Biochem. J. (1989) 258, 397–402 (Printed in Great Britain)

Analysis of progress curves for enzyme-catalysed reactions

Automatic construction of computer programs for fitting integrated rate equations

Ronald G. DUGGLEBY* and Chris WOOD
Department of Biochemistry, University of Queensland, St. Lucia, Queensland 4067, Australia

The computer analysis of progress curves for enzyme-catalysed reactions involves a series of mathematical and computational tasks. The three most daunting of these are the derivation of an integrated rate equation, solving this equation so that the amount of product formed by the reaction at any time can be calculated, and incorporating this solution into a non-linear-regression computer program. This paper describes the basis of a computer program that greatly simplifies the problem. The proposed mechanism is specified in the familiar kinetic constant form, which is automatically translated into a program capable of fitting this mechanism to a series of experimental progress curves. The approach is illustrated for a reversible reaction with one substrate and one product, and tested with some data obtained for the fumarase reaction. A copy of the program has been deposited as Supplementary Publication SUP 50148 (13 pages) at the British Library Document Supply Centre, Boston Spa, Wetherby, West Yorkshire LS23 7BQ, U.K., from whom copies can be obtained on the terms indicated in Biochem. J. (1989) **257**, 5.

INTRODUCTION

The application of progress-curve analysis to an enzyme-catalysed reaction requires that a number of tasks be completed successfully. The most obvious of these tasks, although hopefully not the first that would be attempted, is the experimental one; the experiments must be performed and the data collected.

Next comes the mechanism task; given a particular kinetic mechanism, the differential form of the steady-state rate equation must be obtained. Fortunately, rate equations have been catalogued for a variety of cases [1–4] and it is not difficult to find the appropriate equation in most instances.

A much more difficult task is the integration of this differential rate equation. Until fairly recently, this was one of the major impediments to progress-curve analysis. Integrated equations for selected cases have been published [5–7], but no general equations capable of being tailored to all specific cases have been described. Duggleby & Morrison [8,9] published a general equation for most irreversible reactions involving one substrate and up to two products, which can also be applied to reactions with two substrates in some circumstances. Boeker [10,11] has gone considerably further and has described general equations that are applicable to most mechanisms with two or fewer substrates and products, and both reversible and irreversible reactions.

Even in the very simplest case of an irreversible reaction with only one substrate and that is not inhibited by its products, the integrated rate equation cannot be solved algebraically. That is to say, the integrated rate equation expresses time (t) as a function of the maximum velocity (V_m), the Michaelis constant (K_a), the initial concentration of substrate (A_0), and amount of product (z) formed by the reaction (eqn. 1):

$$t = z/V_m - (K_a/V_m)\cdot\ln(1 - z/A_0) \qquad (1)$$

Given that time is known with little uncertainty whereas z would normally have an associated experimental error, the proper way to proceed is to fit the integrated equation to the data in such a way that some function of the difference between the experimental and predicted value of z is minimized. This in turn requires that the predicted value of z at any time is calculated. The application of the Newton–Raphson method to this solution task has been discussed elsewhere [12].

The fifth, and final, task is that of regression: fitting the integrated rate equation to the experimental data. Since this is impractical except by using a computer program, the solution to the integrated rate equation must be coded appropriately and incorporated into a suitable non-linear-regression computer program.

Except for the experimental task, all the remaining operations must be repeated for each mechanism to be tested, an intimidating prospect that has undoubtedly contributed to the fact that progress-curve analysis has not found wide acceptance.

In the present paper we describe the basis of a computer program that removes virtually all of the work involved in translating a presumed kinetic mechanism into a functioning non-linear-regression-analysis program. The program (AGIRE) accepts a differential rate equation in kinetic constant form and produces the code necessary to fit the integrated form of that equation: it is a computer program that writes a computer program. The procedure is illustrated for a reversible reaction with one substrate and product, and tested with progress-curve data for the reaction catalysed by fumarase. A copy of the program has been deposited as Supplementary Publication SUP 50148 at the British Library Document Supply Centre, Boston Spa, Wetherby, West Yorkshire LS23 7BQ, U.K., or may be obtained from the authors.

THEORY

The problem of taking a differential rate equation for an enzyme-catalysed reaction and converting this into a computer program for fitting the integrated form of this

* To whom correspondence should be addressed.

equation to a set of experimental progress curves can conveniently be considered in three stages. These are the integration, solution and regression tasks described in the Introduction.

Integration

According to Boeker [10,11], unbranched kinetic mechanisms for enzyme-catalysed reactions involving two substrates and products obey a general equation that is no more complex than eqn. (2):

$$\frac{dz}{dt} = \frac{V_f J_{AB}(AB - PQ/K_{eq.})}{J_0 + J_A A + J_B B + J_P P + J_Q Q + J_{AB} AB + J_{AP} AP + J_{AQ} AQ + J_{BP} BP + J_{BQ} BQ + J_{PQ} PQ + J_{ABP} ABP + J_{ABQ} ABQ + J_{APQ} APQ + J_{BPQ} BPQ + J_{ABPQ} ABPQ} \quad (2)$$

In this equation V_f is the maximum velocity in an arbitrarily chosen 'forward' direction and A, B, P and Q represent the concentrations of the substrates and products. $K_{eq.}$ is the overall equilibrium constant and the remaining symbols (J_0, J_A, J_B and so on) represent combinations of kinetic constants. As noted by Boeker [10], there is a redundancy in eqn. (2) and not all the J terms are independent. For example, division by J_{AB} will eliminate one parameter and reduce eqn. (2) to a form consistent with Cleland's [1] notation. We have retained the form used by Boeker [10,11] as the equations presented below are contingent on this form.

Eqn. (2) describes a great variety of reactions, and the difference between various mechanisms is the absence of particular terms. Similarly, one or more of these terms will be missing for simpler reactions having fewer reactants, or those that are irreversible. Except for the unusual situation where the concentrations of A and B are exactly equal, eqn. (2) and its simpler variants integrate to the general form given as eqn. (3):

$$C \cdot t = C_1 \cdot C \cdot z + C_2 \cdot C \cdot z^2/2 + C_3 \cdot C \cdot z^3/3 - C_f \cdot C \cdot \ln(1 - z/z_{eq.}) + C_s \cdot C \cdot \ln[1 - z/(z_{eq.} + D)] \quad (3)$$

Boeker [10,11] then gives expressions for the coefficients of eqn. (3) in terms of the J symbols of eqn. (2) and the initial concentrations of the reactants. It should be noted that ref. [10] contains some errors, and corrections are given in the Appendix of the present paper. Definitions of $z_{eq.}$ and D have also been given by Boeker [13].

To go from a particular kinetic mechanism to an integrated rate equation requires two rounds of substitution. First, the J terms of eqn. (2) are replaced with the groups of kinetic constants that define the mechanism. In the second round of substitution, the definitions of the variables of eqn. (3) are converted into kinetic constants.

For example, a simple reversible reaction with one substrate and one product in which there is no isomerization of the free enzyme is defined by the relationships given as eqns. (4)–(15), where all terms not explicitly defined are taken to be zero:

$$K_{eq.} = V_f K_p / V_r K_a \quad (4)$$

$$J_0 = V_r K_a \quad (5)$$

$$J_A = V_r \quad (6)$$

$$J_P = V_r K_a / K_p \quad (7)$$

$$J_{AP} = 0 \quad (8)$$

$$C = V_f J_A (1 + 1/K_{eq.}) \quad (9)$$

$$C_f \cdot C = J_0 + J_A A_{eq.} + J_P P_{eq.} + J_{AP} A_{eq.} P_{eq.} \quad (10)$$

$$C_1 \cdot C = J_A - J_P + J_{AP}(P_0 - A_{eq.}) \quad (11)$$

$$C_2 \cdot C = J_{AP} \quad (12)$$

$$z_{eq.} = (K_{eq.} A_0 - P_0)/(1 + K_{eq.}) \quad (13)$$

$$A_{eq.} = A_0 - z_{eq.} \quad (14)$$

$$P_{eq.} = P_0 + z_{eq.} \quad (15)$$

These substitutions are entirely mechanical in nature, requiring a great deal of care but almost no thought. This type of symbol manipulation, once correctly programmed into a computer, can be done quickly and without error. This task is the first of two functions of the AGIRE program.

Solution

The solution of eqn. (3) involves finding a value of z that satisfies the equality, for given values of all the remaining variables. Most authors [14–18] have employed the Newton–Raphson method, which involves the use of eqn. (16) and its first derivative with respect to z, eqn. (17):

$$F(z) = -C \cdot t + C_1 \cdot C \cdot z + C_2 \cdot C \cdot z^2/2 + C_3 \cdot C \cdot z^3/3 - C_f \cdot C \cdot \ln(1 - z/z_{eq.}) + C_s \cdot C \cdot \ln[1 - z/(z_{eq.} + D)] \quad (16)$$

$$F'(z) = C_1 \cdot C + C_2 \cdot C \cdot z + C_3 \cdot C \cdot z^2 + C_f \cdot C/(z_{eq.} - z) - C_s \cdot C/(z_{eq.} + D - z) \quad (17)$$

By using an initial estimate of z, $F(z)$ and $F'(z)$ are calculated and a refined estimate of z is calculated by subtracting $F(z)/F'(z)$ from the initial estimate. Provided the first estimate of z falls within certain limits, this refining process yields successive values of z that approach the solution of eqn. (3).

Duggleby [12] has discussed the numerical difficulties that can arise in this process, and these are here summarized briefly. If the initial estimate is below the solution, the first refinement is above the solution and may exceed $z_{eq.}$. In these circumstances, the first of the logarithmic terms in eqn. (16) becomes undefined and the Newton–Raphson method fails. Conversely, if the initial estimate is above the solution, the first refinement is also above, but closer to, the solution and the Newton–Raphson method must succeed. However, if the initial estimate is too close to $z_{eq.}$, $F'(z)$ is so large that many refinement cycles are required to reach the solution. Ideally, then, the initial estimate should be just above the solution; but how is such a value to be chosen when the solution is unknown?

A robust version of the Newton–Raphson method has been described [12] that largely avoids these numerical problems, but this technique is not sufficiently general that it can be applied to the cases considered by Boeker [10,11], in which both logarithmic terms of eqn. (3) occur. More recently, Boeker [14] has described a hybrid method that appears useful. Taking a cue from this work, the following procedure was adopted. First $F(z)$ is calculated at $z = 0$ and at 99.999% of $z_{eq.}$. The solution must lie between 0 and $z_{eq.}$, so if it is found that these two values of $F(z)$ have the same sign then z exceeds 99.999% of $z_{eq.}$ and this value is taken as the solution. Otherwise,

successive z values at 50%, 75%, 87.5%, 93.75% etc. of $z_{eq.}$ are tried until a value of $F(z)$ is found that differs in sign from that at $z = 0$. The estimate of z found in this way must be above the solution and is used to start the Newton–Raphson method.

The second function of the AGIRE program is to generate a series of BASIC statements that accomplish both the location of a suitable initial estimate and its refinement by the Newton–Raphson method. These statements are formulated so as to be consistent with the requirements of the DNRP53 program [19], which is mentioned in the following subsection.

Regression

Cox & Boeker [20] have used non-linear regression to fit a simplified form of eqn. (3) (containing only C_1, C_2 and C_f) to each of a series of progress curves obtained for the reaction catalysed by arginine decarboxylase. A secondary analysis using the known definitions of C_1, C_2 and C_f then allowed values for the various J terms to be determined. Finally, the usual kinetic constants were calculated from the J values.

This three-stage approach can be condensed into a single regression problem by performing an overall fit to the entire collection of progress curves, with the usual kinetic constants as the parameters to be estimated [8,17]. This overall fitting method is the one adopted here.

The DNRP53 computer program [19] is a general non-linear-regression program that is written in BASIC and is suitable for most microcomputers. Although any non-linear regression program can be adapted to progress-curve analysis, the advantage of the DNRP53 program from the present authors' standpoint is familiarity. It requires that the equation to be fitted is inserted at a predefined place and with appropriate syntax. For example, the fitted parameters are referred to as B(1), B(2) and so on. The AGIRE program takes care of all this by generating a series of BASIC statements of exactly the syntax required, and which can be incorporated directly into DNRP53.

EXPERIMENTAL

Fumarase catalyses the reversible hydration of fumarate to malate. Progress curves for this reaction were collected by monitoring the decline in absorbance at 240 nm when the pig heart enzyme (0.09–0.22 unit) was added to 3 ml of solutions containing between 0.05 mM- and 0.75 mM-fumarate. Reactions were carried out at 30 °C and pH 7.5 in a mixed buffer containing 50 mM-Tris acetate and 25 mM-sodium phosphate, conditions under which the enzyme was shown to be stable by the use of Selwyn's [21] test.

Absorbance data were digitized by using a device constructed by Mr. H. R. Johnson, connected to the analogue output of a Perkin–Elmer Lambda 3 spectrophotometer. The digitized data were collected with the use of a Custom Computer Services personal computer running a data-logging program written in this laboratory.

The absorbance at zero time was determined by general progress-curve extrapolation [22], and the concentration of formed malate at each time point was calculated from the change in absorbance, by using a molar absorption coefficient of 2384 $M^{-1} \cdot cm^{-1}$. From each of ten progress curves, 18 or 19 points were selected so as to be approximately evenly spaced in concentration. Some representative data are illustrated in Fig. 1; these 73 experimental points and 112 points from the remaining six progress curves were normalized to a common enzyme concentration by adjusting the time axis and then combined for the analysis.

RESULTS AND DISCUSSION

Except at high concentrations of fumarate, where substrate inhibition is seen (above 3 mM under the assay conditions described above; results not shown), the kinetics of the fumarase reaction may be described by four parameters, namely a maximum velocity and a Michaelis constant for each of the forward and reverse directions. The integrated rate equation is defined by eqns. (3)–(15).

In order to fit this model to experimental data it is necessary to create a suitable computer program. This is a relatively simple task, as can be seen from Fig. 2, which reproduces part of the dialogue with the AGIRE program. After establishing the type of reaction (one substrate, one product, reversible), the user then needs only to define the equilibrium constant and the various J terms as combinations of the parameters to be estimated. The particular parameterization illustrated here corresponds to that shown in eqns. (4)–(8).

The BASIC code generated by this run is shown in Fig. 3. After setting the some constants in lines 5100–5105, Boeker's [10] C, $C_f \cdot C$, $C_1 \cdot C$ and $C_2 \cdot C$ are defined in lines 5106–5109 by the variables BC, BCF, BC1 and BC2. A suitable initial estimate of z is located by lines 5170–5178, and the Newton–Raphson iteration is performed by lines 5180–5182. Values for $F'(z)$ and $F(z)$ are calculated in line 5180 and lines 5185–5187 respectively. Because the AGIRE program is rather general, some unnecessary code may be generated. For example, for this particular model there is no J_{AP} term (Fig. 2), and $C_2 \cdot C$ is zero. Hence the definition of BC2 in line 5109 and the inclusion of BC2 in lines 5180 and 5185 is superfluous and can be removed manually if this is thought to be desirable.

This program segment, when merged with the DNRP53 program, gives a non-linear regression program that will fit the model to the combined data from several progress curves, and this was used to analyse the data obtained for the fumarase reaction. Each experimental point was given a weight inversely proportional to the square root of the initial fumarate concentration, although the results were very similar when all points were equally weighted.

The fit is shown in Table 1. Although approximate values for the kinetic parameters were known from initial-velocity measurements ($V_f = 500$ units/mg, $V_r = 300$ units/mg, $K_a = 0.3$ mM and $K_p = 1$ mM), starting estimates for the fit were deliberately chosen so as to be somewhat different from these expected values, and the initial sum of squares was rather high (see the column labelled *SSQ* in Table 1). Despite this, the program smoothly approached the final values within a few non-linear iterations. The fitted curves are illustrated in Fig. 1.

The final values of the maximum velocities correspond to turnover numbers of $V_f = 9.68 \times 10^4 \pm 0.18 \times 10^4$ min^{-1} and $V_r = 6.99 \times 10^4 \pm 1.48 \times 10^4$ min^{-1}; bearing in mind

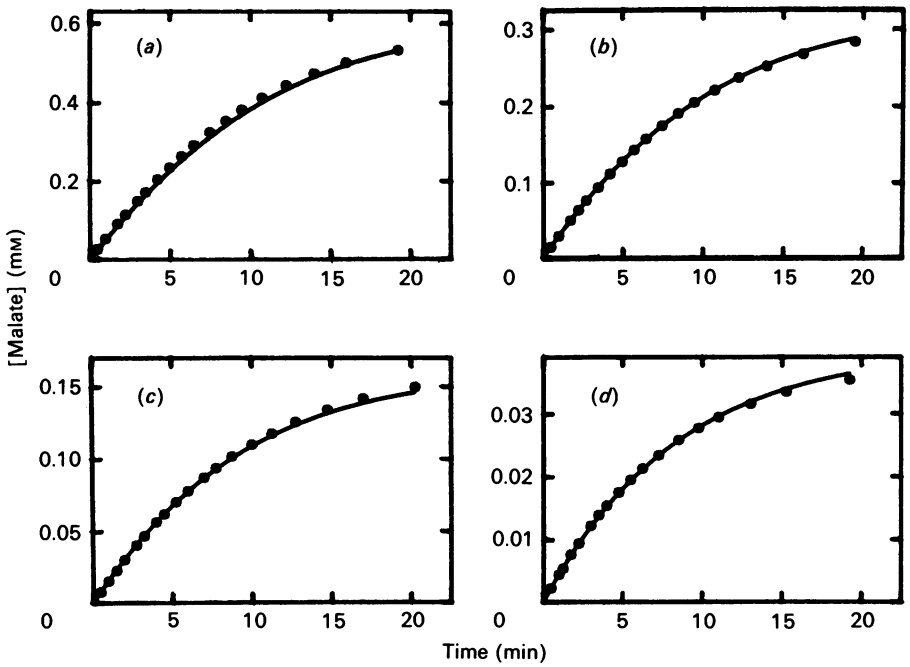

Fig. 1. Progress curves for the fumarase reaction

Data were collected as described in the Experimental section with the following fumarate and fumarase concentrations: (*a*) 0.744 mM-fumarate, 73 munits of fumarase/ml; (*b*) 0.398 mM-fumarate, 52 munits of fumarase/ml; (*c*) 0.199 mM-fumarate, 41 munits of fumarase/ml; (*d*) 0.050 mM-fumarate, 32 munits of fumarase/ml. In each panel the points represent the experimental data and the lines show the overall fit to these four, and six other, progress curves.

```
                    Program AGIRE

          Automatic generation of integrated rate equations

          Please specify the type of reaction catalyzed.

→         Enter the number of substrates (1 or 2): 1
→         Enter the number of  products  (1 or 2): 1
→         Reversible/Irreversible reaction  (R/I): R

          The differential form of the rate equation is taken to
          be

                      VfJa(A-P/Keq)
          v = ------------------------------
              Jo + JaA + JpP + JapAP

          You must supply the combinations of kinetic constants
          which go to make up these J terms and the Haldane
          relationship.

          Term          Combination
          ------------  ------------
→         ( 1) Jo       :VrKa
→         ( 2) Ja       :Vr
→         ( 3) Jp       :VrKa/Kp
→         ( 4) Jap      :
→         ( 5) Keq      :VfKp/VrKa
```

Fig. 2. Dialogue with the AGIRE program

This example shows how a reversible reaction with one substrate and one product, with no isomerization of the free enzyme, would be specified. Responses from the user follow the colon (:) on the lines indicated with an arrow (→).

the sensitivity of fumarase to variations in pH and phosphate concentration [23], these values, as well as the two Michaelis constants, are similar to values reported by other workers [16,23,24]. The equilibrium constant calculated from the Haldane relationship (eqn. 4) is 4.11, which agrees well with a directly measured value from three determinations of 3.87 ± 0.12.

The results shown in Table 1 were achieved by using the program described above run with a BASIC interpreter, and it must be admitted that the running time was rather long, taking a little over 2 h on an IBM-compatible personal computer. Although this may seem excessive, we do not regard this time as disproportionate to the time taken to perform the experiment, which was

```
5100 KEQ=B(1)*B(4)/(B(2)*B(3))
5101 A0=X(2)
5102 P0=X(3)
5103 ZEQ=(KEQ*A0-P0)/(1+KEQ):ZZ=ZEQ
5104 AEQ=A0-ZEQ
5105 PEQ=P0+ZEQ
5106 BC=(B(1))*(B(2))*(1+1/KEQ)
5107 BCF=0+B(2)*B(3)+AEQ*B(2)+PEQ*B(2)*B(3)/(B(4))
5108 BC1=0+B(2)-B(2)*B(3)/(B(4))
5109 BC2=0
5170 Z0=0:Z1=0.99999*ZZ
5171 Z=Z0:GOSUB 5185:G0=G:IF G=0 THEN 5190
5172 Z=Z1:GOSUB 5185:G1=G:IF G*G0>=0 THEN 5190
5175 Z=(Z0+Z1)/2:IF (Z1-Z0)/ZZ<1E-5 THEN 5190
5176 GOSUB 5185:IF G=0 THEN 5190
5177 IF G*G1>0 THEN 5180
5178 G0=G:Z0=Z:GOTO 5175
5180 G1=BC1+Z*BC2+BCF/(ZEQ-Z)
5181 Z1=G/G1:IF ABS(Z1/ZZ)<1E-5 THEN 5190
5182 Z=Z-Z1:GOSUB 5185:GOTO 5180
5185 G=Z*(Z*BC2/2+BC1)
5186 G=G-BCF*LOG(1-Z/ZEQ)
5187 G=G-BC*X(1):RETURN
5190 G=Z:RETURN
```

Fig. 3. Output from the AGIRE program

These BASIC statements are the results from the dialogue shown in Fig. 2. A functional non-linear-regression program for progress-curve analysis of a reversible reaction with one substrate and one product is produced when these statements are combined with the DNRP53 program [19]. The variables B(1), B(2), B(3) and B(4) are the fitted parameters and correspond to V_f, V_r, K_a and K_p respectively. Time, A_0 and P_0 are represented by the symbols X(1), X(2) and X(3), and G symbolizes the fitted variable, z. All other variable names are not used elsewhere in the DNRP53 program and their meaning should be evident from the context.

Table 1. Analysis of fumarase data

Iteration	V_f (units/mg)	V_r (units/mg)	K_a (μM)	K_p (μM)	SSQ (μM)
Initial	300.0	200.0	200.0	200.0	1313.18
1	398.2	159.3	163.5	293.9	36.14
2	458.4	222.9	229.8	468.8	10.54
3	495.8	300.3	281.4	694.5	5.34
4	499.1	347.9	287.2	822.6	4.62
5	498.8	359.6	287.0	851.0	4.60
Final*	498.8	360.3	287.0	852.5	4.60
S.E.	9.1	76.2	8.5	140.3	

* Converged after eight iterations.

Table 2. Run times for the analysis of fumarase data

Environment	Time (min)	Acceleration
Interpreter		
Microsoft GWBASIC 3.10	121.1	1.0
Compilers		
No math co-processor		
Microsoft QuickBASIC 2.0	20.8	6.0
IBM BASCOM 2.0	14.8	8.2
With math co-processor		
Borland Turbo Basic 1.0	4.1	29.8
Microsoft QuickBASIC 4.0	2.9	42.0

about 8 h. Moreover, the analysis time can be shortened substantially by compiling the program, and some representative results obtained with various compilers is shown in Table 2. With either the Microsoft QuickBASIC (version 2.0) compiler or the IBM BASCOM (version 2.0) compiler, neither of which can use a math co-processor, the run time was shortened by approx. 7-fold. When compilers that can use a math co-processor were used, the time required became quite short: 4.1 min for Borland Turbo Basic (version 1.0) and 2.9 min for QuickBASIC (version 4.0). Thus the analysis can be completed in a few minutes, allowing ample opportunity for evaluating alternative kinetic models.

CONCLUSIONS

The AGIRE program described in this paper represents a significant advance in the analysis of progress curves for enzyme-catalysed reactions. The burdensome tasks of deriving an integrated rate equation and creating the computer code necessary to solve and fit this equation have been completely overcome. Instead, the user can work directly with the form that is familiar to enzyme kineticists, the differential rate equation.

The program has been tested and found to produce the correct computer code for both reversible and irreversible reactions involving either one or two substrates and one or two products.

This work was supported by the Australian Research Grants Scheme (Grant no. A08715141).

REFERENCES

1. Cleland, W. W. (1963) Biochim. Biophys. Acta **67**, 104–137
2. Plowman, K. M. (1972) Enzyme Kinetics, pp. 138–153, McGraw-Hill, New York
3. Fromm, H. J. (1975) Initial Rate Enzyme Kinetics, pp. 281–294, Springer-Verlag, Berlin
4. Segel, I. H. (1975) Enzyme Kinetics, pp. 506–749, John Wiley and Sons, New York
5. Huang, H. T. & Niemann, C. (1951) J. Am. Chem. Soc. **73**, 1541–1548
6. Alberty, R. A. & Koerber, B. M. (1957) J. Am. Chem. Soc. **79**, 6379–6382
7. Darvey, I. G. & Williams, J. F. (1964) Biochim. Biophys. Acta **85**, 1–10
8. Duggleby, R. G. & Morrison, J. F. (1978) Biochim. Biophys. Acta **526**, 398–409
9. Duggleby, R. G. & Morrison, J. F. (1979) Biochim. Biophys. Acta **568**, 357–362
10. Boeker, E. A. (1984) Biochem. J. **223**, 15–22
11. Boeker, E. A. (1985) Biochem. J. **226**, 29–35
12. Duggleby, R. G. (1986) Biochem. J. **235**, 613–615
13. Boeker, E. A. (1984) Experientia **40**, 453–456
14. Boeker, E. A. (1987) Biochem. J. **245**, 67–74
15. Fernley, H. N. (1974) Eur. J. Biochem. **43**, 377–378
16. Darvey, I. G., Shrager, R. & Kohn, L. D. (1975) J. Biol. Chem. **250**, 4696–4701
17. Duggleby, R. G. & Morrison, J. F. (1977) Biochim. Biophys. Acta **481**, 297–312

18. Kellershohn, N. K. & Laurent, M. (1985) Biochem. J. **231**, 65–74
19. Duggleby, R. G. (1984) Comput. Biol. Med. **14**, 447–455
20. Cox, T. T. & Boeker, E. A. (1987) Biochem. J. **245**, 59–65
21. Selwyn, M. J. (1965) Biochim. Biophys. Acta **105**, 193–195
22. Duggleby, R. G. (1985) Biochem. J. **228**, 55–60
23. Alberty, R. A., Massey, V., Frieden, C. & Fuhlbrigge, A. R. (1954) J. Am. Chem. Soc. **76**, 2485–2493
24. Frieden, C., Wolfe, R. G. & Alberty, R. A. (1957) J. Am. Chem. Soc. **79**, 1523–1525

APPENDIX

The work described in the main paper hinges upon the relationships given by Boeker [1]. Regrettably, that paper contains some typographical errors that we take this opportunity to correct. Dr. Boeker (personal communication) has confirmed these corrections.

The most important for the present purposes is that, for a reversible reaction with one substrate and two products, the definitions given by Boeker [1] in her Table 1 for $C_1 \cdot C$ and $C_2 \cdot C$ have the wrong sign. The expressions for α and β (Boeker's [1] Appendix), which apply to all reactions irrespective of whether they have one or two substrates and one or two products, also contain some errors. Although these expressions are not used in the present work, they are also corrected here. The definition of α contains a term '$J_B B_e J_P P_e$', which should be changed to read '$J_B B_e + J_P P_e$', and a term '$J_{AB} P_e Q_e$', which should be '$J_{AB} A_e B_e + J_{PQ} P_e Q_e$'. The expression for β contains a term '$+ A_e B_e Q_e$', which should be removed.

REFERENCE

1. Boeker, E. A. (1984) Biochem. J. **223**, 15–22

Received 19 July 1988/19 September 1988; accepted 30 September 1988