the usable of graphic cards, their graphic resolutions and colors which can be handled by DRAWER.EXE.

There are two kinds of input file. The first one creates figures from the data of the other ASCII-files and the second one displays those figures which were created previously by DRAWER.EXE and were stored in a file using a special form.

11.1 Creating Figures Based on Data from ASCII-files

The input data can be found in a file with the suggested extension DRW. The first DOS parameter of the program is the name of this file with or without extension. If the DOS parameter is not given, the program asks for it interactively. The input data of a figure should be stored in this file based on the following syntax:

```
S1(159)
S2(110)
S3(110)
I1 I2 [ | C1  * ]
```
$Curve_1$
$Curve_2$
$\vdots$
$Curve_{I1}$
Range

```
C4  *
```

where *Curve*$_i$ denotes the ith individual curve to be plotted and describes its drawing method. *Range* determines the types and ranges for the axes.

Curve$_i$ has one of the following syntaxes:

Syntax A:
```
Ni I3i I4i I5i I6i{= 0} [ | C2i | C2i I11i   * ]
```

Syntax B:
```
Ni I3i I4i I5i I6i{= 1} S4i(16)   *
```

Syntax C:
```
Ni I3i I4i I5i I6i{= 2} S4i(16) [ | I10i]  *
```

Syntax D:
```
Ni I3i I4i I5i I6i{= 3 or 4} S4i(16) [ | C2i | C2i I11i   *]
```

Range must have one of the following syntaxes:

Syntax A:
```
I7 {= 0} [ |I8|I8 I9 *]
```

Syntax B:
```
I7 {= 0} [ |I8|I8 I9 *]
C3 *
```

Syntax C:
```
I7 {= 1} [ |I8|I8 I9 *]
R1 R2 R3 R4 [ |R5|R5 R6 *]
C3 *
```

Syntax D:
```
I7 {= 1} [ |I8|I8 I9 *]
R1 R2 R3 R4 [ |R5|R5 R6 *]
```

Syntax E:
```
I7 {= 2} [ |I8|I8 I9 *]
R1 R2 R3 R4 [ |R5|R5 R6 *]
R7 I12 R8 R9 I13 R10 *
C3 *
```

Syntax F:
```
I7 {= 2} [ |I8|I8 I9 *]
R1 R2 R3 R4 [ |R5|R5 R6 *]
R7 I12 R8 R9 I13 R10 *
C3 *
```

The meaning of the symbols are the followings:

`S1`, `S2` and `S3` are the main title, the title of X axis and the title of Y axis. If the user does not want to define a title, an empty line must be in the file instead of it.

The main title can includes up to 159 characters. The sub- (see the DOS parameters later in this chapter), X- and Y-titles can have up to 110 characters. Lower resolution may not display the whole strings! The number of the displayed characters depends on the type of the graphics card, however, the user can easily determine it through a little experience.

The Y-title has vertical direction while the other titles have horizontal direction on the figures.

I1 is the number of curves to be plotted.

I2 can be 0, 1, 2 or 3. In the case of 0 or 2, a figure appears with a plain background. If I2 equals to 1 or 3, there will be major grids on a figure.

If a figure represents more curves and I2 is 2 or 3, the user is expected to press a key between drawing each curve. In this way, the user can distinguish them more easily. In the case of 0 or 1, all curves are drawn together without any pause.

C1 determines how the program handles the filenames (N) given in the input file.

- If C1 is 'z' or 'Z', the filenames need not include the pathname and DRAWER.EXE searches the file in the following libraries and sequence: %ZitaRD%, %ZitaEXPR%, %ZitaSIMU%, .\EXPR, .\SIMU and the actual library. In this way an *.DRW file can remain unchanged if a library name of ZITA changes.
- If C1 is another character or it is not indicated in the input file, the program supposes that a filename also contains the name of path or the file in question is in the actual library.

After this the data of the curves to be plotted follow. Each line contains the data of one, and only one curve.

N is the name of the file containing points to be plotted. This name may contain DOS wildcards, therefore, the data points of a curve can derive from more files. These files must have the same structure. In this case, DRAWER.EXE collects the necessary data pairs from the data files into a TDRW*.FIG file and the data from this TDRW*.FIG file are plotted. The sequence of the data follows the unsorted directorial sequence of the original data files referred by N.

I3 and I4 are the ordinal numbers of the columns to be plotted in the file called N. DRAWER.EXE plots the data of the I4th column as a function of the data of the I3th column. The value of either I3 or I4 cannot be larger than 65535.

The columns to be drawn can contain such character string(s) which cannot be interpreted as a real number. In this case, the actual data pair is not taken into consideration. This also means that there can be anything behind the data to be drawn in the data files to be plotted. The only restriction is these characters cannot be interpreted as numbers.

I5 expresses how many lines should be skipped at the beginning of the data file. For example, these lines can contain the initial values of the experimental curves. If the data file does not have such data, I5 must be 0.

I6 determines how the program draws the actual curve.

- If I6 is 0, C2 is a character to be plotted. In this case this character appears at the place of all the points to be plotted in the figure. The point in question is the center of the character. If neither C2 nor I11 are given, the smallest available plus sign will appear at the place of all points.

If I11 is given and its value is higher than 1, only every I11$^{\text{th}}$ data pair will be plotted. In this way, overlapping characters can be avoided.

- If I6 is 1, the points are connected with straight lines. In this case S4 determines the pattern of these lines. It can contain only '0' or '1'. Space cannot occur between the characters so S4 is a binary integer number with 16 digits which contains zeros of no value as well. When the lines are being plotted, a point appears at the place of '1'-s and nothing appears instead of zeros.
- If I6 is 2, the same happens as if I6 would be 1 but DRAWER.EXE fits a cubic spline [16]2 to data before plotting and the values of dependent variable are calculated from this spline. In this way the number of the points connected with straight lines can be increased. I10 determines how many points are calculated between two successive data derived from the file named N. In this case N cannot contain more than 1240 data pairs.
- If I6 is 3, DRAWER.EXE puts the center of C2 characters at the place of all points, carries out a linear regression and draws the best fitted straight line with the pattern determined by S4. Its equation appears at the left bottom of the screen in $y = (a \pm s_a)x + (b \pm s_b)$ form. The average of the residuals (σ) and the correlation coefficient (r) are also indicated. Pressing any key, this equation disappears and the program runs away. If I11 is given and its value is higher than 1, only every I11$^{\text{th}}$ data pair will be plotted and takes part in the calculations.

  If C2 is not given, the smallest available plus sign will appear at the place of all points. In this case, I11 cannot get value, so every point must be taken into consideration.

  If X axis has logarithmic scale (see below at the description of I8 and I9), the fitted equation is $y = (a \pm s_a) \lg x + (b \pm s_b)$. In the case of logarithmic Y axis, this equation is $\lg y = (a \pm s_a)x + (b \pm s_b)$. Finally, if both axes are logarithmic, $\lg y = (a \pm s_a) \lg x + (b \pm s_b)$ is the equation in question.
- The meaning and syntax of I6=4 is the same as that of I6=3 but the $y = (a \pm s_a)x$ equation is supposed instead of the $y = (a \pm s_a)x + (b \pm s_b)$ one.

I7 can be 0,1 or 2. If I7 equals to 0, the range of plotting is chosen automatically. If it is 1 or 2 then R1, R2, R3 and R4 are the minimum and maximum values of X axis and the minimum and maximum values of Y axis, respectively. R1 can be either lower or higher as R2, only the equality is not allowed. It is also true for R3 and R4.

If R5 is given, it will be subtracted from every X-value before plotting. R6 is the same as R5 but it regards to the Y-values. R5 and R6 can be important if the user wants to get the same axes range for different parts of a column.

I7=2 means that the user defines the place of ticks labels, ticks and grids explicitly. R7,R8 and I12 are the first and last tick position and the number of tick intervals for X axis, respectively. R9,R10 and I13 have the same meaning but for the Y axis[1].

I8 and I9 determine the scale of the axes. If I8 is 1, the X axis has a logarithmic scale, otherwise, it has a linear scale. I9 is the same as I8 only it regards to Y axis. If an

[1] It must be mentioned that the program sometimes modifies these values in the case of logarithmic axis.

axis has a logarithmic scale and I7 equals to 0, the data to be plotted cannot be 0.0 or negative numbers. If the plotting range is determined by the user, the data can equal to 0.0 or can be less than 0.0, but they are ignored by DRAWER.EXE. If either I8 or I9 is not indicated, their values are supposed to be zero.

C3 makes possible to save figures to files. It can be 's', 'S', 'r', 'R' or it can be omitted.

- If C3 equals to 's' or 'S', the figure is saved into a PCX image file and DRAWER.EXE goes on running without any pause. The saved figure can be displayed again without the original data. This possibility is detailed in *Section 11.4*.

  'r' or 'R' corresponds to 's' or 'S' except that the image is rotated by clockwise 90 degrees before saving. These options can also be achieved from the keyboard by pressing 'S' or 'R' when a figure is on the screen.

  Naturally, most types of screens can also be copied to a printer by the help of an appropriate resident program if the user has one. In this case, it is worth calling the resident program from AUTOZITA.4#1 file. See also *Section 1.1*!
- The figure is held on the screen until pressing key 'N' or 'E'. The first one begins to plot the next figure while the second one immediately leaves DRAWER.EXE.

C4 can be 'y' or 'n'. If it is 'y', the next lines of the actual *.DRW file contain instructions for creating a new figure. These lines also follow the syntax detailed above. If C4 is 'n', DRAWER.EXE is terminated.

Each test and example contains a *.DRW file.

11.2 Useful DOS parameters

DOS parameters can also be used for many purposes:

- The figures indicate the date and time of the creation until the second DOS parameter contains a colon anywhere.
- If the second DOS parameter is 's', 'S', 'r' or 'R', it causes the same effect as if C3 would be one of these letters. The difference is that C3 concerns only the actual figure but the second DOS parameter concerns every figure to be plotted.
- The second DOS parameter can contain an 'l' or 'L'. A real number may immediately follow it (e.g. 'l5.3'). In this case the figure is saved in LaTeX's picture environment format. The width of the graph is indicated after 'l' in inches. This possibility is detailed in *Section 11.3*.
- The second DOS parameter can also contain an 'g' or 'G'. In this case the corresponding GLE input file of the actual figure is generated. This possibility is detailed in *Section 11.3*.

  The 'g' or 'G' may be followed by an 'Xi' and/or 'Yj' where i and j are the number of X- and Y-subticks between two corresponding ticks, respectively (i.e. i and j are less

by one than the number of subranges). For example, '`gx3y6`' means that GLE input file will be created and GLE will make 4 X-subranges and 7 Y-subranges.

- DOS parameters 3–7 are considered as a subtitle. If one or more of these DOS parameters exist, they appear on every figure under the main title.

11.3 Creating Publication Quality Graphs

Although `DRAWER.EXE` is not really intended to make graphs for publication or presentation, it offers two ways for doing this.

11.3.1 LaTeX pictures

By the help of the second DOS parameter (see *Section 11.2*), the figures can be saved in LaTeX source files as pictures.

The name of the LaTeX file is `FIGxxxx.LTX` where `xxxx` is an integer number with leading zeroes. `DRAWER.EXE` searches the first non-existing `FIGxxxx.LTX` file and it will be created.

The width of the created LaTeX picture is given by the real number (in inches) after '`l`'. If this number is not indicated, the default value is 6.0 inch. The ratio of Y and X axes is equal to the ratio of the screen resolution so it can be modified through the `ZiTaDISPLAYXMAX` and `ZiTaDISPLAYYMAX` environment variables (see *Section 6.1*).

The LaTeX output have some advantages (+) and disadvantages (−):

+ It has a human readable format so anyone can modify it easily who is familiar with LaTeX.
+ The picture can be fitted into LaTeX documents. Moreover, this format can be transformed into other widespread image formats (e.g. PostScript, PCX, MSP, BMP) by several utilities. LaTeX and the utilities can be achieved through anonymous FTP (e.g CTAN sites (`ftp.dante.de`, `ftp.tex.ac.uk`, etc.)) and they are usually freeware or public domain programs.
− LaTeX requires huge memory size so pictures cannot be created from large data sets. This restriction can partially be overridden by some versions of LaTeX e.g. `HTEX386.EXE` in emTeX. PostScript drivers can also resolve this problem.
− Only solid lines are available and they are drawn through emTeX's `\special` commands.
− Subtitles are not indicated automatically in LaTeX's pictures.
− Rotated texts are not available.

11.3.2 GLE Graphs

GLE is a program package for producing publication quality graphs and slides. Its input file contains graph descriptor commands and references of the data files. All files are simple

ASCII-ones without any serious size limitation. Several output format are supported, e.g. dotmatrix and laser printers, plotters, PostScript printers, etc. The program package is freeware and can be downloaded through anonymous FTP[2].

`DRAWER.EXE` is able to create input files for GLE. The name of an GLE input file is `FIGxxxx.GLE` where `xxxx` is an integer number with leading zeroes. `DRAWER.EXE` searches the first non-existing `FIGxxxx.GLE` file and it will be created. The creation of GLE input file have some advantages (+) and disadvantages (−):

+ Anyone can modify it easily who is familiar with GLE.
+ Most parameters of a graph are stored in variables at the beginning of `*.GLE` files so they can be changed easily by non-experienced users, as well.
+ The file consists of several graph modules. They make the axes, ticks, titles and curves independently from each other.
+ An optional key module (it is the GLE term for legend) is also generated.
+ There is not any unavoidable limitation about the number of data, in opposite of LaTeX's picture format.
+ If `DRAWER.EXE` plots a curve from more data files (see the explanation of `N` above) or spline is generated (`I6=2`), the data of the curve are written in a new file. Its name is `xxxxyyyy.DAT` where `xxxx` is defined above and `yyyy` is the ordinal number of the curve in the actual `*.DRW` file.
± The default output is supposed to be a PostScript file. The fontnames at the beginning of a GLE input file must be changed in other cases.
− Even GLE output is created, there are some restrictions in GLE and some incompatibilities between GLE and `DRAWER.EXE`:
  − GLE is not able to handle the redundant rows at the beginning of data files.
  − Missing values can be denoted by only an asterisk in GLE.
  − The data cannot be filtered, so `I11` is not supported by GLE.
  − Up to 20 new markers can be defined.
  − Rows in data files cannot be longer than 350–380 characters.

  All of these problems can be solved by reorganizing the data files and/or `*.DRW` file.

11.4 Redrawing Saved Image Files

It was mentioned in the previous section that the drawing program can save the displayed figures. The name of the file to be saved corresponds to the name of the input `*.DRW` file without the extension. Let this name is `XXXX`! `DRAWER.EXE` searches for that `XXXX.`k file where k has the largest value among the existing `XXXX.`i files. The full name of the file to

[2]Sites: `ftp.std.com/ftp/vendors/emagic/gle` and `ftp.rz.uni-duesseldorf.de/pub/graphics/gle`.

be created is `XXXX.`$k+1$ file. In this way, maximum 999 figure can be stored by the help of one `*.DRW` file.

The raster image format for the saved files is the PC Paintbrush (`PCX`) with two colors. The size of the PCX file equals to the resolution of the screen. In this way the figures created by `DRAWER.EXE` can be exported into word processors, transparencies, graphic programs, etc. One may also want to use the `PCXTIMES.EXE` utility for enlarging the PCX-files.

This possibility is also useful, if a figure is made on the basis of temporary data files. For example, the files containing calculated curves are overwritten during each iteration. If the user wants to see these curves in every iteration but s/he does not want to sit in front of the computer for hours, the figures have to be saved.

If the resolution of the screen is the same during saving a figure and redisplaying it, the figure remains the original one. A monitor with lower resolution does not display the right side and the bottom of the original figure. A monitor with higher resolution shows an empty area at the right side and at the bottom of the screen.

If the user wants to see the figures saved by `DRAWER.EXE` again, another kind of the input file has to be used. Every line of this file includes a filename[3] to be redrawn.

How can the drawing program notice the kind of input file? At first, `DRAWER.EXE` examines the DOS parameters. If the extension of the input file is `DRA` or the second DOS parameter equals to '`+`', the drawing program recognizes that the input file contains only names of files including complete figures. In other cases, `DRAWER.EXE` supposes that the input file has the structure detailed in the previous section.

The previously created figures can be handled in the same way on the screen as if they would be created now. They can be printed or saved again but cannot be changed at all.

There is an example in `CSTRFIGS.DRA` file. If the fitting is carried out by the help of '`Cstr`' DOS command, a new figure is created and saved in each iteration. These figures are in `CSTR.00?` files and can be viewed again by the help of '`Drawer Cstrfigs`' DOS command at any time after the fitting without the original data files.

[3] This name can contain the path as well.

Chapter 12

The Most Efficient Fitting Method

All the possibilities of $\mathcal{ZITA}$ can be achieved through the files and the method were described in the previous chapters. The advantages of this method are the followings:

- The process of fitting can be followed clearly because the fitting is divided in parts and the programs of $\mathcal{ZITA}$ work on a part of the fitting independently. In this way the user has full control over the process of fitting.
- Very special problems can be solved because the user can chose filenames, methods, batch programs without restrictions.

However, there are two disadvantages if the user works this method:

- The user has to create a lot of files to solve a chemical problem. It is usually a time-consuming process. Moreover, some files may have very complicated structure.
- If the user wants to carry out the fitting automatically, he has to know the syntax of DOS batch programs [2].

This is why $\mathcal{ZITA}$ contains an organizer program named `TASK.EXE`. If this program is used, the user has to create only the experimental files, `*.ODE` (or `ODES.PAS`) and `*.DRW` files[1]. `TASK.EXE` generates the `PARAMTRS.DAT`, `*.DAT` and `*.JCD` files, replaces the appropriate `*.BAT` programs, can handle more models together and collects the results. To use this program the user has to take into consideration the following restrictions:

- A *basename* has to be chosen for every task to be solved. It contains maximum four alphanumeric characters. This name is denoted with '`name`' in this chapter.
- The experimental data have to be written into files. The extension of these files must be `EXP`. A filename must be started with the basename and must continued the ordinal number of the file. This number always consists of four digits so it must contain the leading zeroes as well. For example, the name of that file is `name0013.EXP` which contains the 13^{th} experimental curve. If the user wants to use experimental files

[1] Naturally, `*.DRW` file is inessential if the drawing program is not used.

having different structures, the basename of them should be different (see in detail under keystring 'OTHER␣TASK'). Later on, name*i*.EXP means the filename of the i^{th} experimental file.

- The programs of ZITA searches the experimental files in a predefined library. The %ZitaEXPR% DOS environment variable contains the name of it. The experimental files have to be moved into this library before any computation.
- The mathematical model of the mechanism has to be written into name.ODE or ODES.PAS file. If it is ready, ZITA.EXE has to be created by the help of COMPILE.BAT DOS command file.
- TASK.EXE requires one additional file with the recommended extension TSK. It contains those pieces of information concerning the actual task which are not given in name.ODE or ODES.PAS file. TASK.EXE generates the necessary files based on this file. Its structure is detailed later in this chapter.
- TASK.EXE generates the necessary files from the actual *.TSK file and carries out the fitting(s). The following name conventions are valid during this process:
  - The name for the calculated analogous of name*i*.EXP file is name*i*.TMP. This file is renamed to name*i*.SIM, name*i*.TMH or name*i*.TML immediately after the simulation. The first one contains the simulated data calculated with the exact parameter values. The second name is used if one parameter value is increased by a chosen percent during the simulation. The last name is used if one parameter value is decreased by a chosen percent during the simulation. name*i*.SIM exists during a whole iteration but *.TM? files are erased immediately after calling JC.EXE with the appropriate DOS parameters.
  - name.DAT and name.JCD are the name of the necessary input files for ZITA.EXE and JC.EXE, respectively. Only one *.DAT or *.JCD file is necessary. This is because TASK.EXE uses that possibility which are detailed in *Section 9.4* and the filenames listed in the previous paragraph do not contain any special mark.
  - More models can be given if the same *.ODE file belongs to each of them. It means that only the PARAMTRS.DAT files of the models differ from each other. The name of these files are PARAM*i*.DAT during the calculation where i is the ordinal number of the i^{th} model.

If a file having a reserved name exists before the calculation, TASK.EXE erases it without any error message!

- If TASK.EXE finishes a task, the results of it are collected into one file. Its name is MODEL*i*.RES in the user's library where i is the ordinal number of the i^{th} model given in the actual *.TSK file. This file contains the date, the time, some characteristics of the fitting, the contents of RESULT.* files, the content of the last PARAMTRS.DAT file and the most important pieces of information from J*.COL files. The collected files are erased after creating MODEL*i*.RES file.

TASK.EXE protects the existing MODEL*i*.RES files from the accidental deletion. For

example, if MODEL2.RES file exists in the user's library and the user starts a new fitting, the second model of the actual *.TSK file will not be carried out because it would erase the old MODEL2.RES file. It means, *the user has to rename or erase the old* MODEL*.RES *files before a new calculation!*

12.1 Structure of *.TSK files

The user can find *.TSK files in examples 7–10. They show almost all the possibilities. Moreover, User's Guide also contains a complete example in *Text 3.21* on page 54. It is suggested to study this text during reading the next few paragraphs! The keystrings (see below) are written with capital letters in *Text 3.21*.

A *.TSK file contain logical units. The first line of a unit must have a leading '+' character and must contain a *keystring*. Keystrings can be written either with capital letters or small ones. A keystring shows what kind of information is stored in the lines after the first one. These data are interpreted by TASK.EXE. If there are further lines, they are disregarded until a new line occurs with a leading '+' character. The syntax of a line containing a keystring is

+S1**keystring**S2

where S1 and S2 are arbitrary strings or they can be omitted. If a keystring occurs several times in the file, only the last occurrence will be valid with the exception of 'MODEL' keystring. The sequence of the keystrings is arbitrary. *A line cannot contain more than one keystring!*

Several keystrings have a default value. It is valid during a calculation if the actual *.TSK file does not contain a new value. If a keystring does not have default value and *.TSK file does not contain it, TASK.EXE terminates with an error message.

This section details every keystring and their explanations in alphabetical order. The names of the keystrings are framed and the necessary spaces are denoted by '␣' character in them. After a name, the user can find the syntax of the lines belonging to this keystring, the default value if it exists and the names of the other related keystrings. These related keystrings also have information regarding to the keystring in question. Finally, a detailed description can be found. However, if the actual keystring was detailed earlier in the Reference Guide, this section only cites it and does not repeat this explanation.

The syntaxes use the notation system given in *Appendix C*. One further abbreviation is also used in this section: X{Y} means X is defined in Y.

ABSOLUTE,␣RELATIVE

Syntax: C *

Default value: 'R'

See also: OTHER␣TASK

C corresponds to the second DOS parameter of JC.EXE. Its description can be found in *Subsection 10.1.1* (page 119). This value is considered only in the main *.TSK file. If the actual *.TSK file is called from another one, C is disregarded.

ACCOMPLISHED␣RANGE

Syntax: I *

Default value: 0

See also: INNER␣POINT

I can be either 1 or 0. If it is equals to 0, the value of I1{UNIT 3.C, on page 111} can be either 3 or 6. If it is equals to 1, the value of I1{UNIT 3.C, on page 111} is 9.

BASENAME

Syntax: S(4)

Default value: 'Base'

S means the basename of the task to be solved. It was defined in this chapter earlier. S can consist of alphanumeric characters only.

CHARACTERISTIC

Syntax:
$I1_1$ $I2_1$ [$|R1_1|R1_1$ $R2_1]$
$I1_1$ $I2_2$ [$|R1_2|R1_2$ $R2_2]$
⋮ ⋮ ⋮
$I1_n$ $I2_n$ [$|R1_n|R1_n$ $R2_n]$
-1

See also: EXPERIMENTAL␣CURVES
LINES␣MUST␣BE␣SKIPPED

This keystring determines which columns of the experimental and calculated files contain the data for *Eq. (2.3)*.

n is the number of measured characteristics to be compared.

I1 corresponds to I1{*Subsection 10.1.2* on page 120}.

I2 corresponds to I3{*Subsection 10.1.2*}.

R1 corresponds to R1{*Subsection 10.1.2*}. If this value is not indicated, it is considered to be 1.0. Since R1 can have different values for each experimental curve, R1 is valid for those curves which do not have another value. See this possibility in detail under 'EXPERIMENTAL␣CURVES' keystring.

R2 corresponds to R2{*Subsection 10.1.2*}. If this value is not indicated, it is considered to be 1.0. Since R2 can have different values for each experimental curve, R2 is valid for those curves which do not have another value. See this possibility in detail under 'EXPERIMENTAL␣CURVES' keystring.

-1 means that there are no more data belonging to this keystring.

CONTENT␣OF␣THE␣RESULT

Syntax: I1
$I2_1$
$I2_2$
⋮
$I2_{I1}$

The purpose of this part is identical to UNIT 2.C so I1 and I2 correspond to I1{UNIT 2.C, on page 108} and I2{UNIT 2.C, on page 108}, respectively.

However, there are some differences:

- The titles of columns cannot be given here because TASK.EXE does not produce formatted files containing calculated data.
- It is not necessary to write every data into independent lines, only the sequence of the data is essential. These data can be placed into lines less than I1+1 if the integer numbers are separated by space(s).

DIFFERENTIATION,␣METHOD

Syntax: C *

Default value: '+'

See also: OTHER␣TASK
PERCENT

C corresponds to the fourth DOS parameter of JC.EXE. Its description can be found in *Subsection 10.1.1* on page 119. This value is considered only in the main *.TSK file. If the actual *.TSK file is called from another, C is disregarded.

EXPERIMENTAL␣CURVES

Syntax: I_1[|W$R1_1$][|R$R2_1$][|T$R3_1$][|PP_1]
I_2[|W$R1_2$][|R$R2_2$][|T$R3_2$][|PP_2]
⋮
I_{ex}[|W$R1_{ex}$][|R$R2_{ex}$][|T$R3_{ex}$][|PP_{ex}]
-1

See also: CHARACTERISTIC

This keystring determines the experimental curves to be used during the fitting. The sequence of the ordinal numbers is arbitrary, but they are sorted in ascending sequence before any computation.

`ex` is the number of the experimental curves to be used for fitting.

$\texttt{I}_i$ is the ordinal number of the i^{th} used experimental file ($1 \leq \texttt{i} \leq \texttt{ex}$). The leading zeroes can be omitted. For example, $\texttt{I}_i$ equals to 55 in the case of `HOCL0055.EXP` file.

$\texttt{R1}_i$ is the weighting factor of the i^{th} experimental file. The meaning of it corresponds to `R1`{*Subsection 10.1.2* on page 120}. If it is not indicated, this value is considered to be `R1`{Keystring: `CHARACTERISTIC`}.

$\texttt{R2}_i$ is the ratio of the errors of the i^{th} experimental file. The meaning of it corresponds to `R2`{*Subsection 10.1.2* on page 120 }. If it is not indicated, this value is considered to be `R2`{Keystring: `CHARACTERISTIC`}.

$\texttt{R3}_i$ is the initial value of time for the i^{th} experimental file. It corresponds to `R1`{UNIT 3.C, on page 111}. If it is not indicated, this value is considered to be 0.0. If the experimental file also declares the initial time, this declaration supersedes $\texttt{R3}_\text{i}$ (see *Section 8.1*).

$\texttt{P}_\text{i}$ is a list of relative serial numbers of parameters (*See Appendix C*) which helps to avoid the unnecessary computation during fitting. The parameter estimation requires a lot of simulations with systematically changed parameter values. However, if the i^{th} calculated curve does not depend on the value of a parameter, it is unnecessary to calculate the i^{th} curve several times with the changed parameter values. $\texttt{P}_\text{i}$ enumerates those parameters which do not have influence on the i^{th} calculated curve. This list will be translated into the generated `*.DAT` file for `ZITA.EXE` (see UNIT 3.A in *Section 9.3*) and into the `*.JCD` file for `JC.EXE` (see *Subsection 10.1.2*). These input files will direct the executable programs not to carry out the simulation and — otherwise necessary — some calculations.

`-1` means that there are no more data belonging to this keystring.

As the user can see that in `HOCL.TSK` file, more data can be placed into one line separated by space(s). However, spaces cannot be placed within the data! For example:

`55` or `55T0.0` means that the user uses the data of the 55^{th} experimental file and the initial value of time equals to zero. The weighting factor and the ratio equal to the default value since they are not given explicitly.

`23T256.0` means that the user uses the data of the 23^{th} experimental file and the initial value of time equals to 256.0. The weighting factor and the ratio have their default values, as well.

`23W10.0T1.0` or `23T1.0W10.0` mean that the user uses the data of the 23^{th} experimental file, the weighting factor is 10.0, the ratio equals to the default value and the initial value of time equals to 1.0.

`23W10.0R2.0T1.0` or `23R2.0W10.0T1.0` or `23T1.0R2.0W10.0` or ... mean that the user uses the data of the 23^{th} experimental file, the weighting factor is 10.0, the ratio is 2.0 and the initial value of time equals to 1.0. However, `23␣W10.0␣R2.0␣T1.0` *is wrong*!

5P[1,3..8,13,18] directs the programs *not* to calculate the 5^{th} simulated curve when the (ez+tk+1)$^{\text{th}}$, (ez+tk+3–8)$^{\text{th}}$, (ez+tk+13)$^{\text{th}}$ or (ez+tk+18)$^{\text{th}}$ parameter value is changed. The appropriate elements of the Jacobi-matrix (in *.COL files) will be set to zeroes.

FIGURES

Syntax: I *

Default value: '0'

See also: OTHER␣TASK

I determines whether the figures are created by DRAWER.EXE and *.DRW[2] file or not. This value is considered only in the main *.TSK file. If the actual *.TSK file is called from another, I is disregarded.

- If I equals to zero, figures are not created during the fitting.
- If I is greater than zero, the figures belonging to the first, every I$^{\text{th}}$ and the last iterations are created during the fitting. If there is at least one figure to be created, TASK.EXE creates a Basename.DRA file in the user's library. It contains the names of those files which store a figure and were created during the fitting. The subtitles on the figures indicate the ordinal numbers of the models and iterations.These figures can be seen after the fitting by the help of 'DRAWER Basename.DRA' DOS command. Examples 7–9 show applications.

  If the value of I is greater than the maximum number of the iterations (defined in *Subsection 10.2.1*), only the figures of the last iteration are created.
- If I is less then zero, the figures of the first, every abs(I)$^{\text{th}}$ and the last iterations are also created during the fitting. However, these figures are not saved and the fitting can go on only if the user terminates DRAWER.EXE by pressing '*Q*'.

INITIAL␣CONCENTRATION

Syntax A: I1 $I2_1$ $I2_2$ ... $I2_{I1}$ *

Syntax B: I1 $I2_1$ $I2_2$ ... $I2_{I1}$ *
I3 $I4_1$ $I4_2$ ... $I4_{I3}$ *

This keystring determines the sequence of the initial concentrations in the experimental files. The symbols defined here (I1...I4) correspond to I3...I6{UNIT 3.B, on page 110}, respectively.

[2]The user has to write this file before the fitting because TASK.EXE does not create it.

INITIAL␣STEP

Syntax: R * **Default value:** '1.0E-10'

R corresponds to R2{UNIT 2.B, on page 107}.

INNER␣POINT

Syntax: I * **Default value:** '0'

I corresponds to I7{UNIT 3.C, on page 111}.

INTEGRATION␣METHOD

Syntax: I * **Default value:** '21'

I corresponds to I1{UNIT 2.B, on page 107}.

LINES␣MUST␣BE␣SKIPPED

Syntax: I * **Default value:** '1'

See also: 'CHARACTERISTIC'
See also: 'OTHER␣TASK'

I corresponds to I2{*Subsection 10.1.2* on page 120}. Since this number concerns all the experimental files, the user must use 'OTHER␣TASK' keystring if the necessary experimental files have different structure.

MODEL

Syntax: see below **See also:** OTHER␣TASK

The syntax and symbols of this keystring are the same as that of the PARAMTRS.DAT detailed in *Subsection 10.2.1* on page 124. The only exception is the last line. If there is anything behind the maximum number of the iterations (I{*Subsection 10.2.1*}) in the same row, it is interpreted as a main title of the current model and it is written into the first line of the actual MODEL?.RES file. The maximum length for this title is 125 characters.

This keystring is an exception of the general rule. If this keystring can be found several times in the file in question, not only the last one is valid and TASK.EXE carries out the tasks successively. In this way, several tasks can be given in an *.TSK file if the actual *.ODE or ODES.PAS file contains every chemical equation. Disregarding a chemical equation is very simple: its rate constant must be set to zero.

This keystring is considered only in the main *.TSK file. If the actual *.TSK file is called from another, this keystring is disregarded.

NUMBER␣OF␣CHARACTERS

Syntax: I * **Default value:** '80'

I corresponds to I2{UNIT 2.A, on page 106}.

OTHER␣TASK

Syntax:
S1(4)
S2(4)
⋮
Sn(4)
-1

See also: INITIAL␣CONCENTRATION, LINES␣MUST␣BE␣SKIPPED

If the user has such kind of experimental files having different structures, these files cannot be handled with the same basename and *.TSK file. Only those experimental files can have the same basename which do not have different structures.

In this case, the files can be handled with separated *.TSK files. One of them is the *main file* and all the others are called from the main file.

n is the number of *.TSK files to be called from the main file.

Si is the filename without extension of the i^{th} *.TSK file called from the main file. In other words, Si is the basename of the i^{th} group of the experimental files.

-1 means that there are no more data belonging to this keystring.

Not every keystring can be chosen independently from the main file in the called ones. If the keystring is one of the 'ABSOLUTE,␣RELATIVE', 'DIFFERENTIATION␣METHOD', 'FIGURES', 'MODEL', 'OTHER␣TASK', 'PERCENT' or 'PLACE␣OF␣THE␣RESULT␣FILES', it is disregarded in the called files.

An excellent example can be found in example 8 illustrating this possibility.

OUTPUT␣FILES

Syntax: S(3) * **Default value:** 'c'

S can include either 'c', 'v', 'd' or their any combinations. This logical unit determines the kind of the files to be created during a simulation. UNIT 3.D on page 114 details what kinds of files can be calculated. In the cases of 'c', 'v' and 'd', the files containing the

concentrations (and/or their transformations), the rate values and the derivatives of the concentrations with respect to time are generated, respectively. The name for the first kind of file is detailed earlier in this chapter. The names for the others are always `V` and `D`. Their contents can be used in **Transform** procedure (see *Subsection 9.1.1* on page 96) if the `ODES.PAS` file is written or modified manually. Otherwise, this logical unit is useless.

PERCENT

Syntax: `R *` **Default value:** '`1.0`'

See also: '`DIFFERENTIATION␣METHOD`' '`OTHER␣TASK`'

`R` corresponds to the fifth DOS parameter of `JC.EXE`. Its description can be found in *Subsection 10.1.1* on page 119. This value is considered only in the main `*.TSK` file. If the actual `*.TSK` file is called from another, `R` is disregarded.

PLACE␣OF␣THE␣RESULT␣FILES

Syntax: `I *` **Default value:** '`0`'

See also: '`OTHER␣TASK`'

`I` corresponds to `I3`{UNIT 2.A, on page 106}. This value is considered only in the main `*.TSK` file. If the actual `*.TSK` file is called from another, `I` is disregarded.

RELATIVE␣ERROR

Syntax: `R *` **Default value:** '`1.0E-08`'

`R` corresponds to `R1`{UNIT 2.B, on page 107}.

SIGNIFICANT␣DIGITS

Syntax: `I *` **Default value:** '`6`'

`I` corresponds to `I1`{UNIT 2.A, on page 106}.

SPEED␣OPTIMALIZATION

Syntax: `S(3) *` **Default value:** '`no`'

`S` can be either '`yes`' or '`no`'. In *Section 9.4* on page 117 the second item describes a method to avoid some unnecessary simulations. This logical unit instructs `TASK.EXE` to use

this possibility in the case of '`yes`' and prevents the usage of it in the case of '`no`'. If the **Transform** procedure in `ODES.PAS` (see *Subsection 9.1.1* on page 96) is not empty and some fitted parameters take part in the transformations (e.g. molar absorbances to be fitted), the '`yes`' value can speed up the fitting significantly. Otherwise, this setting does not influence anything.

.COL

Syntax: `I *` **Default value:** '`0`'

`I` can be 0 or 1. If `I` equals to zero, `J*.COL` files are erased after `TASK.EXE` finishes the calculation of a model. If `I` equals to 1, `J*.COL` files are kept but they are renamed because the next model also uses these filenames. The new names in the i^{th} model are `*COLM.`i. For example, `J.COL` and `J13.COL` become `JCOLM.3` and `J13COLM.3` after finishing the third model. This possibility can be useful if the user wants to study the sensitivity matrix as well.

12.2 Calling `TASK.EXE`

The first DOS parameter of `TASK.EXE` is the name of `*.TSK` file to be used. If this file has another extension, it must be indicated in the DOS parameter.

The calculation can be interrupted at any time and can be started again. `TASK.EXE` searches that iteration which has not been finished and continues the calculation.

The second parameter can be '`c`' or '`a`' or a number.

- In the case of '`c`', `TASK.EXE` does not carry out the task(s) but generates those files which would be used for solving problems. In this way the user can study or revise them.

  There is an important think about this possibility. The user can think that the best method to create his `*.DAT` file is to generate an appropriate `*.DAT` file with `TASK.EXE` and to revise it. It is possible but the changed file cannot be used by `TASK.EXE`! It can be used only in a DOS command file because `TASK.EXE` regenerates the necessary files before every restarting of calculation.

- An iteration can require a lot of time. In this case, it may be useful if `TASK.EXE` does not start the last iteration again but uses the ready partial results. It can be achieved by an '`a`' value for the second DOS parameter.

  However, this possibility is a bit dangerous. In this case, the user must not change anything in the user's library between the interruption and restarting! Otherwise, there is a small chance (less than 0.01 %) for losing the partial results of the last iteration.

- If the second parameter is a number, the user explicitly instructs `TASK.EXE` which part of the actual iteration has to be carried out. For example, `Task Hurry 3` command means that `TASK.EXE` works with the `Hurry.tsk` file and the third parameter is changed before the next simulation. If the first or the second parameter are to be fitted, the appropriate `J1.COL` and `J2.COL` files are not recalculated.

There can be an additional character in the second DOS parameter followed by the previously detailed ones. If it is 'n' or 'N', the part entitled '`******** Unweighted Standard Deviation of Data:`' is omitted from `MODEL*.RES` files due to the long computing time.

It is very important that the user must not use more DOS parameters than 2! The further DOS parameters are used by `TASK.EXE` for own purposes.

Part III

Appendix

"It is a simple task to make things complex, but a complex task to make them simple."

Meyer's Law [17]

Appendix A

Detailed Mathematical Background

A.1 A General Mathematical Model for Chemical Reaction Mechanisms

The stoichiometry of any chemical reaction can be expressed in the next way:

$$\begin{array}{ccccccc}
\alpha_{11}S_1 + \alpha_{12}S_2 + \cdots + \alpha_{1n}S_n & \xrightarrow{k_1(T,\vec{p}_{k_1})} & \omega_{11}S_1 + \omega_{12}S_2 + \cdots + \omega_{1n}S_n \\
\alpha_{21}S_1 + \alpha_{22}S_2 + \cdots + \alpha_{2n}S_n & \xrightarrow{k_2(T,\vec{p}_{k_2})} & \omega_{21}S_1 + \omega_{22}S_2 + \cdots + \omega_{2n}S_n \\
\vdots \quad \vdots \quad \vdots & & \vdots \quad \vdots \quad \vdots \\
\alpha_{e1}S_1 + \alpha_{e2}S_2 + \cdots + \alpha_{en}S_n & \xrightarrow{k_e(T,\vec{p}_{k_e})} & \omega_{e1}S_1 + \omega_{e2}S_2 + \cdots + \omega_{en}S_n
\end{array}$$

This equation system is denoted more briefly in *Eq. (A.1)*:

$$\sum_{j=1}^{n} \alpha_{ij} S_j \quad \xrightarrow{k_i(T,\vec{p}_{k_i})} \quad \sum_{j=1}^{n} \omega_{ij} S_j \qquad (i = 1 \ldots e) \tag{A.1}$$

where

- e is the number of elementary or mechanistic steps. A mechanistic step is defined as a part of a reaction mechanism. It can consist of several elementary steps but the dependence of its rate on the actual reactant concentration can be given formally by *Eq. (A.2)*. If $\mathcal{ZIA}$ is used, either a negative and/or non-integer order of the reaction with reference to any substance is also allowed.
- n is the number of the species. They can be reactants, intermediates or products concerning the whole mechanism.
- S_j is the j^{th} species.
- α_{ij} is the stoichiometric number of the j^{th} species in the i^{th} step if S_j is supposed to be a reactant in this step.
- ω_{ij} is the stoichiometric number of the j^{th} species in the i^{th} step if S_j is supposed to be a product in this step.
- T is the actual temperature.

$\vec{p}_{k_i}$ is a vector. The rate constants can be expressed as a function of temperature. The elements of this vector are the parameters of that function which is defined as the i^{th} rate constant *vs.* temperature.

If the reaction in question is carried out at a constant temperature, $\vec{p}_{k_i}$ consists of only one element, which is the value of the i^{th} rate constant itself. In this case the temperature cannot be an independent variable.

$k_i(T, \vec{p}_{k_i})$ is the rate constant in the i^{th} step.

Naturally, the laws of conversation of mass, charge and chemical element must be valid in each step.

The rate of the steps can be expressed in the following way according to Guldberg and Waage (1867):

$$\begin{array}{rcccccc} r_1(t) & = & k_1(T,\vec{p}_{k_1}) \cdot C_1^{\nu_{11}}(t) \cdot C_2^{\nu_{12}}(t) \cdot \cdots \cdot C_n^{\nu_{1n}}(t) \\ r_2(t) & = & k_2(T,\vec{p}_{k_2}) \cdot C_1^{\nu_{21}}(t) \cdot C_2^{\nu_{22}}(t) \cdot \cdots \cdot C_n^{\nu_{2n}}(t) \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ r_e(t) & = & k_e(T,\vec{p}_{k_e}) \cdot C_1^{\nu_{e1}}(t) \cdot C_2^{\nu_{e2}}(t) \cdot \cdots \cdot C_n^{\nu_{en}}(t) \end{array}$$

or briefly:

$$r_i(t) = k_i(T,\vec{p}_{k_i}) \prod_{j=1}^{n} (C_j(t))^{\nu_{ij}} \qquad (i = 1 \ldots e) \tag{A.2}$$

The meanings of e, n, T, $\vec{p}_{k_i}$ and $k_i(T, \vec{p}_{k_i})$ in *Eq. (A.2)* are the same as their meanings in *Eq. (A.1)*. The other abbreviations are the following:

t is the independent variable (time).

$r_i(t)$ is the rate of the i^{th} step of the mechanism as a function of time.

$C_j(t)$ is the concentration of S_j species as a function of time if the volume of the reaction mixture does not change during the reaction.

ν_{ij} is the partial order of the i^{th} step concerning S_j species.

The change of the concentrations in time can be expressed as a differential quotients of S_j-s ($j = 1 \ldots n$) with respect to t. It can be formulated from *Eq. (A.1)* and *Eq. (A.2)* in the following way:

$$\begin{array}{rcccl} \dfrac{\mathrm{d}[S_1]}{\mathrm{d}t} & = & C_1'(t) & = & (\omega_{11} - \alpha_{11}) \cdot r_1(t) + (\omega_{21} - \alpha_{21}) \cdot r_2(t) + \cdots + (\omega_{e1} - \alpha_{e1}) \cdot r_e(t) \\ \dfrac{\mathrm{d}[S_2]}{\mathrm{d}t} & = & C_2'(t) & = & (\omega_{12} - \alpha_{12}) \cdot r_1(t) + (\omega_{22} - \alpha_{22}) \cdot r_2(t) + \cdots + (\omega_{e2} - \alpha_{e2}) \cdot r_e(t) \\ \vdots & & \vdots & & \vdots \\ \dfrac{\mathrm{d}[S_n]}{\mathrm{d}t} & = & C_n'(t) & = & (\omega_{1n} - \alpha_{1n}) \cdot r_1(t) + (\omega_{2n} - \alpha_{2n}) \cdot r_2(t) + \cdots + (\omega_{en} - \alpha_{en}) \cdot r_e(t) \end{array}$$

The short form of this *ODEs* is the following:

$$\frac{\mathrm{d}[S_j]}{\mathrm{d}t} = C_j'(t) = \sum_{i=1}^{e}(\omega_{ij} - \alpha_{ij}) \cdot r_i(t) \qquad (j = 1 \ldots n) \tag{A.3}$$

If $r_i(t)$ is substituted from *Eq. (A.2)* into *Eq. (A.3)*, the result is *Eq. (A.4)*:

$$\frac{\mathrm{d}[S_j]}{\mathrm{d}t} = C_j'(t) = \sum_{i=1}^{e}(\omega_{ij} - \alpha_{ij}) \cdot k_i(T, \vec{p}_{k_i}) \prod_{l=1}^{n} (C_l(t))^{\nu_{il}} \qquad (j = 1 \ldots n) \tag{A.4}$$

Eq. (A.4) equation system gives the general mathematical model of a reaction mechanism if the volume is constant and there is not any inflow or outflow during the reaction. Naturally, the *ODEs* given in *Eq. (A.4)* can have another mathematical form — moreover it can become a plain differential equation — if the chemical model is simplified under certain considerations, e.g. steady state principle. Either the complete model or the simplified ones can be given by the next form:

$$\frac{\mathrm{d}[S_j]}{\mathrm{d}t} = C_j'(t) = f_j\left(\overline{\overline{\boldsymbol{\omega\alpha}}}, \overline{\overline{\boldsymbol{\nu}}}, \vec{k}(T), \vec{C}(t)\right) \quad (j = 1 \ldots n) \tag{A.5}$$

where

$\overline{\overline{\boldsymbol{\omega\alpha}}}$ is the matrix the element of which is $(\omega_{ij} - \alpha_{ij})$ $(i = 1 \ldots e,\ j = 1 \ldots n)$ in i^{th} row and j^{th} column.

$\overline{\overline{\boldsymbol{\nu}}}$ is the matrix the element of which is ν_{ij} $(i = 1 \ldots e,\ j = 1 \ldots n)$ in i^{th} row and j^{th} column.

$\vec{k}(T)$ is a vector the elements of which are $k_i(T, \vec{p}_{k_i})$ $(i = 1 \ldots e)$ functions.

$\vec{C}(t)$ is a vector produced from $C_j(t)$ $(j = 1 \ldots n)$ functions.

Eq. (A.5) is an *ODEs* which can usually be solved by only numerical methods[1]. The mathematical model of a mechanism can be given unambiguously by the help of $\overline{\overline{\boldsymbol{\omega\alpha}}}$, $\overline{\overline{\boldsymbol{\nu}}}$ and $\vec{k}(T)$. If they are known, the values of $\vec{C}(t)$ vector function can be calculated at any time. The accuracy of these values is unlimited theoretically but it highly depends on the concrete hardware and software realization in practice.

If the volume of the reaction mixture changes during the reaction, *Eq. (A.4)* must be modified. There are two reasons mainly which can cause changing volume during a reaction:

- There is inflow and/or there is outflow for changing the concentrations in the reaction solution. In this case the

$$C_j'(t) = \sum_{i=1}^{e}\left((\omega_{ij} - \alpha_{ij}) \cdot k_i(T, \vec{p}_{k_i}) \prod_{l=1}^{n} (C_l(t))^{\nu_{il}}\right) - \frac{\frac{\mathrm{d}V^{\text{in}}(t)}{\mathrm{d}t}\left(C_j^{\text{in}} - C_j(t)\right)}{V_0 + V^{\text{in}}(t) - V^{\text{out}}(t)}$$

ODEs is valid, where

[1] ZIA uses the ROW4A-, Gear- and Adams-algorithms to solve an *ODEs*.

V_0 is the initial volume of the reaction mixture.
$V^{\text{in}}(t)$ is the total volume of the entering substance as a function of time.
$V^{\text{out}}(t)$ is the total volume of the outflowing substance as a function of time.
C_j^{in} is the concentration of S_j species in the entering substance[2].

Two special cases of this *ODEs* can be important in practice:

- In the case of continuously fed and stirred tank reactor (CSTR reaction) the

$$V^{\text{in}}(t) = V^{\text{out}}(t) = k \cdot V_0 \cdot t$$

relations are valid, where k is the reciprocal of the residence time. After the substitutions *Eq. (A.4)* has the following form:

$$C_j'(t) = \sum_{i=1}^{e} \left((\omega_{ij} - \alpha_{ij}) \cdot k_i(T, \vec{p}_{k_i}) \prod_{l=1}^{n} (C_l(t))^{\nu_{il}} \right) - k \cdot \left(C_j^{\text{in}} - C_j(t) \right)$$

- If $V^{\text{in}}(t) = k^{\text{in}} \cdot V_0 \cdot t$ and $V^{\text{out}}(t) = k^{\text{out}} \cdot V_0 \cdot t$ but $k^{\text{in}} \neq k^{\text{out}}$, the

$$C_j'(t) = \sum_{i=1}^{e} \left((\omega_{ij} - \alpha_{ij}) \cdot k_i(T, \vec{p}_{k_i}) \prod_{l=1}^{n} (C_l(t))^{\nu_{il}} \right) - \frac{k^{\text{in}} \left(C_j^{\text{in}} - C_j(t) \right)}{1 + \left(k^{\text{in}} - k^{\text{out}} \right) t}$$

formula is valid. If $k^{\text{out}} = 0$, the reactor type is called *semibatch* [18, 19].

- A gas reaction to be carried out under constant pressure can also give changing volume during the reaction. In this case *Eq. (A.4)* follows the

$$C_j'(t) = \sum_{i=1}^{e} \left((\omega_{ij} - \alpha_{ij}) \cdot k_i(T, \vec{p}_{k_i}) \prod_{l=1}^{n} (C_l(t))^{\nu_{il}} \right) - C_j(t) \frac{V'(t)}{V(t)}$$

formula, where $V(t)$ is the volume of the reaction mixture as a function of time.

The reactor types listed above are handled by the program package. The appropriate mathematical formulae for other reactors [18, 19] can also be programmed for the simulations.

A.2 Mathematical Bases of the Program Package

In kinetic experiments, physical-chemical properties of a reaction mixture are measured as a function of time. If the experiments are well-planned, it is possible to deduce the reaction mechanism from the experimental data. These data are usually not concentrations

[2] This value is supposed to be constant by the program package.

— although it is also possible — but functions of the concentrations which can usually be expressed in explicit mathematical form[3]:

$$D_l(t) = D_l\left(\vec{C}(t), \vec{p}_{m_l}\right) \qquad (l = 1 \ldots q) \tag{A.6}$$

where

$\vec{C}(t)$ has the same meaning as described in *Eq. (A.5)*.

q is the number of physical-chemical properties measured at the same initial conditions.

$D_l(t)$ is the value of l^{th} properties as a function of time.

$\vec{p}_{m_l}$ is a vector. It contains the parameters of that mathematical equation which expresses $D_l(t)$ as a function of $\vec{C}(t)$.

In the future $\vec{\mathcal{P}}$ marks that vector which consist of all the elements of $\vec{p}_{k_i}$ $(i = 1 \ldots e)$ and $\vec{p}_{m_l}$ $(l = 1 \ldots q)$ vectors together. In other words, $\vec{\mathcal{P}}$ contains all the parameters which can be found in *Eq. (A.5)* or *Eq. (A.6)*. *The program package can fit the values of the elements of* $\vec{\mathcal{P}}$ *on the basis of measured experimental data.*

If the values of $\overline{\overline{\boldsymbol{\omega\alpha}}}$, $\overline{\boldsymbol{\nu}}$ and $\vec{\mathcal{P}}$ are given in a concrete model, the calculated analogous of every experimental curve can be computed based on *Eq. (A.5)* and *Eq. (A.6)*. The fit of the parameters means the search for the minimum of $\mathcal{SUM}(\vec{\mathcal{P}})$ which is defined as the sum of squares of all the residuals[4]. It is defined by the following expression:

$$\mathcal{SUM}(\vec{\mathcal{P}}) = \sum_{i=1}^{nc} \sum_{j=1}^{mc_i} \sum_{l=1}^{d_{i,j}} \left(\left(D_j(t_{i,j,l}^{\text{expr}}, \vec{\mathcal{P}})^{\text{calc}} - D_{i,j,l}^{\text{expr}}\right) \cdot W_{i,j,l}\right)^2 \tag{A.7}$$

where

$\mathcal{SUM}(\vec{\mathcal{P}})$ is the sum of squares of residuals to be minimized as a function of the parameters to be fitted.

nc is the number of experimental curves measured at different initial conditions.

mc_i is the number of measured characteristics within i^{th} experimental curve.

$d_{i,j}$ is the number of data of j^{th} measured characteristic in i^{th} experimental curve. It means also that an experimental curve is a series of experimental points so the points representing the curve must describe the structure of the curve in question for a successful fitting.

$t_{i,j,l}^{\text{expr}}$ is l^{th} point of time in i^{th} experimental curve concerning j^{th} measured characteristic.

$D_{i,j,l}^{\text{expr}}$ is l^{th} point of j^{th} measured characteristic in i^{th} experimental curve.

[3] The electrode potential, absorbance, conductivity, etc. are these kinds of functions. They depend on the concentrations or the ratio of the concentrations unambiguously.

[4] Residual means the difference between the experimental and calculated data.

$D_j(t_{i,j,l}^{\text{expr}}, \vec{\mathcal{P}})^{\text{calc}}$ is the l^{th} point of j^{th} characteristic in i^{th} calculated curve within the actual iteration. In other words it is the calculated analogous of $D_{i,j,l}^{\text{expr}}$.

$W_{i,j,l}$ is the weighting factor of residuals determined by indices. ZITA sets up a restriction. It must be true that $W_{i,j,l'} = W_{i,j,l''}$ for all $(l', l'') \in \{1 \dots d_{i,j}\}$ values. In the case of relative fitting the factors given by the user are divided by $(D_{i,j}^{\max} - D_{i,j}^{\min})$ where $D_{i,j}^{\max} = \max\{D_{i,j,1}^{\text{expr}} \dots D_{i,j,d_{i,j}}^{\text{expr}}\}$ and $D_{i,j}^{\min} = \min\{D_{i,j,1}^{\text{expr}} \dots D_{i,j,d_{i,j}}^{\text{expr}}\}$.

Eq. (A.7) supposes that the experimental observations of time are exact, only the measured values of the characteristics can have experimental errors. The errors of $t_{i,j,l}^{\text{expr}}$ data can be taken into consideration by orthogonal fitting. In this case the sum of squares of residuals is defined in the following way:

$$\mathcal{SUM}(\vec{\mathcal{P}}) = \sum_{i=1}^{nc} \sum_{j=1}^{mc_i} \sum_{l=1}^{d_{i,j}} \left(\sqrt{\left(\frac{t_{i,j,l}^{\text{calc}} - t_{i,j,l}^{\text{expr}}}{t_{i,j}^{\max} - t_{i,j}^{\min}} \right)^2 + \left(\frac{D_j(t_{i,j,l}^{\text{calc}}, \vec{\mathcal{P}})^{\text{calc}} - D_{i,j,l}^{\text{expr}}}{D_{i,j}^{\max} - D_{i,j}^{\min}} \right)^2} \cdot W_{i,j,l} \right)^2 \tag{A.8}$$

In this equation nc, mc_i, $d_{i,j}$, $t_{i,j,l}^{\text{expr}}$, $D_{i,j,l}^{\text{expr}}$, $D_{i,j}^{\max}$, $D_{i,j}^{\min}$ and $W_{i,j,l}$ have the meanings described under *Eq. (A.7)* and

$t_{i,j}^{\max}$ is the maximum value of time in i^{th} experimental curve concerning j^{th} measured characteristic.

$t_{i,j}^{\min}$ is the minimum value of time in i^{th} experimental curve concerning j^{th} measured characteristic.

$t_{i,j,l}^{\text{calc}}$ is the calculated analogous of $t_{i,j,l}^{\text{expr}}$ concerning j^{th} characteristic in the actual iteration. The next paragraph details how it can be computed.

$D_j(t_{i,j,l}^{\text{calc}}, \vec{\mathcal{P}})^{\text{calc}}$ is the calculated analogous of $D_{i,j,l}^{\text{expr}}$ in the actual iteration. The next paragraph details how it can be computed.

The calculation of $\left(t_{i,j,l}^{\text{calc}}, D_j(t_{i,j,l}^{\text{calc}}, \vec{\mathcal{P}})^{\text{calc}}\right)$ data pair is carried out by minimizing of *Eq. (A.9)* in the experimental point determined by i, j and l indices:

$$\Delta(t_x) = \left(\frac{t_x - t_{i,j,l}^{\text{expr}}}{t_{i,j}^{\max} - t_{i,j}^{\min}} \right)^2 + R_{i,j}^2 \cdot \left(\frac{D_j(t_x, \vec{\mathcal{P}})^{\text{calc}} - D_{i,j,l}^{\text{expr}}}{D_{i,j}^{\max} - D_{i,j}^{\min}} \right)^2 \tag{A.9}$$

If this function is minimized as a function of t_x $\left(\left(\frac{\partial \Delta(t_x)}{\partial t_x}\right)_{t_x = t_{min}} = 0\right)$, the $\left(t_{min}, D_j(t_{min}, \vec{\mathcal{P}})^{\text{calc}}\right)$ data pair is regarded as the computed value.

$R_{i,j}$ is the ratio of errors of experimental time and j^{th} measured characteristic in i^{th} experimental curve. Its values are determined by the user. It can be estimated if the absolute values of the errors of time and characteristic are divided by $\left(t_{i,j}^{\max} - t_{i,j}^{\min}\right)$ and

$\left(D_{i,j}^{\max} - D_{i,j}^{\min}\right)$. In this way the relative error of time and j^{th} characteristic is known and $R_{i,j}$ is the ratio of these two relative errors. If $R_{i,j} \to 0$, $t_{min} \to t_{i,j,l}^{\text{expr}}$ so *Eq. (A.8)* becomes *Eq. (A.7)*. If $R_{i,j} \to \infty$, $D_j(t_{min}, \vec{\mathcal{P}})^{\text{calc}} \to D_{i,j,l}^{\text{expr}}$ so only the experimental values of time can have errors and the characteristics are supposed to be exact. The latter case can be handled unambiguously only if the measured characteristic *vs.* time is a one-to-one function.

Eq. (A.9) can be applied only in that case if the values of $D_j(t_x, \vec{\mathcal{P}})^{\text{calc}}$ can be calculated at any t_x. Since the analytical form of this function is unknown, the computation of the function value is possible only by simulation with arbitrary precision. But practically it is not a good method because the value of t_x has not been known previously so this method would require a lot of simulations and it would become a time-consuming process. That is why the $\mathcal{ZTA}$ uses another method for orthogonal fitting.

The starting data within a curve are $\left(t_{i,j,l}^{\text{expr}}, D_{i,j,l}^{\text{expr}}, D_j(t_{i,j,l}^{\text{expr}}, \vec{\mathcal{P}})^{\text{calc}}\right)$ triple data. Before the search for the minimum these data are converted into the following forms:

$$x_l^{\text{expr}} = \frac{t_{i,j,l}^{\text{expr}} - t_{i,j}^{\min}}{t_{i,j}^{\max} - t_{i,j}^{\min}}, \quad y_l^{\text{expr}} = R_{i,j} \cdot \frac{D_{i,j,l}^{\text{expr}} - D_{i,j}^{\min}}{D_{i,j}^{\max} - D_{i,j}^{\min}}, \quad y_l^{\text{calc}} = R_{i,j} \cdot \frac{D_j(t_{i,j,l}^{\text{expr}}, \vec{\mathcal{P}})^{\text{calc}} - D_{i,j}^{\min}}{D_{i,j}^{\max} - D_{i,j}^{\min}}$$

where the notation is the same as in *Eq. (A.8)*. The distance between the transformed experimental data pair and the transformed calculated curve is computed in the following way:

- In almost every case a parabolic arc can be fitted to three different points marked $\left((x_l^{\text{calc}}, y_l^{\text{calc}}), (x_{l+1}^{\text{calc}}, y_{l+1}^{\text{calc}}), (x_{l+2}^{\text{calc}}, y_{l+2}^{\text{calc}})\right)$ $(l = 1 \ldots d_{i,j} - 2)$[5]. This parabolic arc substitutes the appropriate arc of the calculated curve[6]. In this way *Eq. (A.9)* becomes a maximum four-degree polynomial and the distance of an experimental point from the calculated curve can be computed within the actual parabolic arc.
- The distances of the experimental data pair in question from all the parabolic arcs can be computed by repeating the previous step. The smallest value of them is the $\left(t_{i,j,l}^{\text{calc}}, D_j(t_{i,j,l}^{\text{calc}}, \vec{\mathcal{P}})^{\text{calc}}\right)$ to be determined. If there are several smallest distances that point is chosen by $\mathcal{ZTA}$ which is the nearest $t_{i,j,l}^{\text{expr}}$. If there are several smallest differences, too, that calculated point is used which has the smallest value of time.

On the basis of these facts the ability of the program package can be summarized in the following way:

- The next input data must be given:
    1. The experimental data can be the values of a measured characteristic as a function of time. They are derived from the experimental curves which are taken

[5] If a parabolic arc cannot be fitted to these data, e.g. $x_l^{\text{calc}} = x_{l+2}^{\text{calc}}$, etc., the points in question are connected with lines by $\mathcal{ZTA}$.

[6] In practice, this approach can be applied only in that case if the experimental points reflect the structure of the experimental curve well.

under different initial conditions. The initial concentrations and other important values, e.g. temperature of the experimental curves should also be known[7].

2. The data of the mathematical model for the supposed mechanism are required, too. These data are $\overline{\overline{\boldsymbol{\omega\alpha}}}$ and $\overline{\overline{\boldsymbol{\nu}}}$ matrices and their values cannot be fitted by ZIA!
3. The initial estimated values of the parameters ($\vec{\mathcal{P}}$) are also necessary.

- On the basis of the input data the program package fits the values of $\vec{\mathcal{P}}$ by minimizing the function value of *Eq. (A.7)* or *Eq. (A.8)*[8]. Naturally, fitting can only be successful if the experiments contain *sufficient experimental information* concerning the parameters to be fitted.

If the user describes his own mechanisms by the help of the matrix-formalism presented in *Subsection 9.1.2*, there are two restrictions concerning the experimental curves:

- A concrete curve has to be registered under isotherm conditions but the different curves can be taken at different temperature.
- The validity of Arrhenius-equation is supposed by ZIA if the user fits the parameters of temperature-dependence on rate constants as well.

If the user writes the source text of his own *ODEs* manually, these restrictions are not valid. The value of temperature can change within a concrete curve and the function of $\vec{k}(T)$ vector can have any mathematical form.

[7] The initial concentrations can also be fitted by the program package if their values cannot be determined experimentally.

[8] ZIA uses the Gauss-Newton-Marquardt method to minimize *Eq. (A.7)* or *Eq. (A.8)*

Appendix B

List of Error Messages

This appendix contains all the error messages and their error numbers which can occur during running the programs. The *xxxx* always means a given number and *name* means a concrete filename.

B.1 Runtime Messages and Their Exit Codes Produced by Turbo Pascal

DOS errors

2	File not found.
3	Path not found.
4	Too many open files.
5	File access denied.
6	Invalid file handle.
12	Invalid file access code.
15	Invalid file drive number.
16	Cannot remove current directory.
17	Cannot remove across drives.

I/O errors

100	Disk read error.
101	Disk write error.
102	File not assigned.
103	File not open.
104	File not open for input.
105	File not open for output.
106	Invalid numeric format.

Critical errors

150	Disk is write-protected.
151	Unknown unit. *or* Bad drive request structure length.
152	Drive not ready.
153	Unknown command.
154	CRC error in data.
155	Bad drive request structure length.
156	Disk seek error.
157	Unknown media type.
158	Sector not found.
159	Printer out of paper.
160	Device write fault.
161	Device read fault.
162	Hardware failure.

Fatal errors

200	Division by zero.
201	Range check error.
202	Stack overflow error.
203	Heap overflow error.
204	Invalid pointer operation.
205	Floating point overflow.
206	Floating point underflow.
207	Invalid floating point operation.

B.2 Messages Produced by the Programs of ZITA

Errors produced by all the programs

64 *Name* file does not exist!
69 The DOS environment variable named 'ZitaTEMP' has a wrong value!
70 The DOS environment variable named 'ZitaRD' has a wrong value!
71 The DOS environment variable named 'ZitaSIMU' has a wrong value!
94 The DOS environment variable named 'ZitaEXPR' has a wrong value!

Errors produced by MAKEODE.EXE

59 There is a syntax error in the relations!
62 There are too many equations or species!
79 Tmax <= 0!
81 The cn<=0 and trn=1!
82 The trn<>0 and trn<>1!
83 The cs<0 or cs>4!
84 Number of parameters < number of equations or <= 0!
85 Number of equations <= 0!
92 The *xxxx*. column of input matrix is wrong!
96 Number of species <= 0!

Errors produced by ZITA.EXE

61 There is a wrong initial concentration!
68 2-nd or 3-rd DOS-parameter is wrong!
80 The tc is not in ['t','T','n','N']!
86 Integration was halted by driver at t=*xxxx*.
Eps too small to be attained for the machine precision.
The integration was halted, index=-2.
87 Kflag=-3 from integrator at t=*xxxx*.
Corrector convergence could not be achieved.
Problems appears unsolvable with given input.
The integration was halted, index=-3.
88 Kflag=-2 from integrator at t=*xxxx*.
The requested error is smaller than can be handled.
The integration was halted, index=-2.
89 Kflag=-1 from integrator at t=*xxxx*.
Error test failed with Abs(h)=hmin.
Problems appears unsolvable with given input.
The integration was halted, index=-1.
93 The value of integration method is wrong!
95 Illegal input... (t0-tout)*h0 >= 0.0.
The integration was halted, index=-4.
96 Illegal input... n <= 0.
The integration was halted, index=-4.

97 Illegal input... eps <= 0.0.
The integration was halted, index=-4.
98 Illegal input... index=-4.
The integration was halted, index=-4.
99 Index=-1 on input with (t-tout)*h >= 0.0, t=*xxxx*, tout=*xxxx*
h=*xxxx*. Interpolation was done as a normal return.
Desired parameter changes were not made.
The integration was halted, index=-5.

Errors produced by JC.EXE

63 At least one simulated data has not been found!
91 There is an error in DOS-parameters for input!

Errors produced by GNM.EXE

60 There is not enough memory for the matrices!
65 The environment variable named *name* has
a wrong value or a J*.col file does not exist!
66 The environment variable named *name* does not exist!
67 The value of p cannot be found in ODES.PAS!
90 There is a wrong Hess-matrix!

Errors produced by DRAWER.EXE

58 Duplicated independent value at *xxxx* for splines!
72 The *name* does not exist!
73 Ymin=0.0 for drawing of log(ymin)!
74 Xmin<=0.0 for drawing of log(xmin)!
75 The environment named ZitaDISPLAY contains bad value!
76 The environment named ZitaDISPLAY contains bad value!
The value of graphdriver or graphmode is bad!
77 The environment named ZitaDISPLAY contains bad value!
The value of graphdriver is bad!
78 Xmax=xmin or ymax=ymin!

Appendix C

Notation System Used in Reference Guide

The Reference Guide systematically describes how to make the files by the help of which a given task can be solved. Information about these files is given in separated logical units. These units consist of one or more lines in most cases. Each line can include one or more data which are separated from each other by a space or some spaces. The following symbols are applied:

`C` is a printable character.

`S(x)` is a character string consisting of maximum `x` elements (`x` ≥ 0). One element can be a space or a printable character.

`I` is an integer number.

`R` is a real number.

`:I` is a special integer number with a leading colon. It can be used instead of an `R` in a few cases. In this way the initial concentrations and other special parameters of the experimental curves become fittable parameters. This abbreviation means that the value of the real number is not given explicitly in the actual file but this value is the (`ez`+`tk`+`I`)$^{\text{th}}$ data in the `PARAMTRS.DAT` file, where `ez` is the number of the rate constant and `tk` is the number of the temperature dependent rate constants in the actual model.

`I` is a relative ordinal number concerning the data in the `PARAMTRS.DAT` file. The value of the absolute ordinal number would depend on the value of `ez` and `tk`. Since the relative ordinal number does not depend on them, therefore, the program package uses this kind of ordinal numbers. For example, if the actual model has 5 equations and does not have any temperature dependent rate constants, `:3` instead of an initial concentration means that the eighth row of the `PARAMTRS.DAT` file contains the actual value of the initial concentration in question.

`A` is a number either integer or real.

`N` is a DOS filename.

P is a list of parameter number(s) surrounded by `[]`-s. It can contain serial numbers (e.g. `1`, `423`) and ranges (e.g. `3..7`, `11..11`, `17..56`) separated by commas (no space!). In the ranges, the second number cannot be lower than the first one. For example, `[1,3,5..7,23,25,35..39]` means the parameters 1, 3, 5–7, 23, 25, 35–39 together.

`...` means the existing but not listed data.

⋮ means the existing but not listed data.

`[]` surround a unit that has more syntaxes.

| separates two mutually exclusive choices in a syntax line.

{} surround remarks.

If several data of the same type occur in a logical unit the symbol is followed immediately by an integer number which makes a distinction between them, e.g. `I1`, `I2`, `R3`, `S1(x)`, etc. If the amount of data belonging to the same type depends on the concrete task, the symbols also contain an index which distinguishes data, e.g. $\texttt{I3}_1$, $\texttt{I3}_2$, $\texttt{I3}_j$, $\texttt{R}_1$, etc.

An asterisk (`*`) can stand at the end of a line consisting of symbols. It means that any kind of remark can follow the data, the remarks are ignored by the programs. *If the asterisk is missing, nothing can follow the last data!*

Appendix D

Results of the Tests

This appendix contains the graphical results of tests given by ODEs solver and the circumstances of some runs. The problems are detailed in *Chapter 7* and [13].

D.1 Test A: Robertson's Problem

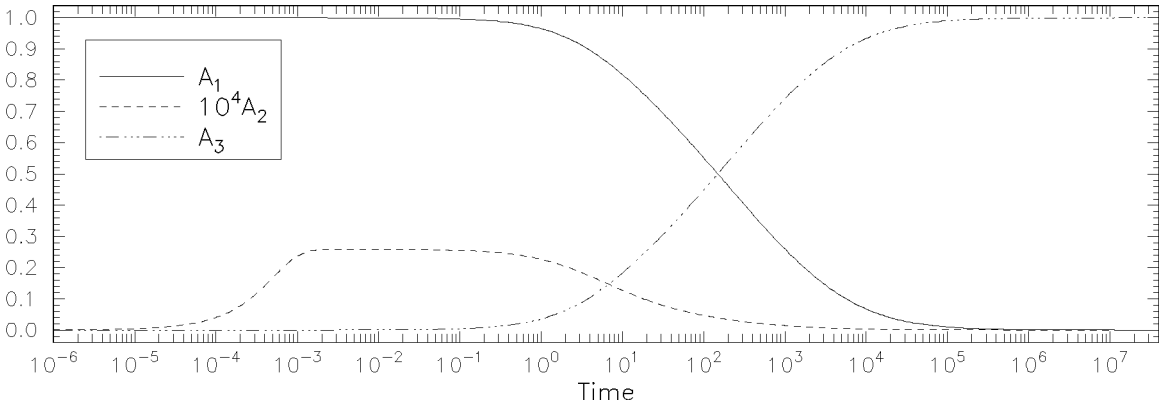


Figure D.1: Result of Robertson's problem.

The contents of `MESSAGES` *file after the computation*		
`The testa.out was calculated within`	`0.71s.`	`6.81s.`
`Method flag:`	`21`	`21`
`Relative error bound:`	`1.000E-03`	`1.000E-08`
`Initial step size:`	`1.000E-06`	`1.000E-10`
`The step size last successfully used:`	`8.447E+05`	`7.697E+05`
`The order last successfully used:`	`1`	`4`
`The cumulative number of steps taken:`	`404`	`4657`
`The cumulative number of Diffun calls:`	`409`	`4731`
`The cumulative number of Jacobian evaluations:`	`100`	`990`
`The number of the output points:`	`101`	`1001`

D.2 Test B: The Field-Noyes Chemical Oscillator

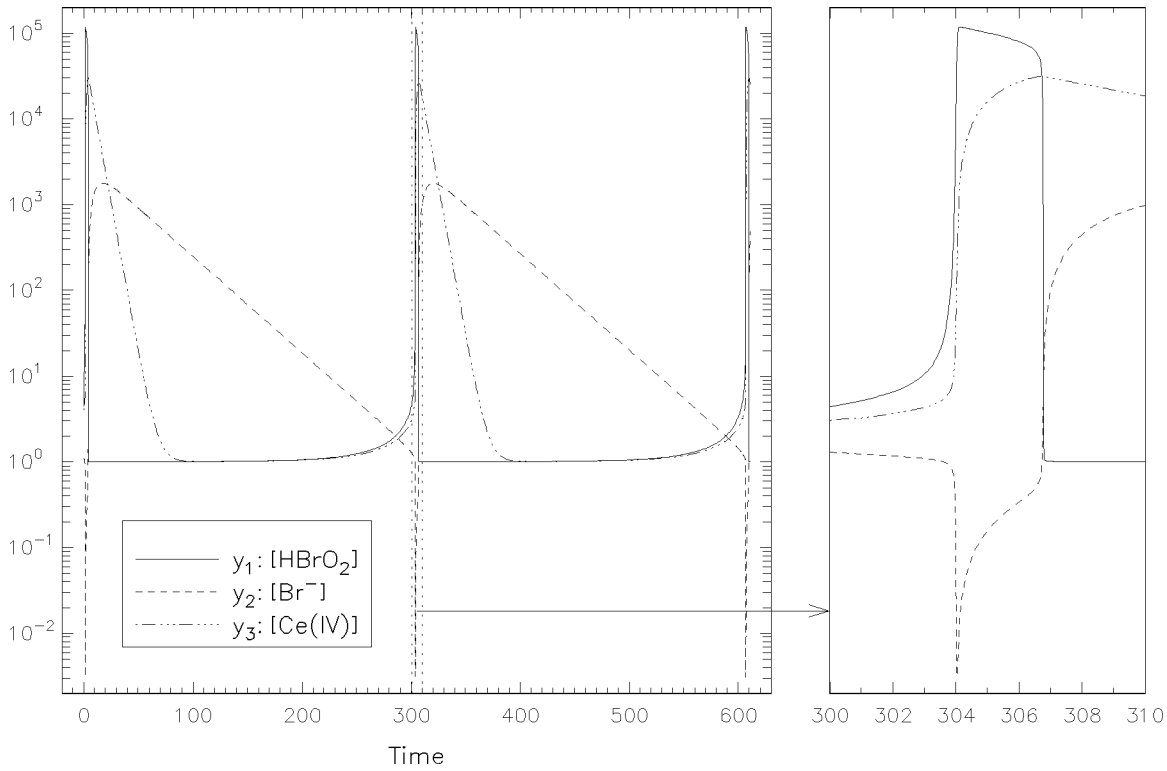


Figure D.2: Result of the Field-Noyes chemical oscillator.

The contents of `MESSAGES` *file after the computation*		
`The testb.out was calculated within`	`4.84s.`	`385.85s.`
`Method flag:`	`21`	`21`
`Relative error bound:`	`1.000E-06`	`1.000E-10`
`Initial step size:`	`1.000E-08`	`1.000E-12`
`The step size last successfully used:`	`9.732E-02`	`7.028E-03`
`The order last successfully used:`	`3`	`4`
`The cumulative number of steps taken:`	`2767`	`276822`
`The cumulative number of Diffun calls:`	`3600`	`276986`
`The cumulative number of Jacobian evaluations:`	`500`	`250745`
`The number of the output points:`	`612`	`61101`

D.3 Test C: Two Species Diurnal Kinetics

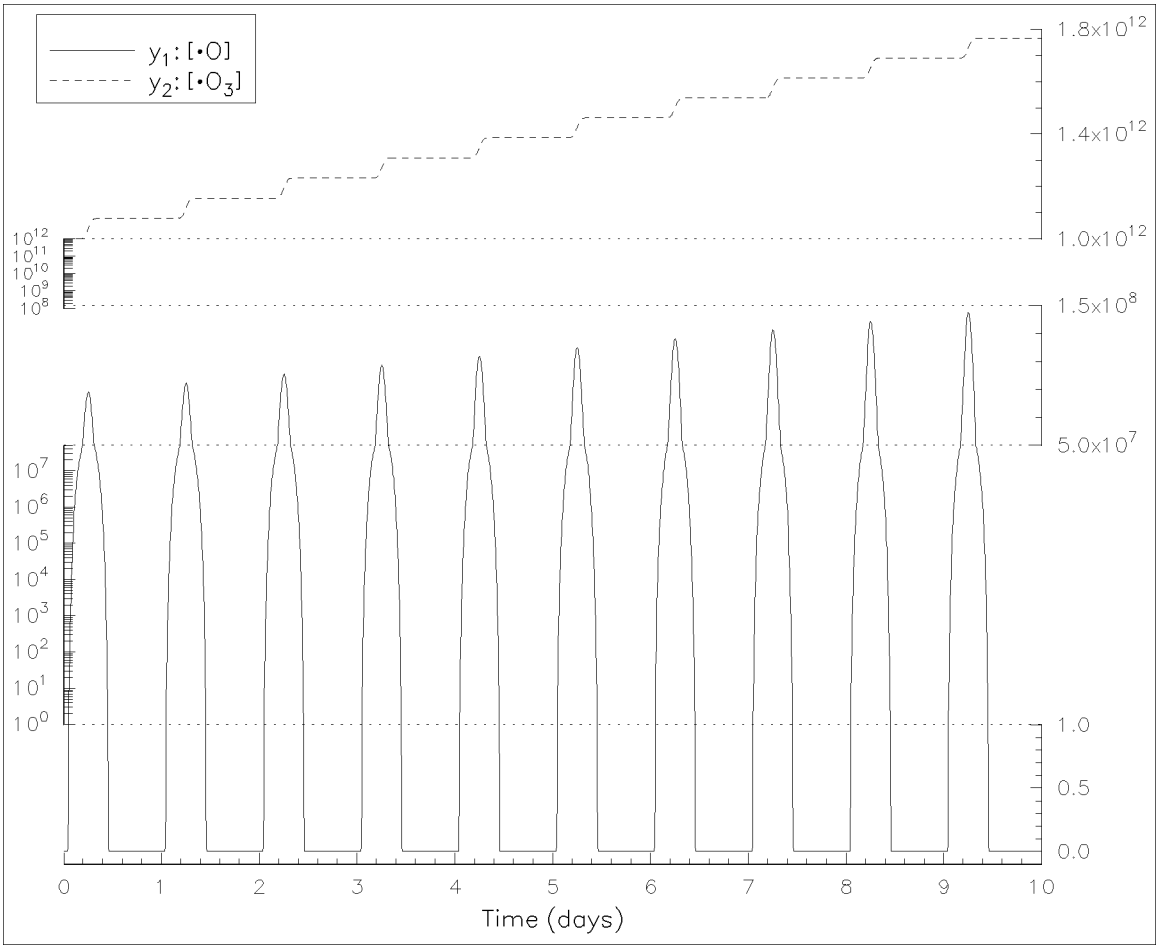


Figure D.3: Result of two species diurnal kinetics.

The contents of MESSAGES *file after the computation*		
The testc.out was calculated within	12.19s.	28.78s.
Method flag:	21	21
Relative error bound:	1.000E-06	1.000E-12
Initial step size (sec):	1.000E-08	1.000E-14
The step size last successfully used (sec):	3.600E+02	3.600E+02
The order last successfully used:	1	1
The cumulative number of steps taken:	7864	21184
The cumulative number of Diffun calls:	8448	30891
The cumulative number of Jacobian evaluations:	1769	3632
The number of the output points:	2401	2401

D.4 Test D: A Kidney Model

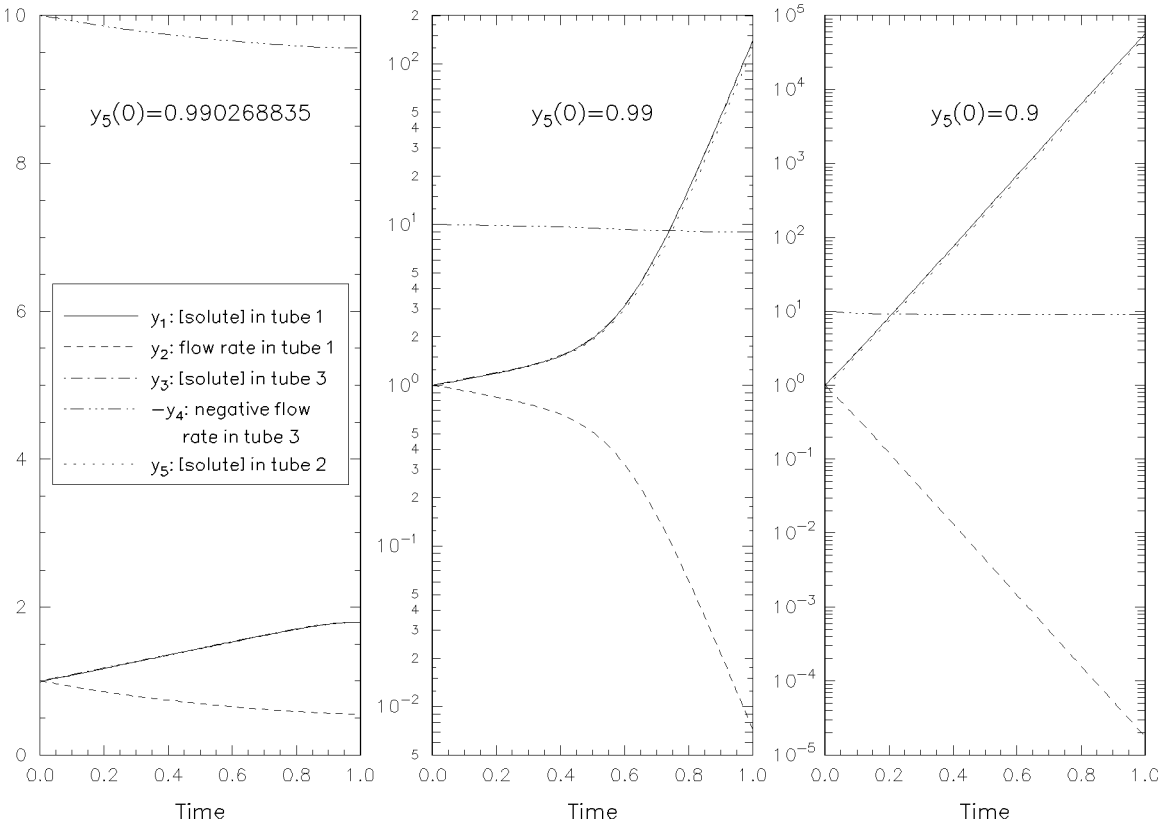


Figure D.4: Results of a kidney model.

The contents of MESSAGES *file after the computation*			
***	1	2	3
The testd***.out was calculated within	0.49s.	1.27s.	41.09s.
Method flag:	21	21	21
Relative error bound:	1.000E-06	1.000E-06	1.000E-06
Initial step size:	1.000E-08	1.000E-08	1.000E-08
The step size last successfully used:	1.213E-02	4.331E-04	1.109E-04
The order last successfully used:	4	3	3
The cumulative number of steps taken:	134	446	15515
The cumulative number of Diffun calls:	179	690	31885
The cumulative number of Jacobian evaluations:	28	122	9432
The number of the output points:	11	11	11

D.5 Test E: A Laser Oscillator Model

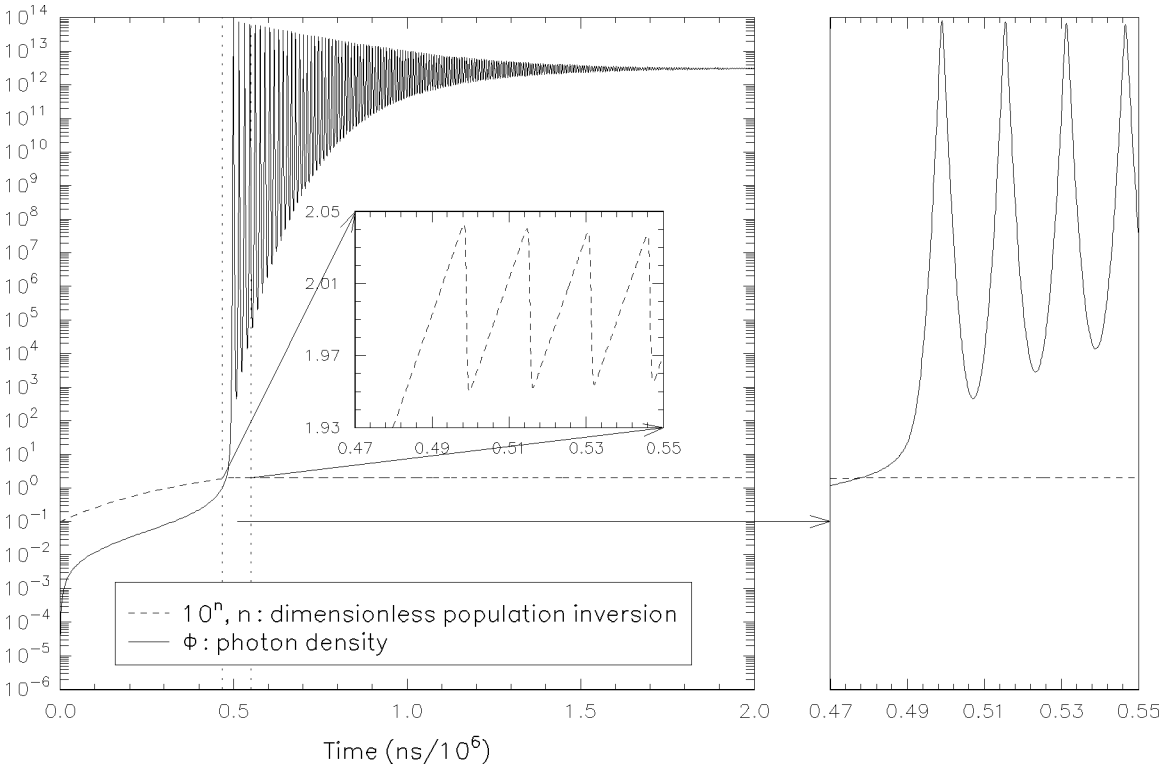


Figure D.5: Result of a laser oscillator model.

The contents of MESSAGES *file after the computation*		
The teste.out was calculated within	33.51s.	584.44s.
Method flag:	21	21
Relative error bound:	1.000E-06	1.000E-16
Initial step size:	1.000E-10	1.000E-14
The step size last successfully used:	9.611E+01	1.242E+00
The order last successfully used:	4	5
The cumulative number of steps taken:	31912	542800
The cumulative number of Diffun calls:	32315	547810
The cumulative number of Jacobian evaluations:	6290	519300
The number of the output points:	5001	20001

Appendix E

Results of the Examples

This appendix contains the most important results of the examples included in ZFIT. Studying them can be useful for understanding the process of fitting better. These results can also be repeated by running the example files. This part gives a possibility to check whether the user's installed version works well. The results calculated by the user cannot be significantly different from data given here because the user may calculate with different accuracy.

Results presented here contain some figures and the contents of the `RESULT.*` and/or `MODEL*.RES` files. The figures show the experimental data pairs and the calculated curves. The interpretation of them are explained in *Chapter 7*. The experimental points are denoted by symbols and the calculated curves are marked by lines. The files contain the parameter estimations and their statistical description. The interpretation of these files are in *Section 10.2* and *Chapter 12*.

E.1 Results of the First Example

The contents of the first and last `RESULT.*` files:

```
ITERATION 1;     WSTD: 3.630577557E-0001;    NMD:136;    MP: 1.000E-0002
******** Initial Parameter Value(s):
p1o   =  3.00000000000000000E-0002
p2o   =  3.00000000000000000E+0000
p3o   =  1.00000000000000000E-0001
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  1.97403528165027235E-0002  1.591E-0003  &      8.06    -1.026E-0002
  p2  3.63277853424576554E+0000  2.853E-0001  &      7.85     6.328E-0001
  p3  6.27585569040004771E-0002  1.475E-0002  &     23.50    -3.724E-0002
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0002
                                    Relative Step: 1.0000
                          Number of Measured Data: 136
                       Number of Fitted Parameters: 3
             Weighted Standard Deviation of Data: 3.63057755700680697E-0001
           Unweighted Standard Deviation of Data: 3.63057755700680697E-0001
Correlation between Measured and Calculated Values: 1.44345847520275755E-0001
                     Coefficient of Determination:-4.60279609805234242E-0001
                                        R-squared: 3.01321188871780848E-0002
                       Model Selection Criterion:-4.22745578007915772E-0001
```

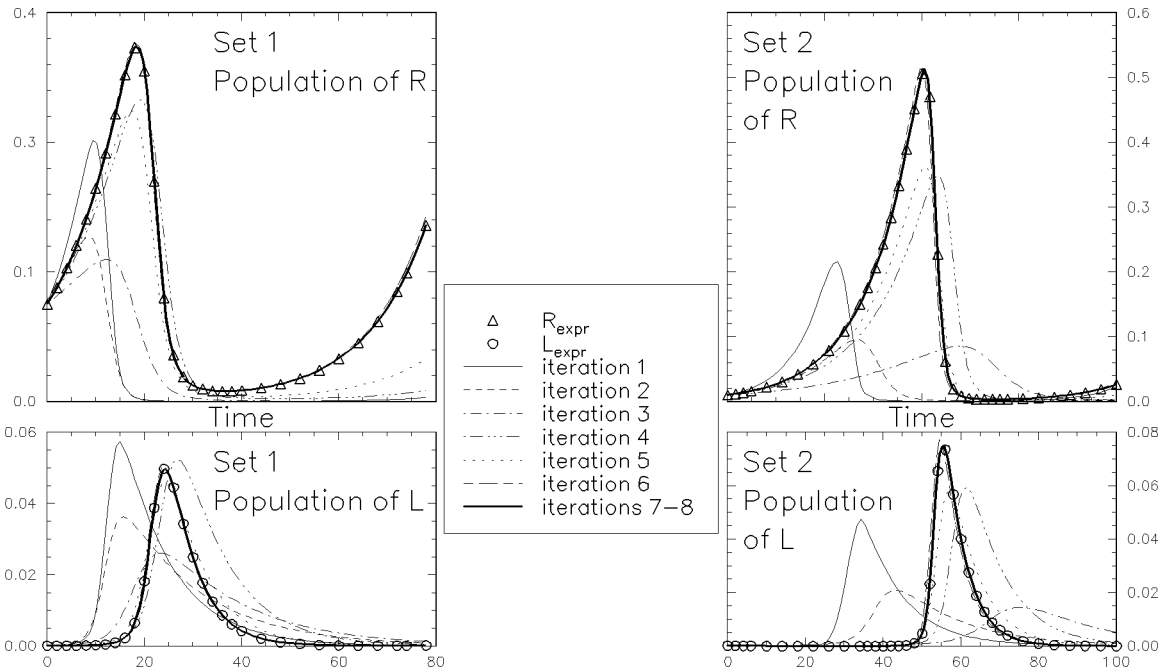


Figure E.1: Experimental and calculated curves in Example 1.

```
******** Residual Analysis:
Serial Correlation: 1.0458835872877467 6E+0001
Heteroscedactivity: 8.9877040321941749 7E-0001
          Skewness:-3.8642660166659236 6E+0000
          Kurtosis: 4.7830971931786520 4E+0000
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p3
  p1 { .570}-.482  .442
  p2  -.400{ .497}-.322
  p3   .346 -.138{ .459}
******** 95% Univariate Confidence Interval(s):
  p1  1.6593972869267710 4E-0002 - 2.2886732763737736 6E-0002
  p2  3.0686234503276627 4E+0000 - 4.1969336181638683 5E+0000
  p3  3.3591255397899220 0E-0002 - 9.1925858410101734 2E-0002
******** 95% Supporting Plane Confidence Interval(s):
  p1  1.5235459909693039 4E-0002 - 2.4245245723312407 6E-0002
  p2  2.8250381219294859 9E+0000 - 4.4405189465620451 0E+0000
  p3  2.0997685086506460 9E-0002 - 1.0451942872149493 E-0001
--------------------------------------------------------------------------------
ITERATION 8;     WSTD: 1.118981212E-0005;    NMD:136;    MP: 1.000E-0008
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  1.9999998660971899 2E-0002  1.507E-0008  &      0.00    -1.435E-0007
  p2  2.0000003542545107 5E+0000  3.742E-0006  &      0.00     7.592E-0006
  p3  1.9999999828433080 8E-0001  6.440E-0007  &      0.00    -1.028E-0006
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0008
                                    Relative Step: 1.0000
                          Number of Measured Data: 136
                       Number of Fitted Parameters: 3
              Weighted Standard Deviation of Data: 1.1189812124593483 8E-0005
            Unweighted Standard Deviation of Data: 1.1189812124593483 8E-0005
Correlation between Measured and Calculated Values: 9.9999999376904080E-0001
                     Coefficient of Determination: 9.9999999861282845 0E-0001
                                        R-squared: 9.9999999907868799 7E-0001
                    Model Selection Criterion: 2.0351881371775238 4E+0001
```

```
******** Residual Analysis:
Serial Correlation: 7.02654568235005959E+0000
Heteroscedactivity: 9.75474286461950264E-0001
          Skewness: 8.00119742084070588E+0000
          Kurtosis: 3.99289242972430031E+0001
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p3
  p1 { .797}-.482  .297
  p2  -.774{ .839} .512
  p3   .724  .784{ .805}
******** 95% Univariate Confidence Interval(s):
  p1  1.99999688633151917E-0002 - 2.00000284586286067E-0002
  p2  1.99999295338031166E+0000 - 2.00000775512870983E+0000
  p3  1.99998724707861234E-0001 - 2.00001271860800381E-0001
******** 95% Supporting Plane Confidence Interval(s):
  p1  1.99999559975763168E-0002 - 2.00000413243674816E-0002
  p2  1.99998975790370673E+0000 - 2.00001095060531476E+0000
  p3  1.99998174815550725E-0001 - 2.00001821753110890E-0001
```

E.2 Results of the Second Example

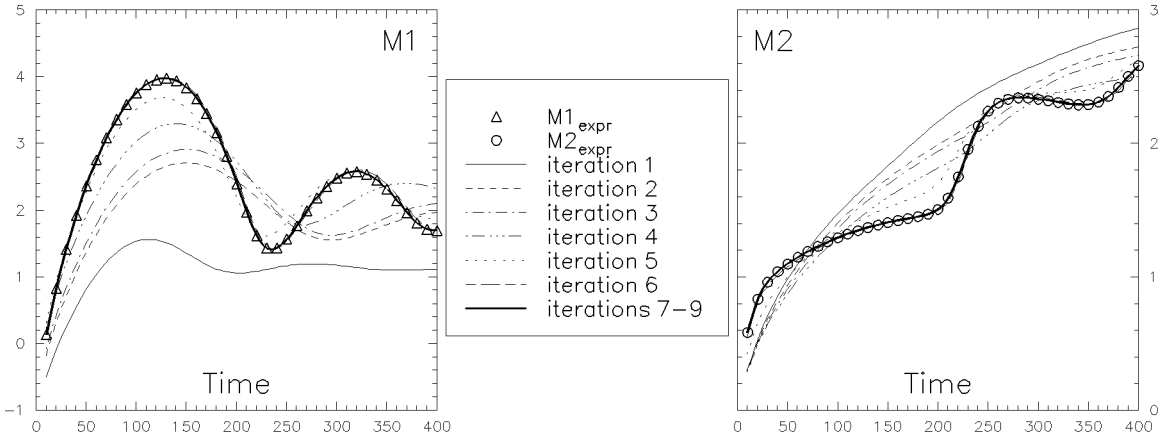


Figure E.2: Experimental and calculated curves in Example 2.

The contents of the first and last `RESULT.*` files:

```
ITERATION 1;     WSTD: 3.238687375E-0001;    NMD:80;    MP: 5.120E+0000
******** Initial Parameter Value(s):
p1o   =  2.50000000000000000E-0002
p2o   =  2.50000000000000000E+0000
p3o   =  5.00000000000000000E-0002
p4o   =  2.00000000000000000E-0001
p5o   =  1.50000000000000000E+0001
p6o   =  2.00000000000000000E+0001
******** Result(s):
       Estimate                    Std. Dev. (abs & %)          dp
  p1  6.46500575256218950E-0003  2.255E-0002  &    348.85    -1.853E-0002
  p2  1.50501666799961693E+0000  1.873E+0000  &    124.44    -9.950E-0001
  p3  5.91003335543614982E-0002  6.001E-0002  &    101.53     9.100E-0003
  p4  3.59134987994330012E-0001  1.724E+0000  &    480.01     1.591E-0001
  p5  1.46630792479197346E+0001  3.914E+0000  &     26.69    -3.369E-0001
  p6  1.81038169058130209E+0001  2.388E+0001  &    131.90    -1.896E+0000
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 5.120E+0000
                                    Relative Step: 1.0000
```

```
                        Number of Measured Data: 80
                    Number of Fitted Parameters: 6
          Weighted Standard Deviation of Data: 3.2386873746551341E-0001
        Unweighted Standard Deviation of Data: 3.2386873746551341E-0001
Correlation between Measured and Calculated Values: 7.7048523388492193E-0001
                 Coefficient of Determination:-2.0574540119960974E-0001
                                    R-squared: 8.5174197449272831E-0001
                      Model Selection Criterion:-3.3709796590170653E-0001
******** Residual Analysis:
Serial Correlation: 8.7632708221878928E+0000
Heteroscedactivity: 7.6752474645879182E-0001
         Skewness:-4.4402279414730930E+0000
         Kurtosis:-1.0873680954600548E+0000
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p3    p4    p5    p6
  p1 { .945}-.698 -.837  .865 -.049  .227
  p2  -.641{ .950} .763 -.492 -.041 -.185
  p3   .526  .848{ .978}-.894  .102 -.268
  p4   .785  .807 -.899{ .979}-.136  .251
  p5   .129 -.326  .077 -.000{ .976}-.950
  p6   .127 -.315  .061 -.000 -.975{ .977}
******** 95% Univariate Confidence Interval(s):
  p1 -3.8425730143057235E-0002 - 5.1355741648181614E-0002
  p2 -2.2226405273043126E+0000 - 5.2326738633035465E+0000
  p3 -6.0337888106998351E-0002 - 1.7853855521572135E-0001
  p4 -3.0721737399220328E+0000 - 3.7904437159106928E+0000
  p5  6.8724472206104115E+0000 - 2.2453711275229058E+0001
  p6 -2.9426451633428488E+0001 - 6.5634085445054530E+0001
******** 95% Supporting Plane Confidence Interval(s):
  p1 -7.5916866904416477E-0002 - 8.8846878409540856E-0002
  p2 -5.3358464688055317E+0000 - 8.3458798048047656E+0000
  p3 -1.6008841822197797E-0001 - 2.7828908533070097E-0001
  p4 -5.9378800426184743E+0000 - 6.6561500186071343E+0000
  p5  3.6599004262101543E-0001 - 2.8960168453218454E+0001
  p6 -6.9122031635776444E+0001 - 1.0532966544740249E+0002
--------------------------------------------------------------------------------
ITERATION 9;     WSTD: 1.886654848E-0006;    NMD:80;    MP: 1.000E-0007
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  2.0000111670013259E-0002  6.980E-0008  &      0.00    -5.508E-0009
  p2  1.9999938524253047E+0000  5.174E-0006  &      0.00     4.883E-0006
  p3  9.9999913496824394E-0002  1.727E-0007  &      0.00     1.613E-0007
  p4  5.0000052555923319E-0001  2.426E-0006  &      0.00    -7.617E-0007
  p5  9.9999820716583892E+0000  1.738E-0005  &      0.00    -5.027E-0006
  p6  4.0000047168311451E+0001  1.113E-0004  &      0.00    -4.089E-0005
******** Goodness-of-Fit Statistics:
                             Marquardt Parameter: 1.000E-0007
                                   Relative Step: 1.0000
                        Number of Measured Data: 80
                    Number of Fitted Parameters: 6
          Weighted Standard Deviation of Data: 1.8866548481837283E-0006
        Unweighted Standard Deviation of Data: 1.8866548481837283E-0006
Correlation between Measured and Calculated Values: 9.9999999980235367E-0001
                 Coefficient of Determination: 9.9999999959083122E-0001
                                    R-squared: 9.9999999994968875E-0001
                      Model Selection Criterion: 2.3769478486559742E+0001
******** Residual Analysis:
Serial Correlation: 8.7597674917206082E-0001
Heteroscedactivity: 2.1030526104072509E+0000
         Skewness: 3.8486089447351734E+0000
         Kurtosis: 5.6138028656732474E+0000
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p3    p4    p5    p6
  p1 { .970}-.928 -.757  .568  .067 -.343
  p2  -.888{ .990} .920 -.690 -.043  .287
  p3   .669  .888{ .981}-.836  .032  .169
  p4   .356  .478 -.728{ .881}-.078 -.085
  p5  -.351 -.137  .084  .000{ .951}-.913
```

```
  p6  -.350 -.103  .047  .000 -.949{ .957}
******** 95% Univariate Confidence Interval(s):
  p1  1.9999972734359673E-0002 - 2.0000250605666843E-0002
  p2  1.9999835534621894E+0000 - 2.0000041513884198E+0000
  p3  9.9999569702331827E-0002 - 1.0000025729131696E-0001
  p4  4.9999569674614437E-0001 - 5.0000535437232201E-0001
  p5  9.9999474776441491E+0000 - 1.0000016665672629E+0001
  p6  3.9999825630207599E+0001 - 4.0000268706415302E+0001
******** 95% Supporting Plane Confidence Interval(s):
  p1  1.9999856700271890E-0002 - 2.0000366639754628E-0002
  p2  1.9999749521366163E+0000 - 2.0000127527139930E+0000
  p3  9.9999282577470061E-0002 - 1.0000054441617873E-0001
  p4  4.9999166389419182E-0001 - 5.0000938722427455E-0001
  p5  9.9999185859609739E+0000 - 1.0000045557355804E+0001
  p6  3.9999640609508258E+0001 - 4.0000453727114644E+0001
```

E.3 Results of the Third Example

The contents of the first and last `RESULT.*` files:

```
ITERATION 1;     WSTD: 3.469040847E-0001;    NMD:136;    MP: 1.000E-0002
******** Initial Parameter Value(s):
p1o   =  3.00000000000000000E-0002
p2o   =  3.00000000000000000E+0000
p3o   =  1.00000000000000000E-0001
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  2.08402488216224994E-0002  1.622E-0003  &      7.78    -9.160E-0003
  p2  3.67102140922351106E+0000  2.903E-0001  &      7.91     6.710E-0001
  p3  6.42644955015661012E-0002  1.553E-0002  &     24.17    -3.574E-0002
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0002
                                   Relative Step: 1.0000
                          Number of Measured Data: 136
                     Number of Fitted Parameters: 3
              Weighted Standard Deviation of Data: 3.46904084675592564E-0001
            Unweighted Standard Deviation of Data: 3.46904084675592564E-0001
Correlation between Measured and Calculated Values: 1.25286560598319419E-0001
                     Coefficient of Determination:-4.60143848574373355E-0001
                                        R-squared: 2.69124391253096836E-0002
                        Model Selection Criterion:-4.22652604346463259E-0001
******** Residual Analysis:
Serial Correlation: 1.02195268489600323E+0001
Heteroscedactivity: 9.39669760187727769E-0001
          Skewness:-4.50598480222917793E+0000
          Kurtosis: 4.31148622653779276E+0000
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p3
  p1 { .569}-.490  .412
  p2  -.431{ .496}-.272
  p3   .332 -.088{ .419}
******** 95% Univariate Confidence Interval(s):
  p1  1.76322813424111820E-0002 - 2.40482695090038807E-0002
  p2  3.09684393600920568E+0000 - 4.24519888243781644E+0000
  p3  3.35509046092147708E-0002 - 9.49780863939174317E-0002
******** 95% Supporting Plane Confidence Interval(s):
  p1  1.62471005426986186E-0002 - 2.54333971005463802E-0002
  p2  2.84893123902454948E+0000 - 4.49311157942247264E+0000
  p3  2.02896926888295856E-0002 - 1.08232983143026170E-0001
--------------------------------------------------------------------------------
ITERATION 8;     WSTD: 3.860470752E-0002;    NMD:136;    MP: 1.000E-0007
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  1.99717786500984004E-0002  5.326E-0005  &      0.27     4.474E-0008
  p2  2.02207660023703541E+0000  1.395E-0002  &      0.69     7.439E-0007
  p3  2.00700400515271498E-0001  2.289E-0003  &      1.14     2.844E-0006
```

```
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0007
                                   Relative Step: 1.0000
                            Number of Measured Data: 136
                         Number of Fitted Parameters: 3
               Weighted Standard Deviation of Data: 3.86047075228918196E-0002
             Unweighted Standard Deviation of Data: 3.86047075228918196E-0002
Correlation between Measured and Calculated Values: 9.90928954425397971E-0001
                      Coefficient of Determination: 9.81917546828137748E-0001
                                         R-squared: 9.87949262486146075E-0001
                           Model Selection Criterion: 3.96869560187914643E+0000
******** Residual Analysis:
Serial Correlation:-2.99150319302475425E+0000
Heteroscedactivity: 1.30144690739890364E+0000
          Skewness: 1.60532425567912851E+0000
          Kurtosis: 1.34824652706455396E+0001
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p3
  p1 { .804}-.510  .270
  p2  -.787{ .848} .515
  p3   .723  .788{ .806}
******** 95% Univariate Confidence Interval(s):
  p1  1.98664512324821290E-0002 - 2.00771060677146717E-0002
  p2  1.99449053546237119E+0000 - 2.04966266501169964E+0000
  p3  1.96173700386638441E-0001 - 2.05227100643904556E-0001
******** 95% Supporting Plane Confidence Interval(s):
  p1  1.98209739969829700E-0002 - 2.01225833032138307E-0002
  p2  1.98257969598064204E+0000 - 2.06157350449342879E+0000
  p3  1.94219206367484622E-0001 - 2.07181594663058374E-0001
```

E.4 Results of the Fourth Example

The contents of the first and last `RESULT.*` files:

```
ITERATION 1;     WSTD: 3.532860958E-0001;    NMD:136;    MP: 1.000E-0002
******** Initial Parameter Value(s):
p1o   =  3.00000000000000000E-0002
p2o   =  3.00000000000000000E+0000
p3o   =  1.00000000000000000E-0001
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  1.97280153152514049E-0002  1.534E-0003  &      7.78    -1.027E-0002
  p2  3.69892855367477383E+0000  2.889E-0001  &      7.81     6.989E-0001
  p3  6.02391742891498421E-0002  1.376E-0002  &     22.85    -3.976E-0002
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0002
                                   Relative Step: 1.0000
                            Number of Measured Data: 136
                         Number of Fitted Parameters: 3
               Weighted Standard Deviation of Data: 3.53286095751125124E-0001
             Unweighted Standard Deviation of Data: 3.53286095751125124E-0001
Correlation between Measured and Calculated Values: 1.36345507987663334E-0001
                      Coefficient of Determination:-5.10911528360752197E-0001
                                         R-squared: 2.10391695955539116E-0002
                           Model Selection Criterion:-4.56830776921608739E-0001
******** Residual Analysis:
Serial Correlation: 1.02788846903423804E+0001
Heteroscedactivity: 9.08058042673020070E-0001
          Skewness:-3.38748720897136332E+0000
          Kurtosis: 5.02691111129724024E+0000
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p3
  p1 { .578}-.495  .452
  p2  -.405{ .513}-.345
  p3   .345 -.157{ .473}
```

```
******** 95% Univariate Confidence Interval(s):
  p1  1.6693277040547624 3E-0002 - 2.2762753589955185 6E-0002
  p2  3.1275921022686598 0E+0000 - 4.2702650050808878 6E+0000
  p3  3.3019175544975605 6E-0002 - 8.7459173033324078 6E-0002
******** 95% Supporting Plane Confidence Interval(s):
  p1  1.5382967623299604 9E-0002 - 2.4073063007203204 9E-0002
  p2  2.8809060737123521 6E+0000 - 4.5169510336371955 0E+0000
  p3  2.1266392450211967 7E-0002 - 9.9211956128087716 6E-0002
--------------------------------------------------------------------------------
ITERATION 8;     WSTD: 4.772514207E-0002;    NMD:136;    MP: 1.000E-0008
******** Result(s):
       Estimate                   Std. Dev. (abs & %)          dp
  p1  1.9829849234768755 2E-0002  6.708E-0005  &      0.34    -3.461E-0008
  p2  2.0262723681648642 0E+0000  1.714E-0002  &      0.85    -7.349E-0006
  p3  1.9549593678643316 6E-0001  3.017E-0003  &      1.54    -9.489E-0007
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0008
                                  Relative Step: 1.0000
                          Number of Measured Data: 136
                     Number of Fitted Parameters: 3
             Weighted Standard Deviation of Data: 4.7725142070825181 5E-0002
           Unweighted Standard Deviation of Data: 4.7725142070825181 5E-0002
Correlation between Measured and Calculated Values: 9.8655655413305983 4E-0001
                    Coefficient of Determination: 9.7242722961918588 0E-0001
                                       R-squared: 9.8213484927351218 5E-0001
                      Model Selection Criterion: 3.5468089265962728 0E+0000
******** Residual Analysis:
Serial Correlation: 2.1233174224599061 7E+0000
Heteroscedactivity: 2.3784276155282521 8E-0001
          Skewness:-3.1260250798685087 8E+0000
          Kurtosis:-9.9062655565619446 E-0002
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p3
  p1 { .811}-.445  .341
  p2  -.782{ .848} .527
  p3   .757  .807{ .831}
******** 95% Univariate Confidence Interval(s):
  p1  1.9697177559926564 8E-0002 - 1.9962520909610945 7E-0002
  p2  1.9923680795870881 6E+0000 - 2.0601766567426402 3E+0000
  p3  1.8952866273677266 4E-0001 - 2.0146321083609366 7E-0001
******** 95% Supporting Plane Confidence Interval(s):
  p1  1.9639893890228637 4E-0002 - 2.0019804579308873 0E-0002
  p2  1.9777292196304117 7E+0000 - 2.0748155166993166 3E+0000
  p3  1.8695217189150097 8E-0001 - 2.0403970168136535 3E-0001
```

E.5 Results of the Fifth Example

The contents of the first and last `RESULT.*` files:

```
ITERATION 1;     WSTD: 1.467381060E-0001;    NMD:200;    MP: 1.000E-0002
******** Initial Parameter Value(s):
p1o   =  1.0000000000000000 0E-0002
p2o   =  4.0000000000000000 0E+0001
p3o   =  1.0000000000000000 0E+0005
p4o   =  1.0000000000000000 0E+0006
******** Result(s):
       Estimate                   Std. Dev. (abs & %)          dp
  p1  1.0718401865128046 2E-0002  4.178E-0003  &     38.98     7.184E-0004
  p2  5.7956944645110911 6E+0001  8.741E+0000  &     15.08     1.796E+0001
  p4  7.8539039439242550 1E+0005  8.523E+0005  &    108.52    -2.146E+0005
```

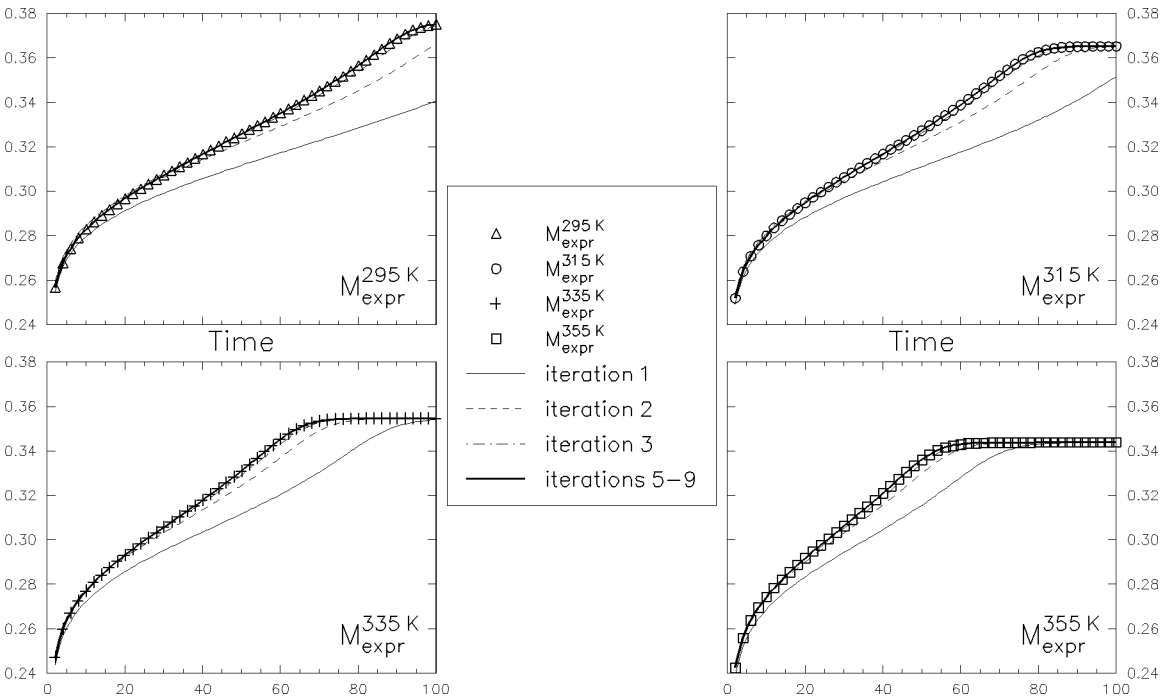


Figure E.3: Experimental and calculated curves in Example 5.

```
******** Goodness-of-Fit Statistics:
                                Marquardt Parameter: 1.000E-0002
                                      Relative Step: 1.0000
                                Number of Measured Data: 200
                            Number of Fitted Parameters: 3
                  Weighted Standard Deviation of Data: 1.46738106049011796E-0001
                Unweighted Standard Deviation of Data: 1.46738106049011796E-0001
Correlation between Measured and Calculated Values: 9.66795751213125032E-0001
                         Coefficient of Determination: 7.89497751612626527E-0001
                                             R-squared: 9.97600810665815813E-0001
                           Model Selection Criterion: 1.52825894474157340E+0000
******** Residual Analysis:
Serial Correlation: 1.39452821297448309E+0001
Heteroscedactivity: 3.99590466811725567E+0000
          Skewness:-8.14072576279946595E+0000
          Kurtosis:-2.33736932976701967E+0000
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p4
  p1 { .998}-.471 -.974
  p2  -.970{ .972} .266
  p4  -.998 -.964{ .998}
******** 95% Univariate Confidence Interval(s):
  p1  2.47991280283886181E-0003 - 1.89568909274172305E-0002
  p2  4.07210140268476934E+0001 - 7.51928752633741297E+0001
  p4 -8.95271869308991607E+0005 - 2.46605265809384261E+0006
******** 95% Supporting Plane Confidence Interval(s):
  p1 -1.06238793010897155E-0003 - 2.24991916603650639E-0002
  p2  3.33100859791812904E+0001 - 8.26038033110405327E+0001
  p4 -1.61790573136003286E+0006 - 3.18868652014488386E+0006
------------------------------------------------------------------------------
ITERATION 9;     WSTD: 2.556543092E-0006;    NMD:200;    MP: 1.000E-0007
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  7.99993646375081520E-0003  6.518E-0008  &      0.00    -1.906E-0008
```

```
  p2  8.00001645031921993E+0001  1.683E-0004  &      0.00     4.064E-0004
  p4  1.30002413357528086E+0006  2.978E+0001  &      0.00    -1.234E+0001
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0007
                                   Relative Step: 1.0000
                          Number of Measured Data: 200
                        Number of Fitted Parameters: 3
              Weighted Standard Deviation of Data: 2.55654309165214812E-0006
            Unweighted Standard Deviation of Data: 2.55654309165214812E-0006
Correlation between Measured and Calculated Values: 9.99999999971162058E-0001
                     Coefficient of Determination: 9.99999999936103491E-0001
                                        R-squared: 9.99999999999271743E-0001
                       Model Selection Criterion: 2.34437563833822497E+0001
******** Residual Analysis:
Serial Correlation: 3.83866460961244865E+0000
Heteroscedactivity:-4.84544628913242251E+0000
          Skewness: 2.10200246406641335E+0001
          Kurtosis: 2.99114527916020316E+0001
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p4
  p1 { .998}-.568 -.985
  p2  -.945{ .955} .425
  p4  -.998 -.933{ .998}
******** 95% Univariate Confidence Interval(s):
  p1  7.99980794107419610E-0003 - 8.00006498642743429E-0003
  p2  7.99998325397789567E+0001 - 8.00004964666054420E+0001
  p4  1.29996541207548847E+0006 - 1.30008285507507325E+0006
******** 95% Supporting Plane Confidence Interval(s):
  p1  7.99975268021651381E-0003 - 8.00012024728511658E-0003
  p2  7.99996898055672402E+0001 - 8.00006392008171585E+0001
  p4  1.29994016360939455E+0006 - 1.30010810354116717E+0006
```

E.6 Results of the Sixth Example

The contents of the last `RESULT1.*` file:

```
ITERATION 10;     WSTD: 1.890277197E-0001;    NMD:136;    MP: 1.000E-0011
******** Result(s):
       Estimate                   Std. Dev. (abs & %)          dp
  p1  1.82143191757796170E-0002  2.012E-0004  &      1.10    -1.686E-0006
  p2  3.13150177242469822E+0000  7.427E-0002  &      2.37    -1.007E-0003
  p5  2.66958995291699909E-0001  1.119E-0002  &      4.19    -2.538E-0004
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0011
                                   Relative Step: 1.0000
                          Number of Measured Data: 136
                        Number of Fitted Parameters: 3
              Weighted Standard Deviation of Data: 1.89027719702776542E-0001
            Unweighted Standard Deviation of Data: 1.89027719702776542E-0001
Correlation between Measured and Calculated Values: 8.37587044299244475E-0001
                     Coefficient of Determination: 6.04145510771008685E-0001
                                        R-squared: 7.37086957785621173E-0001
                       Model Selection Criterion: 8.82590939637597585E-0001
******** Residual Analysis:
Serial Correlation: 9.37650660079877256E+0000
Heteroscedactivity: 9.53527822101977486E-0001
          Skewness: 1.18043734092795845E+0001
          Kurtosis: 3.00433669820094546E+0001
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p5
  p1 { .900}-.474  .526
  p2  -.858{ .880} .381
  p5   .868  .842{ .889}
******** 95% Univariate Confidence Interval(s):
  p1  1.78163156679822946E-0002 - 1.86123226835769394E-0002
```

```
  p2  2.98462598716529731E+0000 - 3.27837755768409913E+0000
  p5  2.44831216612112585E-0001 - 2.89086773971287233E-0001
******** 95% Supporting Plane Confidence Interval(s):
  p1  1.76444696314684930E-0002 - 1.87841687200907409E-0002
  p2  2.92120940650155751E+0000 - 3.34179413834783892E+0000
  p5  2.35277102167276868E-0001 - 2.98640888416122950E-0001
```

The contents of the last RESULT2.* file:

```
ITERATION 11;     WSTD: 8.608866391E-0002;    NMD:136;    MP: 1.000E-0012
******** Result(s):
       Estimate                   Std. Dev. (abs & %)          dp
  p1  2.0568621831720664 4E-0002  1.441E-0004  &      0.70     7.088E-0007
  p3  1.6700167044445579 0E+0000  3.035E-0002  &      1.82     8.167E-0005
  p5  1.7035857059717873 8E-0001  5.347E-0003  &      3.14     2.937E-0005
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0012
                                    Relative Step: 1.0000
                          Number of Measured Data: 136
                       Number of Fitted Parameters: 3
              Weighted Standard Deviation of Data: 8.6088663910215434 5E-0002
            Unweighted Standard Deviation of Data: 8.6088663910215434 5E-0002
Correlation between Measured and Calculated Values: 9.6057422065200969 3E-0001
                     Coefficient of Determination: 9.1789369293797085 7E-0001
                                        R-squared: 9.4546779293393685 2E-0001
                    Model Selection Criterion: 2.4556227967099746 1E+0000
******** Residual Analysis:
Serial Correlation: 1.01143411904484806E+0001
Heteroscedactivity: 1.40798747151212989E+0000
          Skewness:-1.48373965548418083E+0001
          Kurtosis: 2.48654650102896266E+0001
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
       p1    p3    p5
  p1 { .760}-.430  .246
  p3  -.742{ .843} .597
  p5   .694  .803{ .817}
******** 95% Univariate Confidence Interval(s):
  p1  2.02836176881852452E-0002 - 2.08536259752560837E-0002
  p3  1.61000240506833028E+0000 - 1.73003100382078552E+0000
  p5  1.59783483573288745E-0001 - 1.80933657621068732E-0001
******** 95% Supporting Plane Confidence Interval(s):
  p1  2.01605614047679816E-0002 - 2.09766822586733473E-0002
  p3  1.58409002168499264E+0000 - 1.75594338720412316E+0000
  p5  1.55217476602894262E-0001 - 1.85499664591463215E-0001
```

The contents of the last RESULT3.* file:

```
ITERATION 8;     WSTD: 1.989285566E-0005;    NMD:136;    MP: 1.000E-0009
******** Result(s):
       Estimate                   Std. Dev. (abs & %)          dp
  p1  2.0000002937687234 1E-0002  2.798E-0008  &      0.00     4.975E-0008
  p4  1.9999973853626148 4E+0000  6.732E-0006  &      0.00     3.548E-0005
  p5  1.9999966814481438 3E-0001  1.165E-0006  &      0.00     6.286E-0007
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0009
                                    Relative Step: 1.0000
                          Number of Measured Data: 136
                       Number of Fitted Parameters: 3
              Weighted Standard Deviation of Data: 1.9892855660887907 1E-0005
            Unweighted Standard Deviation of Data: 1.9892855660887907 1E-0005
Correlation between Measured and Calculated Values: 9.9999999806023381 5E-0001
                     Coefficient of Determination: 9.9999999561591620 1E-0001
                                        R-squared: 9.9999999708824115 8E-0001
                    Model Selection Criterion: 1.9201167525669453 6E+0001
******** Residual Analysis:
Serial Correlation: 7.87514765705847046E+0000
Heteroscedactivity: 9.32822569810891016E-0001
          Skewness: 7.14347146677293443E+0000
```

```
         Kurtosis: 1.85830093863069619E+0001
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p4    p5
  p1 { .813}-.478  .347
  p4  -.784{ .839} .481
  p5   .749  .785{ .814}
******** 95% Univariate Confidence Interval(s):
  p1  1.9999947603234826OE-0002 - 2.0000058272139642 2E-0002
  p4  1.9999840714632355 6E+0000 - 2.0000106992619941 1E+0000
  p5  1.9999736495553304 2E-0001 - 2.0000197133409572 3E-0001
******** 95% Supporting Plane Confidence Interval(s):
  p1  1.9999923711469696 6E-0002 - 2.0000082163904771 7E-0002
  p4  1.9999783229188281 6E+0000 - 2.0000164478064015 1E+0000
  p5  1.9999637050713853 0E-0001 - 2.0000296578249023 6E-0001
```

E.7 Results of the Seventh Example

The main parts of the `MODEL1.RES` file:

```
Date: 5/11/1994y.  Time: 18:10. Relative residuals were used.
Modifying percent: 1.000E+0000. Differentiation method: +.
ITERATION 1;     WSTD: 5.807734083E-0001;    NMD:922;    MP: 0.000E+0000
******** Initial Parameter Value(s):
p1o   =  1.0000000000000000E+0005
p2o   =  2.0000000000000000E+0006
p3o   =  2.0000000000000000E+0002
p4o   =  2.0000000000000000E+0002
p5o   =  1.0000000000000000E+0001
p6o   =  1.1100000000000000E+0001
p7o   =  1.8000000000000000E+0004
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  4.6809090899754904 8E+0005  1.286E+0005  &     27.47     3.681E+0005
  p2  7.2188768166276765 7E+0007  7.404E+0008  &   1025.59     7.019E+0007
  p3  7.1789976168307915 4E+0003  4.048E+0004  &    563.84     6.979E+0003
  p4  3.9417147735797677 5E+0004  1.217E+0006  &   3087.34     3.922E+0004
-------------------------------------------------------------------------------
    .                                                                        .
    .                                                                        .
    .                                                                        .
-------------------------------------------------------------------------------
ITERATION 11;     WSTD: 3.290022080E-0002;    NMD:922;    MP: 1.000E-0008
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  1.1176681921833043 6E+0006  6.069E+0003  &      0.54    -1.108E+0001
  p2  5.6150438395624697 7E+0005  2.427E+0004  &      4.32    -1.017E+0002
  p3  2.1071515138699834 0E+0003  4.766E+0001  &      2.26    -3.607E-0001
  p4  4.1970070595443786 4E+0004  1.452E+0003  &      3.46     9.979E+0000
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0008
                                   Relative Step: 1.0000
                          Number of Measured Data: 922
                        Number of Fitted Parameters: 4
              Weighted Standard Deviation of Data: 3.2900220804349729 0E-0002
            Unweighted Standard Deviation of Data: 3.2900220804349729 0E-0002
Correlation between Measured and Calculated Values: 9.9471244607593226 0E-0001
                     Coefficient of Determination: 9.8935047187550090 0E-0001
                                        R-squared: 9.9781593106379690 0E-0001
                    Model Selection Criterion: 4.5335629057792292 9E+0000
******** Residual Analysis:
Serial Correlation: 2.5171004165814818 1E+0001
Heteroscedactivity: 6.7915741913453321 7E-0001
         Skewness:-1.4391254397097616 6E+0000
         Kurtosis: 9.2137047054273092 2E+0000
```

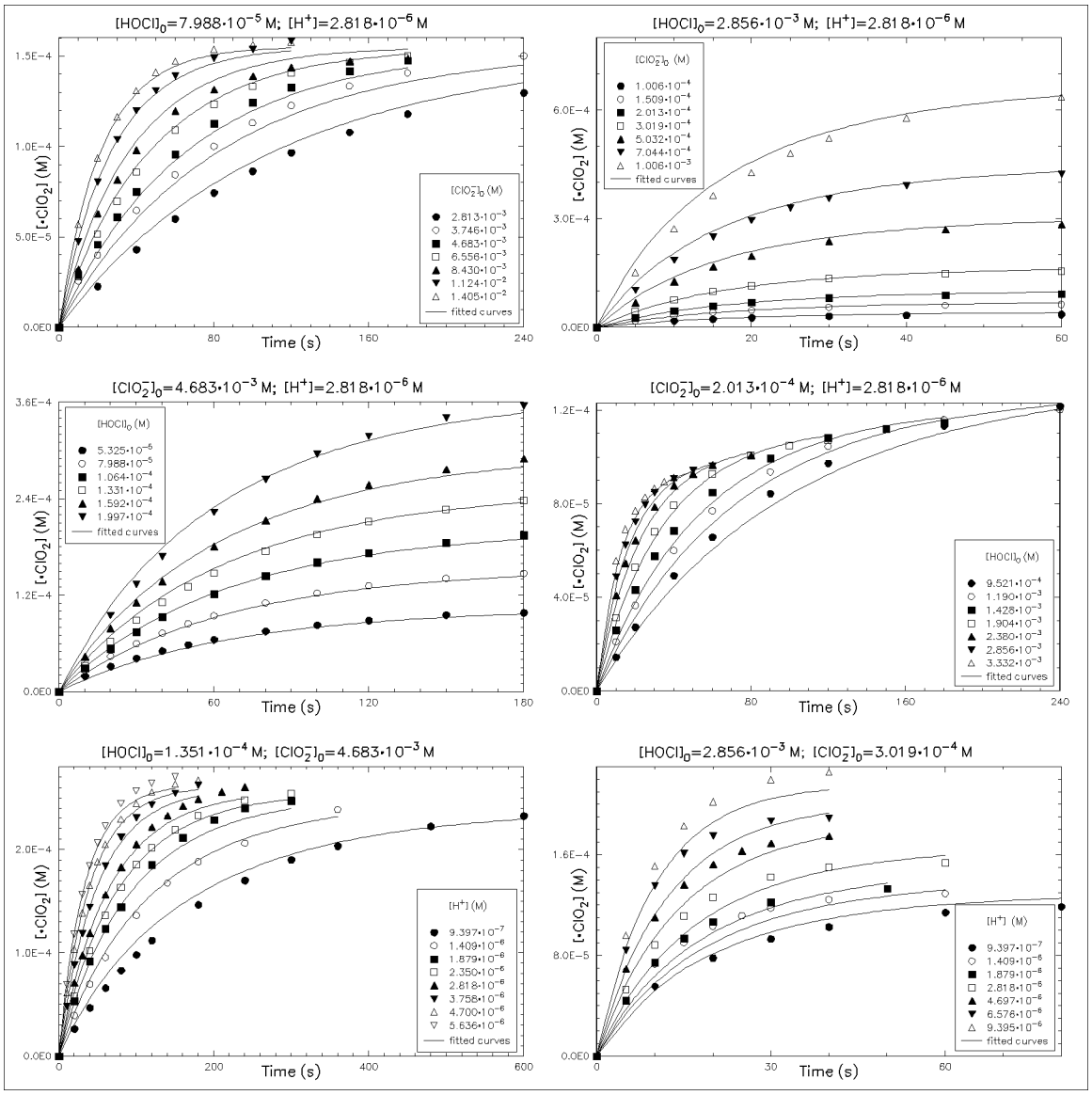


Figure E.4: Experimental and calculated curves in Example 7, first series.

```
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p3    p4
  p1 { .649}-.317 -.057 -.120
  p2  -.630{ .888} .665 -.759
  p3   .147  .412{ .686}-.588
  p4  -.570 -.748 -.055{ .850}
******** 95% Univariate Confidence Interval(s):
  p1  1.10575700324636031E+0006 - 1.12957938112024841E+0006
  p2  5.13863655262468437E+0005 - 6.09145112650025516E+0005
  p3  2.01361908632016654E+0003 - 2.20068394141980027E+0003
  p4  3.91210849511062522E+0004 - 4.48190562397813206E+0004
```

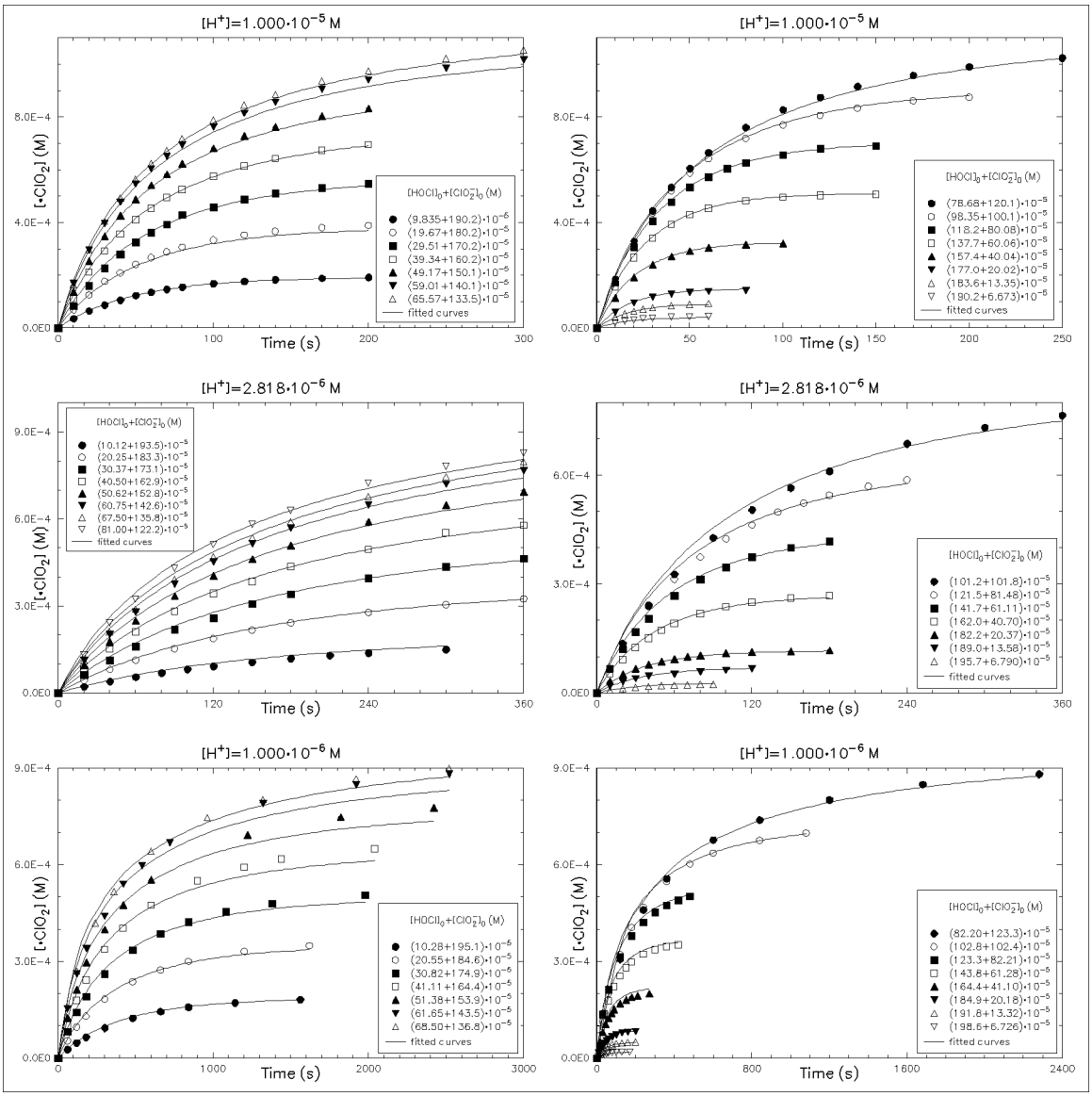


Figure E.5: Experimental and calculated curves in Example 7, second series.

```
******** 95% Supporting Plane Confidence Interval(s):
  p1  1.09893645094607710E+0006 - 1.13639993342053163E+0006
  p2  4.86583752021937229E+0005 - 6.36425015890556724E+0005
  p3  1.96006080450185737E+0003 - 2.25424222323810944E+0003
  p4  3.74897066149193008E+0004 - 4.64504345759682720E+0004
******** The last PARAMTRS.DAT file:
 1.11767926932336616E+0006 * p1
 5.61606078253478758E+0005 * p2
 2.10751216754583897E+0003 * p3
 4.19600912505089282E+0004 * p4
 1.00000000000000000E+0001   p5
 1.11000000000000000E+0001   p6
 1.80000000000000000E+0004   p7

 1.00000E-0008    0.99
```

```
 3.29002083526468578E-0002  1.00000000000000E-0006  1.00000000000000E-0007
12
******** Unweighted Standard Deviation of Data:
   1.22099155169022E-0005 in the case of ABSOLUTE residuals.
   3.29002208043497E-0002 in the case of RELATIVE residuals.
   2.48675908412488E-0002 in the case of ORTHOGONAL residuals.
******** Some pieces of important information from J*.COL file(s):
No.curve Point(s)   Fitting    No.curve Point(s)   Fitting    No.curve Point(s)   Fitting
 1 13 4.07240782926179979E-2   30 13 1.57423741001336888E-2   59  9 2.46645294180903077E-2
 2  9 1.98654047007432047E-2   31 12 1.72668801499736118E-2   60 12 2.86663728199885400E-2
 3 10 1.87357768679272996E-2   32 11 1.15927491103056229E-2   61 10 2.15071780591317771E-2
 4 11 1.40093591123354528E-2   33  9 1.13641854083432843E-2   62 12 3.50388187901211763E-2
 5 13 2.13466545682587504E-2   34  8 1.94567459658693957E-2   63 12 6.18430215019477959E-2
 6 11 1.84832236149922009E-2   35  9 1.73246922192985278E-2   64 12 2.62539432843771508E-2
 7 12 1.68538577451526231E-2   36  9 7.17459315555044663E-2   65 12 3.72671260001480583E-2
 8 11 1.99757176837821613E-2   37 12 6.40028241717535660E-2   66 11 3.14807907028908049E-2
 9 12 2.76316567372554610E-2   38 11 1.82499045258728704E-2   67  7 4.97188566744952609E-2
10 12 1.37018534057893140E-2   39 11 2.16495880603336712E-2   68  9 7.97550010402839333E-2
11 11 1.17519996967102073E-2   40 11 1.59459820055666314E-2   69  7 5.71469723413220393E-2
12 12 2.48639723003647121E-2   41 11 2.16991722195228951E-2   70  8 6.36003127064904674E-2
13 11 1.72256239137960931E-2   42 11 2.30942134104734817E-2   71  8 6.58927965666749511E-2
14 10 2.33753938831044821E-2   43 11 2.55002244425964006E-2   72  7 7.65361420577996720E-2
15 10 4.32898577567354036E-2   44 11 2.64661146594475434E-2   73  7 6.51612787490608082E-2
16 11 2.89078595443236412E-2   45 11 2.89955383246821126E-2   74  7 7.01422304693874214E-2
17 11 2.36125151087114973E-2   46 12 2.76377604149556447E-2   75  8 3.83422732764378363E-2
18 11 2.04031742856177389E-2   47 11 2.63307159391851151E-2   76  8 2.54564971393033593E-2
19 11 3.66726628307374353E-2   48 12 1.96936946151027434E-2   77  8 1.42389342444866630E-2
20 10 1.29011362298691816E-2   49 13 2.00517729967894316E-2   78  8 3.39244729935979378E-2
21 10 1.07580433402153868E-2   50  9 3.90386706602626634E-2   79  9 2.50861334535896345E-2
22 14 1.01007712673132380E-2   51 10 8.35286119691666113E-2   80  9 2.92427434595390251E-2
23 14 2.77041895527882974E-2   52 10 1.99824236638351110E-2   81  9 2.99794955500694566E-2
24 14 1.91974357041789787E-2   53 10 1.96530645769366310E-2   82  9 3.04056391020463078E-2
25 14 1.16760485552368662E-2   54 11 2.24734849177335975E-2   83 10 3.26118611979957451E-2
26 14 1.81664719535158152E-2   55 11 3.29655372423225854E-2   84  9 1.76258656356385521E-2
27 16 2.66579517386904530E-2   56 10 3.37970096214405435E-2   85  9 2.90480407485540074E-2
28 16 2.26980715831676152E-2   57 11 3.76792877894931614E-2   86  9 3.03224786335428081E-2
29 14 2.47255581516852031E-2   58  9 3.11270192657094264E-2   87  9 2.66209387266570298E-2

87 difference quotients equal to zero in J1.COL file.
87 difference quotients equal to zero in J2.COL file.
91 difference quotients equal to zero in J3.COL file.
122 difference quotients equal to zero in J4.COL file.
```

The main parts of the MODEL2.RES file:

```
Date: 5/11/1994y.  Time: 20:17. Relative residuals were used.
Modifying percent: 1.000E+0000. Differentiation method: +.
ITERATION 1;     WSTD: 4.820434436E-0002;    NMD:922;    MP: 0.000E+0000
******** Initial Parameter Value(s):
p1o   =  1.00000000000000000E+0006
p2o   =  5.00000000000000000E+0005
p3o   =  2.00000000000000000E+0003
p4o   =  4.00000000000000000E+0004
p5o   =  1.00000000000000000E+0001
p6o   =  1.11000000000000000E+0001
p7o   =  1.80000000000000000E+0004
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  1.09157563225709138E+0006  1.022E+0004  &      0.94     9.158E+0004
  p2  5.90336645277031321E+0005  3.811E+0004  &      6.46     9.034E+0004
  p3  2.22817046142428879E+0003  8.266E+0001  &      3.71     2.282E+0002
  p4  3.76783130582149575E+0004  2.141E+0003  &      5.68    -2.322E+0003
  p7  2.57242840907848366E+0005  7.532E+0005  &    292.80     2.392E+0005
-------------------------------------------------------------------------------
    .                                                                        .
    .                                                                        .
    .                                                                        .
-------------------------------------------------------------------------------
ITERATION 11;     WSTD: 3.207518371E-0002;    NMD:922;    MP: 0.000E+0000
```

```
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  1.0921538872356674 5E+0006  6.908E+0003  &      0.63    -4.822E+0001
  p2  6.0733034496561514 6E+0005  2.797E+0004  &      4.61     4.692E+0002
  p3  2.2455455098124913 3E+0003  5.204E+0001  &      2.32     8.383E-0001
  p4  3.7715882568930117 8E+0004  1.307E+0003  &      3.47    -2.083E+0001
  p7  3.0028014332966562 1E+0005  4.344E+0004  &     14.47     4.512E+0002
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 0.000E+0000
                                   Relative Step: 1.0000
                          Number of Measured Data: 922
                        Number of Fitted Parameters: 5
               Weighted Standard Deviation of Data: 3.2075183714306088 1E-0002
             Unweighted Standard Deviation of Data: 3.2075183714306088 1E-0002
Correlation between Measured and Calculated Values: 9.9495370049839980 8E-0001
                     Coefficient of Determination: 9.8988891645090765 0E-0001
                                        R-squared: 9.9792635849844610 0E-0001
                          Model Selection Criterion: 4.5832770887343494 7E+0000
******** Residual Analysis:
Serial Correlation: 2.5314051326170688 0E+0001
Heteroscedactivity: 5.9142694003937911 9E-0001
          Skewness:-2.2812819532513411 4E+0000
          Kurtosis: 8.8524864331388931 8E+0000
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p3    p4    p7
  p1 { .762}-.397 -.259  .136 -.518
  p2  -.627{ .897} .695 -.766  .309
  p3   .167  .452{ .742}-.636  .427
  p4  -.590 -.755  .016{ .885}-.501
  p7  -.676 -.553  .290 -.631{ .790}
******** 95% Univariate Confidence Interval(s):
  p1  1.0785972683922374 2E+0006 - 1.1057105060790974 8E+0006
  p2  5.5244001731715427 5E+0005 - 6.6222067261407601 7E+0005
  p3  2.1434054629414702 8E+0003 - 2.3476855566835123 7E+0003
  p4  3.5149940250187969 5E+0004 - 4.0281824887672266 1E+0004
  p7  2.1502104960940887 0E+0005 - 3.8553923704992237 3E+0005
******** 95% Supporting Plane Confidence Interval(s):
  p1  1.0691197242328973 3E+0006 - 1.1151880502384375 7E+0006
  p2  5.1406573607363816 4E+0005 - 7.0059495385759212 8E+0005
  p3  2.0719985128872309 2E+0003 - 2.4190925067377517 3E+0003
  p4  3.3356068790271259 3E+0004 - 4.2075696347588976 3E+0004
  p7  1.5541571327507253 6E+0005 - 4.4514457338425870 6E+0005
******** The last PARAMTRS.DAT file:
 1.0922021084227772E+0006 * p1
 6.0686118265582722E+0005 * p2
 2.2447072556035969 0E+0003 * p3
 3.7736707615902484 1E+0004 * p4
 1.0000000000000000E+0001   p5
 1.1100000000000000E+0001   p6
 2.9982897045658429 9E+0005 * p7

 0.00000E+0000    0.99
 3.2075152914277683 0E-0002  1.00000000000000E-0006  1.00000000000000E-0007
12
******** Unweighted Standard Deviation of Data:
   1.06346601580458E-0005 in the case of ABSOLUTE residuals.
   3.20751837143061E-0002 in the case of RELATIVE residuals.
   2.37405095693519E-0002 in the case of ORTHOGONAL residuals.
******** Some pieces of important information from J*.COL file(s):
No.curve Point(s)   Fitting    No.curve Point(s)   Fitting    No.curve Point(s)   Fitting
 1 13 3.6887723795548662 7E-2   30 13 1.3177353260095086 7E-2   59  9 1.8607671826058309 7E-2
 2  9 1.6700333825855839 7E-2   31 12 1.3175342006921060 9E-2   60 12 3.0773242000144129 3E-2
 3 10 2.3252656295320512 7E-2   32 11 9.0606862063593818 1E-3   61 10 2.8705296438493954 4E-2
 4 11 1.3942602608211882 9E-2   33  9 1.0800315893408104 5E-2   62 12 4.0099471348074488 8E-2
 5 13 2.5059254876798312 6E-2   34  8 2.3744110863347835 4E-2   63 12 6.2453245116375196 6E-2
 6 11 2.1394915388029947 4E-2   35  9 1.3551440279321990 0E-2   64 12 2.0040980122596443 0E-2
 7 12 1.9471870897543399 8E-2   36  9 6.6998614637438240 5E-2   65 12 4.3415346129969450 9E-2
 8 11 2.2731901367198716 6E-2   37 12 5.7476867975588824 6E-2   66 11 4.0130975545911123 3E-2
 9 12 3.3409335540411035 3E-2   38 11 1.4901002053048299 4E-2   67  7 4.2677091011987953 5E-2
```

```
10 12 1.91580990282388915E-2   39 11 1.99993632265243831E-2   68  9 8.08189274785818128E-2
11 11 1.22763527427843255E-2   40 11 1.49564484472015311E-2   69  7 5.73936634323048572E-2
12 12 2.03669400402840132E-2   41 11 1.91828343430354244E-2   70  8 6.34903071444911898E-2
13 11 1.80701708660159961E-2   42 11 1.90887310834883501E-2   71  8 6.42923797268010916E-2
14 10 2.19092870595271219E-2   43 11 2.28439968717339355E-2   72  7 7.46717036882220480E-2
15 10 3.64847707284615635E-2   44 11 1.87544991585789572E-2   73  7 6.07223795185187009E-2
16 11 3.41701751635409488E-2   45 11 2.61533504008616722E-2   74  7 6.39047536992251521E-2
17 11 2.91125062802438169E-2   46 12 2.57850751524730126E-2   75  8 3.48941154443407996E-2
18 11 1.47145006635555859E-2   47 11 2.69488417615256084E-2   76  8 2.34814241768523307E-2
19 11 3.12489352392545358E-2   48 12 1.90924359664185980E-2   77  8 1.41385136394070133E-2
20 10 1.54038864430747717E-2   49 13 2.08434229139888880E-2   78  8 3.81678161810048592E-2
21 10 1.38173612359567736E-2   50  9 3.72940343887314944E-2   79  9 3.10517811040769458E-2
22 14 7.73727634455562131E-3   51 10 8.04416218331299666E-2   80  9 3.70844258017603210E-2
23 14 2.93533866145790556E-2   52 10 1.59067041233136744E-2   81  9 2.96574806664247702E-2
24 14 1.75273310998589717E-2   53 10 2.03684786488404410E-2   82  9 3.09135453400461418E-2
25 14 9.62658284053349270E-3   54 11 2.41894260115905605E-2   83 10 3.34170039264636619E-2
26 14 1.28423352280835475E-2   55 11 3.40724412640433770E-2   84  9 1.82674594013321901E-2
27 16 1.73940588119227191E-2   56 10 3.34742472841666638E-2   85  9 3.02757205979499591E-2
28 16 1.14513217526686185E-2   57 11 3.38572366323216395E-2   86  9 3.21903056944394543E-2
29 14 9.98569284189643823E-3   58  9 2.57300656036497571E-2   87  9 2.88606095274641636E-2

87 difference quotients equal to zero in J1.COL file.
87 difference quotients equal to zero in J2.COL file.
91 difference quotients equal to zero in J3.COL file.
114 difference quotients equal to zero in J4.COL file.
108 difference quotients equal to zero in J7.COL file.
```

E.8 Results of the Eighth Example

The main parts of the `MODEL1.RES` file (relative fitting is used):

```
Date: 6/11/1994y.  Time: 22:47. Relative residuals were used.
Modifying percent: 5.000E+0000. Differentiation method: -.
ITERATION 1;     WSTD: 3.080295368E-0001;    NMD:302;    MP: 1.000E-0003
******** Initial Parameter Value(s):
p1o   =  2.50000000000000000E-0002
p2o   =  2.50000000000000000E+0000
p3o   =  1.00000000000000000E-0001
p4o   =  2.00000000000000000E-0001
p5o   =  1.50000000000000000E+0001
p6o   =  2.00000000000000000E+0001
******** Result(s):
       Estimate                   Std. Dev. (abs & %)          dp
  p1  1.98227097463840792E-0002  8.683E-0004  &      4.38    -5.177E-0003
  p2  2.61218433480815670E+0000  1.325E-0001  &      5.07     1.122E-0001
  p3  1.15140050359845464E-0001  1.558E-0002  &     13.53     1.514E-0002
  p4  3.09692989720801048E+0000  7.636E+0000  &    246.56     2.897E+0000
  p5  7.31119361921267634E+0000  1.535E+0000  &     20.99    -7.689E+0000
  p6  5.33861750835512916E+0001  5.033E+0001  &     94.27     3.339E+0001
-------------------------------------------------------------------------------
    .                                                                        .
    .                                                                        .
    .                                                                        .
-------------------------------------------------------------------------------
ITERATION 22;     WSTD: 7.937020965E-0003;    NMD:302;    MP: 1.000E-0023
******** Result(s):
       Estimate                   Std. Dev. (abs & %)          dp
  p1  2.00112851002880768E-0002  8.656E-0006  &      0.04     1.879E-0009
  p2  2.00095170139755068E+0000  1.448E-0003  &      0.07     4.304E-0005
  p3  2.00235664450194831E-0001  1.416E-0004  &      0.07     5.928E-0006
  p4  4.92629974497659187E-0001  1.252E-0002  &      2.54    -1.841E-0004
  p5  1.00013551461008858E+0001  4.776E-0002  &      0.48    -1.563E-0004
  p6  3.99701841277301278E+0001  2.059E-0001  &      0.52     1.139E-0003
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0023
                                   Relative Step: 1.0000
```

```
                          Number of Measured Data: 302
                     Number of Fitted Parameters: 6
          Weighted Standard Deviation of Data: 7.93702096485750429E-0003
        Unweighted Standard Deviation of Data: 7.93702096485750429E-0003
Correlation between Measured and Calculated Values: 9.99697424799042122E-0001
                  Coefficient of Determination: 9.99392443683456502E-0001
                                     R-squared: 9.99723071635434344E-0001
                    Model Selection Criterion: 7.36633058557918496E+0000
******** Residual Analysis:
Serial Correlation:-1.51227329710982159E-0001
Heteroscedactivity:-1.12605932942576768E+0000
         Skewness: 1.20798501698099243E+0002
         Kurtosis: 1.03178093322787893E+0003
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p3    p4    p5    p6
  p1 { .742}-.630 -.059 -.072  .015 -.007
  p2  -.738{ .848} .565  .062  .134 -.238
  p3   .466  .689{ .745}-.264  .238 -.289
  p4   .149  .282 -.382{ .393}-.057  .035
  p5  -.113 -.163  .107 -.000{ .819}-.813
  p6  -.170 -.223  .026 -.000 -.806{ .829}
******** 95% Univariate Confidence Interval(s):
  p1  1.99942515138048584E-0002 - 2.00283186867712951E-0002
  p2  1.99810283955582128E+0000 - 2.00380056323928008E+0000
  p3  1.99956986946429336E-0001 - 2.00514341953960325E-0001
  p4  4.67999831137771198E-0001 - 5.17260117857547177E-0001
  p5  9.90736348252543994E+0000 - 1.00953468096763317E+0001
  p6  3.95650753739345647E+0001 - 4.03752928815256908E+0001
******** 95% Supporting Plane Confidence Interval(s):
  p1  1.99803463696668342E-0002 - 2.00422238309093193E-0002
  p2  1.99577720868662104E+0000 - 2.00612619410848032E+0000
  p3  1.99729492234044382E-0001 - 2.00741836666345279E-0001
  p4  4.47893337775562144E-0001 - 5.37366611219756231E-0001
  p5  9.83063462503691886E+0000 - 1.01720756671648528E+0001
  p6  3.92343701746124804E+0001 - 4.07059980808477752E+0001
******** The last PARAMTRS.DAT file:
 2.00112832208576324E-0002 * p1
 2.00090865827073803E+0000 * p2
 2.00229736866451760E-0001 * p3
 4.92814031368457965E-0001 * p4
 1.00015114004745713E+0001 * p5
 3.99690453128928716E+0001 * p6

 1.00000E-0023    0.99
 7.93727551786200724E-0003  5.00000000000000E-0005  5.00000000000000E-0005
25
******** Unweighted Standard Deviation of Data:
   3.92989811920573E-0003 in the case of ABSOLUTE residuals.
   7.93702096485750E-0003 in the case of RELATIVE residuals.
   3.79893020953727E-0003 in the case of ORTHOGONAL residuals.
******** Some pieces of important information from J*.COL file(s):
No.curve Point(s)   Fitting
   1        31 6.57741096238320408E-4 Rcon0001.Exp
   2        37 1.21669963561693745E-3 Rcon0002.Exp
   3        31 5.97269143694143128E-4 Lcon0001.Exp
   4        37 2.23859230549255530E-2 Lcon0002.Exp
   5        49 4.59400627104174962E-4 Pot0001.Exp
   6        42 4.77984562307000634E-4 Pot0002.Exp
   7        42 3.58831019693751326E-4 Abs0001.Exp
   8        33 1.25977061447253567E-4 Abs0002.Exp

8 difference quotients equal to zero in J1.COL file.
8 difference quotients equal to zero in J2.COL file.
8 difference quotients equal to zero in J3.COL file.
211 difference quotients equal to zero in J4.COL file.
229 difference quotients equal to zero in J5.COL file.
229 difference quotients equal to zero in J6.COL file.
```

The main parts of the `MODEL1.RES` file (orthogonal fitting is used):

```
Date: 5/11/1994y.  Time: 21:58. Orthogonal residuals were used.
Modifying percent: 5.000E+0000. Differentiation method: -.
ITERATION 1;     WSTD: 1.998352150E-0001;    NMD:302;    MP: 1.000E-0003
******** Initial Parameter Value(s):
p1o   =  2.50000000000000000E-0002
p2o   =  2.50000000000000000E+0000
p3o   =  1.00000000000000000E-0001
p4o   =  2.00000000000000000E-0001
p5o   =  1.50000000000000000E+0001
p6o   =  2.00000000000000000E+0001
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  2.16065737794886020E-0002  2.579E-0003  &     11.93    -3.393E-0003
  p2  1.98868538059881471E+0000  1.603E-0001  &      8.06    -5.113E-0001
  p3  2.00862147608334109E-0001  2.796E-0002  &     13.92     1.009E-0001
  p4  9.64380115690881054E-0001  2.109E+0000  &    218.69     7.644E-0001
  p5  9.81045288416485529E+0000  1.412E+0000  &     14.39    -5.190E+0000
  p6  5.17543150523956526E+0001  3.364E+0001  &     65.00     3.175E+0001
-------------------------------------------------------------------------------
    .                                                                       .
    .                                                                       .
    .                                                                       .
-------------------------------------------------------------------------------
ITERATION 7;     WSTD: 3.783744225E-0003;    NMD:302;    MP: 1.000E-0008
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  2.00577686456494147E-0002  2.894E-0005  &      0.14     6.690E-0007
  p2  1.99914712078148821E+0000  3.472E-0003  &      0.17    -6.107E-0004
  p3  2.00239880528354111E-0001  2.223E-0004  &      0.11    -4.736E-0005
  p4  4.89460922076294186E-0001  1.395E-0002  &      2.85     1.120E-0003
  p5  1.00084508306025443E+0001  2.569E-0002  &      0.26     7.508E-0004
  p6  3.99221312070692023E+0001  1.400E-0001  &      0.35    -1.032E-0002
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0008
                                    Relative Step: 1.0000
                          Number of Measured Data: 302
                       Number of Fitted Parameters: 6
              Weighted Standard Deviation of Data: 3.78374422489878639E-0003
            Unweighted Standard Deviation of Data: 3.78374422489878639E-0003
Correlation between Measured and Calculated Values: 9.99937844693063139E-0001
                     Coefficient of Determination: 9.99875147095335432E-0001
                                        R-squared: 9.99975590234609174E-0001
                        Model Selection Criterion: 8.94863917693654633E+0000
******** Residual Analysis:
Serial Correlation:-3.98353816789781547E-0001
Heteroscedactivity: 1.47176917924599436E-0002
         Skewness:-1.19732859130365165E+0002
         Kurtosis: 1.01949685297966676E+0003
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
        p1    p2    p3    p4    p5    p6
  p1 { .768}-.707 -.157 -.064  .204 -.264
  p2  -.728{ .861} .535 -.010 -.105  .212
  p3   .376  .685{ .787}-.478  .043 -.161
  p4   .137  .308 -.539{ .571}-.047  .175
  p5   .165  .210 -.180 -.000{ .718}-.697
  p6   .082  .290 -.300 -.000 -.701{ .754}
******** 95% Univariate Confidence Interval(s):
  p1  2.00008101077810668E-0002 - 2.01147271835177626E-0002
  p2  1.99231456106050163E+0000 - 2.00597968050247480E+0000
  p3  1.99802453961741151E-0001 - 2.00677307094967070E-0001
  p4  4.62011416018119043E-0001 - 5.16910428134469328E-0001
  p5  9.95790293271389061E+0000 - 1.00589987284911979E+0001
  p6  3.96465654195811027E+0001 - 4.01976969945573019E+0001
******** 95% Supporting Plane Confidence Interval(s):
  p1  1.99543127550520457E-0002 - 2.01612245362467837E-0002
  p2  1.98673689085901002E+0000 - 2.01155735070396641E+0000
  p3  1.99445366541276484E-0001 - 2.01034394515431738E-0001
  p4  4.39603373024205846E-0001 - 5.39318471128382525E-0001
  p5  9.91663882155865381E+0000 - 1.01002628396464347E+0001
```

```
  p6  3.94216109158243457E+0001 - 4.04226514983140589E+0001
******** The last PARAMTRS.DAT file:
 2.00570996831819495E-0002 * p1
 1.99975780601264324E+0000 * p2
 2.00287236281126627E-0001 * p3
 4.88340776805839172E-0001 * p4
 1.00076999814229628E+0001 * p5
 3.99324508780085504E+0001 * p6

 1.00000E-0008    0.99
 3.78369736246296181E-0003  5.00000000000000E-0005  5.00000000000000E-0005
25
******** Unweighted Standard Deviation of Data:
   1.39041132579629E-0002 in the case of ABSOLUTE residuals.
   8.64438680565818E-0003 in the case of RELATIVE residuals.
   3.78374422489879E-0003 in the case of ORTHOGONAL residuals.
******** Some pieces of important information from J*.COL file(s):
No.curve Point(s)   Fitting
   1        31 6.14252689520257170E-4 Rcon0001.Exp
   2        37 7.14659996171927703E-4 Rcon0002.Exp
   3        31 2.48511151053658014E-4 Lcon0001.Exp
   4        37 1.06304888187030171E-2 Lcon0002.Exp
   5        49 3.80680270348613911E-4 Pot0001.Exp
   6        42 4.83911772611861887E-4 Pot0002.Exp
   7        42 1.86706359203941983E-4 Abs0001.Exp
   8        33 4.10869882195531949E-4 Abs0002.Exp

6 difference quotients equal to zero in J1.COL file.
6 difference quotients equal to zero in J2.COL file.
6 difference quotients equal to zero in J3.COL file.
211 difference quotients equal to zero in J4.COL file.
229 difference quotients equal to zero in J5.COL file.
229 difference quotients equal to zero in J6.COL file.
```

E.9 Results of the Ninth Example

The main parts of the MODEL1.RES file:

```
Date: 5/11/1994y.  Time: 22:14. Relative residuals were used.
Modifying percent: 1.000E+0000. Differentiation method: +.
ITERATION 1;     WSTD: 4.997845823E-0001;    NMD:135;    MP: 6.400E-0002
******** Initial Parameter Value(s):
p1o   =  1.00000000000000000E+0001
p2o   =  1.00000000000000000E-0002
p3o   =  1.00000000000000000E+0003
p4o   =  1.00000000000000000E+0001
p5o   =  1.00000000000000000E+0001
p6o   =  1.00000000000000000E+0001
p7o   =  1.00000000000000000E+0003
p8o   =  1.00000000000000000E+0001
p9o   =  2.00000000000000000E+0001
p10o   =  1.00000000000000000E+0003
p11o   =  1.00000000000000000E+0001
******** Result(s):
       Estimate                   Std. Dev. (abs & %)          dp
  p1  3.37571238263252608E+0000  2.839E+0000  &     84.10    -6.624E+0000
  p2  7.32543696282715429E-0003  4.612E-0003  &     62.96    -2.675E-0003
  p3  1.04893113241082277E+0003  3.318E+0002  &     31.63     4.893E+0001
  p4  1.04110071558602615E+0001  6.064E+0002  &   5824.88     4.110E-0001
  p5  1.20354183237456137E+0001  1.302E+0002  &   1081.54     2.035E+0000
  p6  6.99131275945768823E+0001  4.170E+0002  &    596.41     5.991E+0001
  p7  1.01033365539151393E+0003  6.182E+0002  &     61.18     1.033E+0001
  p8  4.63454288327397548E+0001  1.358E+0002  &    293.10     3.635E+0001
  p9  2.11474688613961658E+0001  2.985E+0002  &   1411.71     1.147E+0000
 p10  7.17469891781646194E+0001  4.905E+0001  &     68.37    -9.283E+0002
 p11  5.54402687672015727E+0002  3.670E+0003  &    662.05     5.444E+0002
-------------------------------------------------------------------------------
```

```
    .                                                                       .
    .                                                                       .
    .                                                                       .
--------------------------------------------------------------------------------
ITERATION 7;     WSTD: 4.299938412E-0004;    NMD:135;    MP: 1.000E-0007
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  5.00022027772441285E+0000  2.973E-0003  &      0.06    -1.735E-0005
  p2  4.99885394046403325E-0003  2.409E-0006  &      0.05    -1.564E-0008
  p3  9.99377133079436694E+0002  1.901E-0001  &      0.02    -1.934E-0003
  p4  1.49951236627649372E+0002  4.314E-0001  &      0.29     7.725E-0005
  p5  1.94808541949505180E+0001  1.162E-0001  &      0.60     6.082E-0004
  p6  6.99677325207298394E+0001  3.636E-0002  &      0.05     1.055E-0004
  p7  8.99293458926060799E+0002  3.822E-0001  &      0.04     2.139E-0004
  p8  9.85054787691966687E+0000  3.737E-0002  &      0.38     2.316E-0004
  p9  2.97747240013889023E+0001  1.813E-0001  &      0.61    -1.311E-0003
 p10  2.48795679379729399E+0002  3.035E-0001  &      0.12     4.023E-0003
 p11  6.99620666664844445E+0002  7.736E-0002  &      0.01    -1.251E-0004
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0007
                                   Relative Step: 1.0000
                          Number of Measured Data: 135
                      Number of Fitted Parameters: 11
             Weighted Standard Deviation of Data: 4.29993841234660204E-0004
           Unweighted Standard Deviation of Data: 4.29993841234660204E-0004
Correlation between Measured and Calculated Values: 9.99999363041891173E-0001
                    Coefficient of Determination: 9.99998714805400123E-0001
                                       R-squared: 9.99999673801261218E-0001
                        Model Selection Criterion: 1.34016374485267333E+0001
******** Residual Analysis:
Serial Correlation:-6.19512353927432711E-0001
Heteroscedactivity: 2.83164065906206046E-0001
         Skewness: 2.49099994872724294E+0000
         Kurtosis:-1.07742219728255094E+0000
******** Correlation Matrix (rij\Ri\Sij):
(Others Are Fixed\Dependence from Others\Others Are Fitted)
       p1    p2    p3    p4    p5    p6    p7    p8    p9   p10   p11
  p1 { .964}-.750  .514  .514  .399 -.506 -.854  .389 -.435 -.011 -.444
  p2   .027{ .961}-.394 -.356 -.272  .174  .921  .094  .381 -.175  .214
  p3   .404  .017{ .755}-.164  .108 -.256 -.444  .192 -.224 -.002 -.225
  p4   .425  .032 -.638{ .770} .007 -.274 -.420  .226 -.220 -.018 -.236
  p5   .310  .029 -.345 -.404{ .556}-.215 -.323  .179 -.170 -.015 -.184
  p6  -.495  .419 -.000 -.000 -.000{ .785} .162 -.449  .194  .091  .280
  p7  -.491  .811 -.000 -.000 -.000 -.631{ .974}-.144  .407 -.108  .303
  p8   .119  .634 -.000 -.000 -.000 -.354 -.422{ .766}-.120 -.165 -.277
  p9  -.314 -.098 -.000 -.000 -.000 -.000 -.000 -.000{ .759}-.543  .040
 p10  -.314 -.172 -.000 -.000 -.000 -.000 -.000 -.000 -.676{ .724}-.135
 p11  -.331 -.145 -.000 -.000 -.000 -.000 -.000 -.000 -.403 -.424{ .622}
******** 95% Univariate Confidence Interval(s):
  p1  4.99433973289984828E+0000 - 5.00610082254897741E+0000
  p2  4.99408847732658489E-0003 - 5.00361940360148161E-0003
  p3  9.99001113102839227E+0002 - 9.99753153056034161E+0002
  p4  1.49097963448822839E+0002 - 1.50804509806475905E+0002
  p5  1.92511020692061332E+0001 - 1.97106063206949027E+0001
  p6  6.98958192305019015E+0001 - 7.00396458109577773E+0001
  p7  8.98537526928862577E+0002 - 9.00049390923259022E+0002
  p8  9.77663054060599819E+0000 - 9.92446521323333554E+0000
  p9  2.94161709191627114E+0001 - 3.01332770836150933E+0001
 p10  2.48195453880282851E+0002 - 2.49395904879175946E+0002
 p11  6.99467661756780018E+0002 - 6.99773671572908872E+0002
******** 95% Supporting Plane Confidence Interval(s):
  p1  4.98674762038097951E+0000 - 5.01369293506784618E+0000
  p2  4.98793599767998882E-0003 - 5.00977188324807767E-0003
  p3  9.98515650266173959E+0002 - 1.00023861589269943E+0003
  p4  1.47996340014898321E+0002 - 1.51906133240400423E+0002
  p5  1.89544792150082971E+0001 - 2.00072291748927388E+0001
  p6  6.98029751476319103E+0001 - 7.01324898938277685E+0001
  p7  8.97561576408875331E+0002 - 9.01025341443246268E+0002
  p8  9.68119912206732715E+0000 - 1.00198966317720066E+0001
  p9  2.89532588208011426E+0001 - 3.05961891819766621E+0001
```

```
 p10  2.47420529164216288E+0002 - 2.50170829595242509E+0002
 p11  6.99270123856361018E+0002 - 6.99971209473327872E+0002
******** The last PARAMTRS.DAT file:
 5.00023762724249100E+0000 * p1
 4.99886957979311296E-0003 * p2
 9.99379066758451757E+0002 * p3
 1.49951159381634192E+0002 * p4
 1.94802459783641258E+0001 * p5
 6.99676270678285029E+0001 * p6
 8.99293245016294676E+0002 * p7
 9.85031631509172702E+0000 * p8
 2.97760346035253195E+0001 * p9
 2.48791656107394111E+0002 * p10
 6.99620791755849940E+0002 * p11

 1.00000E-0007    0.99
 4.37349350621319385E-0004  1.00000000000000E-0004  1.00000000000000E-0004
12
******** Unweighted Standard Deviation of Data:
   2.89679120724762E-0002 in the case of ABSOLUTE residuals.
   4.29993841234660E-0004 in the case of RELATIVE residuals.
   3.01710243991275E-0004 in the case of ORTHOGONAL residuals.
******** Some pieces of important information from J*.COL file(s):
No.curve Point(s)   Fitting
   1        20 2.56353445621752186E-4 Ex90001.Exp
   2        17 4.87459611088272473E-4 Ex90002.Exp
   3        36 4.94677684074549791E-4 Ex90003.Exp
   4        28 3.26646115515086801E-4 Ex90004.Exp
   5        20 4.00796217038594987E-4 Ex90005.Exp
   6        14 4.27902767219412150E-4 Ex90006.Exp

0 difference quotient equals to zero in J1.COL file.
0 difference quotient equals to zero in J2.COL file.
98 difference quotients equal to zero in J3.COL file.
98 difference quotients equal to zero in J4.COL file.
99 difference quotients equal to zero in J5.COL file.
71 difference quotients equal to zero in J6.COL file.
71 difference quotients equal to zero in J7.COL file.
73 difference quotients equal to zero in J8.COL file.
102 difference quotients equal to zero in J9.COL file.
101 difference quotients equal to zero in J10.COL file.
101 difference quotients equal to zero in J11.COL file.
```

E.10 Results of the Tenth Example

The main parts of the MODEL1.RES file:

```
Date: 5/11/1994y.  Time: 22:16. Relative residuals were used.
Modifying percent: 1.000E+0000. Differentiation method: +.
ITERATION 1;     WSTD: 3.531716360E-0001;    NMD:33;    MP: 6.400E-0001
******** Initial Parameter Value(s):
p1o   =  2.00000000000000000E-0002
p2o   =  2.08700000000000000E-0002
p3o   =  2.00000000000000000E-0002
--------------------------------------------------------------------------------
    .                                                                          .
    .                                                                          .
    .                                                                          .
--------------------------------------------------------------------------------
ITERATION 5;     WSTD: 3.703420540E-0003;    NMD:33;    MP: 1.000E-0004
******** Result(s):
       Estimate                  Std. Dev. (abs & %)          dp
  p1  5.00261808914963792E-0003  1.888E-0005  &      0.38     2.523E-0007
  p3  6.10363610809073483E-0003  3.219E-0005  &      0.53    -1.672E-0007
******** Goodness-of-Fit Statistics:
                              Marquardt Parameter: 1.000E-0004
                                    Relative Step: 1.0000
```

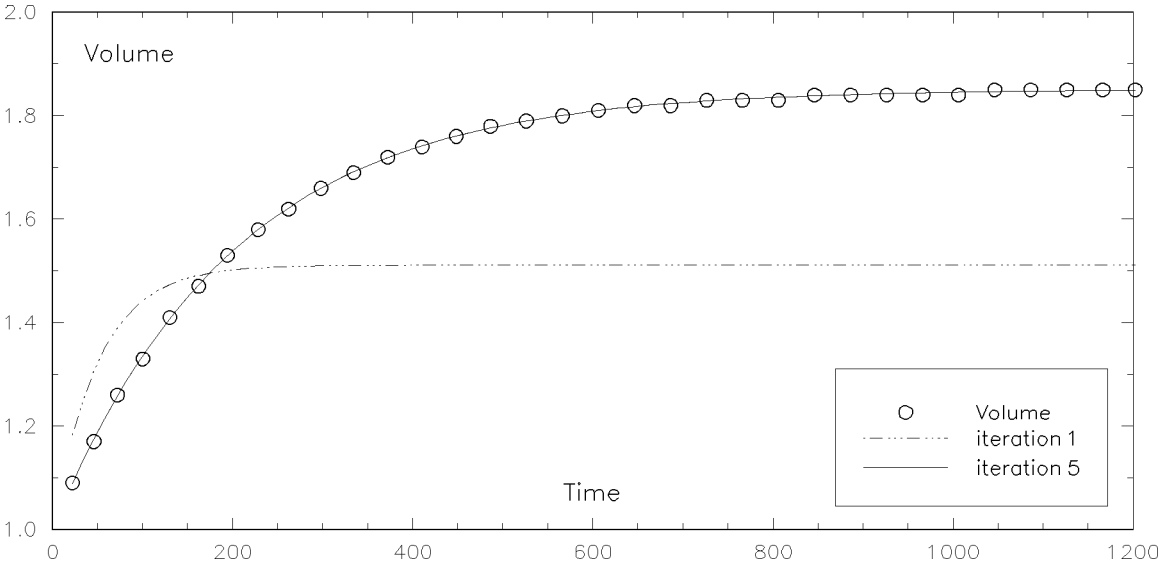


Figure E.6: Experimental and calculated curves in Example 10.

```
                        Number of Measured Data: 33
                    Number of Fitted Parameters: 2
          Weighted Standard Deviation of Data: 3.70342054020525113E-0003
        Unweighted Standard Deviation of Data: 3.70342054020525113E-0003
Correlation between Measured and Calculated Values: 9.99918617178672811E-0001
                 Coefficient of Determination: 9.99836408817663572E-0001
                                   R-squared: 9.99997437060700208E-0001
                  Model Selection Criterion: 8.59692791173513416E+0000
******** Residual Analysis:
Serial Correlation:-9.51484243343276017E-0001
Heteroscedactivity:-1.12688984423930942E+0000
          Skewness: 1.24732955131677311E+0000
          Kurtosis:-1.04475680870712976E+0000
******** Correlation Matrix (rij\Ri\Sij):
        p1    p3
  p1 { .681} .681
  p3   .681{ .681}
******** 95% Univariate Confidence Interval(s):
  p1  4.96415591869582044E-0003 - 5.04108025960345541E-0003
  p3  6.03806557432148511E-0003 - 6.16920664185998455E-0003
******** 95% Supporting Plane Confidence Interval(s):
  p1  4.95407285184692110E-0003 - 5.05116332645235475E-0003
  p3  6.02087590282056920E-0003 - 6.18639631336090046E-0003
******** The last PARAMTRS.DAT file:
5.00236578837499061E-0003 * p1
 3.47661967004675925E-0002
6.10380329953240754E-0003 * p3

1.00000E-0004    0.99
4.27264829019076787E-0003  5.00000000000000E-0005  5.00000000000000E-0005
12
******** Unweighted Standard Deviation of Data:
   2.81459961055599E-0003 in the case of ABSOLUTE residuals.
   3.70342054020525E-0003 in the case of RELATIVE residuals.
   3.01555799467202E-0003 in the case of ORTHOGONAL residuals.
******** Some pieces of important information from J*.COL file(s):
No.curve Point(s)   Fitting
   1        33 3.58944173352414392E-3 Ex100001.Exp
0 difference quotient equals to zero in J1.COL file.
0 difference quotient equals to zero in J3.COL file.
```

Bibliography

[1] *Turbo Pascal, Reference Guide, Version 5.0.*
Borland International; Scotts Valley, 1989.

[2] *Microsoft MS-DOS, User's Reference, Version 3.30.*
Microsoft Corporation, 1987.
Microsoft MS-DOS, User's Guide and Reference, Version 5.0.
Microsoft Corporation, 1991.

[3] Hindmarsh, A. C. *Gear... Ordinary Differential Equation System Solver.*
UCID-30001 rev. 3, Lawrence Livermore Laboratory,
P.O.Box 808, Livermore, CA 94550, Dec. 1974.

[4] Hindmarsh, A. C. *Linear Multistep Methods for Ordinary Differential Equations... Method Formulations, Stability, and the Methods of Nordsieck and Gear.*
UCRL-51186 rev. 1, Lawrence Livermore Laboratory,
P.O.Box 808, Livermore, CA 94550, March 1972.

[5] Hindmarsh, A. C. *Construction of Mathematical Software, part III... The Control of Error in the Gear Package for Ordinary Differential Equations.*
UCID-30050 part 3, Lawrence Livermore Laboratory,
P.O.Box 808, Livermore, CA 94550, August 1972.

[6] Gottwald, B. A.; Wanner, G. *A reliable Rosenbrock-integrator for stiff differential equations.*
Computing **26**, 355-360, (1981).

[7] Bard, Y. *Nonlinear Parameter Estimation.*
Academic Press, New York, 1974.

[8] Himmelblau, D. M. *Applied Nonlinear Programming.*
McGraw-Hill, New York, 1972.

[9] Leggett, D. J. *Computational Methods for the Determination of Formation Constants.*
Plenum Press, New York, 1987.

[10] Meyer, J. *Die stufenweise Verseifung der Ester zweibasischer Säuren. II.*
Zeitschrift für physikalische Chemie **67**, 257-301, (1909).

[11] Frost, A. A.; Schwemmer, W. C. *The Kinetics of Competitive Consecutive Second-Order Reactions: The Saponification of Ethyl Adipate and of Ethyl Succinate.*
Journal of American Chemical Society **74**, 1298-1273, (1952).

[12] Schwemmer, W. C.; Frost, A. A. *A Numerical Method for the Kinetic Analysis of Two Second Order Reactions.*
Journal of American Chemical Society **74**, 4151-4152, (1951).

[13] Byren, G. D.; Hindmarsh A. C. *Stiff ODE Solvers: A Review of Current and Coming Attractions.*
Journal of Computation Physics **70**, 1-62 (1987).

[14] Peintler, G.; Nagypál, I.; Epstein, R. I. *Kinetics and Mechanism of the Reaction between Chlorite Ion and Hypochlorous Acid.*
Journal of Physical Chemistry **94**, 2954-2958 (1990).

[15] Eigen, M.; Kustin, K. *The Kinetics of Halogen Hydrolysis.*
Journal of American Chemical Society **84**, 1355-1361, (1962).

[16] Carl de Boor *A Practical Guide to Splines.*
Applied Mathematical Sciences **27**,
Springer-Verlag, New York, Heidelberg, Berlin, 1978.

[17] Arthur Block *Murphy's Law, Book Two.*
Suffolk, Methuen, 1980.

[18] Rutherford Aris *Elementary Chemical Reactor Analysis.*
Prentice-Hall, Inc., Englewood Cliffs, NJ, 1969.

[19] Denbigh, K. G.; Turner, J. C. R. *Chemical Reactor Theory.*
Cambridge University Press, Cambridge, 1971.

[20] Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; Vetterling, W. T. *Numerical Recipes in Pascal, The Art of Scientific Computing.*
Cambridge University Press, Cambridge, 1989.

[21] Akaike, H. *An Information Criterion (AIC).*
Math. Sci. **1**, No. 4: 5-9, (1962).

[22] Norris, A. C. *Computational Chemistry*
John Wiley & SOns Ltd., New York, 1981.

[23] Rabitz, H. *Chemical Dynamics and Kinetics Phenomena as Revealed by Sensitivity Analysis Techniques*
Chemical Reviews **87**, 101-112 (1987).

Index

Z