ACTA CHEMICA SCANDINAVICA 21 (1967) 791–800

A Compiler for Digital Computation in Chemical Kinetics and its Application to Oscillatory Reaction Schemes

P. LINDBLAD

Department of Information Processing, H. C. Ørsted Institute, University of Copenhagen

H. DEGN

Chemistry Laboratory III, H. C. Ørsted Institute, University of Copenhagen, Denmark

The compiler described may be used for calculating the concentrations as functions of time for any system of first and second order reactions. The computer input is the reaction scheme, the rate constants and the initial concentrations, expressed in the conventional chemical notation so that no special knowledge on computer programming is required. The calculation is performed by means of a Monte-Carlo method. The results of the calculation may be presented as curves showing the concentrations as functions of time. Some hypothetical reaction schemes containing autocatalytic steps are used as examples.

A common problem in chemical kinetics is to calculate concentrations as functions of time for a postulated set of reactions when the initial concentrations and reaction rate constants are specified. Traditionally the problem is approached by expressing the information contained in the reaction scheme in differential equations and then proceeding to solve these equations. Unfortunately only differential equations originating from relatively simple reaction schemes can easily be solved. In more complicated cases solutions can be obtained by means of analog or digital computers.

It would be desirable for the chemist to be able to use the computer without having any training in computer programming. To make this possible we must have a language in which the chemist can easily state his problem and a compiler which will translate the statements into a computer program. As to the language the chemists conventional symbol language is perfectly suitable. This language is extremely simple, since it only consists of an arrow, the plus sign and arbitrary letters serving as names of substances. If we restrict ourselves to first and second order reactions, only the following four expressions can occur in a reaction scheme

$$\begin{aligned}
&A \rightarrow B\\
&A + B \rightarrow C\\
&A \rightarrow B + C\\
&A + B \rightarrow C + D
\end{aligned}$$

Two or more identical letters may appear in one expression. The meaning of these expressions is that the substances on the left side are converted into the substances on the right side at a rate proportional to the concentrations of the substances on the left side. In the present paper we shall describe a compiler which allows a digital computer (GIER) solution of any problem which can be formulated in terms of the above four expressions.

The above symbols may be looked upon as a stenographic notation of what is observed to happen in a chemical reaction, but it does not presuppose any specific way of calculating the various concentrations as functions of time. Given the language, one can freely choose a mathematical model which is convenient for calculation, *i.e.* we do not necessarily have to use differential equations. When a computer is used in the calculation of numerical results the most direct approach is to simulate the reaction by means of a stochastic model. It is possible to invent many different models of chemical reactions based on stochastic principles. One such model has been described in detail by Schaad.[1] The model described in the present paper is quite different from that of Schaad, and it has the advantage of requiring much less memory capacity in the computer.

DESCRIPTION OF THE MODEL

As the compiler is not restricted to be used for calculations in chemical kinetics a somewhat abstract description of the mathematical model and the adopted language will be given.

We assume that the system we want to study can be in certain states, and that each of these states can be described by a set of integers, $m_1, m_2, \ldots, m_M$, called the parameters. To each parameter is associated a name. (In reaction kinetics the parameters will be the numbers of molecules of different substances.) Furthermore we assume that the system may undergo a sequence of transformations of the following types:

1) parameter m_i is reduced by one and m_k increased by one for arbitrary i and k,
2) parameters m_i and m_j are reduced by one and m_k increased by one for arbitrary i, j, and k,
3) parameter m_i is reduced by one and m_k and m_l increased by one for arbitrary i, k, and l,
4) parameters m_i and m_j are reduced by one and m_k and m_l increased by one for arbitrary i, j, k, and l.

The probability that the system undergoes a certain transformation is proportional to m_i if the transformation is of type 1 or 3, and proportional to $m_i m_j$ in case 2 or 4 if $i \neq j$ and to $m_i(m_i-1)$ if $i = j$.

The probability that no transformation occurs, *i. e.* identity transformation, is 1 minus the sum of all probabilities of proper transformations. By definition, time is proportional to the number of transformations, including identity transformations, which have occurred.

A description of the system is a list of transformation rules, the proportionality constants entering into the probabilities and the initial values of the parameters. In the description we use the following notation: The transformation by which the parameter a is reduced by one and the parameter b is increased by one is denoted

$$a \rightarrow b,k$$

where k is the proportionality constant, and similarly

$$\begin{aligned} &a + b \rightarrow c,k \\ &a \rightarrow b + c,k \\ &a + b \rightarrow c + d,k \end{aligned}$$

We signify that the parameter a has the initial value h by

$$a = h$$

If a name appears in the list of transformations but the initial value of it is not defined, it is taken to be 0.

When the compiler has been loaded into the computer followed by a description of a system the computer will do the following:

For each name appearing in the description the computer will choose a storage-location for holding the initial value of that name.

For each transformation rule the computer will choose locations for holding the addresses of the locations corresponding to the names appearing on the left and right side of the $\rightarrow$ sign, a location for the proportionality constant, and a location which during the calculations will hold the probability that this transformation will be executed.

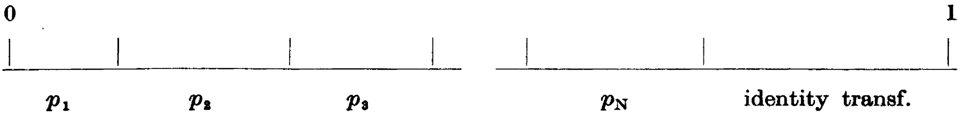
When this has been done the computer will proceed as follows:

1) calculate the probability, p_i, corresponding to each transformation rule, T_i, using the addresses and proportionality constant mentioned above, and store this in the location reserved for that purpose.

2) generate a random number, r, $0 < r \leq 1$, and determine from the set of transformation rules, T_i, that particular T_I which satisfies

$$\sum_{i=1}^{I-1} p_i < r \text{ and } \sum_{i=1}^{I} p_i > r$$

if such a T_I exists.



This procedure may be visualized as follows: We divide the real axis from 0 to 1 into intervals corresponding to each transformation rule in such a way that the i^{th} interval is of length p_i, and we then execute the I^{th} transformation if r belongs to the I^{th} interval.
3) if a transformation rule satisfying 2) is found, add one to the parameters appearing on the right hand side of → in that rule and subtract one from those appearing on the left side.
4) Add one to the number of transformations executed and repeat the process from 1).

If, for example, we take a first order reaction the number of reactant and product molecules may be used as parameters, denoting the reactant by A and the product by B. A description of this system ready to be loaded into the computer is

$$\mathrm{A} \rightarrow \mathrm{B},\ 0.0001$$
$$\mathrm{A} = 1000$$

The machine will calculate the initial probability of transformation as 0.1 and if the first random number is < 0.1, the state will be changed to $\mathrm{A} = 999$, $\mathrm{B} = 1$. After the first transformation the probability will be 0.0999 or 0.1 depending on whether the first random number < 0.1 or > 0.1 *etc.*

During the calculation on a particular reaction scheme the sum of transformation probabilities may increase or decrease several orders of magnitude. A low sum of probabilities is, however, uneconomical since it means that an excessive number of identity transformations are calculated. A mechanism which will automatically optimize the sum of transformation probabilities was, therefore, introduced in the compiler. If during the calculation the sum of probabilities exceeds one the computer will make a correction by dividing all proportionality constants by two and counting two transformations as one in the following calculations. On the other hand if the sum of probabilities falls below a certain limit (arbitrarily fixed to 0.35) the calculation speed is increased by the reverse correction.

COMPARISON WITH THE DIFFERENTIAL EQUATION MODEL

The model used is a discrete time Markov-process with stationary transition probabilities. The result of our computation is a sample function of this process. If many sample functions of the above first order reaction were calculated and the mean values of the parameters were found these would satisfy the equation $\mathrm{A}\ (\text{mean}) = \mathrm{A}_0\ (1-k)^n$, where A_0 is the initial value of A, k is the proportionality constant and n is the number of transformations. To compare this with the solution of the differential equation for the first order reaction we observe that if the relationship between time and the number of transformations in the Markov-process is given by $t = c{\cdot}n$, where c is a proportionality constant, and if the conventional rate constant of the reaction is k' then $k'{\cdot}c = k$. Substituting in the above expression we get $\mathrm{A} = \mathrm{A}_0\ (1-k'{\cdot}c)^{t/c}$. If we let c tend to zero, that is we represent an interval of time by an increasing number of transformations, we get

$$\lim_{c \to 0}(A) = \lim_{c \to 0}(A_0(1 - k' \cdot c)^{t/c}) = A_0e^{-k't} = A_0e^{-kn}$$

When a similar limit argument is used to the second order reaction the expression for the mean values is not identical with the solution of the differential equation model. Renyi [2] has shown that for a second order reaction the mean values will satisfy a differential equation which differs from the one usually used to describe a second order reaction by having an extra term. The relative importance of this extra term will, however, tend to zero as the number of molecules gets large. A comparison of the stochastic mean values and the classical deterministic expressions for reversible chemical reactions involving second order steps has been done by Darvey *et al.*[3]

RELIABILITY OF THE CALCULATION PROCEDURE

It is intuitively clear (and can be verified by experiments on the computer) that the relative random deviation of the sample function from the mean decreases as the initial values of the parameters increases. Since, when the present model is used, the load on the memory in the computer depends on the number of transformation rules and not on the values of the parameters the technical limit of the value of each parameter is very high. Consequently the error of the calculations can be reduced to any desired level by increasing the values of the parameters. However, if the parameters are multiplied by a factor q then the calculation time is also multiplied by q, but, according to our experience, the error is only divided by approximately the square root of q. Thus the cost in computing time of reducing the error is considerable.

We have not been able to obtain any general theoretical evaluation of the error as a function of the values of the parameters. Our practical experience is that for simple reaction schemes with a few first or second order reactions an error of about 1—2 % and a calculation time of 10—100 min is obtained when the initial values of the parameters are about 10 000. This conforms with the results reported by Schaad.

In addition to the simple reaction schemes the compiler has been successfully tested on more complex cases including polymerisation processes with up to 100 simultaneous reactions and the relaxation of vibrationally excited states in a gas. In the following we shall show examples of the use of the compiler on different oscillatory reaction schemes none of which have been analytically solved. In one case, the Volterra scheme, a numerical evaluation of the error is presented.

REACTION SCHEME WITH ONE AUTOCATALYTIC REACTION

The following reaction scheme, invented by Lotka,[4] was the first oscillatory scheme appearing in the chemical literature.

$$\begin{aligned} a &\rightarrow A \\ A &\rightarrow B \\ B &\rightarrow C \end{aligned} \qquad \text{(I)}$$

[a] is taken to be constant, and the second reaction is postulated to be autocatalytic. Although Lotka was unable to calculate the time evolution of the reaction scheme, he found that a stationary state where [A] = k_3/k_2 would eventually be reached. He further noted that there was a possibility for a damped oscillation while the stationary state was approached.

The above scheme cannot be used directly as input for calculation since it would be interpreted as three first order reactions. However, the additional assumptions, that the first reaction is of zeroth order and that the second reaction is autocatalytic, can easily be expressed in reaction equations. The zeroth order reaction is written

$$\mathrm{a} \rightarrow \mathrm{A} + \mathrm{a}$$

and the autocatalytic reaction is written

$$\mathrm{A} + \mathrm{B} \rightarrow 2\mathrm{B}$$

A slow zeroth order reaction supplying B must be added to the reaction scheme, since the autocatalytic reaction requires initiation. After these modifications scheme (I) appears as

$$\begin{array}{l} \mathrm{a} \rightarrow \mathrm{A} + \mathrm{a} \\ \mathrm{A} + \mathrm{B} \rightarrow 2\mathrm{B} \\ \mathrm{B} \rightarrow \mathrm{C} \\ \mathrm{a} \rightarrow \mathrm{B} + \mathrm{a} \end{array} \qquad \text{(I)}$$

In Fig. 1 is shown the result of a computation of the time evolution of scheme (I). It is seen to be in accord with the theory, *i.e.* [A] approaches the value k_3/k_2 through a damped oscillation.

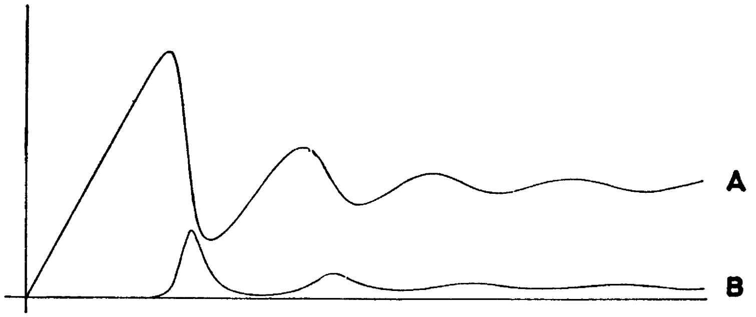


Fig. 1. Time evolution of a reaction scheme containing a zeroth order reaction supplying a substance, A, which is consumed by an autocatalytic reaction (branched chain reaction).

Computer input:	a → A + a	0.9
	A + B → B + B	$5_{10}-9$
	B → C	$2_{10}-4$
	a → B + a	$_{10}-4$
	a = 1	

REACTION SCHEMES WITH TWO AUTOCATALYTIC REACTIONS

A reaction scheme with two autocatalytic reactions, originally invented by Lotka,[5] was studied by Volterra,[6] who interpreted the substances as animal species devouring each other. The Volterra scheme is

$$\begin{aligned} &A_1 \rightarrow 2A_1 \\ &A_1 + A_2 \rightarrow A_2 \\ &A_1 + A_2 \rightarrow 2A_2 \\ &A_2 \rightarrow M \end{aligned} \qquad \text{(II)}$$

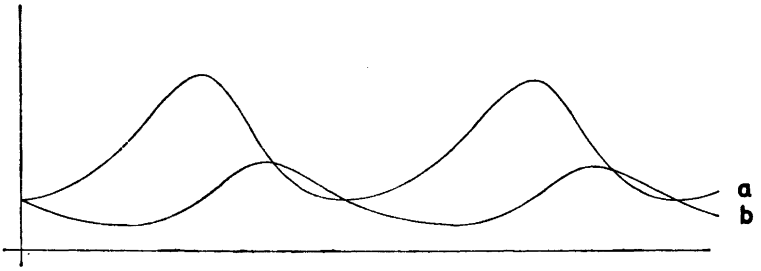


Fig. 2. Time evolution of reaction scheme containing a sequence of two autocatalytic reactions (Volterra scheme).

Computer input:	a → a + a	$_{10}-4$
	a + b → b + b	$_{10}-8$
	a + b → b	$_{10}-8$
	b → c	$_{10}-4$
	a = 5000	
	b = 5000	

Fig. 2 shows the result of a computation of the time evolution of the Volterra scheme (II). Volterra showed that the trajectory of $[A_1]$ and $[A_2]$ in the phase plane is a closed curve. The reliability of the computation may, therefore, be estimated by examining how closely the representative point returns to the initial point. For this purpose the time evolution of the Volterra scheme (II) was computed for several different rate constants and initial concentrations. The result of these calculations is shown in Table 1.

The Volterra scheme is essentially a sequence of two autocatalytic reactions. A somewhat different combination of two autocatalytic reactions, possibly

Table 1. Deviation of recurrence of the initial point in the Volterra process at different rate constants and initial concentrations. The error is defined as the distance between the initial point and the nearest point after one cycle divided by the distance from the origin in the phase plane (A_1 *versus* A_2.)

k_1	k_2	k_3	k_4	a	b	error
$4_{10}-5$	$4_{10}-9$	$4_{10}-9$	$4_{10}-5$	10 000	10 000	0.006
$2_{10}-5$	$_{10}-9$	$_{10}-9$	$4_{10}-5$	18 000	18 000	0.04
$4_{10}-5$	$_{10}-9$	$_{10}-9$	$2_{10}-5$	18 000	18 000	0.02
$4_{10}-5$	$_{10}-9$	$_{10}-9$	$2_{10}-5$	15 000	15 000	0.01

having a closer relation to reality, is obtained when the phenomenon of quadratic branching is included in the reaction scheme. Quadratic branching or positive chain interaction was postulated by Semjonow [7] for the explanation of certain phenomena in chain reaction kinetics. The idea is that free radicals constituting the chain center may interact to produce new radicals which propagate the chain reaction more effectively than the previous ones. The following reaction scheme contains a zeroth order reaction supplying a substance B which is consumed by a quadratically branched chain reaction.

$$\begin{aligned} A &\rightarrow B + A \\ B + X &\rightarrow 2X \\ 2X &\rightarrow 2Y \\ B + Y &\rightarrow 2X \\ X &\rightarrow C \\ Y &\rightarrow D \\ E &\rightarrow X + E \end{aligned} \qquad \text{(III)}$$

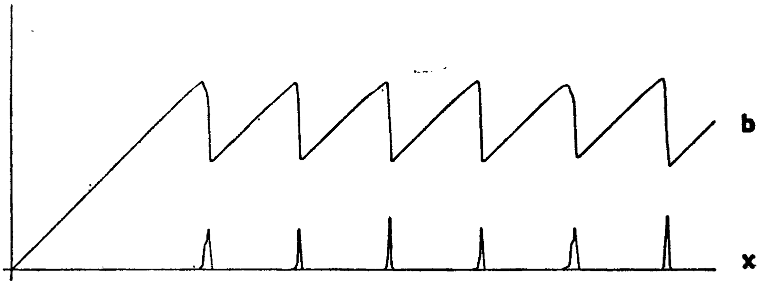


Fig. 3. Time evolution of a reaction scheme containing a zeroth order reaction supplying a substance, b, which is consumed by a quadratically branched chain reaction.

Computer input:

$a \rightarrow b + a$	$_{10}-1$
$b + x \rightarrow x + x$	$_{10}-9$
$x + x \rightarrow y + y$	$5_{10}-7$
$b + y \rightarrow x + x$	$1.5_{10}-9$
$x \rightarrow c$	$4_{10}-4$
$y \rightarrow d$	$4_{10}-4$
$a \rightarrow x + a$	$_{10}-3$
$a = 1$	

Fig. 3 shows the time evolution of this scheme. The curve is very similar to the curve obtained when the concentration of iodine is recorded as a function of time in the oscillating reaction of iodate and hydrogen peroxide described by Bray.[8] One of the authors (H.D.)[9] has collected some experimental evidence that this oscillation is caused by the intermittent action of a branched free radical chain reaction. Quadratic branching would be an explanation of the sustained oscillation in this reaction.

REACTION SCHEMES CONTAINING MORE THAN TWO AUTOCATALYTIC REACTIONS

Lotka's idea of a sequence of two autocatalytic reactions was extended to a series of several autocatalytic reactions by Moore,[10] who examined the following general scheme

$$\begin{array}{l} w \rightarrow W \\ W + A_1 \rightarrow 2A_1 \\ A_1 + A_2 \rightarrow 2A_2 \\ \cdots\cdots \\ \cdots\cdots \\ A_{n-1} + A_n \rightarrow 2A_n \\ A_n \rightarrow P \end{array} \qquad \text{(IV)}$$

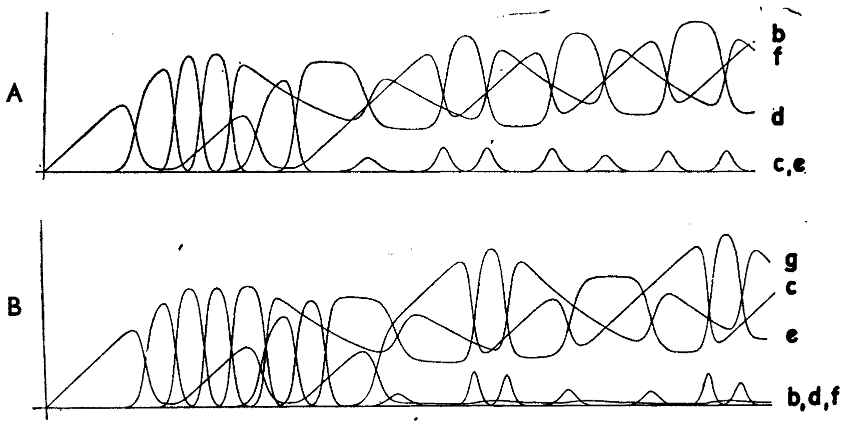


Fig. 4. Time evolution of a reaction scheme containing a zeroth order reaction supplying a substance, b, which is consumed by a sequence of (A) four and (B) five autocatalytic reactions.

Computer input: (A)	$a \rightarrow b + a$	0.9	(B)	$a' \rightarrow b + a$	0.9
	$b + c \rightarrow c + c$	$5_{10}-9$		$b + c \rightarrow c + c$	$5_{10}-9$
	$c + d \rightarrow d + d$	$5_{10}-9$		$c + d \rightarrow d + d$	$5_{10}-9$
	$d + e \rightarrow e + e$	$5_{10}-9$		$d + e \rightarrow e + e$	$5_{10}-9$
	$e + f \rightarrow f + f$	$5_{10}-9$		$e + f \rightarrow f + f$	$5_{10}-9$
	$f \rightarrow g$	$_{10}-5$		$f + g \rightarrow g + g$	$5_{10}-9$
	$a \rightarrow c + a$	$_{10}-4$		$g \rightarrow h$	$_{10}-5$
	$a \rightarrow d + a$	$_{10}-4$		$a \rightarrow c + a$	$_{10}-4$
	$a \rightarrow e + a$	$_{10}-4$		$a \rightarrow d + a$	$_{10}-4$
	$a \rightarrow f + a$	$_{10}-4$		$a \rightarrow e + a$	$_{10}-4$
	$a = 1$			$a \rightarrow f + a$	$_{10}-4$
				$a \rightarrow g + a$	$_{10}-4$
				$a = 1$	

Moore was able to calculate the time evolution of such systems by means of a differential analyser if the simplifying assumption [W] = constant was made. When the kinetic compiler is used such a restriction is not necessary. In Fig. 4 is shown the time evolution of two reaction schemes of the type (IV) containing a series of four and five autocatalytic reactions, respectively. A distinction between cases with an even and an odd number of autocatalytic substances was stressed by Moore. The difference between these two cases is clearly seen from the curves. In the case of an even number of autocatalytic substances the precursor (b) and the even numbered autocatalysts constitute a group of substances with high concentrations whereas the odd numbered autocatalysts have low concentrations when the stationary oscillation has been attained. This situation is reversed when the total number of autocatalysts is odd. The distribution of even and odd numbered autocatalysts followed the same rule in reaction schemes containing 1, 2, 3, and 6 autocatalytic substances.

CONCLUDING REMARKS

After the different trials it is concluded that the compiler is extremely flexible and easy to use. The only disadvantage is that the calculation time is rather high. The curves in Figs. 1—4 occupied the GIER computer for about 2, 2, 11, 18, and 18 h, respectively. It should be noted, however, that a similar compiler may be constructed for any digital computer, and there exist computers which would do the same calculations 10—50 times faster than the GIER. The calculation time would, in such case, always be within the limits required for practical work.

Acknowledgements. We wish to thank Dr. Bj. Svejgaard, Head of the Department of Information Processing, for helpful discussions and Professor Thor A. Bak for comments on the manuscript.

REFERENCES

1. Schaad, L. J. *J. Am. Chem. Soc.* **85** (1963) 3588.
2. Renyi, A. *Magyar Tud. Akad. Alkalm. Mat. Int. Kozl.* **2** (1954) 93.
3. Darvey, I. G., Ninham, B. W. and Staff, P. J. *J. Chem. Phys.* **45** (2145) 1966.
4. Lotka, A. *Z. physik. Chem. (Leipzig)* **72** (1910) 508.
5. Lotka, A.' *J. Am. Chem. Soc.* **42** (1920) 1595.
6. Volterra, V. *Théorie Mathématique de la Lutte pour la Vie,* Gauthier-Villars, Paris 1931.
7. Semjonow, N. N. *Einige Probleme der chemischen Kinetik und Reaktionsfähigkeit,* Akademie-Verlag, Berlin 1961.
8. Bray, W. C. *J. Am. Chem. Soc.* **43** (1921) 1262.
9. Degn, H. *Acta Chem. Scand.* **21** (1967). *In press.*
10. Moore, M. J. *Trans. Faraday Soc.* **45** (1949) 1098.

Received November 28, 1966.