CHAPTER VI

THE DESIGN AND EXECUTION OF SIMULATION EXPERIMENTS

1. The Promises and Pitfalls of Simulations.

The question that is being examined in this thesis concerns the reliability of the rate coefficients and other parameters pertaining to consecutive, first-order reactions, particularly when they number four, as estimated from experimental data containing random errors. Quasi-linear methods, such as that of residuals, which can give only approximate estimates, are kept out of purview, the analysis being confined to non-linear methods that are appropriate to the situation. These necessarily involve computer-based, iterative techniques, which lead to the "optimal", i.e., the best under the circumstances, estimates of the parameters.

The most direct way of checking the correctness of a parameter evaluated from experimental data is to compare it with its value known accurately either from theory or from an independent experiment. Thus, in a two-step, consecutive, first-order reaction under scheme 1, the rate coefficient evaluated for the second step can be checked by an independent kinetic run performed with the intermediate, B. Such independent corroboration of the findings, at least once in a while, is desirable and must be considered mandatory, in all research. But when the intermediate is non-isolable, this recourse is not available. Another method would be to compare the rate coefficients resolved from, say, titrimetric data with those from absorbance data. This could be adopted if the absorbance data are more reliable, i.e., accurate, than the volumetric. Unfortunately, this is often not
so and different estimation techniques suffer from similar disadvantages. There are, indeed, limitations to checking every scientific result.

One avenue of assessing the accuracy of experimental results, when their true values are unknown, is offered by the technique of simulations. Simulations are feigned or imaginary experiments designed to imitate real ones, these can be carried on paper if the numerical calculations involved are simple enough and on a computer if they are complicated. Developed over the last century, and increasingly adopted over the last few decades under the ever-expanding field of operations research for a variety of purposes including the testing of hypotheses, planning, designing and forecasting, they have become a powerful tool in the hands of workers in the physical as well as the social sciences and engineering. A few applications worth mentioning are the use of flight simulators in aviation to train pilots, simulated detonation of nuclear weapons to study their efficiency and impact, in war games to evolve and assess alternative military strategies, and in testing theories of evolution of stars and galaxies. These are fields where the actual conduct of experiments is prohibitively costly, destructive or just impossible. When simulations are carried out on computers, the results are available in a very short time at practically little cost with no damage to either personnel or materials. The savings effected in terms of inputs of time, material and energy are enormous. The most attractive feature about simulations is that they can be carried out for ranges of values of the parameters and variables that are out of reach in real experiments. They have been much more resorted to by chemical engineers and technologists than by the chemist, in fact, their application in the field of chemical kinetics, particularly of consecutive reactions, has been sparse.

The most important condition in simulation work is that the system equation must be properly and accurately defined, as otherwise the results will just be wrong, leading to wrong conclusions. The simulation “experiment” must also be properly designed and executed and the computer program which
performs the simulation exercise must have been written faultlessly. The experiment can be replicated any number of times, not usually feasible in actual work and the worker can have at hand a large number of outputs. The analysis and interpretation of the outputs are then up to the ability of the worker.

It all seems highly promising with simulations, offering hopes of tackling intricate problems at a click of the mouse. But Nature always has the last laugh. the so-called system equations that have been formulated by the human mind to describe nature are actually idealizations and approximations, there are so many imponderables in real situations that no human intelligence can take into account, and, whatever be the enormity of the computer, simulation results can often be at variance with reality. The best that the worker can do is to be careful in the design and execution of the simulations and in the interpretation of the results and then hope that these are as close to reality as possible. Simulation results must be interpreted in the statistical sense rather than the mathematical. This is particularly so when the number of replications is limited and not huge.

2. The Stages in the Design and Execution of Simulations in this Work.

Having decided to adopt simulations to seek an answer to the problem under investigation in this thesis, it was necessary to design the experiments in a way that they, while being economic, would provide answers to as many questions as possible. The execution of the experiments, it was desired, should be simple and at the same time uncompromising in the quality of the outputs.

The design and execution of simulation experiments adopted in this work are essentially the same as those employed by Sivaramakrishnan in 1991 [45] to establish the applicability of the point specific linear regression method developed by him for the resolution of the rate coefficients of two-step, consecutive, first-
order reactions from product and co-product concentration data. Since his method employed linear regression only, he had used an electronic calculator, rather than a computer, in his work.

The following stages become implicit:

1. Selection of the form of the system equation to be employed depending on the species and the mode of monitoring the progress of the consecutive reaction,

2. Specifying the ranges and selecting representative values of the parameters (rate coefficients and system property constants) to be covered in the simulations, also the range of the variable, i.e., reaction time, needed to cover various extents of the reaction;

3. Selection of representative sets of random, normal deviates for producing the replications;

4. Generation of accurate kinetic data from the system equation for the representative combinations of the various parameters over the range of the reaction, to be covered, followed by the incorporation of random, normal deviates of each set at different levels of variance, then rounding off to four significant digits, so as to obtain a set of replications that would resemble actual experimental data, and,

5. Subjecting the simulated data to a suitable iterative routine to optimize the various parameters, the outputs being checked for convergence, and also for whether the minimum in the chosen norm has been reached.

Stages 4 and 5 are performed together on the basis of a computer program. The optimized outputs of the parameters, called recoveries, are then to be
tabulated and discussed on the basis of the initially assumed values of the same in order to draw conclusions about the feasibility and efficiency of the optimization process for the type of reaction and the mode of monitoring it.


If the reaction is monitored in terms of the concentration of the intermediate B, the product C or the co-product D, they are governed by eqs (5) – (7), respectively.

Simulations were carried out also for the three-step, consecutive, first-order reaction, monitored in terms of the concentration of the product E, given by eq (8), in order to examine the feasibility of resolution of the three rate coefficients by optimization in three dimensions.

For two-step reactions, where the kinetics are monitored by recording changes in a system property, such as the absorbance, simulations in this work were restricted to the situation where the number of parameters is reduced to four, as in eqs (13) and (14), and to examine whether Acton’s contention (Ch 5) can be fully established. Here the optimization is performed in four dimensions.

In the cases of product and co-product formation, respectively, it was chosen to minimize the errors in the functions, Y = (1 – γ) and in Y = (2 – δ), rather than the errors in γ and in δ, because the right-hand side of the equations for the functions contain only two exponential terms with no off-sets; these functions can be considered more natural in describing their formation than γ and δ themselves. Thus, the logarithms of (1 – γ) and (2 – δ) are the quantities employed in determining the rate coefficients by quasi-linear methods, including that of residuals. At any rate, minimizing the error in the function (a – x), where a is a constant, is equivalent to minimizing the error in x itself. To be doubly
sure, separate programs were written for minimizing the errors in \((1 - \gamma)\) and in \(\gamma\), and when optimization was done, the outputs were almost identical - the sum of deviations squared differed at the most by one point in the seventh decimal place. The estimates of the rate coefficients agreed to the fourth significant digit, or beyond. Similarly, with the system property data, and with the data of the product data in three-step reactions, errors were minimized in \(Y = |Z - Z_r|\) and in \(Y = (1 - E/A_0)\). However, in the case of data of the intermediate in two-step reactions, the minimization was effected in \(\beta\) itself, as the right-hand side of the equation for \(\beta\) contains only two exponential terms with no off-set.


The parameters, viz., the individual rate coefficients and property constants of the various species in the reaction mixture can each theoretically lie in the range 0 to \(\infty\). The rate coefficients associated with very slow steps are close to zero, while those associated with very fast steps have high numerical values. Similarly, the molar absorbance of a species can be close to zero or be numerically very high, of the order of \(1E+5\). Some restrictions regarding the choice of these parameter values are necessary to keep the number of simulations within manageable limits.

Considering the two-step, consecutive, irreversible, first-order reaction, where only two rate coefficients are involved, if \(k_{\text{fast}}\) is over one hundred times \(k_{\text{slow}}\), the kinetics become very close to that of simple first-order and so there is no point in carrying out simulations for such reactions. For simulations with product formation data under this scheme, because of the symmetry of eqn (6), leading to identical kinetic curves for reactions in which the two rate coefficients are interchanged among the two steps, there is need to consider only cases of \(k\) in the range 1 to 100; simulations for cases in which \(< 1\) can be dispensed with as being redundant. Simulations are therefore carried out for fourteen representative
values of $\kappa$ within this range. But, with the co-product or intermediate concentration data, as there is no such symmetry in their equations, simulations have to be carried out for the range of $\kappa$ from 0.01 to 100.

Similar judicious specifications of the rate coefficients are made in the three-step, consecutive, irreversible, first-order reaction, restricting to the range $k(\text{fastest})/k(\text{slowest}) = 1$ to 100, with the other rate coefficient lying in between.

When absorbance readings are simulated, the molar absorbances are so selected as to make the readings scaled to lie in the range $0 - 1.0$ Abs. Since the molar absorbances of three species enter the equation and they can vary widely, or, equivalently, the pre-exponential parameters, $P$ and $Q$ in eq (13), can also vary widely, assuming negative values as well, some discrimination has to be employed in specifying them so as to keep the burden of simulation work tolerable and yet permit definitive conclusions to be drawn. This required a standardization or normalization of eq (13), the specifics of which are dealt with in the next chapter.

5. Random, Normal Deviates for Simulation of Kinetic Data.

Using an appropriate system equation, kinetic data can be generated, on the computer to seven significant digits, covering any extent of reaction for assumed, hypothetical values of the two rate coefficients. But these data will be "accurate" Real experiments are characterized by the presence of random errors in the data, assuming that systematic errors have been traced out and eliminated or adjusted for. These random errors are, in most experiments, belong to the normal or Gaussian distribution, and are assumed to be so in the context of the present work. Therefore, in order to make the generated data resemble those in an actual experiment, random, normal deviates (RND's) are added to the accurate data at selected levels of the standard error, $\sigma$. 

83
The generation of a random, normal number first requires the use of a set of uniform random numbers, \( r_i \), lying in the range \([0, 1]\) These, in turn, can be obtained by a truly random process, such as drawing lots from "slips in hat", as described by Wagner [83] (pp. 927-928). The Rand Corporation, using a specially-designed electronic experiment to perform such a truly random experiment, did generate a table of a million random digits in 1955(\(q \nu\)).

A random, normal deviate, \( V_j \), of mean 0 and variance 1, can be also generated from a sequence of such uniform random numbers by one of many algorithms, the most commonly employed one being

\[
V_j = \sum_{i=1}^{12} r_i - 6 \tag{54}
\]

The Rand Corporation volume also lists one hundred thousand random, normal deviates.

The use of tables is considered to be cumbersome by many workers, who prefer to employ computer-generated sequences of uniform random numbers instead, by using one of many available algorithms, which are described and discussed in [81] (Ch. 7). Though such computer-generated random numbers have most of the statistical properties of true random numbers, since they are generated by a deterministic algorithm, they may suffer from series correlations which are not evident at first sight and are, therefore, not truly random. Hence, they are known as pseudo-random numbers. One such routine, RANDU, which was very widely used and even installed on IBM mainframe computers for many years, was later found to suffer from a sequential correlation and so fell into disuse. Better algorithms are described in [81].

However, in order to avoid any possibility of series correlation, it was decided not to generate pseudo-random numbers on the computer, but instead
resort to the use of true random numbers, or, rather, the true random, normal
deviates generated therefrom. These were taken from a book of statistical tables
[108], which contains extracts from the larger volume by the RAND Corporation
referred to above.

Since twenty-four readings are generated to cover up to the 80% extent of
a reaction in this work, a sequence containing the same number of random,
normal deviates is required for incorporation into the readings. The first RND was
located by a purely random method by entering one of the pages of [108], also
randomly chosen, and further twenty-three RND’s were recorded in a sequence,
reading either upwards or downwards in the same column in the table, the manner
of reading also being randomly decided. These RND’s are to be incorporated in
the “accurate” readings generated from the system equation at the various selected
error levels, after multiplying by 0.001 for the error level of 0.1%, 0.002 for that
of 0.2%, and so on, the error-incorporated readings are then rounded off to four
decimal places to resemble real data.

If a real experiment is repeated a number of times under as far identical
conditions as possible, taking readings at the same reaction times in each
repetition, the readings at a given instant will differ slightly from one another, at
the same error-level, in the different repetitions. This is because the exact
magnitude of the random error entering a given reading is unpredictable both in
sign and magnitude and so will vary in each repetition – that is exactly why they
are defined as being random. The data of the different runs will lead to estimates
of the rate coefficients which differ, slightly or significantly, depending on the
exact sequence of random errors in each run. Under these circumstances, in order
that the inferences based on the results of simulations are of general validity, it is
essential that each simulation is repeated a number of times. This is achieved by
randomly selecting different sequences of RND’s and incorporating them into the
accurate readings, to obtain a number of replications of the simulated experiment.
For the present work, it was decided to employ six such sequences, randomly
chosen from [108]; these are listed as data at the end of the program for the simulations using product concentration data in the two-step reaction

While these RND's are from a population with zero mean and unit variance, their actual sample mean and sample standard deviation differ from the population values, because they are relatively small samples. Those sequences which have a sample standard deviation much greater from the population standard deviation of unity can be expected to produce estimates of the parameters which are less accurate than the ones with a lower sample standard deviation, because the actual magnitudes of the errors entering the readings will be larger in the former case. A more important decider of the accuracy of the estimates of the parameters is the trend in the exact sequence of the RND's. If there are many outliers, they tend to affect the accuracy of the estimates of the parameters, especially when they are crowded together at one end of the sequence; when the sequence contains a "run" of RND's, say six or more in a sequence, all of the same sign, the results are poorer. Such a sequence may be considered to be "rogue", and the resulting data represent those of an experiment where the circumstances have just been unfortuitous - something which can happen to any good experimenter on a bad day. If he had performed just this particular run alone, he would be blissfully ignorant of the nature of his results, had he performed just two runs, and if their results varied widely, that would leave him bewildered as to which one was the more accurate, hence, the insistence on an experimenter performing triplicate, not mere duplicate, runs of each of his experiments.

For the purpose of the present study, twenty-seven sequences of RND's were initially selected randomly from the handbook of statistics. Preliminary simulations involving product formation data showed that these sequences of RND's fell broadly into three categories. Some of these were "well-behaved", i.e., gave good recoveries of the parameters, some were moderately so and the rest "rogue" in that they gave relatively wild estimates of the parameters. In order to
keep the down size of the whole exercise, it was decided to restrict the number of
detailed simulations to six replications for every experiment, basing on two of
each of the above three sub-classes of sequences. The first of the six sets of
RND’s used in the present study was the same as used by Sivaramakrishnan in his
own work [45]. The other five sequences used in the present work were newly
chosen and had not been employed by him

6. The Nelder-Mead Downhill Simplex Minimization Routine of
Parameter Optimization.

The next stage is to subject the simulated data containing random, normal
errors to a non-linear optimization routine in order to recover the parameters. The
first of the two routines employed in this study is a non-gradient, direct search
method, that employs elementary geometric concepts in reaching the minimum in
the error (hyper)surface. The method and the algorithm have been discussed in
detail in [81] and [75]. This is, in fact, a much neglected method, and has been
very little used in chemical problems, as already noted, Carlson and Provdre [102]
have used this routine for the evaluation of the rate coefficients of a scheme in
which a first-order reaction is followed by one of second order

More recently, Bhaskara Raju [60], in these laboratories, has employed the
Nelder-Mead routine to evaluate the equilibrium constants of the formation of
charge-transfer complexes between chloranil and substituted anilines, and also
their molar absorbances, from absorbance versus concentration data collected in
the charge-transfer region of the spectrum.

The simplex method considered here has nothing to do with the simplex
method, due to G. Dantzig (1947), of solving linear programming problems,
though the latter too employs the geometrical concept of a simplex
The "Sequential Simplex" method of optimization of the parameters of a non-linear equation was first formulated by Spendley et al. in 1962 [109]. Nelder and Mead in 1965 described a more efficient version of the simplex method for minimizing a function of \( n \) variables, which is adopted for the present work.

The simplex is a polyhedron, i.e., a geometrical figure, with \((n+1)\) vertices in \( n \)-dimensions, with all their interconnecting line segments, polygonal faces, etc. Thus, in two dimensions, the simplex is a triangle, bounded by its three sides. In three dimensions, it is a tetrahedron, bounded by four triangular faces, and in four-dimensional space, it has five vertices, bounded by five four-dimensional hyperfaces. The last and higher simplexes cannot be visualized, but are amenable to mathematical treatment. While Spendley et al., who first formulated the method, considered only regular polyhedra, Nelder and Mead adopted it also to polyhedra which need not be regular. The simplex is non-degenerate, i.e., encloses a finite inner \( n \)-dimensional volume. If any point of a non-degenerate simplex is taken as the origin, then the \( n \) other points define vector directions that span the \( n \)-dimensional vector space.

Each dimension of the space containing the simplex corresponds to one of the parameters to be optimized. The values of the objective or trial function, i.e., the sum of the squares of the deviations, \( S \), if mapping is done, have to be represented along an extra, \((n+1)\)th, dimension, they lie on a surface in the case of optimization of just two parameters, or a hypersurface in the case of three or more parameters, the error surface/hypersurface can be mapped employing contours.

The Nelder-Mead routine starts with not just a single point, but with \( n+1 \) points, together defining an initial simplex. If one of these is \( P_0 \), then the other \( n \) points may be defined by

\[
P_i = P_0 + \lambda e_i
\]  \( \quad \ldots (55) \)
where the $e_i$'s are $n$ unit vectors and $\lambda$ is a constant which is a guess of the problem's characteristic length scale; or, it need not be kept constant and different $\lambda$'s can be chosen for each vector direction.

The value of the objective function is calculated at each vertex of the simplex. In the case where the LS norm is employed, as is done in this study, it is given by

$$
S = \sum_{i=1}^{N} [Y_i - f(A_1, A_2, \ldots, A_n, t)]^2
$$

(56)

where $Y_i$'s are the individual readings that total $N$ in the experiment, and $Y = f(A_1, A_2, \ldots, A_n, t)$ is the form of the system equation which expresses the system property $Y$ as a function of the $n$ system parameters, $A_1, A_2, \ldots, A_n$ (including the rate coefficients) and time, $t$. The term within the square bracket then corresponds to the residual given by each reading for the assumed values of the parameters.

The points are then arranged in the order of increasing values of the function. Then a series of step is taken, most of the steps just moving the "worst" point, where the value of the function is the highest, along the line joining it to the point of intersection of the medians of the opposite face of the simplex, in search of a "better" point at which the value of the function is lower. This may require a mere 'reflection', an 'expansion' following the reflection, or a 'contraction' in the transverse direction thereafter, depending on the circumstances that obtain. The new point is then substituted, according to the circumstances, for one of the points of the original simplex to constitute a new simplex and the process is repeated.

Each iteration is commenced with a so-called reflection step. (This term is a misnomer, since it is not true reflection, in the geometric sense, of the worst point,
through the foot of the perpendicular, about the opposite face, rather, it is an inversion of the worst point through the point of intersection of the median lines of the opposite face. However, the term reflection is employed in the present work to be in line with the accepted literature. A reflection step conserves the volume (area in the case of a triangle) of the simplex and thus maintains its non-degeneracy. If the optimization problem is only in two dimensions, the simplex is a triangle and the worst point is reflected through the mid point of the opposite side, if in three dimensions, the worst vertex is reflected through the point of intersection of the median lines of the opposite, triangular face of the tetrahedron, and, in four dimensional optimization, through the point of intersection of the median lines of the opposite hyperface. The co-ordinates of the new point can be found by a simple geometrical exercise in each case.

If a reflection step finds a better point (than the best of the original simplex), the method takes a larger step, the line being extended, usually by an equal distance (as in the present work) to reach a new point, which is examined for the value of the function thereat. If this proves to be a better point than the "worst", it is substituted for the latter to constitute the new, expanded simplex. This is termed a reflection and expansion step.

The process is repeated with the new simplex formed as a result of the reflection or reflection-expansion step. The vertices are arranged again in the order of decreasing function values and the worst point replaced by a better point to form yet another simplex, during these exercises, the simplex moves down the valley in the error surface, often changing direction until it reaches the valley floor. At this stage, a mere reflection or reflection-expansion will not suffice to find a better point, the routine now contracts the simplex in the transverse direction, seeking a better point which may lie, usually mid-way, between the worst vertex and the point of intersection of the median lines of the opposite face. This way, the contracted simplex tries to ooze down the valley to its deeper reaches.
In the end, it contrives to "pass through the eye of a needle" ([81] p. 403) by contracting itself in all directions towards the best point, finally pulling itself in around the minimum in the error surface.

It may happen sometimes that an anomalous step somewhere might have resulted in the simplex in getting stuck in a ledge and not being able to move further down towards the minimum, to circumvent such a contiguity, Press et al advise a restart to the whole program, reinitializing n of the n+1 vertices of the simplex again, the other being the claimed minimum.

Press et al. aptly describe the movements of the simplex towards the minimum as "delightful", being similar to those of a multi-dimensional amoeba, squeezing itself ever down tortuous contours of the error surface, now expanding, then shrinking and thus constantly propelling itself towards the target, viz, the deepest point.

To ascertain when the minimum has been reached and to terminate the program, a tolerance value is generally prescribed. As Press et al. point out, tolerance criteria can be delicate in any multi-dimensional minimization problem "without bracketing and with more than one independent variable (implying, parameter to be optimized), one no longer has the option of repairing a certain tolerance for a single independent variable." To overcome this difficulty, no tolerance is prescribed in the programs employed in this thesis, instead, the program is allowed to run until constancy in the optimized parameters, as well as in the objective function, to the extent of the machine precision (which is seven significant digits, working in single precision arithmetic) is observed. This can be objected to on the ground that many more iterations have to be carried out, but with the high-speed computers now readily available, this is no serious objection. Admittedly, such a recourse of prolonging the iterations than is absolutely necessary, is not very aesthetic or economical, but has the advantage of being
practicable in a complex situation. It is a moot question, with theoretical implications, whether constancy to seven significant digits in the parameters and the function minimum makes real sense while the input data have a precision of four significant figures at the most, but Press et al. (p. 404) have approved of this strategy of the tolerance being of the order of the machine precision, at the same time suggesting that it may be set slightly higher in order not to be diddled by round-off

The sequences of steps in each iteration, as followed in the present work, are illustrated with appropriate equations for the case of two-dimensional optimization, where the simplex is a triangle:

1. Define the starting triangular simplex by specifying three points, P\(_0\), P\(_1\) and P\(_2\).
2. Calculate the value of the objective function, f, at each of these vertices.
3. Arrange the vertices in the increasing order of the value of the function f and redesignate them as f\(_S\), f\(_M\) and f\(_G\), such that f\(_S\) < f\(_M\) < f\(_G\).
4. Determine the co-ordinates of the point R, found by 'reflecting' (actually by extending the worst vertex by equal distance past the mid point of the opposite side). These are given by

\[
X_R = X_S + X_M - X_G
\]

and

\[
Y_R = Y_S + Y_M - Y_G
\]

5. Calculate f\(_R\) at R.
6. If f\(_R\) < f\(_S\), carry out the 'expansion' step, moving past R to locate the point E given by

\[
X_E = 1.5 X_S + 1.5 X_M - 2 X_G
\]

and

\[
Y_E = 1.5 Y_S + 1.5 Y_M - 2 Y_G
\]
7. Calculate \( f_E \) at \( E \).

8. If \( f_E < f_R \), replace \( G \) in the original simplex by \( E \) to reconstitute it.

9. If \( f_S \leq f_R < f_M \), replace \( G \) by \( R \).

10. If \( f_G > f_R \leq f_M \), perform the 'contraction' step to find the point \( C \) given by

\[
X_E = 0.75 X_S + 0.75 X_M - 0.5 X_G
\]

and
\[
Y_L = 0.75 Y_S + 0.75 Y_M - 0.5 Y_G
\]

If \( f_R > f_G \), \( C \) is given

\[
X_E = 0.25 X_S + 0.25 X_M - 0.5 X_G
\]

and
\[
Y_F = 0.25 Y_S + 0.25 Y_M - 0.5 Y_G
\]

In all cases where \( f_R > f_G \), replace \( G \) by \( R \) first, irrespective of whether subsequent replacements are to be made.

Repeat the cycle of operations with the reconstituted simplex until no further improvement, i.e., decrease, is found in the value of \( f \).

At this stage the simplex is shrunk in all directions towards its best point

\[
X_{M(NEW)} = 0.5 X_S + 0.5 X_M,
\]

and
\[
Y_{M(NEW)} = 0.5 Y_S + 0.5 Y_M,
\]
\[
X_{G(NEW)} = 0.5 X_S + 0.5 X_G,
\]

and
\[
Y_{G(NEW)} = 0.5 Y_S + 0.5 Y_G
\]

93
The shrunk simplex is subjected again to the iterative routine for achieving further minimization, if feasible.

It is possible to accelerate the process by increasing the step size by a factor $F (>1)$; this is effected by multiplying the coefficients of $X$ and $Y$ in the right-hand sides of the above equations by $F$. This strategy has not been adopted in the present work.

The above equations, giving the co-ordinates of the points obtained by reflection, expansion, contraction, etc., pertain to optimization in two dimensions where the simplex is a triangle. For optimization in higher dimensions, such as in three dimensions involving a tetrahedron, similar equations can be written using analytical geometry. But with each higher dimension an additional vertex is incorporated in the simplex.

As indicated earlier, the Nelder-Mead routine always reaches a minimum, even if it is a local one. When the error level in the data is low, the error surface is free of local minima in the vicinity of the global minimum and in such situations the method finds the true minimum; when the error level increases, the error surface becomes contorted with the appearance of local minima, besides the global one, particularly in the case of optimization in higher dimensions; in such circumstances, the simplex may land in the local minimum nearest to its starting point, but the trajectory may get altered by a number of factors and it may eventually land in a minimum farther away. All the minima can, however, be accessed by the simplex from appropriate starting points ($q, v$).

The Nelder-Mead is admittedly not very efficient in terms of the function evaluations that it requires and is therefore slow in reaching the minimum. As Press et al. [81] point out, Powell's method [89] is almost surely faster in all applications, but the downhill simplex may frequently be the best method to use "to get something working quickly" for a problem whose computational burden is
small. This turns out to be particularly so when the number of parameters to be optimized becomes large. It is also recognized as being especially suited for noisy functions [71] (p 114) Since fairly high levels of noise (random errors in the data) are employed in the simulated data in this work, the Nelder-Mead becomes the apt choice.

One defect in the Nelder-Mead method is that, while estimating the various parameters, it does not give the standard errors of these estimates. While this is certainly not to be relished from the statistical point of view, insofar as a kineticist is content with obtaining estimates of the rate coefficients that fit his data best, this is not a matter of serious concern.

7. The Deming Method of Optimization of Parameters.

Being a gradient method, the Deming routine is much faster and reaches the minimum in far fewer steps than does the non-gradient, Nelder-Mead. But, as with most of the faster gradient methods, it does so only if the initial inputs of the parameters are close to their final estimates; it works well when there are just two parameters to be optimized, but with increasing number of parameters, this requirement becomes increasingly more stringent, as even modestly differing inputs may lead to convergence failure.

The method, described by Deming in his 1938 treatise, “Statistical Adjustment of Data” [92] is a statistical adaptation of the well-known Newton method (also called the Newton-Raphson method) of function minimization in numerical analysis.

In one dimension, Newton’s method is applied to determine the root or solution of a function of a variable x, written in the form f(x) = 0. Algebraically,
the method derives from the Taylor series expansion of a function in the neighbourhood of a point,

\[ f(x + \delta) \approx f(x) + f'(x) \delta + f''(x) \delta^2 + \ldots \]

For small values of \( \delta \), and for well-behaved functions, the terms beyond the linear become insignificant and can be omitted, so that \( f(x + \delta) = 0 \) implies

\[ \delta = -\frac{f(x)}{f'(x)} \]

Hence the Newton-Raphson formula by which the root can be found iteratively,

\[ x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \]

where \( x_i \) is the input at the \( i \)-th iteration and \( f'(x_i) \) is the derivative of \( f(x) \) with respect to \( x \) and represents the gradient of the function at the point \( x \). The method, therefore, requires the evaluation of both the function \( f(x) \) and its derivative \( f'(x) \) at arbitrary points \( x \). The formula consists geometrically of extending the tangent line at a current point \( x_i \) until it crosses zero, then setting the next guess \( x_{i+1} \) to the abscissa of that zero-crossing [81] (p. 355) Convergence is quadratic, i.e. rapid in the neighbourhood of a root, with the number of significant digits approximately doubling with each step. This very strong convergence property makes Newton-Raphson the fastest root-finding algorithm and the method of choice for any function whose derivative is continuous and non-zero in the neighbourhood of a root and can be evaluated efficiently.

When the initial guess of the root is far from the true root, however, the higher-order terms in the Taylor series are significant and cannot be omitted, the iteration then gives gross and meaningless corrections, leading to convergence failure. If an iteration places a trial guess near a local minimum or maximum, however small, the first derivative nearly vanishes and the next iteration sends the
solution into limbo with very little hope of recovery. So, like many powerful tools, Newton-Raphson can be destructive when used in inappropriate circumstances.

The Newton-Raphson, initially formulated for one-dimensional search, can be readily generalized to multiple dimensions, but convergence failure becomes more frequent.

Whereas the Newton–Raphson is designed to seek the zero, i.e., the solution of a function of one (or more) variables in numerical analysis, the Deming method seeks to find the minimum of a function of one (or more) parameters. For this purpose, the various parameters are treated as variables whose values can be adjusted till the minimum is reached. The minimum corresponds, in the case of a single parameter, to the point where the derivative or the gradient of the function with respect to the parameter becomes zero, and it is this zero, which is sought to be located by adopting the Newton-Raphson. When there are many parameters to be optimized, the problem translates into one of finding a zero of an N-dimensional gradient vector. For problems in more than two dimensions, one needs to find points mutually common to N unrelated zero-contour hypersurfaces, each of dimension N-1. This is unfortunately not the equivalent of the relatively simple task of finding a zero of a function of N variables. This is because, as Press et al. ([81], p. 376) point out, "the components of a gradient vector are not independent, arbitrary functions. Rather, they obey so-called integrability conditions that are highly restrictive. Put crudely, [one] can always find a minimum by sliding downhill on a single slope. The test of 'downhillness' is thus one-dimensional. There is no analogous conceptual procedure for finding a multidimensional root, where 'downhill' must mean simultaneously downhill in N separate function spaces, thus allowing a multitude of trade-offs, as to how much progress in one dimension is worth compared with progress in another".
The above limitation is rather drastic, as it reduces the workability of the Deming routine to regions of the vector space close to where the roots (i.e., the optimized parameters) lie. Further, as any other efficient routine (i.e., when it works), the Deming method has a tendency to come to rest on global and local minima indiscriminately.

The Deming routine starts with the deviations in the function $Y$, defined in Sec 3,

$$F_0 = Y_i - \hat{Y} \quad . \quad (57)$$

where $Y_i$ is the function corresponding to the $i$-th reading observed at time $t$, (represented as $X(i)$ in the programs) and $\hat{Y}$ is the value of the function $Y$ predicted by the inputs of the parameters. The trial function to be minimized, according to the least squares norm, is

$$S = \sum (F_0)^2 = \sum (Y_i - \hat{Y})^2 \quad . \quad (58)$$

which is the sum of the squares of the deviations for all the $N$ readings.

The computations become very simple if all the readings are associated with errors of the same variance, as are assumed to be throughout this work, then, weighting of the data can be discarded altogether. For optimization in two-dimensions, i.e., in the cases of the product, co-product and intermediate data only the two rate coefficients, whose inputs and estimates are represented by $J$ and $K$ in the programs, are to be optimized. The routine proceeds by treating the variables $t$ and $Y$, as constants and the parameters $J$ and $K$ as adjustable.

Defining the partial derivatives of $F_0$ with respect to $J$ and $K$ as

\[ \text{98} \]
\[
F_J = \frac{\partial F_0}{\partial J}
\] . (59)

and

\[
F_K = \frac{\partial F_0}{\partial K}
\] (60)

the values of which are computed for the inputs of \( J \) and \( K \), followed by the six summations

\[
\sum (FJ)^2, \sum FJ.FK, \sum (FK)^2, \sum FJ.F0, \sum FK.F0 \text{ and } \sum (F0)^2 \quad (= S)
\]

for the \( N \) readings. The corrections \( JJ \) and \( KK \) to be effected in the inputs of \( J \) and \( K \) are given by

\[
JJ = \frac{[\left( \sum FJ.F0 \right) \sum (FK)^2 - \left( \sum FK.F0 \right) \left( \sum FJ FK \right)] / Q}{Q} \quad (61)
\]

and

\[
KK = \frac{[\left( \sum FK.F0 \right) \sum (FJ)^2 - \left( \sum FJ.F0 \right) \left( \sum FK FK \right)] / Q}{Q} \quad (62)
\]

where

\[
Q = \sum (FJ)^2 \sum (FK)^2 - \left( \sum FJ FK \right)^2 \quad (63)
\]

The new values of \( J \) and \( K \), to be used in the next iteration, are

\[
J \text{ (new)} = J - JJ \times F \quad (64)
\]

and

\[
K \text{ (new)} = K - KK \times F \quad . (65)
\]

where \( F \) is a factor, usually taken as unity, but can be altered to modify the step size.

In most cases, \( J \) and \( K \) converge to their optimal estimates in 6 - 10 iterations. If divergence takes place, the program may be restarted reducing the step size, by putting \( F = 0.1 \) or 0.5. The program may converge from the initial
inputs with such a reduced step size, or may not; if it does not, a different pair of J and K may be tried as the inputs.

The standard errors in the estimates of J and K are given by

\[ DJ = \sqrt{\left[ T \sum (FK)^2 / Q (N - 2) \right]} \]  \hspace{1cm} (66)

and \[ DK = \sqrt{\left[ T \sum (FJ)^2 / Q (N - 2) \right]} \]  \hspace{1cm} (67)

where \[ T = \sum (F0)^2 - KK \sum JJ F0 - JJ \sum J \cdot K \cdot 0 \] and \[ \sum (F0)^2 \]  \hspace{1cm} (68)