Thermochimica Acta, 149 (1989) 239–249
Elsevier Science Publishers B.V., Amsterdam – Printed in The Netherlands

BASIC PROGRAM TO DISCRIMINATE AMONG MECHANISMS OF HETEROGENEOUS SOLID–GAS DECOMPOSITION

T. COSEAC and E. SEGAL

Department of Physical Chemistry and Electrochemical Technology, Faculty of Chemical Technology, Polytechnic Institute of Bucharest, Bulevardul Republicii 13, Bucharest (Romania)

(Received 2 January 1989)

ABSTRACT

The authors present a BASIC program which selects the conversion function which best describes α,T data, using the iterative method developed by Urbanovici and Segal (Thermochim. Acta., 141 (1989) 9). The program, which has been tested for various model curves, enables discrimination among various mechanisms of solid–gas decomposition.

INTRODUCTION

One of the most important problems concerning the kinetics of heterogeneous solid–gas decomposition consists in finding conversion functions, not necessarily given by the reaction order model, to describe the reaction adequately. Among the attempts to solve this problem using a computer program one has to mention the work of Reich and Stivala [1]. The present work introduces another computer program for discrimination among conversion functions characteristic of various reactions.

MATHEMATICAL FORMULATION OF THE PROBLEM

The derivation of the basic equation has been given in a previous paper [2]. In its most general form, the conversion function $f(x)$ is written as

$$\ln f(x) = \sum_{i=3}^{5} n_i e_i \tag{1}$$

where x is the degree of conversion, $e_3 = \ln(1-x)$, $e_4 = x$, $e_5 = \ln[-\ln(1-x)]$, and n_i are the exponents of $(1-x)$, x and $-\ln(1-x)$, respectively, in $f(x)$. For $n_4 = n_5 = 0$, n_3 is the reaction order. For the general form of the conversion function, $n_1 \equiv n$, $n_2 \equiv m$, $n_3 \equiv p$.

The kinetic parameters n_i, pre-exponential factor A and activation energy E are evaluated through minimization of the sum S_2

$$S_2 = \sum_{\lambda=1}^{N} \left(\sum_{i=1}^{6} n_i e_{i\lambda} \right)^2 \tag{2}$$

where $n_1 = \ln A$, $n_2 = E$, n_3, n_4, $n_5, \ldots$ have known meanings, $n_6 = 1$, $e_{1\lambda} = 1$, $e_{2\lambda} = -1/RT_\lambda$, $e_{3\lambda} = \ln(1-x)$, $e_{4\lambda} = \ln x_\lambda$, $e_{5\lambda} = \ln[-\ln(1-x)]$, and $e_{6\lambda} = -\ln(\Delta x_\lambda / \Delta t_\lambda)$. N is the number of intervals considered on the curve (x,T), $\Delta x_\lambda = x_{i+1} - x_i$ is the width of the interval λ, and Δt_λ is the time interval corresponding to the progress of the reaction with $\Delta\alpha_\lambda$.

The variables x_λ and T_λ are defined by applying the average theorem to the integrals of conversion and temperature on the interval λ

$$\int_{x_i}^{x_{i+1}} \frac{\mathrm{d}x}{f(x)} = \frac{\Delta x_\lambda}{f(x_\lambda)} \tag{3}$$

$$\int_{T_i}^{T_{i+1}} \mathrm{e}^{-E/RT} \, \mathrm{d}T = \Delta T_\lambda \, \mathrm{e}^{-E/RT_\lambda} \tag{4}$$

Minimization of the sum S_2 with respect to n_i ($i = 1$–5) leads to the system of equations

$$\sum_{i=1}^{6} n_i \sum_{\lambda=1}^{N} e_{i\lambda} e_{j\lambda} = 0 \quad j = 1\text{–}5 \tag{5}$$

For known x_λ and T_λ, the values of the kinetic parameters n_i ($i = 1$–5) can be obtained by solving system (5). As x_λ and T_λ are unknown, their values together with the values of the kinetic parameters can be obtained as follows: starting from an arbitrary set of x_λ and T_λ values (e.g. the values corresponding to the middle of the interval), a set of n_i values is obtained by solving system (5). These values are then used to evalaute another set of x_λ and T_λ values by solving eqns. (3) and (4), and so on. The calculations continue until the difference between the successive values of one of the parameters is less than a given value. Thus, in the u order approximation, the approximate values of the parameters n_i, $n_i^{(u)}$ are obtained by solving the system

$$\sum_{i=1}^{6} n_i^{(u)} \sum_{\lambda=3}^{N} e_{i\lambda}^{(u-1)} e_{j\lambda}^{(u-1)} = 0 \quad j = 1\text{–}5 \tag{6}$$

the values of x_λ and T_λ being obtained from the equations

$$\int_{x_i}^{x_{i+1}} \frac{\mathrm{d}x}{\prod_{i=3} (\exp e_i)^{n_i^{(u)}}} = \frac{\Delta x_\lambda}{\prod_{i=3} \left(\exp e_{i\lambda}^{(u+1)}\right)^{n_i^{(u)}}} \tag{7}$$

and

$$\int_{T_i}^{T_{i+1}} \mathrm{e}^{-n_2^{(u)}/RT} \, \mathrm{d}T = \Delta T_\lambda \, \mathrm{e}^{-n_2^{(u)}/RT_\lambda^{(u+1)}} \tag{8}$$

Factorization of the conversion function $f(x)$ in eqn. (7) enables us to make the computation program shorter. The value $x_{\lambda}^{(u+1)}$ to be determined from eqn. (7) appears implicitly in the factors $e_{i\lambda}^{(u+1)}$. The calculations are continued until the value of one of the kinetic parameters n_i fulfils the condition

$$\left| n_i^{(u+1)} - n_i^{(u)} \right| < \varepsilon \tag{9}$$

For simple cases with one or two factors in the conversion function, the number of equations in system (5) is correspondingly reduced. The use of such functions has to be considered, as the general form (eqn. (1)) leads in some cases to poorly conditioned systems of equations.

THE PROGRAM

The flow chart of the program is given in Fig. 1.

Listing

```
5 REM Program to discriminate among mechanisms of solid–gas de-
  composition
10 CLS : PAPER 7 : INK 1
20 DIM e$ (5,10) : DIM a (5,6) : DIM c(6) : DIM b$ (7)
30 DEF FNS (t,x) = t * EXP(−x) * (x↑3 + 18 * x↑2 + 88 * x +
   96)/(x↑4 + 20 * x↑3 + 120 * x↑2 + 120)
40 PRINT INK 7; PAPER 1; "SELECTION OF THE CONVERSION
   FUNCTION"
50 PRINT' "f(x) = (1 − x)↑n" : INPUT b$ (1)
60 PRINT' "f(x) = x↑m" : INPUT b$ (2)
70 PRINT' "f(x) = (−LN(1 − x))↑p" : INPUT b$ (3)
80 PRINT' "f(x) = ((1 − x)↑n) * x↑m) : INPUT b$ (4)
85 PRINT' "f(x) = ((1 − x)↑n) * ((−LN(1 − x))↑p)" : INPUT b$ (5)
90 PRINT' "f(x) = (x↑m) * ((−LN(1 − x)↑p)" : INPUT b$ (6)
100 PRINT' "f(x) = ((1 − x)↑n) * (x↑m) * ((−LN(1 − x))↑p)": INPUT
    b$ (7)
110 CLS : PRINT INK 7; PAPER 1; "INPUT DATA:": PRINT' "−
    initial weight" ' "− final weight" ' "− number of intervals" ' "− error
    1" ' "− error 2" ' "− error 3" ' "− convergence parameter" ' "points
    (weight, temperature, time)
120 INPUT m0, mf, n, e1, e2, e3, ind; DIM m(n + 1):DIM t(n + 1):DIM
    v(n + 1):DIM u(n + 1):DIM f(n):DIM z(n):DIM x(n):DIM y(n):DIM
    p(n):DIM q(n):DIM r(n)
130 FOR i = 1 TO n + 1:PRINT i: INPUT m(i),t(i),v(i):LET t(i) = t(i) +
    273 : NEXT i
```

```
140 LET dm = mf − m0 :LET u(1) = (m(1) − m0)/dm
150 FOR i = 1 TO n: LET ip = i + 1:LET u(ip) = (m(ip) − m0)/dm:LET
    f(i) = (u(ip))/2:LET y(i) = u(ip) − u(i):LET z(i) = (t(ip) − t(i)))/2:LET
    q(i) = t(ip) − t(i):LET r(i) = v(ip) + v(i):NEXT i
```

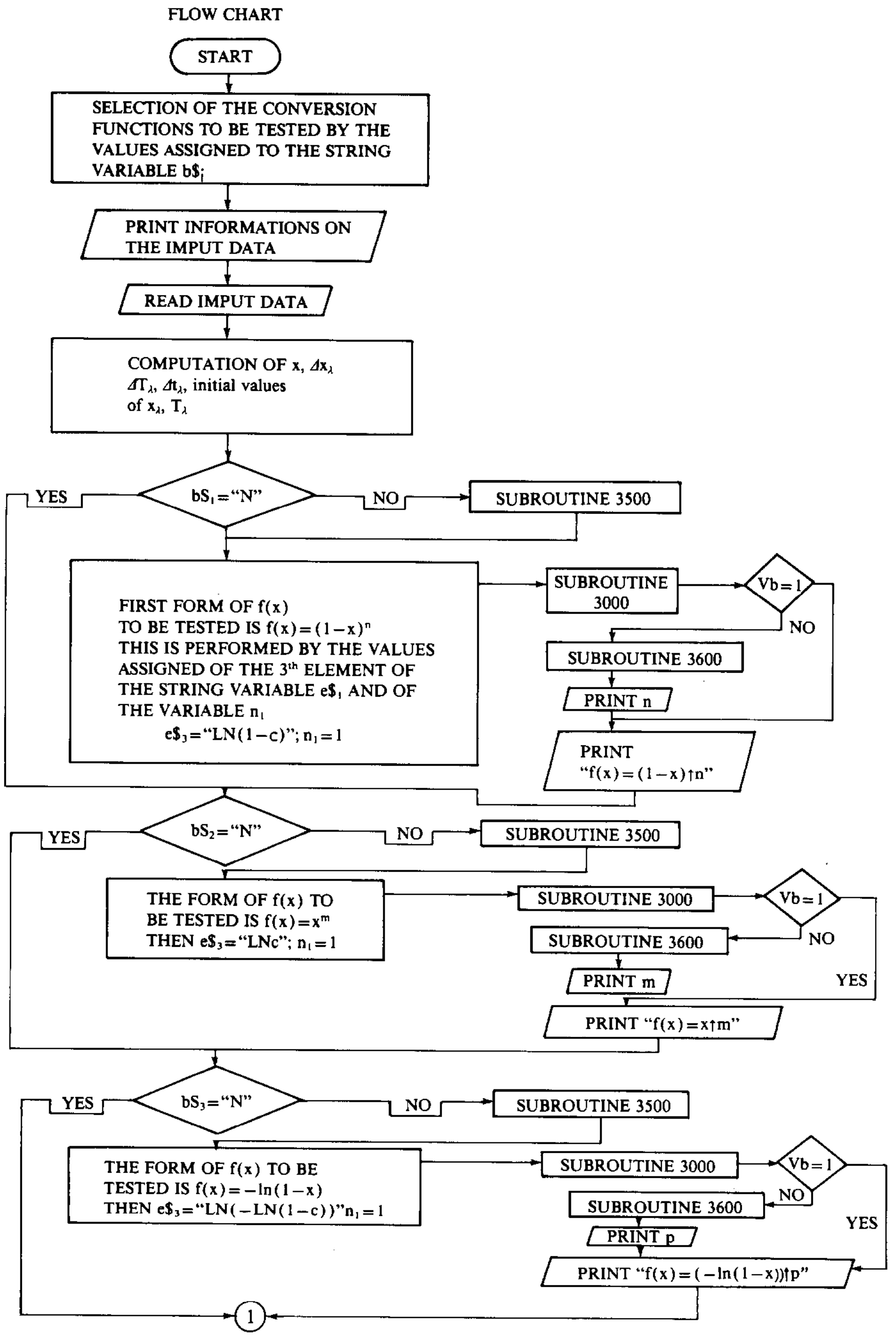

Fig. 1. Flow chart of the program.

```
155 IF b$ (1) = "N" THEN GO TO 210
160 GO SUB 3500
170 LET e$ (3) = "LN(1 - c)":LET n1 = 1: GO SUB 3000
180 IF vb = 1 THEN GO TO 200
190 GO SUB 3600
195 PRINT "n = "; a(3,n3)
200 PRINT INK 7; PAPER 1; "f(x) - (1 - x)↑n"
210 STOP
220 IF e$ (2) = "N" THEN GO TO 280
230 GO SUB 3500
```

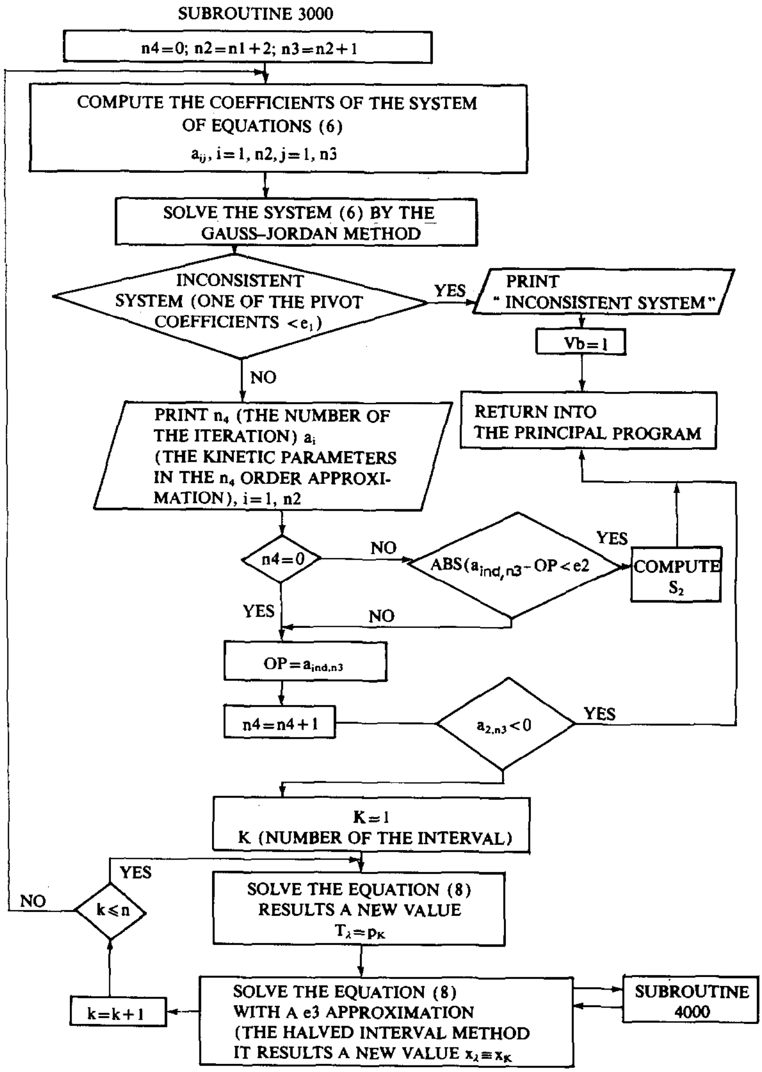

Fig. 1. (continued).

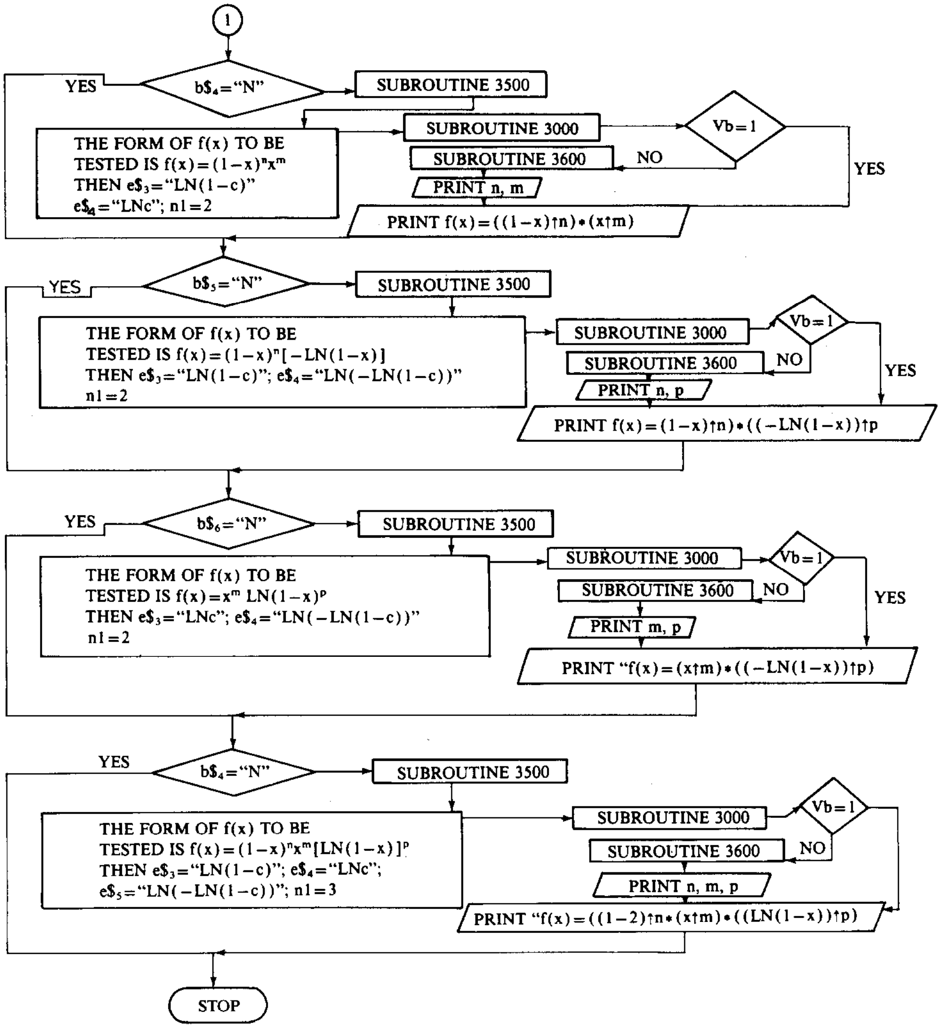

Fig. 1. (continued).

```
240 LET e$ (3) = "LNc":LET n1 = 1: GO SUB 3000
250 IF vb = 1 THEN GO TO 270
255 GO SUB 3600
260 PRINT "m = "; a(3,n3)
270 PRINT INK 7; PAPER 1; "f(x) = x↑m"
280 STOP
290 IF b$ (3) = "N" THEN GO TO 350
300 GO SUB 3500
310 LET e$ (3) = "LN(-LN(1 - c))":LET n1 = 1:GO SUB 3000
320 IF vb = 1 THEN GO TO 340
325 GO SUB 3600
```

```
330 PRINT "p = "; a(3,n3)
340 PRINT INK 7; PAPER 1; "f(x) = (-LN(1 - x))↑p
350 STOP
360 IF b$ (4) = N" THEN GO TO 430
370 GO SUB 3500
380 LET e$ (3) = "LN(1 - c)":LET e$ (4) = "LNc":LET n1 = 2:GO SUB
    3000
390 IF vb = 1 THEN GO TO 420
400 GO SUB 3600
410 PRINT "n = "; a(3,n3)' "m = "; a(4,n3)
420 PRINT INK 7; PAPER 1; "f(x) = ((1 - x)↑n) * (x↑m)"
430 STOP
440 IF b$ (5) = "N" THEN GO TO 430
450 GO SUB 3500
460 LET e$ (3) = "LN(1 - c)":LET e$ (4) = "LN(-LN(1 - c))":LET n1
    = 2:GO SUB 3600
470 IF vb = 1 THEN GO TO 500
480 GO SUB 3600
490 PRINT "n = "; a(3,n3)' "p = "; a(4,n3)
500 PRINT INK 7; PAPER 1; "f(x) = ((1 - x)↑n) * ((-LN(1 - x))↑p)"
510 STOP
520 IF b$ (6) = "N" THEN GO TO 590
530 GO SUB 3500
540 LET e$ (3) = "LNc":LET e$ (4) = "LN(-LN(1 - c))":LET n1 =
    2:GO SUB 3000
550 IF vb = 1 THEN GO TO 580
560 GO SUB 3600
570 PRINT "m = "; a(3,n3)' "p = "; a(4,n3)
580 PRINT INK 7; PAPER 1; "f(x) = (x↑m) * ((-LN(1 - x))↑p)
590 STOP
600 IF b$ (7) = "N" THEN GO TO 670
610 GO SUB 3500
620 LET e$ (3) = "LN(1 - c)":LET e$ (4) = "LNc":LET e$ (5) = "LN
    (-LN(1 - c))" :LET n1 = 3:GO SUB 3000
630 IF vb = 1 THEN GO TO 660
640 GO SUB 3600
650 PRINT "n = "; a(3,n3)' "m = "; a(4,n3)' "p = "; a(5,n3)
660 PRINT INK 7; PAPER 1; "f(x) = ((1 - x)↑n) * (x↑m) * ((-LN(1 -
    x))↑p)"
670 STOP
3000 FOR i = 1 TO n:LET x(i) = f(i):LET p(i) = z(i):NEXT i:LET n4 =
    0:LET n2 = n1 + 2:LET n3 = n2 + 1
3010 FOR i = 1 TO n2:FOR j = 1 TO n3:LET a(i,j) = 0:NEXT j:NEXT i
3020 FOR k = 1 TO n
```

```
3030 LET c(1) = 1:LET c(2) = −1/1.986/p(k):LET c(n3) = LN(y(k)/r(k))
3040 FOR j = 3 TO n2:LET c = x(k):LET c(j) = VAL e$ (j):NEXT j
3050 FOR i = 1 TO n2:FOR j = 1 TO n3:LET a(i,j) = a(i,j) + c(i) * c(j):
     NEXT j:NEXT i
3060 NEXT k
3070 LET k = 1:LET vb = 0
3080 IF k > = n2 THEN GO TO 3110
3090 LET amax = ABS(a(k,k)):LET imax = k:LET kp1 = k + 1
3094 FOR i = kp1 TO n2:IF amax > = ABS(a(i,k)) THEN GO TO 3100
3098 LET amax = ABS(a(i,k)):LET imax = i
3100 NEXT i
3104 IF imax = k THEN GO TO 3110
3108 FOR j = k TO n3:LET at = a(imax,j):LET a(imax,j) = a(k,j):LET
     a(k,j) = at:NEXT j
3110 IF ABS a(k,k) < = e1 THEN GO TO 3190
3120 LET div = a(k,k)
3130 FOR j = k TO n3:LET a(k,j) = a(k,j)/div:NEXT j
3140 FOR i = 1 TO n2:LET mult = a(i,k):IF i = k THEN GO TO 3160
3150 FOR j = k TO n3:LET a(i,j) = a(i,j) − mult * a(k,j):NEXT j
3160 NEXT i
3170 IF k > = n2 THEN GO TO 3200
3180 LET k = k + 1:GO TO 3080
3190 CLS:PRINT "INCONSISTENT SYSTEM":LET vb = 1:GO TO 3450
3200 PRINT'n4:FOR i = 1 TO n2:PRINT a(i,n3):NEXT i:IF n4 = 0 THEN
     GO TO 3220
3210 IF ABS (a(ind,n3) − op) < = e2 THEN GO TO 3400
3220 LET n4 = n4 + 1:LET op = a(ind,n3):IF a(2,n3) < 0 THEN GO TO
     3450
3230 LET k = 1:LET l1 = FNS (t(k), a(2,n3)/1.986/t(k)):LET c1 = u(k):
     GO SUB 4000
3240 LET 1m1 = 1/prod
3250 FOR k = 1 TO n:LET kp = k + 1
3260 LET 12 = FNS(t(kp), a(2,n3)/1.986/t(kp)):LET int1 = l2 − l1:LET l1
     = l2:LET p(k) = a(2,n3)/1.986/(LN q(k) − LN int1)
3270 LET c1 = f(k): GO SUB 4000
3280 LET 1m2 = 4/prod:LET c1 = u(kp):GO SUB 4000
3290 LET 1m3 = 1/prod:LET int2 = (1m1 + 1m2 + 1m3)/6:LET 1m1 =
     1m3
3300 LET xl1 = u(k):LET x12 = u(kp)
3305 LET x1m = (xl1 + x12)/2
3310 LET c1 = x1m:GO SUB 4000
3320 LET sgm = SGN(int2 − 1/prod)
3330 LET c1 = xl1:GO SUB 4000
3340 LET sgl = SGN(int2 − 1/prod)
```

```
3350 IF INT (sgm - sgl) = 0 THEN GO TO 3370
3360 LET x12 = xlm:GO TO 3380
3370 LET xl1 = xlm
3380 IF ABS (x12 - xl1) > e3 THEN GO TO 3305
3390 LET x(k) = (xl1 + x12)/2: NEXT k:GO TO 3010
3400 LET S2 = 0
3410 FOR k = 1 TO n:LET S1 = a(1,n3) - a(2,n3)/1.986/p(k) - LN(y(k)/
     r(k))
3420 FOR i = 3 TO n2:LET c = x(k):LET S1 = S1 + VAL e$ (i):NEXT i
3430 LET S2 = S2 + ABS(S1)↑2
3440 NEXT k
3450 RETURN
3500 CLS:PRINT AT 10, 10; INK 7; PAPER 1; "COMPUTER IS
     WORKING"
3510 RETURN
3600 CLS:PRINT "S2 = "; S2'"A"; EXP(a(1,n3))'"E = "; a(2,n3)
3610 RETURN
4000 LET prod = 1
4010 FOR i = 3 TO n2:LET c = cl:LET PROD = (EXP(VAL e$ (i)))
     ↑a(i,n3):NEXT i
4020 RETURN
```

Input data

As the program has been written such that the computer displays the input data together with information concerning their meanings, it is not necessary to give here a list of the identifiers used in the program for various magnitudes in relationships (1)–(9). We shall mention only that e_1 is the limit value under which, after diagonalization of the matrix of system (6), a pivot element is considered as being practically equal to zero; $e_2 \equiv \varepsilon$ from the convergency condition (9); and e_3 is the approximation of the value $T_\lambda^{(u+1)}$ from eqn. (8). The identifier "ind" is used to select the kinetic parameter for the convergency condition (9). Thus, the input values 1 and 2 for "ind" represent kinetic parameters with respect to which the row of n values is considered to be convergent with the logarithm of the pre-exponential factor and the activation energy, and the values 1, 2, 3 represent one of the exponents which appear in the conversion function in the order n, m, p.

Output data

After each iteration the computer displays the number of the iteration and the values of the kinetic parameters, in the following order: ln A; E; the exponents in the conversion function used, in the order n, m, p. If for a given conversion function system (6) is inconsistent, the computer displays

"INCONSISTENT SYSTEM". After condition (9) has been fulfilled, the optional values of the kinetic parameters are displayed.

Structure of the program

The structure of the program is clearly indicated by the flow chart and the listing. No flow charts have been given for subroutines 3500, 3600 and 4000, which, respectively, display on the screen the message "COMPUTER IS WORKING" while the calculations are being performed, determine the optional values of S_2, A and E, and calculate the function $f(x)$ for a given value of x.

The integral from the LHS of eqn. (7) is calculated using Simpson's method. The integral from eqn. (8) is calculated using the Senum and Yang [3] approximation. The use of the string variable $e\$_i$ enables us to define the elements of ln $f(x)$ for a given particular form of $f(x)$, and to realize a unique sequence for the calculation of the coefficients of system (6), of the value of $f(x)$ and of the sum S_2, regardless of the form of $f(x)$. This has led to a considerable simplification of the program.

If the conversion function is chosen erroneously it is possible to obtain a negative value for the activation energy. In such cases it is necessary to introduce another form of the conversion function, previously selected, by means of the instruction "CONTINUE".

Use of the program

First, the forms of the conversion function are chosen. The functions are displayed successively on the screen. If the operator is not interested in the use of a given function the letter "N" should be entered. Any other α-numeric value assigned to the string $b\$_i$ keeps the form i of the conversion function for the calculation of the kinetic parameters. After the input data have been introduced and the calculations performed, the values of the kinetic parameters as well as of the sum S_2 are displayed for the first form of the conversion function. A different form of the conversion function can then be introduced by means of the instruction "CONTINUE".

RESULTS

The program was checked using the data $x(T)$ given in Table 1 of our previous communication [2], as well as for a model curve corresponding to $f(x) = (1 - x)^2$. In both cases we could discriminate among the conversion functions through the minimum values of S_2.

Recommendation

We recommend 10^{-15}–10^{-20} as input values for e_1, and $10^{-2}\ \Delta x$ for e_3. Convergency with respect to E is reached for values of e_2 between 10 and 100 cal mol^{-1}.

Among the values of x which limit the intervals Δx_λ, the values $x = 0$ and $x = 1$ should not be used.

REFERENCES

1 L. Reich and S.S. Stivala, Thermochim. Acta, 94 (1985) 413.
2 E. Urbanovici and E. Segal, Thermochim. Acta, 141 (1989) 9.
3 G.I. Senum and R.T. Yang, J. Therm. Anal., 11 (1977) 445.