Downloaded from orbit.dtu.dk on: Dec 18, 2015

Technical University of Denmark

CHEMSIMUL: A simulator for chemical kinetics

Kirkegaard, Peter; Bjergbakke, E.

Publication date:
1999

Document Version
Publisher final version (usually the publisher pdf)

Link to publication

Citation (APA):
Kirkegaard, P., & Bjergbakke, E. (1999). CHEMSIMUL: A simulator for chemical kinetics. (Denmark. Forskningscenter Risoe. Risoe-R; No. 1085(EN)).

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Risø–R–1085(EN)

CHEMSIMUL: A Simulator for chemical kinetics

Peter Kirkegaard and Erling Bjergbakke

January 1999; Revised December 1999

Risø National Laboratory, Roskilde, Denmark
January 2000

Abstract CHEMSIMUL is a computer program system for numerical simulation of chemical reaction systems. It can be used for modeling complex kinetics in many contexts, in particular radiolytic processes with pulse trains. It contains a translator module and a module for solving the resulting coupled nonlinear ordinary differential equations. An overview of the program system is given, and its use is illustrated by examples. A number of special features are described, in particular a method for verifying the mass balance. Moreover, the document contains a complete User's Guide for running CHEMSIMUL on a PC or another computer. Finally, the mathematical implementation is discussed.

ISBN 87-550-2474-2
ISBN 87-550-2475-0 (Internet)
ISSN 0106–2840

Information Service Department · Risø · 2000

Contents

Preface *4*

1 Introduction *5*
1.1 Overview *5*
1.2 CHEMSIMUL development and related work *5*
1.3 Chemical kinetics with pulse radiolysis *6*
1.4 Units *6*

2 Translating reactions to differential equations *8*
2.1 Simple example with one reaction *8*
2.2 Sample case from combustion *8*

3 Output possibilities *9*
3.1 Result file *9*
3.2 Graphical output and plot expressions *11*
3.3 Formal printout of differential equations *11*

4 Miscellaneous features *11*
4.1 The stoichiometric mass balance *11*
4.2 Check of electro neutrality *12*
4.3 Variation of rate constant with temperature *12*
4.4 Adiabatic processes *12*
4.5 Third body reactions *13*
4.6 Maximum values and half-lives for species *14*
4.7 External pulse-train irradiation *14*
4.8 Mixed particle irradiation *14*

5 User's Guide for CHEMSIMUL Version Dec 1999 *16*
A Running the program *16*
B Description of general part of input file *17*
C Simulation commands *18*
D Description of the experimental file *23*
E Sample data set *23*
F Isotope input and output *25*

6 Hints for solving special problems *27*
6.1 How to mark a reaction *27*
6.2 Maintaining a constant concentration of a solute *27*
6.3 Maintenance of gas/liquid equilibrium *28*
6.4 Transport, dissolution and diffusion. *28*

7 Simulation examples *29*
7.1 Fricke dosimeter *29*
7.2 Oregonator *29*
7.3 Further references to works using CHEMSIMUL *30*

8 Mathematical implementation *30*
8.1 Translation principle for reactions *30*
8.2 Stoichiometric balance check by linear programming *31*
8.3 Solution of the ODE system *31*
8.4 The Jacobian *32*
8.5 Implementation of plot expressions *32*

9 Updates since previous versions *34*
9.1 Since 1993 *34*
9.2 Since 1998 *34*

10 Computer requirements *35*
10.1 Computer platforms *35*
10.2 MINICHEM *35*

References *35*

Preface

The CHEMSIMUL program system was created several decades ago by the pioneering work of Ole Lang Rasmussen at Risø National Laboratory in Denmark. Since then the program has evolved and expanded alongside with its practical use by chemists in Denmark and elsewhere.

In 1998 Risø National Laboratory and chemists at CEA Saclay in France initiated a collaboration with the aim of updating CHEMSIMUL with a set of new features, and consolidating some of the old ones.

For fruitful inspiration in the development of CHEMSIMUL we thank Dr. P. Bouniol, Dr. C. Corbel, and Dr. B. Hickel, all at CEA Saclay. Moreover, Dr. Frank Markert, Risø, has contributed with valuable comments to the report.

The present December 1999 edition corrects some minor errors in the January edition. In addition, some new facilities were added to the program, causing the documentation to be revised accordingly.

Peter Kirkegaard, e-mail peter.kirkegaard@risoe.dk
Erling Bjergbakke, e-mail erb@gundsoe.dk

1 Introduction

Chemical transformations taking place in the natural environment as well as in industrial processes are in general very complex. Even in well designed laboratory experiments it is often difficult to study elementary chemical reactions without interference from simultaneous side reactions. Therefore computer simulation has become a powerful tool for the analysis of complex reactions, which are fundamental for understanding atmospheric chemistry, combustion chemistry and air pollution problems, and for the development of new technologies.

1.1 Overview

The program system CHEMSIMUL was developed at Risø National Laboratory as the result of a close co-operation between chemists and applied mathematicians. CHEMSIMUL is a computerized simulator of chemical kinetics with the following main components:

- Module for input of reaction equations in chemical notation
- Automatic translator from chemical equations to differential equations
- Solution of system of ordinary differential equations
- Output routines
- Miscellaneous facilities

The simulation results will be concentrations of the chemical species in the reaction system as functions of time. These can be given in tabular and graphical form.

CHEMSIMUL is tailor-made to simulation of radiation chemistry, where bursts of electron beams or γ-rays are modeled by rectangular source pulses of finite time duration, or trains of such pulses. Heat generation from the reactions may also be taken into account. There are a number of other special features to be described in the following, notably a method for checking the mass balance.

1.2 CHEMSIMUL development and related work

Chemists at Risø and elsewhere have used CHEMSIMUL and its predecessors for many years as a simulation tool supporting their experimental work. Rasmussen and Bjergbakke [1] give a historical account of the development of software for kinetics simulation at Risø over several decades, right from the beginning in 1966 when analogue methods were still prevailing. They point out the close connection with establishment of numerical techniques for fast and accurate integration of "stiff" non-linear differential equations. In the chemists' language stiffness means that the kinetic system has a wide range of relaxation times for perturbations. Stiff methods for Ordinary Differential Equations (ODE) were introduced at Risø shortly after 1971 when Gear [2] published his DIFSUB code. They replaced the classical fifth-order Runge-Kutta methods. Eventually, the ODE software in CHEMSIMUL was again replaced, first by a revised DIFSUB, then by EPISODE (Hindmarsh and Byrne [3]), and still later by the LSODA program (Hindmarsh [3], Petzold [4]), which is the solver used today.

It was early realized that the construction of the ODEs from the reaction equations, which was originally done manually, should be automated, and this led to the development of the translation module in CHEMSIMUL. It was a design criterion for the system that the chemist could express the reaction processes to be studied in familiar chemical nomenclature.

The early versions of CHEMSIMUL were written in ALGOL. The old documentation by Rasmussen and Bjergbakke [1] relates to this language. Since 1986 PASCAL and MODULA-2 were in use as programming languages for mainframe versions, but from 1992 CHEMSIMUL has been entirely FORTRAN based. In 1997 it was converted to FORTRAN 90

with the result that most of the restrictions on problem size were alleviated. The PC is now the most important computer platform for running CHEMSIMUL at Risø and elsewhere; however, the program may run under any system that supports FORTRAN 90, as for example workstations under Unix or Digital/Compaq VMS.

Other computer programs are available with application fields more or less overlapping with CHEMSIMUL. Curtis and Sweetenham [5] have constructed the Harwell simulator FACSIMILE (= Flow and Chemistry Simulator). It has broader scope than CHEMSIMUL, being a general system for time evolution problems. It also solves partial differential equations and mixed algebraic-differential systems. Braun *et al.* [6] have developed the simulator program ACUCHEM, Deuflhard *et al.* [7] the program LARKIN, and Kee *et al.* [8] the Sandia program CHEMKIN. These programs are alike CHEMSIMUL in some respects, apart from CHEMSIMUL's facilities for radiation chemistry and thermodynamics.

1.3 Chemical kinetics with pulse radiolysis

CHEMSIMUL is constructed to simulate homogeneous kinetics in monophase. Only zero-, first-, and second-order reactions are allowed. Reactions of higher order must be emulated by suitable first- and second-order reactions.

When preparing input data for CHEMSIMUL, the reactions are written in normal chemical notation, e.g.

$$\mathrm{A} = \mathrm{B} + \mathrm{C}; \quad k_1 \tag{1}$$

for a first-order reaction, and

$$\mathrm{A} + \mathrm{D} = \mathrm{E} + \mathrm{F}; \quad k_2 \tag{2}$$

for a second-order reaction; k_1 and k_2 are the rate constants for these reactions.

An important facility in CHEMSIMUL is its ability to simulate radiolytic processes. These are zero-order reactions, whereby a species A is produced by radiation, say from electron beams or γ-rays. The rate of production is proportional to the dose rate $D(t)$:

$$\frac{d[\mathrm{A}]}{dt} = c\,G(\mathrm{A})D(t). \tag{3}$$

Here, $G(\mathrm{A})$ is the so-called *G-value* for radiolytic production of the given species A with concentration [A], and c is the *conversion constant* whose value will be given in Section 1.4, where units are discussed.

G-values can be negative as well as positive. When the radiation produces species with positive G-values, a corresponding negative value for $G(\mathrm{H_2O})$ should be included in order to preserve the mass balance.

Though the concept of G-values is primarily designed for calculation of irradiation yields, it can be used for zero-order reactions in general. In this way other types of reactions can be simulated.

CHEMSIMUL normally assumes that the dose rate $D(t)$ is a rectangular pulse:

$$D(t) = \begin{cases} D_0 & 0 < t < t_\mathrm{r} \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where t_r is the time of radiation; this may be generalized to a train of equidistant rectangular pulses as described in Section 4.7. It is also possible to handle cases with a sum of exponentially decaying dose rates. This is discussed in Section 4.8.

1.4 Units

CHEMSIMUL allows a free choice of units in the computations, although the program originally was designed for kinetics in condensed phase with the traditional units: concentration in $mol \times dm^{-3}$, rate constants in s^{-1} and $mol^{-1} \times dm^3 \times s^{-1}$ for first and second order reactions, respectively, activation energy in $kcal \times mol^{-1}$, heat formation in $kcal \times mol^{-1}$,

and specific heat capacity in $kcal \times mol^{-1} \times K^{-1}$, cf. Sections 4.3 and 4.4. Now CHEMSIMUL can be used with other units as well, for example the standard gas kinetics units: concentration in $molecules \times cm^{-3}$ and rate constant in $molecules^{-1} \times cm^3 \times s^{-1}$ for second order reactions. Please observe that all units must be changed.

For convenience we state below some common units and physical constants that may be useful when working with CHEMSIMUL:

Units:

Energy and heat:	$1\,J = 10^7\,erg$
	$1\,eV = 1.6022.10^{-19}\,J$
	$1\,kcal = 4186\,J$
Absorbed dose:	$1\,Gy\,(\text{gray}) = 1\,J \times kg^{-1} = 10^{-1}\,krad$
	$1\,krad = 10^5\,erg \times g^{-1} = 10^{-2}\,J \times g^{-1} = 2.389.10^{-3}\,kcal \times kg^{-1}$

Constants:

Avogadro's number:	$N_\mathrm{A} = 6.023.10^{23}\,molecules \times mol^{-1}$
Gas constant:	$R = 8.315\,J \times mol^{-1}K^{-1} = 1.9864\,kcal \times mol^{-1}K^{-1}$

The definition of the G-value introduced in (3) of Section 1.3 is *molecules* per $100\,eV$. The conversion constant c in (3) depends on the units for concentration and dose. Choosing these as $mol \times dm^{-3}$ and *krad*, respectively, we find c by computing $d[\mathrm{A}]/dt$ for $D(t) = 1\,krad \times s^{-1} = 10^{-2} J \times g^{-1} \times s^{-1} = 10^{17}/1.6022\,eV \times g^{-1} \times s^{-1}$:

$$\frac{d[\mathrm{A}]}{dt} = G(A) \times \frac{10^{15}}{1.6022}\,molecules \times g^{-1}s^{-1} =$$

$$G(A) \times \frac{10^{15}\rho}{1.6022}\,molecules \times cm^{-3}s^{-1} = G(A) \times \frac{10^{18}\rho}{1.6022 \times 6.023.10^{23}}\,mol \times dm^{-3}s^{-1}. \quad (5)$$

In irradiation of water where $\rho = 1\,g \times cm^{-3}$ we thus obtain

$$c = 1.036.10^{-6}, \quad (6)$$

which is the default setting in CHEMSIMUL. If the unit for absorbed dose is changed to *Gy*, then c must be changed to $1.036.10^{-7}$, and if ρ is different from $1\ g \times cm^{-3}$, c should likewise be changed. Such changes of the conversion constant are easily accommodated for in CHEMSIMUL by the command `CONVERT`, see Section C of the User's Guide.

In Section 4.3 it is mentioned that CHEMSIMUL accepts rate constants in the modified Arrhenius form (14). If you use this facility, you should note that the gas constant must be entered in the input file either as $R = 1.9864.10^{-3}\,kcal \times mol^{-1} \times K^{-1}$ if the activation energy E_a is entered in unit $kcal \times mol^{-1}$, or as $R = 8.315\,J \times mol^{-1} \times K^{-1}$ if E_a is entered in unit $J \times mol^{-1}$. See also Section C of the User's Guide.

We have collected some frequently used CHEMSIMUL units in Table 1 below, where each line represents a consistent set of units.

Concentration	Rate Constants		Gas Constant	Activation Energy	Heat of Reaction	Specific Heat Capacity
`CON`	1.order	2.order	`R`	`EA`	`Q`	`HCV`
$mol \times dm^{-3}$	s^{-1}	$mol^{-1} \times dm^3 \times s^{-1}$	$kcal \times mol^{-1} \times K^{-1}$	$kcal \times mol^{-1}$	$kcal \times mol^{-1}$	$kcal \times mol^{-1} \times K^{-1}$
$mol \times dm^{-3}$	s^{-1}	$mol^{-1} \times dm^3 \times s^{-1}$	$J \times mol^{-1} \times K^{-1}$	$J \times mol^{-1}$	$J \times mol^{-1}$	$J \times mol^{-1} \times K^{-1}$
$molecules \times cm^{-3}$	s^{-1}	$molecules^{-1} \times cm^3 \times s^{-1}$	$kcal \times mol^{-1} \times K^{-1}$	$kcal \times mol^{-1}$	$kcal \times molecules^{-1} \times cm^3$	$kcal \times molecules^{-1} \times cm^3 \times K^{-1}$
$molecules \times cm^{-3}$	s^{-1}	$molecules^{-1} \times cm^3 \times s^{-1}$	$J \times mol^{-1} \times K^{-1}$	$J \times mol^{-1}$	$J \times molecules^{-1} \times cm^3$	$J \times molecules^{-1} \times cm^3 \times K^{-1}$

Table 1. Consistent sets of units for CHEMSIMUL

2 Translating reactions to differential equations

When explaining the translation in CHEMSIMUL from chemical reactions to differential equations we shall first use a very simple reaction scheme with only one reaction, and then a more realistic sample case from combustion.

2.1 Simple example with one reaction

Consider the reaction

$$\mathrm{R1} + \mathrm{R2} \rightarrow \mathrm{R3} + \mathrm{R4} \tag{7}$$

with 4 *species* (2 *reactants* and 2 *products*). We assume that the reaction proceeds according to the law of mass action with the rate constant k. Suppose also that the chemical medium is irradiated either by an electronic beam or by γ-rays, such that the species R2 and R3 are produced by this radiation with yields determined by $G(\mathrm{R2})$ and $G(\mathrm{R3})$, respectively, cf. Sections 1.3 and 1.4. Then the resulting differential equations for the concentrations are

$$\begin{aligned}
\frac{d[\mathrm{R1}]}{dt} &= -k[\mathrm{R1}][\mathrm{R2}] \\
\frac{d[\mathrm{R2}]}{dt} &= -k[\mathrm{R1}][\mathrm{R2}] + c\,G(\mathrm{R2})D(t) \\
\frac{d[\mathrm{R3}]}{dt} &= k[\mathrm{R1}][\mathrm{R2}] + c\,G(\mathrm{R3})D(t) \\
\frac{d[\mathrm{R4}]}{dt} &= k[\mathrm{R1}][\mathrm{R2}]
\end{aligned} \tag{8}$$

Here $[\cdot]$ denotes concentration, $D(t)$ is the dose rate at time t (normally a rectangular pulse or a pulse train), and c is the conversion constant.

The reaction between R1 and R2 is a second-order reaction, while the radiolytic production of R2 and R3 are zero-order reactions. (CHEMSIMUL does not treat third-order reactions or higher directly, so these must be emulated by lower-order reactions, cf. Section 4.5.) We see that a single chemical reaction equation is described by a system of Ordinary Differential Equations (ODE), which are non-linear in the concentrations. Starting with the time $t = 0$ and the initial values of the reactant concentrations $[\mathrm{R1}]_0$, $[\mathrm{R2}]_0$, $[\mathrm{R3}]_0$ and $[\mathrm{R4}]_0$ we can integrate the system up to some final time $t = t_{\mathrm{end}}$.

Very small chemical systems such as (7) can be simulated by making a direct write-up of the ODE system, but for larger systems this would be tedious and error-prone. Therefore CHEMSIMUL has a module for automatic translation of the chemical reactions to differential equations.

2.2 Sample case from combustion

Let us now consider a more realistic reaction system discussed in [9] for modeling a H_2–O_2 combustion process. This example will be used repeatedly as a sample case:

$$\begin{aligned}
&(\mathrm{R1}) && \mathrm{H} + \mathrm{H} && \rightarrow && \mathrm{H_2} \\
&(\mathrm{R2}) && \mathrm{H} + \mathrm{O_2} && \rightarrow && \mathrm{HO_2} \\
&(\mathrm{R3}) && \mathrm{H} + \mathrm{HO_2} && \rightarrow && \mathrm{OH} + \mathrm{OH} \\
&(\mathrm{R4}) && \mathrm{HO_2} + \mathrm{HO_2} && \rightarrow && \mathrm{H_2O_2} + \mathrm{O_2} \\
&(\mathrm{R5}) && \mathrm{OH} + \mathrm{OH} && \rightarrow && \mathrm{H_2O_2} \\
&(\mathrm{R6}) && \mathrm{H} + \mathrm{OH} && \rightarrow && \mathrm{H_2O}
\end{aligned} \tag{9}$$

$$\text{(R7)} \qquad \mathrm{OH + HO_2} \rightarrow \mathrm{H_2O + O_2}$$

$$\text{(R8)} \qquad \mathrm{OH + H_2} \rightarrow \mathrm{H_2O + H}$$

The corresponding part of the CHEMSIMUL input file, with rate constants, reads:

```
RE1: H+H=H2; A=4.0E7
RE2: H+O2=HO2; A=4.5E8
RE3: H+HO2=OH+OH; A=6.5E10
RE4: HO2+HO2=H2O2+O2; A=2.0E9
RE5: OH+OH=H2O2; A=4.0E9
RE6: H+OH=H2O; A=1.0E10
RE7: OH+HO2=H2O+O2; A=6.0E10
RE8: OH+H2=H2O+H; A=4.0E3
```
(10)

We note the similarity with the chemical notation in (9). When processing a reaction system as this, CHEMSIMUL scans and "digests" all the kinetic equations, symbol by symbol. On encounter it tabulates every new species and includes its name in the current set of symbols. It also stores the reaction rates. After the scan phase an assembly phase is invoked. The technical details are discussed in Section 8.1. Here we shall only show the outcome of the translation process, where we for illustration use CHEMSIMUL's ODE printout feature (Section 3.3):

```
D[H]/DT   = - 2*K1*H*H - K2*H*O2 - K3*H*HO2 - K6*H*OH + K8*OH*H2
            + G(H)*CONST*DOSE
D[H2]/DT  =   K1*H*H - K8*OH*H2
D[O2]/DT  = - K2*H*O2 + K4*HO2*HO2 + K7*OH*HO2
D[HO2]/DT =   K2*H*O2 - K3*H*HO2 - 2*K4*HO2*HO2 - K7*OH*HO2
D[OH]/DT  =   K3*H*HO2 + K3*H*HO2 - 2*K5*OH*OH - K6*H*OH - K7*OH*HO2
            - K8*OH*H2
D[H2O2]/DT=   K4*HO2*HO2 + K5*OH*OH
D[H2O]/DT =   K6*H*OH + K7*OH*HO2 + K8*OH*H2
```
(11)

The complete CHEMSIMUL input data file for the sample case is shown in Section E of the User's Guide.

3 Output possibilities

When running CHEMSIMUL, the program will produce a result file. Moreover, it has facilities for presenting the output in graphical form. We shall give a short description of each of these features, illustrating with the sample data set in Section E of the User's Guide.

3.1 Result file

The name of the result file will be the name of the input file with the extension replaced by `.res`. The first part of the file is an echo of the CHEMSIMUL input data as prepared according to the User's Guide, Section 5. Then follows the integration results in tabular form. These comprise the concentrations of all reacting species as functions of time (possibly supplemented by heat production and temperature for adiabatic processes, cf. Section 4.4).

By and large the CHEMSIMUL output is self-explanatory. We show below the result file for our H_2–O_2 combustion case in Section 2.2:

```
##############################
CHEMSIMUL COMPILED:    DEC 1999
##############################
```

```
INPUT DATA FILE: h2-o2
DATE OF COMPUTATION:  07-DEC-1999 15:58
---------------------------------------
CASE: 5 mbar O2 + 100 mbar H2 + Ar to 1 atm. k8=4000

NO. OF REACTION EQUATIONS =   8
NO. OF G-VALUES           =   1
NO. OF CONCENTRATIONS     =   2

REACTION EQUATION SYSTEM
------------------------
RE1:H+H=H2;A=4.0E7
RE2:H+O2=HO2;A=4.5E8
RE3:H+HO2=OH+OH;A=6.5E10
RE4:HO2+HO2=H2O2+O2;A=2.0E9
RE5:OH+OH=H2O2;A=4.0E9
RE6:H+OH=H2O;A=1.0E10
RE7:OH+HO2=H2O+O2;A=6.0E10
RE8:OH+H2=H2O+H;A=4.0E3

PLOT EXPRESSIONS
----------------
PE1:HO2

G - VALUES
----------
G(H)=1.0

START CONCENTRATIONS
--------------------
CON(O2)=2.0E-4
CON(H2)=4.0E-3

NO. OF TERMS IN DIFFERENTIAL SYSTEM =  25
NO. OF SPECIES                     =   7

NO. OF RADIATION PERIODS.........=        1
TOTAL DOSE.......................= 9.00E-01
RADIATION TIME FOR EACH PERIOD...= 5.00E-09
NO. OF RESULTS DURING RADIATION..=        2

TOTAL NO. OF RESULTS.............=       12
MAXIMUM INTEGRATION TIME.........= 1.00E-03
RELATIVE INTEGRATION TOLERANCE...= 1.00E-03

                            RESULT TABLE
------------------------------------------------------------------------
   TIME       H         H2        O2        HO2        OH        H2O2
 0.00E+00  0.00E+00  4.00E-03  2.00E-04  0.00E+00  0.00E+00  0.00E+00
 2.50E-09  4.66E-07  4.00E-03  2.00E-04  5.25E-11  2.75E-15  4.45E-21
 5.00E-09  9.32E-07  4.00E-03  2.00E-04  2.10E-10  3.43E-14  9.92E-20
 1.00E-04  1.80E-11  4.00E-03  2.00E-04  2.80E-07  4.40E-08  3.29E-08
 2.00E-04  1.74E-12  4.00E-03  2.00E-04  2.22E-07  9.88E-09  4.53E-08
 3.00E-04  4.94E-13  4.00E-03  2.00E-04  1.98E-07  2.80E-09  5.40E-08
 4.00E-04  1.58E-13  4.00E-03  2.00E-04  1.82E-07  8.99E-10  6.12E-08
 5.00E-04  5.53E-14  4.00E-03  2.00E-04  1.69E-07  3.14E-10  6.73E-08
 6.00E-04  2.08E-14  4.00E-03  2.00E-04  1.58E-07  1.18E-10  7.27E-08
 7.00E-04  7.99E-15  4.00E-03  2.00E-04  1.48E-07  4.70E-11  7.73E-08
 8.00E-04  3.53E-15  4.00E-03  2.00E-04  1.40E-07  1.98E-11  8.15E-08
 9.00E-04  1.55E-15  4.00E-03  2.00E-04  1.33E-07  8.72E-12  8.52E-08
 1.00E-03 -4.28E-17  4.00E-03  2.00E-04  1.26E-07  4.01E-12  8.86E-08

                            RESULT TABLE
------------------------------------------------------------------------
   TIME      H2O
 0.00E+00  0.00E+00
 2.50E-09  1.06E-20
 5.00E-09  3.28E-19
 1.00E-04  2.71E-07
 2.00E-04  3.05E-07
 3.00E-04  3.12E-07
 4.00E-04  3.14E-07
```

```
 5.00E-04  3.15E-07
 6.00E-04  3.15E-07
 7.00E-04  3.15E-07
 8.00E-04  3.15E-07
 9.00E-04  3.15E-07
 1.00E-03  3.15E-07

EXECUTION TIMES FOR CHEMSIMUL (SECS)
INPUT       SETUP    INTEGRATION    OUTPUT             TOTAL
 0.08        0.04        0.40        0.09               0.60
```

3.2 Graphical output and plot expressions

CHEMSIMUL also produces a plot table file whose name will be the name of the input file, with the extension replaced by `.tbl`.

The plot table file has one column for time, and one column for each of the "plot expressions" that are specified in the input file.

Plot expressions (`PE...`) are discussed in Section C of the User's Guide, and in Section 8.5. A plot expression may be just a single species concentration, as `PE1:HO2` in our sample case. However, the concept is much more versatile, as it supports all the 5 operations $+$, $-$, $\times$, $/$, and `^` (exponentiation), together with the common mathematical functions. This is a useful facility, when a researcher wants to plot a curve which is directly comparable with certain experimental results. An example is the measurement of extinction

$$E = (\varepsilon_A[\mathrm{A}] + \varepsilon_B[\mathrm{B}])\ell \tag{12}$$

where ε_A and ε_B are the extinction coefficients for species A and B, respectively, ℓ is the optical path length, and $[\cdot]$ denotes concentration.

CHEMSIMUL has an interface to the public-domain program GNUPLOT [10], which is in widespread use on many different computers. For each run, CHEMSIMUL generates a command file `gnu.cmd`, which you may execute by the following GNUPLOT command:

```
gnuplot gnu.cmd
```

GNUPLOT will then plot the simulation results from the CHEMSIMUL plot table file.

Hardcopies can be produced by prepending suitable `set terminal` and `set output` commands to the file `gnu.cmd` [10] (e.g. `set terminal postscript`).

As the plot table file is just a normal text file, you may of course use it in connection with other graphics programs as well.

Two examples of plots produced by CHEMSIMUL/GNUPLOT are shown in Section 7.

3.3 Formal printout of differential equations

The program can give a formal print-out of the differential equation system to be solved. This may be a useful tutorial facility for understanding reaction kinetics. An example of this was shown at the end of Section 2.2. The reaction rates are labelled `K`n corresponding to reaction `RE`n. The ODE print-out is activated by the command `DIFFEQ`, cf. Section C of the User's Guide.

4 Miscellaneous features

In the following a number of additional CHEMSIMUL features will be described.

4.1 The stoichiometric mass balance

The accurate solution of the differential equation system describing the chemical reactions requires an overall conservation of the chemical mass balance. The numerical integration

scheme itself preserves the mass balance, cf. Section 8.2, so there is no need to check for mass balance continually. We use instead a static consistency check. If the check fails, the simulation is halted before integration; this practice has proved useful for detecting input errors in the write-up of the reaction equations. CHEMSIMUL recognizes each species as an entity with a name. This means that a formal stoichiometric balance does not imply an atom-to-atom balance. But it is possible to perform a *partial* consistency check based on the *stoichiometric matrix* **A**. This matrix is defined formally in Section 8.1, but its meaning is clear from the following example, where the reactions of the H_2–O_2 combustion case (9) are written in matrix form as a balance equation:

$$\begin{pmatrix} -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & 1 & -2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -2 & 1 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & -1 & 0 & 1 \\ 1 & -1 & 0 & 0 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{H} \\ \mathrm{H2} \\ \mathrm{O2} \\ \mathrm{HO2} \\ \mathrm{OH} \\ \mathrm{H2O2} \\ \mathrm{H2O} \end{pmatrix} = \mathbf{0} \tag{13}$$

It must be possible to satisfy this equation by a set of strictly positive values for the masses of all the elements of the species vector.

The mathematical implementation of the balance check is discussed in Section 8.2.

4.2 Check of electro neutrality

As explained in Section B of the User's Guide, CHEMSIMUL admits species names with charge designators as e.g. `FE[+++]` or `OH[-]`. Such a convention enables the program to check the electro neutrality both in the individual reactions and for the totality of G-values. Like the mass balance check, this feature may be useful in catching errors in the input data file.

4.3 Variation of rate constant with temperature

CHEMSIMUL accepts rate constants in the modified Arrhenius form

$$k = AT^{\beta} \exp(-E_{\mathrm{a}}/(RT)) \tag{14}$$

where T is the absolute temperature, E_{a} the activation energy, R the gas constant, and β an empirical exponent. The factor T^{β} is included for gas phase kinetics.

The components A, β, and E_{a} are entered as `A`, `B`, and `EA`, respectively, in the record description of the reaction in the input file. See also Section B of the User's Guide. For proper choice of units, see Section 1.4.

4.4 Adiabatic processes

It is possible to simulate an adiabatic reaction at constant volume. In particular this means that the total heat formation Q_{tot} and the temperature T can be simulated by two extra equations. If some of the reaction rates k_r assume the modified Arrhenius form (14), then there will be a coupling between (15–16) and the kinetic differential equations. If, on the other hand, all the k_r are constant, then (15–16) will be completely decoupled from the kinetic equations.

The source of the heat may come from the chemical reactions as well as from the irradiation. The specific heat capacity, $c_v(\mathrm{R}_s)$, must be known for each species R_s, and the specific heat of reaction, q_r, must be known for each reaction r. In this context q_r is defined as $-\Delta H$ (H = enthalpy), which means that q_r is positive for an exothermic reaction.

The differential equations governing $Q_{\rm tot}$ and T are

$$\frac{dQ_{\rm tot}}{dt} = \sum_{r=1}^{M} k_r[\mathrm{R}_j][\mathrm{R}_k]\, q_r + \alpha D'(t) \tag{15}$$

and

$$\frac{dT}{dt} = \frac{dQ_{\rm tot}/dt}{\sum_{s=1}^{N} c_v(\mathrm{R}_s)\,[\mathrm{R}_s]} = \frac{\sum_{r=1}^{M} k_r[\mathrm{R}_j][\mathrm{R}_k]\, q_r + \alpha D'(t)}{\sum_{s=1}^{N} c_v(\mathrm{R}_s)\,[\mathrm{R}_s]}, \tag{16}$$

respectively, where M is the number of reactions and N the number of species in the system; $D'(t)$ is the dose rate, and α is the *heat production coefficient* for converting dose to heat.

The proper units are $kcal \times mol^{-1} \times K^{-1}$ or $J \times mol^{-1} \times K^{-1}$ for c_v and $kcal \times mol^{-1}$ or $J \times mol^{-1}$ for q_r, cf. Table 1 in Section 1.4.

In (15) R_j and R_k are the second-order reactants attached to the reaction r with rate constant k_r. For a first-order reaction we should set $[\mathrm{R}_k] = 1$.

The specific heats q_r are entered in the input records for the corresponding reactions r as `Q = ...`, see Section B of the User's Guide. The specific heat capacities $c_v(\mathrm{R}_s)$ are given for each species by the command `HCV = ...`, see Section B of the User's Guide.

The dose rate $D'(t)$ may be given by an external radiation pulse, cf. (3) and (4) in Section 1.3. Alternatively it may come from an internal source in the presence of decaying mixed-particle irradiation, in which case $D'(t) = \sum_{i=1}^{m} e^{-\lambda_i t}(D_i^{\alpha} + D_i^{\beta} + D_i^{\gamma} + D_i^{n})$, cf. (23-28) in Section 4.8.

CHEMSIMUL will only calculate the heat production and the subsequent rise in temperature if at least one command `HCV(S) = ...` is present in the input file, where `S` is a species with non-vanishing initial concentration. Moreover, the irradiation heat contribution will only be estimated when the gas constant `R` is included in the input file (or entered on-line by request); here `R` just serves to tell the computer in which units you are working, so that it can calculate α in (15–16).

4.5 Third body reactions

Reactions of higher order than 2 should not be considered in condensed phase, and the present version of CHEMSIMUL does not accept reactions of higher order.

However, in gas phase many reaction rates are dependent on a "third body" to absorb surplus reaction energy, as in the reactions

$$\mathrm{A} + \mathrm{B} + \mathrm{M} \rightarrow \mathrm{P} + \mathrm{M} \tag{17}$$

and

$$\mathrm{A} + \mathrm{M} \rightarrow \mathrm{C} + \mathrm{D} + \mathrm{M} \tag{18}$$

These reactions are formally third and second order reactions, but actually they are pseudo-second and pseudo-first order, respectively, because the concentration [M] does not change during the reaction. This means that the differential equation for the third-order reaction

$$\frac{d[\mathrm{P}]}{dt} = k[\mathrm{A}][\mathrm{B}][\mathrm{M}] \tag{19}$$

can be reduced to

$$\frac{d[\mathrm{P}]}{dt} = k_{\mathrm{M}}[\mathrm{A}][\mathrm{B}] \tag{20}$$

where

$$k_{\mathrm{M}} = k[\mathrm{M}]. \tag{21}$$

The substitution (21) will be done automatically for all reactions containing M (both pseudo-first and pseudo-second order) by using the input command `THIRD(M)` = [M], see Section C of the User's Guide.

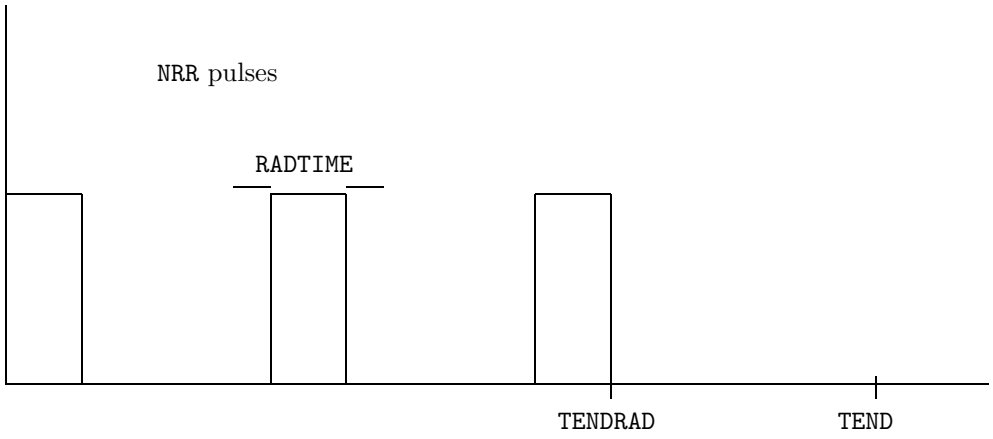
4.6 Maximum values and half-lives for species

From the computed simulation curves CHEMSIMUL may estimate maximum-values and half-lives for all the species. The latter are stated as period of formation as well as period of decay, both related to the maximum, if there is such a maximum.

4.7 External pulse-train irradiation

When CHEMSIMUL simulates an external radiation source, the normal situation is to consider a single rectangular pulse of time duration t_r = `RADTIME`, cf. (4) in Section 1.3.

However, CHEMSIMUL also admits a train of such pulses, as shown in the sketch below:



The pulse train consists of n_{rr} = `NRR` individual pulses each of duration t_r = `RADTIME`. The first pulse begins at time zero, and the last pulse ends at time t_{end}^{rad} = `TENDRAD`. See also Section C of the User's Guide.

4.8 Mixed particle irradiation

In the study of radiation chemistry for storage of radioactive waste it is necessary to take the time variation of dose rate as well as radiation yields into account.

The present version of CHEMSIMUL is able to simulate a mixture of exponentially decaying dose rates.

Let m isotopes be given. Each of them is a potential emitter of α-, β-, γ-, and neutron-radiation. With isotope i we assign the following parameters:

$$\tau_i = \text{period (half-life) in } s\text{. May be } \infty \text{ (default).} \quad (22)$$

$$\lambda_i = \text{decay rate} = \ln 2/\tau_i \text{ in } s^{-1}\text{. May be 0 (default).} \quad (23)$$

$$D_i^{\alpha} = \text{initial } \alpha \text{ dose rate} \quad (24)$$

$$D_i^{\beta} = \text{initial } \beta \text{ dose rate} \quad (25)$$

$$D_i^{\gamma} = \text{initial } \gamma \text{ dose rate} \quad (26)$$

$$D_i^{n} = \text{initial neutron dose rate} \quad (27)$$

The units for the α-, β-, γ-, and neutron dose rates are $krad \times s^{-1}$ or $Gy \times s^{-1}$. Moreover, we shall define G-values for the four types of radiation for each species R_s ($s = 1, \ldots, N$): G_s^{α}, G_s^{β}, G_s^{γ}, G_s^{n}. We are now able to compute the production rate for R_s from all these

radiation types at time t:

$$\frac{d[\mathrm{R}_s]}{dt} = \sum_{i=1}^{m} e^{-\lambda_i t}(G_s^{\alpha} D_i^{\alpha} + G_s^{\beta} D_i^{\beta} + G_s^{\gamma} D_i^{\gamma} + G_s^{n} D_i^{n}) \times c, \tag{28}$$

where c is the conversion constant introduced in Sections 1.3 and 1.4. When setting up the differential equations, the sum in (28) should be added to the right-hand side (cf. (8), (11), and (56)).

For input prescriptions concerning mixed particle irradiation you should consult Section F of the User's Guide, where also a partial sample output is shown.

5 User's Guide for CHEMSIMUL Version Dec 1999

The present User's Guide is written as an almost self-contained manual within the frames of the complete CHEMSIMUL document.

A Running the program

CHEMSIMUL runs on PC, UNIX, VMS, and other systems, cf. Section 10. The main platform is the PC where it normally runs in a DOS-window under Win95, Win98, or WinNT. You may run CHEMSIMUL by just typing:

```
chem
```

Then the program answers with the text:

```
ENTER NAME OF INPUT FILE:
```

You now type the input file name, including its extension (normally `.dat`), e.g. `mycase.dat` (when the extension is `.dat` it may be omitted); the output file will then be named `mycase.res`. The PC version also admits the use of command line input, i.e. you may type:

```
chem mycase
```

If the input file is not available in your current directory, or is corrupted, the program aborts with a message like:

```
INPUT FILE COULD NOT BE OPENED (ERROR CODE =     105)
*** THIS CASE IS DELETED ***
```

(the actual error code may be system dependent), and the program must be restarted. CHEMSIMUL checks all the input data, and if errors are found the program is discontinued with an error message in plain text on the computer screen. The error messages should be sufficient for identifying the kind of errors and correcting them in the input file. When the input file is accepted, the program types:

```
REACTION SYSTEM IS BALANCED - PROGRAM CONTINUES -
WAIT UNTIL STATUS BAR BELOW IS FILLED WITH ASTERISKS...
```

The status bar reflects the current fraction of the simulation that is already completed. It is useful for estimating the progress of the integration procedure for long simulations. A snapshot of the status bar may look as follows:

```
**************************  .     .     .     .     .     .     .     .
```

After the status bar is completely filled with asterisks, the program continues:

```
EXPERIMENTAL FILE CAN BE PLOTTED TOGETHER WITH THE PLOT EXPRESSION(S):
PE1:(expression)
PE2:(expression)
ENTER FILE NAME (RETURN FOR NO EXPERIMENTAL FILE):
```

If you wish to compare the simulation with experimental data you now type the name of the file. If you have no experimental file you just type a ⟨RETURN⟩.

After the CHEMSIMUL run you may plot the results by GNUPLOT or by other means. You invoke GNUPLOT by the following command:

```
gnuplot gnu.cmd
```

Here `gnu.cmd` is a file created by CHEMSIMUL that causes GNUPLOT to read and display the data file `mycase.plt`, also created by CHEMSIMUL, where `mycase` was the name of your problem. (For hardcopies see Section 3.2; other graphics programs may of course be substituted for GNUPLOT to display the data in `mycase.plt`.)

Often it is convenient to launch the running of CHEMSIMUL and subsequently GNUPLOT by a single command file, say `chemrun.bat`. In that case you may type:

```
chemrun mycase
```

to invoke both the simulation and the plotting.

B Description of general part of input file

Preliminaries

All input data are identified by well-defined character strings. The reaction equations always begin with `RE`, the start concentration of reactants always with `CON`, etc.. The order of the input data can be chosen freely according to the user's taste. The program is case insensitive for command typing, but case sensitive for typing the species names in the reaction equations.

Only plain ASCII text characters are applied, so you may just use your favorite editor to edit the CHEMSIMUL input files.

Reaction equations

Each reaction equation is prefixed with an identification `RE`n`:`, where `RE` means `R`eaction `E`quation and n is the equation identification number. The reaction equation terminates with a semicolon (`;`) followed by the thermodynamics constants `A`, `EA`, `B` (Section 4.3), and `Q` (Section 4.4), separated by a comma (`,`). Interspersed blanks for enhancing readability are allowed. Example:

```
RE3: H[+]+OH[-]=H2O; A=2.35E13, EA=3
RE2: H+HO2=OH+OH; A=2.5E11, EA=1.9, Q=38.3
RE64: 2*OH=H2O2; A=6.0E9
```

The stoichiometric constants are integers, while the rate constants are real numbers; their values can be entered in free format. CHEMSIMUL supports "scientific notation" for exponents, using the symbol E (or D) for the exponent, i.e. 1.4E11 means $1.4.10^{11}$. The maximum order of a chemical reaction to be simulated is limited to 2, i.e. up to 2 reactants can be written on the left hand side of the reaction equations, but there are virtually no restrictions on the number of products on the right hand side. The maximum number of characters in a species (i.e. reactant or product) is 16. The are no bounds on the number of reactions or species.

The terms `A`, `EA` and `B` are related to the modified Arrhenius expression for the rate constant given in Section 4.3, $k = AT^{\beta}\exp(-E_{\mathrm{a}}/(RT))$, where T is the temperature in Kelvin: `A` $= A$, `EA` $= E_{\mathrm{a}}$, and `B` $= \beta$. The factor T^{β} is included for gas phase kinetics, with β being an empirical exponent. A is the so-called frequency factor, and E_{a} the activation energy.

If only `A` is included in a reaction then its value will be just the rate constant.

The term `Q` corresponds to the specific heat formation q_r for reaction r, as described in Section 4.4.

Default values for `EA`, `B`, and `Q` are zero, while the input of `A` is mandatory for all reactions.

Names of species

Names of species may be chosen freely, apart from certain restrictions. The main rule is that they should be alphanumeric and begin with a letter; they may contain small and

capital letters and are interpreted in a case-sensitive way. However, species names may also be qualified by square brackets containing strings of `+` or `-`, like `FE[+++]` or `OH[-]`. This feature enables CHEMSIMUL to check the electro neutrality. It is also allowed to use square brackets as numerical markers of species names, like `A[2]`. The two modes may be combined, but not within a single pair. Thus `A[2][+]` would be legal but not `A[2+]`.

Concentration values

The command `CON(S) = ...` gives the numerical value of the start concentration for the reactant S. Example:

```
con(OH[-]) = 1.0E-7
```

Default values of the concentrations are 0, i.e. those chemical species in the reaction equations for which no start concentrations are given, are automatically assigned the initial concentration zero. The concentration unit is $mol \times dm^{-3}$ or $molecules \times cm^{-3}$.

G values

If the reaction system is irradiated e.g. by high-energetic electrons or by gamma rays, then the G-values must be specified. G-values were discussed in Sections 1.3 and 1.4. They are in units of *molecules* per $100eV$. If say the reactant OH is produced by an irradiation given by the total dose D *krad* during the radiation time t_{r}, then the rate of change of [OH] measured in $mol \times dm^{-3}$ is given by:

$$\frac{d[\mathrm{OH}]}{dt} = c \times G(\mathrm{OH}) \times D \,/\, t_{\mathrm{r}}, \tag{29}$$

where $c = 1.036.10^{-6}$ is the conversion constant. (The numerical value of c depends on density and units (Section 1.4) and may be reset by the command `CONVERT` (Section C); if e.g. Gy is used, then you should set $c = 1.036.10^{-7}$.) The G-values default to 0 in CHEMSIMUL. Example:

```
G(OH) = 2.7
```

Heat capacity values

In Section 4.4 it was discussed how CHEMSIMUL can deal with adiabatic processes. When simulating such a system, the specific heat capacity `HCV` $= c_v(\mathrm{R}_s)$ should be entered for the species involved as well as the reaction heat, `Q` $= q_r$, for all reactions. The proper units are $kcal \times mol^{-1} \times K^{-1}$ or $J \times mol^{-1} \times K^{-1}$ for `HCV` and $kcal \times mol^{-1}$ or $J \times mol^{-1}$ for `Q`. By default the specific heat capacity values are set to zero. Example:

```
hcv(H2O) = 0.018
```

Hint: In diluted aqueous solutions it is sufficient to enter `HCV` for water.

C Simulation commands

In the following we have collected all the various commands in CHEMSIMUL that in some sense are related to the simulation process. They are ordered alphabetically. Note that some of them are obligatory and are marked accordingly. Also note that an asterisk (`*`) in front of a command makes the command inoperative. For example:

```
*rstart
```

CASE:

This begins an identifying text for the case to be simulated. Example:

```
case: SIMULATION OF NOX.
```

It is possible to continue the case text over a second line. In that case the first character on the second line must be an asterisk (*).

CONVERT

Sets the conversion constant c (Sections 1.3 and 1.4), if you need another value than the default value $c = 1.036.10^{-6}$. Example:

```
convert = 1.036E-7
```

Recall that the conversion constant depends on the unit for dose:

Irradiation dose	Conversion constant
TOTALDOSE	CONVERT
krad	1.036E-6 (default)
Gy	1.036E-7

Table 2: Value of the conversion constant c

DIG

This command defines the number of significant digits in the result table of CHEMSIMUL. By default `DIG = 3`. Example:

```
dig = 5
```

DIFFEQ

If this command is written in the input file, then the differential equation system will be printed. By default the system is not printed. See also Sections 3.3 and Sections 2.2. Example:

```
diffeq
```

ENDDATA

Obligatory.
This command ends the data for a simulation, and commands after `ENDDATA` are ignored. Example:

```
enddata
```

EPS

The command `EPS =` ε_{rel} gives the relative accuracy ε_{rel} in the integration routine (LSODA). Its default value is $1.0.10^{-3}$. Example:

```
eps = 1.0e-4
```

Hint: Normally, but not always, the computing time increases when `EPS` *is made smaller. Certain difficult cases may require a very small value of* `EPS` *to prevent integration failures.*

FSTSTP

The command `FSTSTP =` h_0 gives the initial integration step h_0, measured in seconds. The program has automatic step size control and is able to estimate its own initial step by default. Example:

```
fststp = 1.0e-8
```

Hint: To see the actual first step, use the `MATHINFO` *command. The experienced user may sometimes save time by setting a larger* `FSTSTP`*.*

HMAX

The command `HMAX =` $h_{\max}$ gives the maximum allowed length $h_{\max}$ of an integration step in seconds. By default the integrator is free to use as large steps as it wants. Example:

```
hmax = 0.001
```

Hint: To see the maximal step actually applied by the code, use the `MATHINFO` *command. The experienced user may sometimes want to set a smaller* `HMAX` *to enhance the precision.*

MATHINFO

When this command is present, some statistics about the integration are printed in the result file. Furthermore, an additional file `message.ode` is created with a few more, rather technical, details. Example:

```
mathinfo
```

MAXCON

To find the maximum concentration value $[\mathrm{S}]_{\max}$, together with the formation and decay half-lives (in *s*) for a species S (Section 4.6), you should type the command `MAXCON(S)`. Example:

```
maxcon(H[+])
```

MODE

This command defines the thermodynamic mode of the system. For `MODE = ISO`, the system is considered isothermal, while for `MODE = ADI` it is adiabatic. By default CHEMSIMUL assumes `MODE = ISO` unless there is at least one non-zero `HCV` value, in which case `MODE = ADI` is assumed. Example:

```
mode = adi
```

NORMALIZE

With this command CHEMSIMUL will normalize the plot by setting the maximum value of the calculated plot expressions equal to 1.0. Experimental files cannot be plotted when `NORMALIZE` is used. Example:

```
normalize
```

NRR

This command may be used in case of irradiation. `NRR` states the number of individual pulses in a pulse train. Default value is 1 in case of irradiation and otherwise 0. Thus the command is only needed when there are more than 1 pulse. Example:

```
nrr = 5
```

OUTTYPE

Defines scaling type of the plot. There are four different types. By default `OUTTYPE = 1`.
`OUTTYPE = 1`, linear time and concentration scales.
`OUTTYPE = 2`, linear time, logarithmic concentration scales.
`OUTTYPE = 3`, logarithmic time, linear concentration scales.
`OUTTYPE = 4`, logarithmic time and concentration scales.

Example:

```
outtype = 3
```

PE

CHEMSIMUL supports the so-called *plot expressions*. These admit general arithmetic expressions in the species concentrations and/or heat (`HEAT`) and temperature (`TEMP`) to be plotted as functions of time. Allowed operators are sum (+), difference (−), product (∗), and exponentiation (`^`) (cf. Section 8.5). Furthermore, the stanadard mathematical functions `cos`, `exp`, `log` (natural logarithm), `log10` (logarithm with base 10), `sin`, `sqrt`, can be used. Plot expressions are initiated by `PE` followed by a number (in sequence), and a colon. Example:

```
PE1: OH
PE2: TEMP-273
PE3: 210*H2O2+1800*O2[-]-FE[+++]^1.5/220
```

When an invalid operation occurs during evaluation of a plot expression, e.g. a zero division, a warning will be printed on first instance only and the result cleared to zero.

PLONPA

Defines the maximum number of curves (plot expressions) on each plot. The command is also used to seperate plots with corresponding experimental data. By default `PLONPA =` *number of plot expressions*. Example:

```
plonpa = 2
```

`GNUPLOT` generates automatically the maximum value on the y-axis according to the highest calculated value of any plot expression in a plot. If, however, an experimental file is included, then the maximum value on the y-axis will be chosen according to the highest experimental value.

PRINTS

Normal case: The command `PRINTS =` n defines the total number of output lines in the result table after time-zero, such that $n + 1$ lines are printed. Default value is $n = 0$. Each line contains the time and the concentration results belonging to that time.
Multi-pulse case `NRR` > 1: Here `PRINTS` $*$ `NRR` becomes the total number of output line intervals. Example:

```
prints = 25
```

R

The gas constant in the units you chose. This command must be used if you use activation energy expressions in the reactions (Section 4.3). It is also needed for calculation of the heat production from the irradiation. Example:

```
r = 1.9864E-3
```

RADPRS

This command should be used in case of external irradiation. `RADPRS=` n_r defines the number n_r of output line intervals in the result table during an irradiation pulse. Default value is 0. Note: `RADPRS` must satisfy the condition $0 \leq$ `RADPRS` $\leq$ `PRINTS`. See also `PRINTS`. Example:

```
radprs = 5
```

RADTIME

This command should be used in case of irradiation. `RADTIME=` t_r gives the irradiation time t_r in seconds for the rectangular pulse (or for each pulse if there are more than 1). Default value is zero. Example:

```
radtime = 1E-8
```

RSTART

With this command CHEMSIMUL will produce a restart file holding the concentration of all species at `TEND`. This file contains the concentration value for a continuation of the simulation. The restart file has the name of the input file with the extension `.sav`. The continuation is made by using the same `.dat` file, only with the data in the restart file inserted at the end, just before the end command `ENDDATA`. When the program reads the `.dat` file, then the input of the concentration values are overwritten by the values from the restart file. Example:

```
rstart
```

If this command is used e.g. in the input file `nox.dat`, then the restart file gets the name `nox.sav`. The continuation is made by pasting the `.sav` file at the end (just before `ENDDATA`) of the `.dat` file that you want to use for the simulation.

The restart facility may e.g. be used in the simulation of circulating cooling water in reactors where the different stages in the cooling cycle present different reaction conditions like dose rate and temperature. The results from the simulation of the first stage is used as start concentrations in the input file for the second stage, the results from the second stage as input in the third stage, and so on, until the cycle is completed.

TEMP

The command `TEMP =` T defines the temperature T of the reaction system in degree Kelvin (K). By default $T = 300$. In a themodynamic simulation, where the temperature is supposed to vary, `TEMP` means the initial value of the temperature. Example:

```
temp = 455
```

TEND

Obligatory.
The command `TEND =` t_{end} gives the end time t_{end} in s for the simulation. Example:

```
tend = 12
```

TENDRAD

This command is used in case of pulse-train irradiation. `TENDRAD` $= t_{end}^{rad}$ is given in s and is equal to the trailing edge of the last pulse as described in Section 4.7. See also the command `NRR`. `TENDRAD` is obligatory when `NRR` > 1. When `NRR` $= 1$ `TENDRAD` is automatically set to `RADTIME` and need not be set by the user. Example:

```
tendrad = 2.0e-5
```

THIRD

The command `THIRD(M) =` [M] includes a third-body species M in the rate constant for its reaction as explained in Section 4.5. Example:

```
third(M) = 0.013
```

TOTALDOSE

The command `TOTALDOSE =` D gives the total irradiation dose D in *krad* during the irradiation time for the rectangular pulse (or the total train for more than 1 pulse). Example:

```
totaldose = 3.7
```

If `TOTALDOSE` is entered in Gy, the the conversion constant c must be set to $1.036.10^{-7}$ by the command `CONVERT`.

Use of the command `TOTALDOSE` precludes the use of the `ISOTOPES` block decribed in Section F.

UNBAL

The command `UNBAL` enables the user to overrule the halting of the simulation when a stoichiometric unbalance has been detected. The check itself can never be switched off. WARNING: This facility should only be applied in exceptional cases; reliable simulation results are only guaranteed when the mass balance is fulfilled. Example:

```
unbal
```

D Description of the experimental file

The name of the experimental file can be chosen freely. The file contains pairs of experimental data in two columns separated by an optional comma and at least one space. The time is in the first column and the measured value (corresponding to a plot expression PE) in the second. Example:

```
0 0
2E-6 2.4E-4
4E-6 3.2E-4
6E-6 3.6E-4
```

E Sample data set

The complete input data file for the sample combustion test that produced the output in Section 3.1 is listed below:

```
RE1: H+H=H2; A=4.0E7
RE2: H+O2=HO2; A=4.5E8
RE3: H+HO2=OH+OH; A=6.5E10
RE4: HO2+HO2=H2O2+O2; A=2.0E9
RE5: OH+OH=H2O2; A=4.0E9
RE6: H+OH=H2O; A=1.0E10
RE7: OH+HO2=H2O+O2; A=6.0E10
RE8: OH+H2=H2O+H; A=4.0E3
G(H)=1.0
```

```
TOTALDOSE=0.9
CON(O2)=2.0E-4
CON(H2)=4.0E-3
RADTIME=5.0E-9
PRINTS=12
RADPRS=2
TEND=1.0E-3
PE1:HO2
*DIFFEQ
CASE: 5 mbar O2 + 100 mbar H2 + Ar to 1 atm. k8=4000
ENDDATA
```

F Isotope input and output

We shall here give the format of input data for the mixed-particle irradiation facility described in Section 4.8. All isotope data must be confined to a single block in the data file. This block begins with the command `ISOTOPES =` m, where m is the total number of isotopes, and it ends with the command `END ISOTOPES`. The isotope block may be positioned freely in the data file before `ENDDATA`, and its data may come in arbitrary order.

With reference to Section 4.8 we write `PERIOD(...)`, `DA(...)`, `DB(...)`, `DG(...)`, `DN(...)` for τ_i, D_i^α, D_i^β, D_i^γ, D_i^n, respectively. Within the parantheses we write an identification for the isotope, which can be numeric or alphabetic, as `PERIOD(1)` or `PERIOD(Am241)`. Moreover we write `GA(R)`, `GB(R)`, `GG(R)`, `GN(R)` for the G-values G_s^α, G_s^β, G_s^γ, G_s^n, for a given species R = R$_s$. By default all periods τ_i are ∞. Thus if half life is not entered in the input data there will be no decay, and mixed-particle irradiation in reactors can be simulated.

Comment lines within the isotope blocks are accepted.

Note that use of the `ISOTOPES` block precludes the use of the command `TOTALDOSE` in Section C.

The preparation of input for a isotope block should be clear from the following example:

```
Isotopes = 5
PERIOD(Am241)=1.365E10
PERIOD(Ba137m)=9.515E8
PERIOD(Ce144)=2.462E7
PERIOD(Cm244)=5.715E8
PERIOD(Co57)=2.348E7
DA(Am241)=1.E-4
DG(Am241)=1.6983E-9
DG(Ba137m)=2.1469E-4
DG(Ce144)=9.4689E-7
DG(Cm244)=1.4916E-10
DG(Co57)=9.5811E-7

GA(E[-])=0.06
GA(H)=0.21
GA(OH)=0.24
GA(H2O2)=0.985
GA(HO2)=0.22
GA(OH[-])=0
GA(H3O[+])=0.06
GA(H2O)=-2.71
GA(H2)=1.3

GG(E[-])=2.8
GG(H)=0.55
GG(OH)=3.0
GG(H2O2)=0.6
GG(HO2)=0
GG(OH[-])=0.5
GG(H3O[+])=3.3
GG(H2O)=-8.0
GG(H2)=0.425
End Isotopes
```

Next we show how these data will appear in the input-echo part of the result file:

```
ISOTOPE INPUT
-------------
                                I N I T I A L  D O S E  R A T E S
ISOTOPE       PERIOD         A           B           G           N
************************************************************************
Am241      1.36500E+10  1.00000E-04              1.69830E-09
Ba137m     9.51500E+08                           2.14690E-04
Ce144      2.46200E+07                           9.46890E-07
Cm244      5.71500E+08                           1.49160E-10
Co57       2.34800E+07                           9.58110E-07
************************************************************************

                                I S O T O P I C  G - V A L U E S
SPECIES                        A           B           G           N
************************************************************************
E[-]                       0.060000                 2.800000
H                          0.210000                 0.550000
OH                         0.240000                 3.000000
OH[-]                                               0.500000
H2O2                       0.985000                 0.600000
HO2                        0.220000
H2                         1.300000                 0.425000
H2O                       -2.710000                -8.000000
H3O[+]                     0.060000                 3.300000
************************************************************************
```

At the end of the result file you will find a table of integrated doses originating from the decaying isotopes. It looks as follows:

```
             INTEGRATED DOSE FROM DECAYING ISOTOPES
-----------------------------------------------------------------------
    TIME       ADOSE       BDOSE       GDOSE       NDOSE
 0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
 1.3149E+04  1.3149E+00  0.0000E+00  2.8480E+00  0.0000E+00
 2.6298E+04  2.6298E+00  0.0000E+00  5.6960E+00  0.0000E+00
 3.9447E+04  3.9447E+00  0.0000E+00  8.5439E+00  0.0000E+00
 5.2596E+04  5.2596E+00  0.0000E+00  1.1392E+01  0.0000E+00
 6.5745E+04  6.5745E+00  0.0000E+00  1.4240E+01  0.0000E+00
  .
  .
  .
 3.0900E+06  3.0898E+02  0.0000E+00  6.6829E+02  0.0000E+00
 3.1032E+06  3.1029E+02  0.0000E+00  6.7113E+02  0.0000E+00
 3.1163E+06  3.1161E+02  0.0000E+00  6.7397E+02  0.0000E+00
 3.1295E+06  3.1292E+02  0.0000E+00  6.7681E+02  0.0000E+00
 3.1426E+06  3.1424E+02  0.0000E+00  6.7964E+02  0.0000E+00
 3.1558E+06  3.1555E+02  0.0000E+00  6.8248E+02  0.0000E+00
```

6 Hints for solving special problems

6.1 How to mark a reaction

If you want to know how much material is passing through a single reaction you can mark the reaction without sacrificing the material balance: The reaction

$$\texttt{RE1: A} + \texttt{B} = \texttt{C} + \texttt{D};\ \texttt{A} = k_1 \tag{30}$$

is marked like this:

$$\texttt{RE1: A} + \texttt{B} = \texttt{AB};\ \texttt{A} = k_1 \tag{31}$$

$$\texttt{RE2: AB} + \texttt{M} = \texttt{C} + \texttt{D} + \texttt{N};\ \texttt{A} = k_2 \tag{32}$$

$$\texttt{CON(M)} = M \tag{33}$$

To ensure that `RE1` is rate determining during the whole simulation you must choose high values for [M] and k_2:

$$[\text{M}] \times [\text{AB}] \times k_2 \gg [\text{A}] \times [\text{B}] \times k_1 \tag{34}$$

The simulated concentration [N] represents the yield of this reaction.

6.2 Maintaining a constant concentration of a solute

The program is designed to preserve the mass balance unconditionally, so if you want to maintain a constant concentration for solute `S` you may use either of two possible methods:

6.2.1 The equilibrium method

You must assign a high `DUMMY` concentration in equilibrium with `S`:

$$\texttt{CON(DUMMY)} = M \tag{35}$$

$$\texttt{RE1: DUMMY} = \texttt{S};\ \texttt{A} = a \tag{36}$$

$$\texttt{RE2: S} = \texttt{DUMMY};\ \texttt{A} = b \tag{37}$$

Choose a value for the rate constant a (e.g. 100) and calculate b from the equation:

$$[\text{DUMMY}] \times a = [\text{S}] \times b \tag{38}$$

The concentration $\texttt{CON(DUMMY)} = M$ must be so high that the change in concentration during the simulation is insignificant.

6.2.2 The conservation method

This method recreates S whenever S is consumed in a reaction, and removes S whenever S is produced in a reaction. Consider the reactions:

$$\begin{gathered}\texttt{RE1: S} + \texttt{N} = \texttt{S}[-] + \texttt{N}[+];\ \texttt{A} = a \\ \texttt{RE2: C} + \texttt{B} = \texttt{S} + \texttt{D};\ \texttt{A} = b\end{gathered} \tag{39}$$

Assign a high `DUMMY` concentration as in (35). Change the equations as follows:

$$\begin{gathered}\texttt{RE1: S} + \texttt{N} = \texttt{S}[-] + \texttt{XN}[+];\ \texttt{A} = a \\ \texttt{RE2: C} + \texttt{B} = \texttt{DUMMY} + \texttt{D};\ \texttt{A} = b\end{gathered} \tag{40}$$

and add the reaction

$$\texttt{RE3: XN}[+] + \texttt{DUMMY} = \texttt{S} + \texttt{N}[+];\ \texttt{A} = c \tag{41}$$

Choose the rate constant c so high that S is produced much faster in `RE3` than it is consumed in `RE1`.

6.3 Maintenance of gas/liquid equilibrium

If you have a gas phase (volume = g) in equilibrium with a liquid phase (volume = ℓ) and you want to maintain the equilibrium of for example oxygen during the simulation, this can be accomplished in the following way:

First determine the equilibrium concentration ratio of liquid oxygen O_2 and gas phase oxygen $O_{2,gas}$:

$$\frac{[O_2]}{[O_{2,gas}]} = r \tag{42}$$

Then calculate [DUMMY] as the oxygen concentration if the gas phase has the same volume as the liquid phase:

$$[\text{DUMMY}] = [O_2] \times \frac{g}{\ell} \tag{43}$$

Finally determine the rate constants a and b in the equilibrium reactions

$$\texttt{RE1: DUMMY} = \texttt{O2; A} = a \tag{44}$$

$$\texttt{RE2: O2} = \texttt{DUMMY; A} = b \tag{45}$$

Choose the value for a (e.g. 100) and calculate b from the equations:

$$[\text{DUMMY}] \times a = [O_2] \times b \tag{46}$$

and

$$\frac{[O_2]}{[\text{DUMMY}]} = r \tag{47}$$

6.4 Transport, dissolution and diffusion.

Problems with transport, dissolution, and diffusion, can be solved by using a first or zero order reaction. In case of diffusion in or out of the solution a first order reaction should be used:

$$\texttt{RE1: H2} = \texttt{DUMMY; A} = f(D) \tag{48}$$

where f is a function of the diffusion constant D.

This method can also be used to modify adiabatic systems with transport of heat in or out of the system by using a first order reaction with a `Q` value:

$$\texttt{RE1: S} = \texttt{DUMMY; A} = a\texttt{; Q} = q \tag{49}$$

Then the heat transport will be determined by:

$$\frac{dQ}{dt} = a \times [\texttt{M}] \times q \tag{50}$$

In case of dissolution or precipitation two zero order reactions can be used:

$$\texttt{G(solute)} = a \tag{51}$$

and

$$\texttt{G(solid)} = -a \tag{52}$$

If you use a total dose D_{tot} *krad*, the irradiation time t_{r} *s* and `G(solute) = 1`, then

$$\frac{d[\texttt{solute}]}{dt} = c\,\frac{D_{\text{tot}}}{t_{\text{r}}}\ mol \times dm^{-3} \times s^{-1} \tag{53}$$

where $c = 1.036 \cdot 10^{-6}$ is the conversion constant.

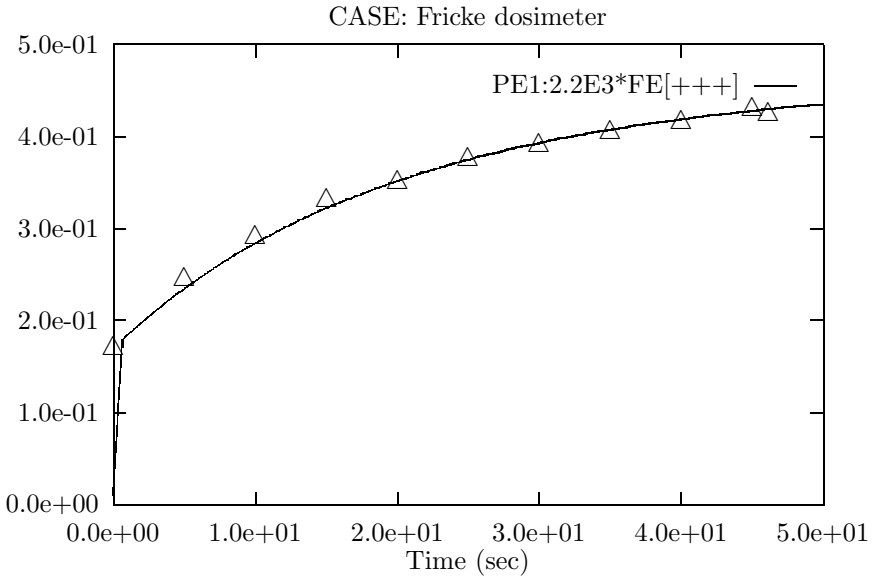
A saturated solution in equilibrium with solid matter is treated as the gas equilibrium. Transport in form of addition into the system must be solved according to the actual method of addition. In case of a continuous addition during the the whole computation time either a zero order reaction (G-value) or a first-order reaction can be used. If the addition is instantaneous at time t you can simulate the reactions until time t using the `RSTART` command. Then add the concentration of the new solute to the `.SAV` file and enter the `.SAV` concentrations into your `.DAT` file as new start concentrations for the continued simulation.

7 Simulation examples

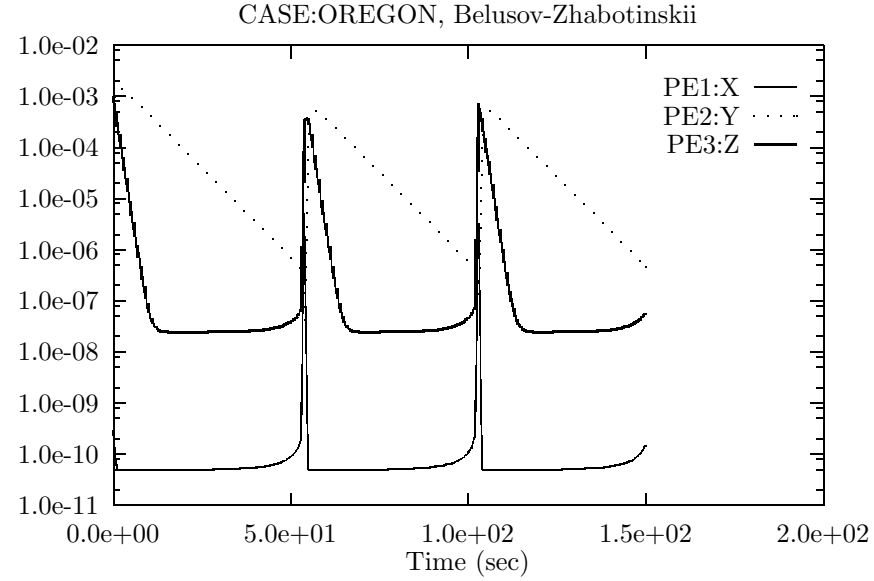
Rasmussen and Bjergbakke [1] give several examples of simulations using an earlier version of CHEMSIMUL. Their examples span from dosimetry and NO_x processes in the atmosphere to the study of radiolysis of ground water from spent nuclear fuel. They also simulate classical processes, notably the Belusov-Zhabotinskij oscillating reaction, which has gained renewed interest as a chemical example of a dynamic system with the typical nonlinear features of limit cycles, period doubling, and eventually chaos.

Here we shall only illustrate two of these test runs with graphical output obtained with CHEMSIMUL plot expressions and GNUPLOT. Details for these examples are found in Rasmussen and Bjergbakke [1].

7.1 Fricke dosimeter



7.2 Oregonator



7.3 Further references to works using CHEMSIMUL

Moreover, CHEMSIMUL was used in numerous research projects through the last decades. To give a few references to relevant publications, we mention the papers by Christensen and Bjergbakke [11, 12, 13], by Hart *et al.* [14], and by Bjergbakke *et al.* [15, 16].

8 Mathematical implementation

In the following we shall discuss the principal features in the mathematical implementation of the CHEMSIMUL program. Readers without special interest for the CHEMSIMUL mathematics may skip this chapter.

8.1 Translation principle for reactions

To explain the mechanism of the CHEMSIMUL translator, let us assume a general chemical system being specified with M reactions involving altogether N species. We write the reaction system in the following way:

$$\mathrm{Re}(r): \quad \alpha_j \times \mathrm{R}_j + \alpha_k \times \mathrm{R}_k \to \sum \alpha_\ell \times \mathrm{R}_\ell, \quad r = 1, \ldots, M, \tag{54}$$

where R_s is the chemical symbol for species s, $s = 1, \ldots, N$. The left-hand side is empty for a zero-order reaction. Otherwise α_j, α_k, α_ℓ are non-negative integers satisfying $1 \le \alpha_j + \alpha_k \le 2$. For a second-order reaction we may have $j \ne k$ or $j = k$; in both cases $\alpha_j + \alpha_k = 2$. Each reaction $\mathrm{Re}(r)$ is assigned a rate constant k_r. Moving the reactant terms in (54) to the right-hand side, we may assign with every species s in reaction r a *stoichiometric coefficient* a_{rs}, which is positive for a product and negative for a reactant. Setting unassigned entries $a_{rs} = 0$, we obtain the *stoichiometric matrix* $\mathbf{A} = \{a_{rs}\} \in \mathbb{Z}^{M \times N}$. This enables us to express the reaction system in compact vector-matrix notation as

$$\mathbf{A}\mathbf{v} = \mathbf{0}, \tag{55}$$

where the N-dimensional *species vector* $\mathbf{v}$ contains the distinct species names R_s, $s = 1, \ldots, N$. The resulting general ODE system for the species concentrations has the form

$$\frac{d[\mathrm{R}_s]}{dt} = \sum_{j,k} n_{jks} M_{jks} [\mathrm{R}_j][\mathrm{R}_k] + \sum_j n_{js} M_{js} [\mathrm{R}_j] + c\, G_s D(t), \quad s = 1, \ldots, N. \tag{56}$$

The right-hand side of (56) is the sum of a quadratic form coming from the second-order reactions, a linear form from the first-order reactions, and a term coming from the radiolytic zero-order reactions. The 3-dimensional tensors n_{jks} and M_{jks} contain the stoichiometric coefficients a_{rs} and the reaction rates k_r, respectively. For a given s, the sum is taken over pairs (j, k) for which $(\mathrm{R}_j, \mathrm{R}_k)$ is a reacting pair in some reaction $r = r(j, k)$ involving R_s on either side, with $n_{jks} = a_{rs}$ and $M_{jks} = k_r$. A similar interpretation holds for the linear form. For simplicity we here assume that the zero-order term has the form $c\, G_s D(t)$, where c is the conversion constant, G_s the G-value for species s, and $D(t)$ the dose rate at time t. (In case of mixed-particle irradiation the zero-order term is given by the right-hand side of (28).)

Given a computerized write-up like (10) of the reactions (54), CHEMSIMUL is now able to assemble the ODE system (11) or (56) from all the individual pieces in the reactions. On encounter it tabulates every new species and includes its name in the symbol set S. It also stores the rate constants k_r. After the scan phase an assembly phase is invoked. Here CHEMSIMUL constructs a table with a line for each term on the right-hand side of the total ODE system. Entry 2 in the line contains the stoichiometric coefficients n_{jks}, while the other entries contain address pointers to 1) the species s, 3) the reaction r, 4) $[\mathrm{R}_j]$, and 5) $[\mathrm{R}_k]$ (for first-order processes we may set $[\mathrm{R}_k] = 1$). This pointer table gives a complete description of the ODE system and permits the evaluation of its Jacobian.

8.2 Stoichiometric balance check by linear programming

The accurate solution of the ODE system in chemical kinetics requires an overall conservation of the mass balance, as pointed out by Ridler *et al.* [17]. This balance may be affected by two main factors, the reaction mechanism and the integration process. Being a linear multistep method, the numerical integration scheme in CHEMSIMUL has the implicit property of mass preservation, as shown by Rosenbaum [18, 19]. Hence it is sufficient to make a static consistency check, and as stated in Section 4.1, such a check should be based on the demand of existence of strictly positive solutions to the balance equation (13) or (55) where $\mathbf{A}$ is the stoichiometric matrix and $\mathbf{v}$ the species vector.

Ridler *et al.* [17] proposed a heuristic checking method based on Gaussian elimination in integer arithmetic on $\mathbf{A}$, and that method was adopted by Rasmussen and Bjergbakke in [1]. Later we have replaced it by a mathematically more stringent method: By the homogeneity of (55) we can replace the demand of positive values by a demand of all values be ≥ 1. Thus, if $\mathbf{A} = \{a_{ij}\}$, it must be possible to find solutions $y_j \geq 1$ to the equations

$$\sum_{j=1}^{N} a_{ij} y_j = 0, \quad i = 1, \ldots, M. \tag{57}$$

On writing $y_j = x_j + 1$ we may replace (57) by the system

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \tag{58}$$

where $\mathbf{x} = \{x_j\}$ and $\mathbf{b} = \{b_i\}$ with $b_i = -\sum_{j=1}^{N} a_{ij}$. To decide the feasibility of (58) is a standard linear-programming problem. It can be solved efficiently by the simplex method.

8.3 Solution of the ODE system

The ODE systems in chemical kinetics are non-linear and often autonomous. As mentioned in Section 1.2 the corresponding initial-value problems may be of the stiff type. After many years of experience we are still in favor of using the LSODA solver from Hindmarsh' ODEPACK collection, written by Petzold [4] and Hindmarsh [20]. This code switches automatically between stiff and nonstiff methods, starting with a nonstiff (Adams) method.

When the stiff (BDF) procedure is invoked, the Jacobian df/dy will be needed. Experience shows that it is more efficient to supply an analytical Jacobian than to let LSODA compute it internally by difference approximations. For kinetics problems df/dy is normally simple and cheap to calculate. There might, however, be a concern about space, if we want to solve really big problems, because the Jacobian is treated as a full matrix.

LSODA admits a combined relative and absolute error control: It controls the vector $\mathbf{e} = \{e_i\}$ of estimated local errors in the solution $\mathbf{y} = \{y_i\}$ by the restriction $\|\mathbf{r}\|_\infty \leq 1$, where $\mathbf{r} = \{r_i\}$ is given by $r_i = e_i/w_i$ with $w_i = \varepsilon_i^{\mathrm{rel}}|y_i| + \varepsilon_i^{\mathrm{abs}}$. Normally CHEMSIMUL will not need this flexibility; usually fixed relative and absolute tolerances $\varepsilon_i^{\mathrm{rel}} = \varepsilon_{\mathrm{rel}}$ and $\varepsilon_i^{\mathrm{abs}} = \varepsilon_{\mathrm{abs}}$, $\forall i$, give good results in practice.

Simulations over very long time spans may exploit the restart facility provided by LSODA.

In radiolytic simulations, where the radiation burst induces a constant source term in the finite time slot $[0, t_{\mathrm{r}}]$ and then disappears, we found that the best way of using the solver was to select the one-step mode in combination with a critical time barrier $t = t_{\mathrm{r}}$ (ITASK = 5 in LSODA). Integration beyond t_{r} proceeds *ab initio* from $t = t_{\mathrm{r}}$.

Other useful options in LSODA are specification of a minimum step length $h_{\min}$ (not exploited in the present CHEMSIMUL version), a maximum step length $h_{\max}$, and an initial step length h_0 to be attempted instead of the internally computed default value. On combining these possibilities judiciously we can solve quite hard kinetics problems efficiently. In rare cases, where a species approaches zero concentration very rapidly, we have seen traces of instability with negative concentrations. The standard remedy here is to decrease the tolerance $\varepsilon_{\mathrm{rel}}$ and/or to restrict $h_{\max}$.

8.4 The Jacobian

When LSODA switches to its stiff integration procedure, it will need the Jacobian $d\mathbf{f}/d\mathbf{y}$ of the right hand side of the ODE system

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}, t) \tag{59}$$

This is a square matrix whose order m equals the dimension of $\mathbf{y}$. As mentioned in Section 8.3 it is better to supply an analytical Jacobian than to rely on an internal numerical estimation within LSODA. Let us first consider the case where the vector $\mathbf{y}$ contains concentrations only. In that case $m = N$, i.e. the number of species, and it is easy to compute $d\mathbf{f}/d\mathbf{y}$ from (56); see also the ODE system (11) for our sample case. We find for the (s, k)-element:

$$\Big(\frac{d\mathbf{f}}{d\mathbf{y}}\Big)_{sk} = \sum_j (n_{jks}M_{jks} + n_{kjs}M_{kjs})[\mathrm{R}_j] + n_{ks}M_{ks}. \tag{60}$$

For the interpretation of the j-sum, see similar comments in Section 8.1.

Incorporation of the adiabatic process descibed in Section 4.4 induces an augmentation of the Jacobian with two extra rows and columns, such that now $m = N + 2$. Row and column $m - 1$ refer to the total heat formation Q_{tot}, while row and column m refer to the temperature T. The entire column $m - 1$ becomes zero. The first N entries of row $m - 1$ are computed from (15):

$$\Big(\frac{d\mathbf{f}}{d\mathbf{y}}\Big)_{m-1,\ell} = \sum_{r=1}^{M} k_r q_r (\delta_{j\ell}[\mathrm{R}_k] + \delta_{k\ell}[\mathrm{R}_j]), \quad \ell = 1, \ldots, N, \tag{61}$$

where the species indices $j = j(r)$ and $k = k(r)$ depend on the reaction r, and where δ_{ij} is the Kronecker delta. For first-order reactions r the corresponding summand in (61) should read $k_r q_r \delta_{j\ell}$. If the reaction rates k_r are independent of the temperature T, then the $(m - 1, m)$-entry is zero. If, on the other hand, there is a T-dependence through (14), then

$$\Big(\frac{d\mathbf{f}}{d\mathbf{y}}\Big)_{m-1,m} = \sum_{r=1}^{M} \frac{\partial k_r}{\partial T}[\mathrm{R}_j][\mathrm{R}_k] q_r, \tag{62}$$

where the temperature derivative of $k_r = k$ is found from (14):

$$\frac{\partial k}{\partial T} = AT^{\beta} \exp(-E_{\mathrm{a}}/(RT)) \Big(\frac{\beta}{T} + \frac{E_{\mathrm{a}}}{RT^2}\Big). \tag{63}$$

For the last row we find from (16):

$$\Big(\frac{d\mathbf{f}}{d\mathbf{y}}\Big)_{m\ell} = \frac{1}{S_{\mathrm{cap}}^2} \Big(\Big(\frac{d\mathbf{f}}{d\mathbf{y}}\Big)_{m-1,\ell} S_{\mathrm{cap}} - S_{\mathrm{heat}} c_v(\mathrm{R}_\ell) \Big), \quad \ell = 1, \ldots, N, \tag{64}$$

where S_{heat} and S_{cap} are the numerator and denominator, respectively, in (16). Moreover

$$\Big(\frac{d\mathbf{f}}{d\mathbf{y}}\Big)_{mm} = \frac{1}{S_{\mathrm{cap}}} \Big(\frac{d\mathbf{f}}{d\mathbf{y}}\Big)_{m-1,m}. \tag{65}$$

Finally, the first N entries of the last column are computed from (56):

$$\Big(\frac{d\mathbf{f}}{d\mathbf{y}}\Big)_{sm} = \sum_{j,k} n_{jks} \frac{\partial M_{jks}}{\partial T}[\mathrm{R}_j][\mathrm{R}_k] + \sum_j n_{js} \frac{\partial M_{js}}{\partial T}[\mathrm{R}_j], \quad s = 1, \ldots, N, \tag{66}$$

where the temperature derivative of the reaction rates are evaluated as in (63).

The dose terms in (56) or (28) bear no influence on the Jacobian.

8.5 Implementation of plot expressions

When describing the implementation of plot expressions (cf. `PE` in Section C of the User's Guide) we use an example:

```
PE1: 4711.77 + (log10(OH[-])/7.45)^1.5
```

The interpretation of such an expression has two phases, a lexicographical analysis and a successive processing. The lexicographical analysis decomposes the plot expression into a list of *atoms*. Each atom is a constant, a species name, an arithmetic operator, or a function name. The 7 possible operators are those of elementary algebra: $*$, /, $+$, $-$, ^, (, and), where ^ stands for exponentiation; depending on context $+$ and $-$ can be infix or unary. The 6 standard mathematical functions cos, exp, log, $\log_{10}$, sin, and $\sqrt{\ }$ are supported. The string of atoms is ended with an "empty atom", which is considered as an extra operator. In the above example we get the following table of atoms, where species are supposed to be assigned definite concentration values:

Atom#	Atom	Type
1	`4711.77`	Constant
2	`+`	Operator
3	`(`	Operator
4	`log10`	Function
5	`(`	Operator
6	`OH[-]`	Species
7	`)`	Operator
8	`/`	Operator
9	`7.45`	Constant
10	`)`	Operator
11	`^`	Operator
12	`1.5`	Constant
13	empty	Operator

We process this table atom by atom using common techniques from computer science: We create and maintain two stacks **V** and **O**, the first for values (operands) and the second for operators. Moreover we have chosen to use a separate stack **F** for the mathematical functions.

An operand may be a constant, a concentration value, or a computation result. A fresh operand always pushes **V** and enters its top. Likewise a fresh function name pushes **F** and enters its top.

Let N be a new operator and T the current top-of-stack operator. If $N =$ "(", it pushes **O** and enters its top. If $|\mathbf{O}| = 0$ the same happens: N pushes **O** and goes to its top. If $|\mathbf{O}| > 0$ we observe whether $T =$ "(" $\wedge\, N =$ ")". If so, T is devoured and **O** is popped. We now check whether a function call waits on top of **F**; if so it is executed and **F** popped too. Then a new atom enters. If N was not "(", it will be compared with T according to priorities π: If $\pi(N) \leq \pi(T)$, a unary or binary operation takes place on the current top value(s) and **O** is popped, while the top value is updated with the current result. If the operation was binary, **V** is popped, too. After popping **O** we go back in the algorithm to the point where we checked if $|\mathbf{O}| = 0$. If, on the other hand, $\pi(N) > \pi(T)$, then **O** is pushed and N stored on its top.

After storing the operator N, a new atom will be processed. The algorithm terminates when the empty operator enters the top of **O**. When this happens we must have $|\mathbf{V}| = 1$.

We use the following operator priority list (from low to high):

$$\{\text{empty}\}, \quad \{(\ ,\ )\} \quad \{\text{infix}+,\ \text{infix}-\}, \quad \{*,\ /\}, \quad \hat{}, \quad \{\text{unary}+,\ \text{unary}-\}. \tag{68}$$

The rules given above are fairly standard for processing expressions in programming languages and calculators. Quite general formulas can be treated in this way, like for instance

$$(+A + 0.19 * B * (B - 3) * C * Q - \sqrt{J/40}) * (-E/F * 0.2 + 1/G\hat{}3) \tag{69}$$

9 Updates since previous versions

CHEMSIMUL has undergone a number of revisions and enlargements since the original 1984 edition, documented by Rasmussen and Bjergbakke [1].

The previous CHEMSIMUL version dates back to 1993. By and large, this was a FORTRAN translation of the 1984 ALGOL code, with few new facilities, notably the modified Arrhenius rate constants (Section 4.3) and inclusion of adiabatic processes (Section 4.4).

The present 1999 edition, written in FORTRAN 90, has many new and improved features compared to preceding versions:

9.1 Since 1993

- The program is made independent of specific units (Section 1.4).
- The mass balance check is made mathematically correct by linear programming technique (Section 8.2).
- The diagnostics for input errors are improved and made more precise.
- There is no limit on the number of reactions.
- There is no limit on the number of species.
- There is no limit on the number of terms in differential equations.
- Decaying mixed-particle radiation can be handled (Section 4.8).
- There is a test for electro neutrality for each reaction (Section 4.2).
- There is a test for electro neutrality for the totality of G values (Section 4.2).
- Gas constant R entered in input file when needed (Section C of the User's Guide).
- Conversion constant c may be modified in the input (Section C of the User's Guide).
- An erroneous evaluation of the analytical Jacobian for adiabatic processes was corrected.
- Case-sensitive interpretation of species names.
- Year 2000 protection.

9.2 Since 1998

- A train of pulses can be handled by the new command `NRR`.
- Heat production from external irradiation as well as from isotope radiation can be taken into account. (Section 4.4).
- The program may compute and print integrated doses from decaying isotopes.
- Isotope identification need no longer be numeric, but can also be alphabetic.
- Decaying mixed-particle input is included in the input echo.
- Comment lines made possible also within isotope block.
- Exponentiation operator is allowed in plot expressions (Section C of the User's Guide).
- Mathematical functions allowed in plot expressions: cos, exp, log, $\log_{10}$, sin, $\sqrt{\ }$ (Section C of the User's Guide).
- Improved layout and grouping of input echo in result file; redundant items removed.

- Status bar on screen showing progress of simulation.
- Command-line input possible for PC version, e.g. `chem h2-o2`.
- New command `UNBAL` for overruling rejection of unbalanced systems (not recommended).
- New command `MODE` for choosing between adiabatic and isothermal simulation.
- Problem case text may be continued on a second line (Section C of the User's Guide).
- Mathematical integration command `EPS` is no longer needed.
- Mathematical integration command `FSTSTP` is no longer needed.
- Mathematical integration command `HMAX` is no longer needed.
- `PRINTS` no longer obligatory (now only `TEND` obligatory).
- Negative results no longer suppressed in result file.
- Miscellaneous minor improvements.

10 Computer requirements

CHEMSIMUL is written in Standard FORTRAN 90 and should therefore be able to run on many different computer systems. If you are interested in acquiring CHEMSIMUL, you should contact the authors. A PC demo version MINICHEM is freely available on request (Section 10.2).

10.1 Computer platforms

The main platform for running CHEMSIMUL is the PC (any 386, 486, Pentium or newer). It is also running on an Alpha platform under Digital VMS, and under several UNIX systems, e.g. IBM and HP.

A graphical interface is made to the bannerware program GNUPLOT [10], which is in widespread use. However, the CHEMSIMUL plot tables should allow you to use any other graphics system you might prefer.

The standard version of CHEMSIMUL has, in principle, no limits on the number reactions, species, etc.. Nevertheless, the specific hardware you are using might put its own restriction on problem size by limiting the total amount of allocatable data storage.

10.2 MINICHEM

MINICHEM is a limited PC version of CHEMSIMUL for demonstration purpose. It is freely available on request from the authors and may be freely redistributed. MINICHEM has the same features as CHEMSIMUL, but problem size parameters are restricted in the following way:

Maximum number of reactions	10
Maximum number of species	20
Maximum number of isotopes	1

Otherwise MINICHEM runs precisely as CHEMSIMUL does, and the present manual therefore applies to MINICHEM, too.

References

[1] O. L. Rasmussen and E. Bjergbakke, "CHEMSIMUL — a program package for numerical simulation of chemical reaction systems," Tech. Rep. Risø-R-395(EN), Risø National Laboratory, DK-4000 Roskilde, Denmark, 1984.

[2] C. W. Gear, "Algorithm 407, DIFSUB for solution of ordinary differential equations," *Commun. ACM*, vol. 14, p. 185, 1971.

[3] A. C. Hindmarsh and G. D. Byrne, "EPISODE: An experimental package for the integration of systems of ordinary differential equations," Tech. Rep. UCID-30112, Livermore, 1975.

[4] L. R. Petzold, "Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations," Tech. Rep. SAND80-8230, Sandia, 1980.

[5] A. R. Curtis and W. P. Sweetenham, "FACSIMILE release H user's manual," Tech. Rep. AERE R 11771, Harwell, 1985.

[6] W. Braun, J. T. Herron, and D. Kahaner, *ACUCHEM/ACUPLOT - A computer program for modelling complex reaction systems.* Gaithersburg, Maryland: National Bureau of Standards, 1986.

[7] P. Deuflhard, G. Bader, and U. Nowak, "LARKIN — a software package for the numerical simulation of LARge systems arising in chemical reaction KINetics," in *Modelling of chemical reaction systems* (K. H. Ebert, P. Deuflhard, and W. Jäger, eds.), (Berlin, Heidelberg, NY), Springer-Verlag, 1981.

[8] R. J. Kee, J. A. Miller, and T. H. Jefferson, "CHEMKIN: A general-purpose, problem-independent, transportable, FORTRAN chemical kinetics code package," Tech. Rep. SAND80-8003, Sandia, 1980.

[9] O. L. Rasmussen, E. Bjergbakke, P. Pagsberg, and P. Kirkegaard, "CHEMSIMUL — software and methods for numerical simulation of chemical reaction systems," in *Industrial Mathematics Week*, (Trondheim), pp. 174–181, Norwegian Institute of Technology (NTH), 1992. Paper presented by P. Kirkegaard.

[10] T. Williams and C. Kelley, *GNUPLOT — An Interactive Plotting Program*, 1993. Manual for Version 3.5.

[11] H. Christensen and E. Bjergbakke, "Alfa-radiolysis of aqueous solutions," in *Mat. Res. Soc. Symp. Proc.*, vol. 50, p. 401, Materials Research Society, 1986.

[12] H. Christensen and E. Bjergbakke, "Application of CHEMSIMUL for groundwater radiolysis," *Nuclear and Chemical Waste Management*, vol. 6, p. 265, 1986.

[13] H. Christensen and E. Bjergbakke, "Radiation induced dissolution of UO2," in *Mat. Res. Soc. Symp. Proc.*, vol. 84, (Boston), Materials Research Society, 1987.

[14] E. J. Hart, K. Sehested, E. Bjergbakke, and J. Holcman, "Gamma-ray initiated chain decomposition of aqeuous ozone solutions," *Radiat. Phys. Chem.*, vol. 29-5, p. 399, 1987.

[15] E. Bjergbakke, Z. D. Draganic, K. Sehested, and I. G. Draganic, "Radiolytic products in waters. Part I: Computer simulation of some radiolytic processes in the laboratory," *Radiochimica Acta*, vol. 48, p. 65, 1989.

[16] E. Bjergbakke, Z. D. Draganic, K. Sehested, and I. G. Draganic, "Radiolytic products in waters. Part II: Computer simulation of some radiolytic processes in nature," *Radiochimica Acta*, vol. 48, p. 73, 1989.

[17] G. M. Ridler, P. F. Ridler, and J. G. Sheppard, "A systematic method of checking of systems of chemical equations for mass balance," *J. Phys. Chem.*, vol. 81, p. 2435, 1977.

[18] J. S. Rosenbaum, "Conservation properties of numerical integration methods for systems of ordinary differential equations," *J. Comp. Phys.*, vol. 20, p. 259, 1976.

[19] J. S. Rosenbaum, "Conservation properties of numerical integration methods for systems of ordinary differential equations. 2.," *J. Phys. Chem.*, vol. 81, p. 2362, 1977.

[20] A. C. Hindmarsh, "LSODE and LSODI, two new initial value ordinary differential equation solvers," *ACM-SIGNUM Newsletter*, vol. 15, no. 4, pp. 10–11, 1980.

Bibliographic Data Sheet **Risø–R–1085(EN)**

Title and author(s)

CHEMSIMUL: A simulator for chemical kinetics

Peter Kirkegaard and Erling Bjergbakke

ISBN	ISSN
87–550–2474–2	0106–2840

Dept. or group	Date
	December 1999

Groups own reg. number(s)	Project/contract No.

Pages	Tables	Illustrations	References
37	2	3	20

Abstract (Max. 2000 char.)

CHEMSIMUL is a computer program system for numerical simulation of chemical reaction systems. It can be used for modeling complex kinetics in many contexts, in particular radiolytic processes with pulse trains. It contains a translator module and a module for solving the resulting coupled nonlinear ordinary differential equations. An overview of the program system is given, and its use is illustrated by examples. A number of special features are described, in particular a method for verifying the mass balance. Moreover, the document contains a complete User's Guide for running CHEMSIMUL on a PC or another computer. Finally, the mathematical implementation is discussed.

Descriptors INIS/EDB

RADIOLYSIS; CHEMICAL REACTION KINETICS; COMPUTERIZED SIMULATION; COMPUTER PROGRAM DOCUMENTATION; PULSE TECHNIQUES; DIFFERENTIAL EQUATIONS

Available on request from:
Information Service Department, Risø National Laboratory
(Afdelingen for Informationsservice, Forskningscenter Risø)
P.O. Box 49, DK–4000 Roskilde, Denmark
Phone (+45) 46 77 46 77, ext. 4004/4005 · Fax (+45) 46 77 40 13