COMPUTER PROGRAMS FOR CHEMISTRY

VOLUME II

DELOS F. DETAR, Volume II Editor
The Florida State University

W. A. BENJAMIN, INC.
New York *Amsterdam* *1969*

COMPUTER PROGRAMS FOR CHEMISTRY

VOLUME II

CONTRIBUTORS TO VOLUME II

DeLos F. DeTar
Department of Chemistry and
Institute of Molecular Biophysics
The Florida State University
Tallahassee, Florida 32306

REMECH
RMCHSS
TSTMCH

Carleton E. DeTar
Department of Chemistry and
Institute of Molecular Biophysics
The Florida State University
Tallahasee, Florida 32306

REMECH

PREFACE

Application of computer techniques to chemical research is causing a fundamental revolution comparable to that stemming from NMR, gas chromatography, and other new tools.

Effective use of computers requires powerful and versatile programs or software. The design of a fine program is a work of art on a par with the design of an elegant instrument. The analogy is remarkably close: while a chemist must be able to design and assemble his own experimental set up, he has neither the time nor the expert knowledge to construct his pH meters, his spectrophotometers, and so on. Likewise chemists will frequently design simple computer programs but usually will not have the time nor the detailed knowledge required for creating a sophisticated general program.

The purpose of *Computer Programs for Chemistry* is to make available in convenient form a wide range of programs of interest to chemists, and by providing a medium for publication, to encourage the devising and application of quantitative mathematical models to problems in chemistry. This publication has been organized by an editorial board which serves without compensation. Programs have been screened and tested before publication.

At the outset we were uncertain about the probable shelf life of the programs. Obviously this will vary, but it is now our belief that most programs will be useful for many years. Errors in coding and minor revisions can usually be handled by means of Correction Decks. The major reason for scrapping a program is that the computations are no longer of interest, and this reason does not apply to the programs in these volumes. Recoding to increase speed, to decrease storage requirements, or to enhance convenience will be worthwhile only for programs which are very heavily used. Recoding to permit extension

to larger systems is likely to be of limited interest, and in most cases can be made available through a Conversion Deck.

The investment in developing, debugging, and otherwise stabilizing a comprehensive computer program is considerable. It is difficult to evaluate costs, partly because accurate records are not generally available, partly because the proven value of a program grows with use, and partly because it is not at all clear where to stop counting costs of debugging and of validation. What is the appropriate dollar value for the time of a research scientist who spends several months developing a general program? Finally there is a question as to how many preliminary versions of a program should be included in the cost estimates. Subject to such uncertainties the cost of developing and debugging the programs in one volume of this series is of the order of $50,000–$100,000. An important purpose of *Computer Programs for Chemistry* is to preserve the results of these heavy research investments and to make them readily available to others.

Programming standards are undergoing important changes. The initial efforts of most programmers were directed toward immediate results with little thought for convenience to other users or to future use by the author. Present trends are toward longer range goals. The trauma suffered by many programmers who have been required to shift from one computing system to another has sharply emphasized the advantages of compatibility. Sophisticated programmers now prefer to use a higher language such as FORTRAN IV wherever possible, even though assembly language programs are usually more efficient for a specific job.

Another particularly welcome trend is elimination of booby traps. The practice of some programmers to read an index such as NOPT under an I5 format where the maximum permissible value of NOPT is, say, 4 can lead to chaotic results. Such practice is frowned upon, and careful programmers also arrange to test all such indexes to be sure they are within range. It is good practice to test also whether subscripts of arrays are within permissible limits.

Almost any general program tends to be rather complex to use properly. Good programming practice devotes considerable effort toward simplifying the input so as to make the problems of preparing error-free data cards as convenient as possible. For instance, parallel programs should accept data in similar formats. There is an advantage in starting fields in Columns 1, 11, 21, and so on, whenever practical. It is then possible to use the automatic tabulator programming facilities of the card punches.

Another important practice mentioned in Volume I is gradually

being implemented. This is the effective treatment of the problem of bad input data, where an I is punched in place of a 2, for example. It is very wasteful of computer time to have to read a tray of 4000 data cards twenty times because there happen to be twenty errors which can be caught just one at a time. Yet this has been happening at most computing centers in the past.

It is practical to modify system input subroutines to provide for recovery. Perhaps the software packages of the major computers will soon provide this option. To use these modified routines effectively it is necessary to incorporate any required error routines into the program itself. Our own practice has been to reserve the first location in blank COMMON as an error box. This may be referred to as INPERR, for example. INPERR is zeroed, and after a READ statement under any but A FORMAT, INPERR is tested. If zero, then the READ was performed correctly. If nonzero, then there was an error, and the variable involved has been set to 0. If this can cause trouble, the program must take appropriate error action. Examples of programs which include this feature are LSKIN1 and TSTMCH.

As with all improvements, there is a considerable lag between recognition and implementation. The date of appearance of a program in this series does not reflect clearly the date of writing. The perceptive user will note a great many deficiencies. However, the possible improvements do not warrant the expense of recoding and revalidating. It is our expectation that future programs including those now under construction will incorporate most of the types of improvements mentioned here.

DELOS F. DETAR, *Editor*

Tallahassee, Florida
October 1968

BRIEF DESCRIPTIONS OF PROGRAMS

RMCHSS AND REMECH

Both programs treat reaction mechanisms. RMCHSS is a relatively short and somewhat limited program that will run on small computers. REMECH is a versatile general program suitable for reaction mechanisms of virtually any complexity.

REMECH accepts a reaction mechanism consisting of a set of chemical equations plus assigned values for all rate and equilibrium constants plus a set of initial concentrations. By a process of numerical integration REMECH provides tables of concentration data as a function of time. Another way to describe the purpose of REMECH is to say that it computes the theoretical curves about which the experimental data are supposed to fall if the mechanism and associated reaction constants are correct.

These programs do not replace other techniques. Instead they provide the tool for finding precisely what predictions are made by a given mechanism. With REMECH it is not necessary to restrict attention to those limiting conditions which are accessible to treatment by hand.

REMECH computes the concentrations of the components of complex mixtures of polyprotic acids and bases, and it computes the concentrations of the components of complex general equilibria.

In addition to equilibria, REMECH treats complex systems of steady state intermediates, such as may occur in coupled enzyme systems. A given chemical equation may have any number of reactants or products. The capacity of the program as distributed is 250 equations and 150 compounds.

RMCHSS treats systems with steady state intermediates, including certain enzyme problems. The equations are written in numerical form; this requirement was added to simplify the programming. As distributed, each equation is limited to two reactants and two products. Although it is a relatively simple program, RMCHSS is capable of treating a variety of complex reaction mechanisms.

TAPE COPIES OF PROGRAMS IN VOLUME II

All programs in Volume II are available on a master tape. The tape consists of files of unblocked 80 character records in standard character code of the type produced by the IBM 1401 on half inch tape. Recording density is 556 bits per inch. If appropriate editing programs are available, the tape can be used directly as input. More commonly the tape will be used to punch the source cards and these will be edited by hand.

Tape copies of the contents of Volume II are provided at a total cost which includes the tape plus the cost of loading at a data processing center. Orders should be placed with W. A. Benjamin, Inc., Two Park Avenue, New York, New York 10016.

TABLE OF CONTENTS OF THE TAPE

```
TABLE OF CONTENTS OF THE TAPE
FILE  1
   HEADER
FILE  2
   TAPE CONTENTS
FILE  3
  CORRECTION AND CONVERSION DECKS FOR PROGRAMS IN PREVIOUS VOLUMES
FILE  4
   RMCHSS FORTRAN IV CDC
FILE  5
   RMCHSS CORRRECTION AND CONVERSION DECKS
FILE  6
   RMCHSS DATA
FILE  7
   REMECH FORTRAN IV CDC
   THE FOLLOWING SUBROUTINES HAVE SPECIAL CDC CODING
      BNJBCD
      GBCDBN
      IBCDBN
      ITYPE
      LOCSET
      PACK
      ERA
      (THESE SEVEN AND SMAX ARE AT THE END OF THIS FILE)
   SMAX IS A DUMMY, AND MAY BE USED AS IS WITH ANY SYSTEM
FILE  8
   CDC ASCENT VERSIONS OF BNJBCD,GBCDBN,IBCDBN,ITYPE, AND PACK
FILE  9
   REMECH CORRECTION AND CONVERSION DECKS
FILE 10
   REMECH DATA
FILE 11
   TSTMCH FORTRAN IV CDC
FILE 12
   TSTMCH DATA
FILE 13
END OF TAPE MESSAGE

   (NOTE - FILES ARE SEPARATED BY EOF (TAPE) MARKS WHILE LAST FILE IS
   TERMINATED BY TWO TAPE MARKS)
```

CONTENTS

CHAPTER 1

Correction and Conversion Decks for Programs in Volume I

The method of using correction and conversion decks was described in Chapter 1 of Volume I. These are editing decks: if the serial number of an edit card matches that of a card already present, then the edit card replaces (or deletes if so indicated) the matching card of the original deck. If there is no match, the edit card is to be inserted in proper sequence.

When Volume I was in preparation, the original plan was to carry out occasional revisions of the master tape for Volume I. For several reasons this approach seems less desirable now, and the present plan is not to change the master tape. One important consideration is the great amount of work required in preparing and rechecking a master tape. Another is the difficulty and expense of maintaining an inventory of revised tapes.

It will always be possible to identify the level of revision of a given tape by the date given in the tape header. If that date is later than the publication date of the corresponding volume of this series, then the tape incorporates revisions. If the date of the tape header is earlier than the publication date of the volume, then the tape is the original.

Each volume of *Computer Programs for Chemistry* will contain a section listing the latest set of correction and conversion decks. Those for Volume I are listed in Figure 1-1 and appear on the master tape for Volume II.

```
CORRECTION AND CONVERSION DECKS FOR PREVIOUS VOLUMES


C       CORRECTION DECK FOR LSKIN1
C      ESTKF REPLACED BY EXTERNAL FUNCTION (BEGINS AT AI131400) TO
C      PERMIT CHECK FOR ILLEGAL ARGUMENTS ETC.
C      ESTKF(X,Y,Z,DT) = ALOG((Y - X)/(Z - Y))/DT    (NOW EXTERNAL)          AI116700
C
C      PURPOSE OF FOLLOWING CHANGES IS TO PROVIDE DIAGNOSTIC FOR NEGATIVE
C      OR ZERO ARGUMENT (K SET TO -0.)
       IF(PARAM(1)) 1110,1110,1010
 1110 WRITE(6,7500) PARAM(1)                                                  AI121810
 7500 FORMAT(23H ERROR. RATE CONSTANT =, E10.2/ 65H CHECK VALUE ENTERED       AI124005
     1AND CHECK DATA POINTS FOR INITIAL ESTIMATES.)                           AI124010
       IERR=IERR+1                                                            AI124015
       GO TO 1100                                                             AI124020
       FUNCTION ESTKF(X,Y,Z,DT)                                               AI124025
C      TO REPLACE THE INTERNAL FUNCTION, CARD AI116700.                       AI131400
C      ESTK RETURNS -0. IF NEGATIVE ARGUMENT FOR LOG OR DT=0                  AI131410
       IF((Z-Y).EQ.0.) GO TO 5                                                AI131420
       W=(Y-X)/(Z-Y)                                                          AI131500
       IF(W) 5,5,20                                                           AI131600
     5 ESTKF=-0                                                               AI131700
    10 RETURN                                                                 AI131800
    20 IF(DT) 5,5,30                                                          AI131900
    30 ESTKF=ALOG(W)/DT                                                       AI132000
       GO TO 10                                                               AI132100
       END                                                                    AI132200
C                                                                             AI132300
C      END OF CORRECTION DECK FOR LSKIN1.
C
C      CONVERSION DECK FOR PLOTLN, VOLUME 1
C      CONVERSION TO ELIMINATE NEED FOR SUBRCUTINE BNJBCD. ANY COMPUTER.
C      SUITABLE FOR LSKIN1, LSKIN2, ACTENG, AND MANY OTHER APPLICATIONS
C      OF PLOTLN.
C
C      THERE ARE TWO EFFECTS OF THIS ELIMINATION. (1) IT IS IMPOSSIBLE TO
C      WRITE SCALES ON THE RIGHT UNLESS NX = 121. (2) IN SOME CASES THE
C      X SCALES WILL NOT LINE-UP NICELY WITH THE GRID LINES.
C      IN ALL OTHER RESPECTS PLOTLN RUNS NORMALLY.
C
       DATA H6,H121/1H6,3H121/
  180 FORMT3(5)=FM6                                                           AF 16220
  183 FORMT3(2)=FM3                                                           AF 24300
  187 FORMT2(2)=H6                                                            AF 24600
  253 FORMT(2)=H121                                                           AF 25000
C                                                                             AF 28600
C      END OF PLOTLN CONVERSION TO ELIMINATE BNJBCD.
```

FIGURE 1-1

CHAPTER 2

RMCHSS

Author: DeLos F. DeTar, Department of Chemistry and Institute of Molecular Biophysics, The Florida State University, Tallahassee, Florida 32306.

Checked by: Herbert A. Moore, Jr., and Andrew Streitwieser, Jr., Department of Chemistry, University of California, Berkeley, California 94720.

2-1 INTRODUCTION

RMCHSS is a relatively simple program which nevertheless treats a variety of complex reaction mechanisms including those with steady state intermediates. It is also useful for enzyme problems where there is a single enzyme and a single level of protonation. Several enzyme complexes may be present. One reaction may incorporate a catalyst. There is no provision for equilibrium or acid-base systems. A given equation may have at most two reactants and two products.

RMCHSS is adaptable to small computers; furthermore, the indexing technique illustrates the principles of the more complex indexing in REMECH. The method of using the program is illustrated by an en-

zyme example. Other examples and a more detailed description of RMCHSS are available.[1] Directions have been published for writing simple computer programs capable of treating a wide variety of complex mechanisms which do not involve steady state intermediates.[2,3]

2-2 THEORETICAL PRINCIPLES

These have been described in detail elsewhere.[1] The theoretical principles are also described under REMECH.

2-3 INPUT TO RMCHSS

CARD 1

Cols. 1–12 Date (2A6)
Cols. 13–72 Header (10A6)

CARD 2

Cols. 1–2 Number of compounds. NCPDS. Maximum set at 20. (I2)
Cols. 6–7 Number of steady state intermediates. NSS (I2)
Cols. 11–12 Number of reactions. NR (I2)
Cols. 16–17 Serial number of catalyzed reaction. NRCAT (I2)
Cols. 21–22 Serial number of the compound which is the catalyst. NCPCAT (I2)

CARD 3

Cols. 1–6 Name of first compound. CPDNAM(1). A6. List steady state compounds first.
Cols. 11–16 Name of second compound. Rest are 21–26, and so on, seven maximum per card.
If more than seven compounds, additional cards are used here.

CARD 4

Cols. 1–2 Index number of compound which is first reactant. JRE (I, 1) I2