Thermochimica Acta, 233 (1994) 47–61
Elsevier Science B.V., Amsterdam
SSDI 0040-6031(93)01459-G

Standard polynomial method for derivatives: Application to non-isothermal and isothermal kinetics and computer programming for the Sharp–Wentworth and Freeman–Carroll methods

S.G. Viswanath [a,*], S.S. Umare [a] and M.C. Gupta [b]

[a] *Laxminarayan Institute of Technology, Nagpur University, Nagpur-440 010 (India)*
[b] *Department of Chemistry, Nagpur University, Nagpur-440 010 (India)*

(Received 6 May 1993; accepted 10 June 1993)

Abstract

A method is suggested for obtaining polynomials from experimental data. These polynomial expressions are differentiated to yield first-order derivatives, and these derivatives are used for computing non-isothermal and isothermal kinetic parameters. A single computer program is presented for the Sharp–Wentworth and Freeman–Carroll methods.

INTRODUCTION

Since Freeman and Carroll [1] suggested a method for the kinetic analysis of thermoanalytical data, several methods have been proposed. Of these, the methods of Coats and Redfern [2] and of Sharp and Wentworth [3] are widely used. The former is an integral method, and the latter and the Freeman–Carroll are differential methods.

BASIC [4–6] and FORTRAN-77 [7] programs have been reported for the Coats–Redfern method. Schempt et al. [8] suggested a FORTRAN-II program for calculating non-isothermal kinetic parameters for an *n*th-order polynomial using a least-squares fit. Carrasco and Costa [9] reported a method using non-linear expansions. Gonzalez and Havel [10] used a polynomial of $\leq$4th degree for the estimation of Arrhenius equation parameters from non-isothermal kinetic data. A number of procedures have been reported for calculating the non-isothermal kinetic parameters. The data are fitted into a pre-chosen polynomial and the coefficients are calculated by the least-squares method. The obtained polynomial is differentiated to obtain the derivatives at a given time or temperature. Sestak et al. [11] compared

[*] Corresponding author.

the manually calculated parameters with those calculated by a FORTRAN program and concluded that the FORTRAN programming method is less accurate, leading to large errors.

Merely obtaining a correlation coefficient of one or nearly one for a chosen polynomial does not mean that the polynomial represents a given function in all respects. When approximating a given function $f(x)$ by a polynomial function, $\phi(x)$, one may ask: how should the closeness of the approximation be measured?; and what are the criteria for the best polynomial approximation to the function? Ideally, the curve generated by the polynomial should be smooth with all the points falling on or nearly on the curve; the error between the calculated and the observed values should be very small, and the correlation coefficient should be nearly one. If these conditions are satisfied then one can expect excellent results.

OBTAINING THE POLYNOMIAL

Let C be the fraction decomposed and Y represent either time t or temperature T; then C may be expressed as a function of Y

$$C^m = x_1 + x_2Y + x_3Y^2 + \cdots + x_nY^{n-1} \tag{1}$$

where n is the number of total data points and m is a real number which may be varied until the three criteria cited above are satisfied. Generally m is varied between -1 and $+1$ but not zero; $x_1, x_2, \ldots, x_n$ are the coefficients of the polynomial. If for n data points, $C_1, C_2, \ldots, C_n$ are different values of the dependent variable C, and $Y_1, Y_2, \ldots, Y_n$ are corresponding values of the independent variable Y, we get n equations, i.e.

$$\begin{aligned}
&x_1 + x_2Y_1 + x_3Y_1^2 + \cdots + x_nY_1^{n-1} = C_1^m\\
&x_1 + x_2Y_2 + x_3Y_2^2 + \cdots + x_nY_2^{n-1} = C_2^m\\
&\vdots\\
&x_1 + x_2Y_n + x_3Y_n^2 + \cdots + x_nY_n^{n-1} = C_n^m
\end{aligned}$$

Each term in each equation and the C terms are considered as the elements of a matrix, then an $n \times (n+1)$ matrix is obtained

$$\begin{bmatrix}
1 & Y_1 & Y_1^2\cdots Y_1^{n-1} & C_1^m\\
1 & Y_2 & Y_2^2\cdots Y_2^{n-1} & C_2^m\\
\vdots & & & \\
1 & Y_n & Y_n^2\cdots Y_n^{n-1} & C_n^m
\end{bmatrix}$$

When the Gaussian elimination method is applied, the lower diagonal

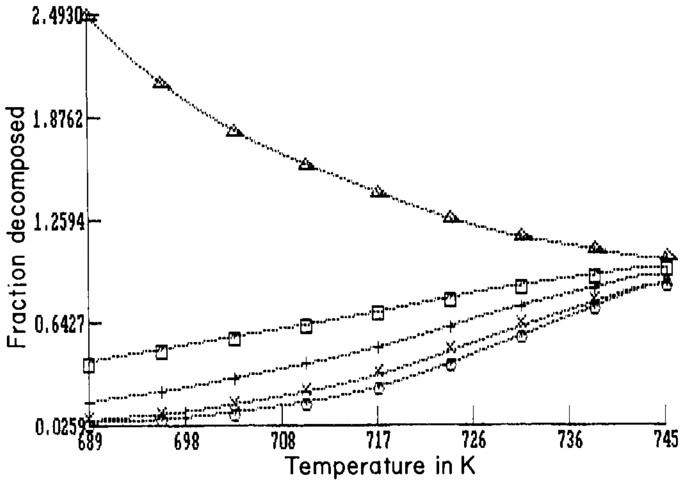

Fig. 1. Computer-generated curves using different *m* values in the polynomial: ○, 1.00, ×, 0.75; +, 0.50, □, 0.25; and △, −0.25.

elements become zero. When the method is applied from bottom to top, the upper diagonal elements become zero, and then the matrix is

$$\begin{bmatrix} x'_1 & \emptyset & \emptyset \cdots \emptyset & C'_1 \\ \emptyset & x'_2 & \emptyset \cdots \emptyset & C'_2 \\ \vdots & & & \\ \emptyset & \emptyset & \emptyset \cdots x'_n & C'_n \end{bmatrix}$$

The coefficients of the polynomial equation are obtained as $x_1 = C'_1/x'_1$, $x_2 = C'_2/x'_2, \ldots, x_n = C'_n/x'_n$.

To illustrate the method, the data for the decomposition of calcium oxalate are taken from Schempt et al. [8]. Using the polynomial, intermediate points are generated through a computer program and curves are drawn. These curves are shown in Fig. 1 for various values of *m* when temperature is used as the independent variable. Table 1 gives the absolute average percent error between the observed and calculated values, and the correlation coefficients for various *m* values. Similar curves are obtained using time as the independent variable. When time is used as the variable, the error is negligible and the correlation coefficient is 1.000.

FREEMAN-CARROLL METHOD

In this method, the following linear relationship is used to estimate the order of reaction *n* and the activation energy E_a [1]

$$\frac{\Delta \ln(\mathrm{d}C/\mathrm{d}t)}{\Delta \ln(1-C)} = n - \frac{E_a \Delta(1000/T)}{R\Delta \ln(1-C)} \tag{2}$$

where $\mathrm{d}C/\mathrm{d}t$ is the time derivative and Δ represents the difference between

TABLE 1

Errors and correlation coefficients for various m values

No.	m	% error	Correlation coefficient
1	1.00	0.937	1.000
2	0.75	0.387	1.000
3	0.50	0.424	1.000
4	0.25	0.033	1.000
5	−0.25	0.081	1.000
6	−0.50	0.069	1.000
7	−0.75	0.038	1.000
8	−1.00	0.010	1.000

two points. Equation (2) can be expressed as a temperature derivative $\mathrm{d}C/\mathrm{d}T$ as

$$\frac{\Delta \ln(\mathrm{d}C/\mathrm{d}T)}{\Delta \ln(1-C)} = n - \frac{E_a \Delta(1000/T)}{R\,\Delta \ln(1-C)} \tag{3}$$

For a linear rate of heating β

$$\mathrm{d}C/\mathrm{d}T = (\mathrm{d}C/\mathrm{d}T)(\mathrm{d}T/\mathrm{d}t) = \beta(\mathrm{d}C/\mathrm{d}T) \tag{4}$$

The last data point is used to find Δ values. When the last point is negative, the penultimate point is used instead. But the Δ values obtained from successive points or from the first or second points do not give good results. From the linear plot of the left-hand-side as a function of $\Delta(1000/T)/\Delta \ln(1-C)$, E_a and n are estimated from eqns. (2) and (3). Plots are shown in Figs. 2 and 3. The n and E_a values for different values of m in the polynomial for the temperature and time derivatives are summarized in Tables 2 and 3 respectively.

SHARP–WENTWORTH METHOD

This method uses the equation [3]

$$[\ln[\mathrm{d}C/\mathrm{d}t]/(1-C)^n] = [\ln(A) - (E_a/R)](1000/T) \tag{5}$$

where $(\mathrm{d}C/\mathrm{d}t)/(1-C)^n$ is the specific reaction rate constant, k. However, if the temperature derivative $\mathrm{d}C/\mathrm{d}T$ is used instead of $\mathrm{d}C/\mathrm{d}t$, then k can be written

$$\ln k = [\ln(A/\beta) - (E_a/R)](1/T) \tag{6}$$

where A is the pre-exponential factor.

The value $\ln(k)$ is plotted against $(1000/T)$, which gives a straight line when the value of n is assumed. The A and E_a values from eqns. (5) and (6) are recorded in Tables 2 and 3 respectively, whereas Figs. 4 and 5 are the

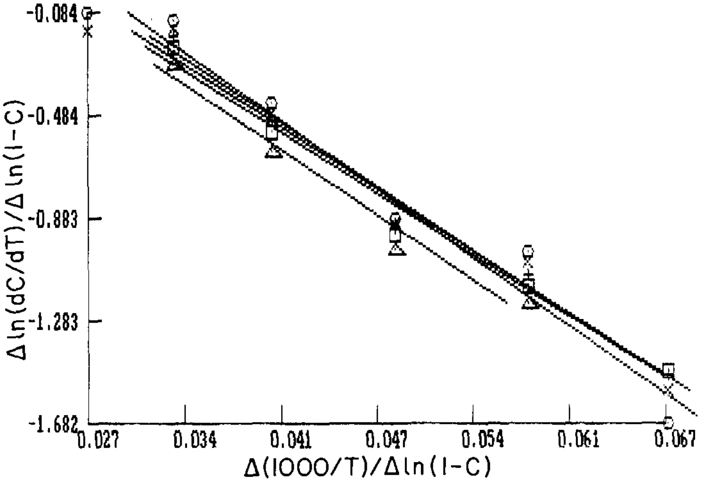

Fig. 2. Temperature derivative plots from the Freeman–Carroll method with different *m* values in the polynomial: ○, 1.00; ×, 0.75; +, 0.50; □, 0.25; and △, −0.25.

respective linear plots. In these, the overall order of reaction *n* is assumed to be one.

ISOTHERMAL KINETICS

In isothermal kinetics, the following relationship is used

$$(-\mathrm{d}c/\mathrm{d}t) = kc^n \qquad (7)$$

A linear plot of $\ln(-\mathrm{d}c/\mathrm{d}t)$ versus $\ln c$ gives *n* and $\ln k$. Figure 6 shows such plot for the reduction of ferric chloride with stannous chloride. Table 4 summarises the order of reaction and reaction rate constant for (A) the reduction of Fe(III) with Sn(II); (B) the saponification of ethyl acetate; and (C) the isomeric change of *N*-chloroacetanilide.

RESULTS AND DISCUSSION

Examination of Fig. 1 and Table 1 shows that for an *m* value of 0.25 or 0.50, the fit to the polynomial is excellent and average percent errors are small. Even though the fit for the polynomial having an *m* value of −0.25 also satisfies these conditions, the shape of the curve is altogether different. Out of these, the *m* value of 0.5 provides the best fit polynomial for the calculation of the kinetic parameters for the decomposition of calcium oxalate using the temperature derivative.

For the polynomials using time as the independent variable, similar curves are obtained, with the average percent error being zero. But in both cases, the correlation coefficients are found to be 1.000. From Tables 2 and 3, it is observed that the order of the reaction is fairly constant for the

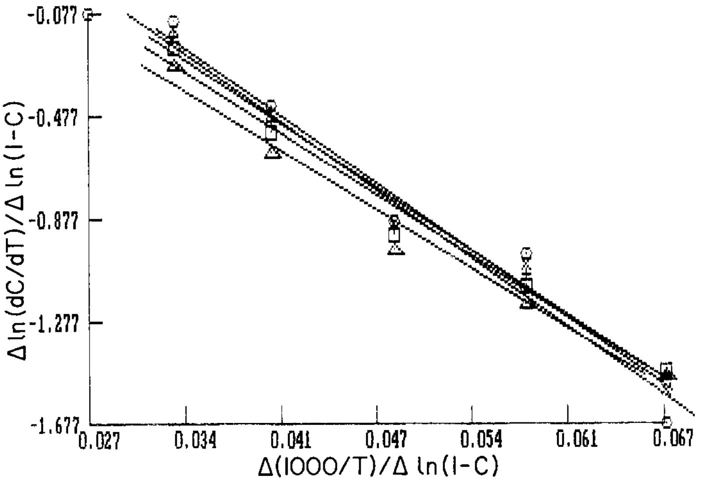

Fig. 3. Time derivative plots from the Freeman–Carroll method with different *m* values in the polynomial: ○, 1.00; ×, 0.75; +, 0.50; □, 0.25; and △, −0.25.

temperature or time derivative, while the activation energy is not consistent in case of the Freeman–Carroll method. However, the activation energy values are fairly constant with *m* values from 0.50 to −0.25 when the Sharp–Wentworth method is used. For *m* values of 0.5 and 0.25, the temperature and time derivatives give similar values of kinetic parameters by both methods and the activation energy values agree within 2% error. So it appears that the order of the reaction could be determined by the Freeman–Carroll method, and using this value for the *n*, the activation energy may be determined from the Sharp–Wentworth method.

Data on the thermal decomposition of solid complexes formed by the ligand 2-carboxyl-2′-hydroxyl-3′,5′-dimethyl azobenzene (CHDMA) with

TABLE 2

Non-isothermal kinetic parameters with different values of *m* of the polynomial equation using the temperature derivative

No.	*m*	Freeman–Carroll method		Sharp–Wentworth method	
		n	E_a/kJ mol^{-1}	E_a/kJ mol^{-1}	$A \times 10^{-19}$/s^{-1}
1	1.00	1.12	331.14	319.44	18
2	0.75	0.94	301.43	309.47	3.4
3	0.50	1.01	310.84	305.02	1.6
4	0.25	0.92	299.97	303.58	1.3
5	−0.25	0.93	316.27	304.21	1.5
6	−0.50	—	—	305.86	1.9
7	−0.75	—	—	306.57	2.2
8	−1.00	—	—	306.58	2.2

TABLE 3

Non-isothermal kinetic parameters with different values of m of the polynomial equation using the time derivative

No.		Freeman–Carroll method		Sharp–Wentworth method	
	m	n	$E_a/\mathrm{kJ\ mol^{-1}}$	$E_a/\mathrm{kJ\ mol^{-1}}$	$A \times 10^{-19}/\mathrm{s^{-1}}$
1	1.0	1.12	331.70	319.13	17
2	0.75	1.13	328.86	309.27	3.3
3	0.50	1.01	311.55	304.85	1.6
4	0.25	0.93	300.79	303.35	1.3
5	−0.25	0.79	288.69	305.60	1.9
6	−0.50	—	—	304.43	1.5

different metal ions [12] have been fitted to the present polynomial method. Figures 7 and 8 give the graphs for the Freeman–Carroll and Sharp–Wentworth methods, respectively. The non-isothermal kinetic parameters determined from these two methods are presented using the temperature derivative in Table 5. The order of reaction obtained by the Freeman–Carroll method is zero for the thermal decomposition of all the solid complexes, which is in agreement with the results reported by Choudhari [12]. For the calculation of these parameters by the Sharp–Wentworth method, the order of reaction is taken as zero.

Figure 9 shows two TG curves: for calcium oxalate and for the iron(II) complex of CHDMA. When the first and last points are joined, these curves may be broadly classified into two categories: below the line and concave; and above the line and convex. For the concave curve, the derivatives are

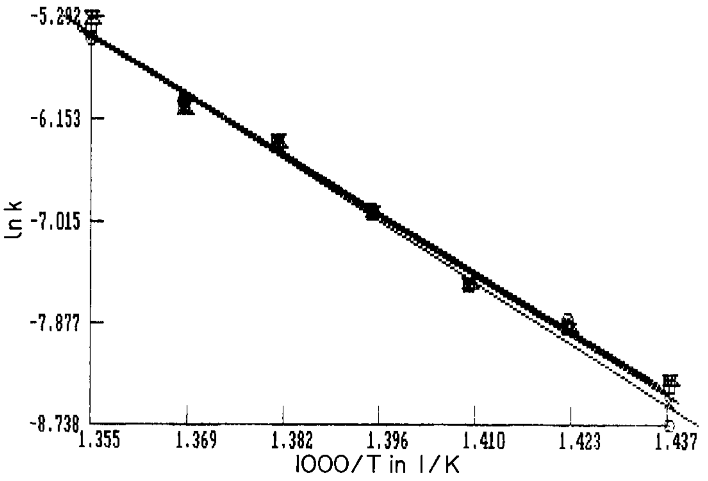

Fig. 4. Temperature derivative plots from the Sharp–Wentworth method with different m values in the polynomial: ○, 1.00; ×, 0.75, +, 0.50; □, 0.25; △, −0.25; and ▽, −0.50.

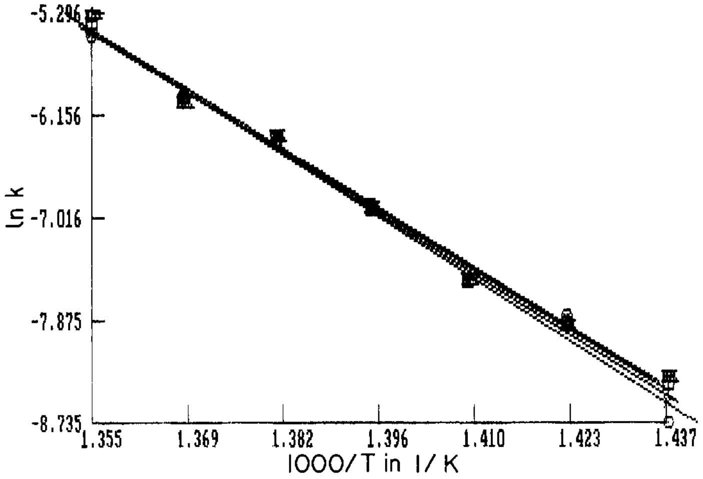

Fig. 5. Time derivative plots from the Sharp–Wentworth method with different *m* values in the polynomial: ○, 1.00; ×, 0.75; +, 0.50, □, 0.25; △, −0.25; ▽, −0.50.

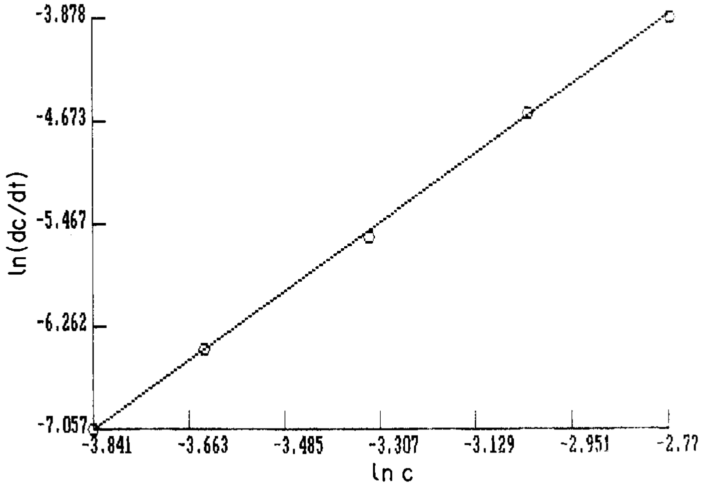

Fig. 6. Plot for isothermal kinetics using the proposed polynomial method.

TABLE 4

Order of reaction *n* and specific reaction rate constant *k* for isothermal kinetics of certain reactions

Reaction[a]	*n*	*k*	
		Present method	Integral method
A	2.99	83.73	85.50
B	2.18	0.0070	0.0068
C	1.00	0.0208	0.0202

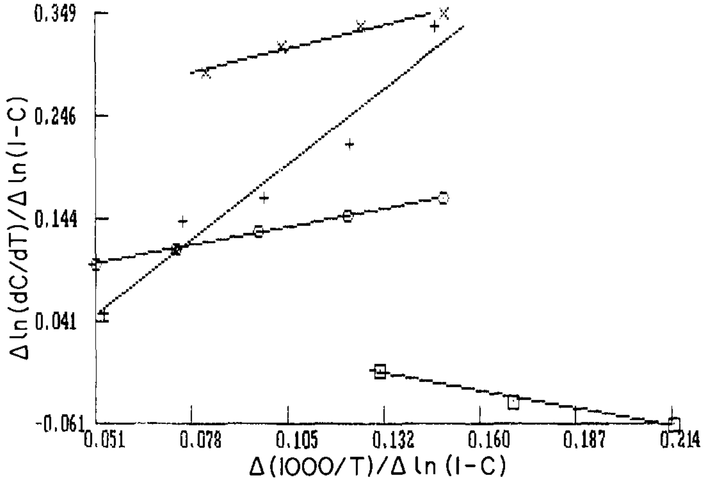

Fig. 7. Temperature derivative plots from the Freeman-Carroll method for CHDMA complexes of: ○, Co(II); ×, Fe(II); +, Mn(II); and □, Ni(II).

in ascending order while in the case of the convex curve, these are in descending order. Points falling above and below the line for concave and convex curves respectively are omitted when the kinetic parameters are determined. The shape of the curve may depend on the mechanism of the thermal decompostion.

Some attempts have been made using a non-linear rate of heating to obtain non-isothermal kinetic parameters for the decomposition of the Ni(II) CHDMA complex. The mass loss and temperature were recorded at regular intervals of time, and the time derivative was obtained using the

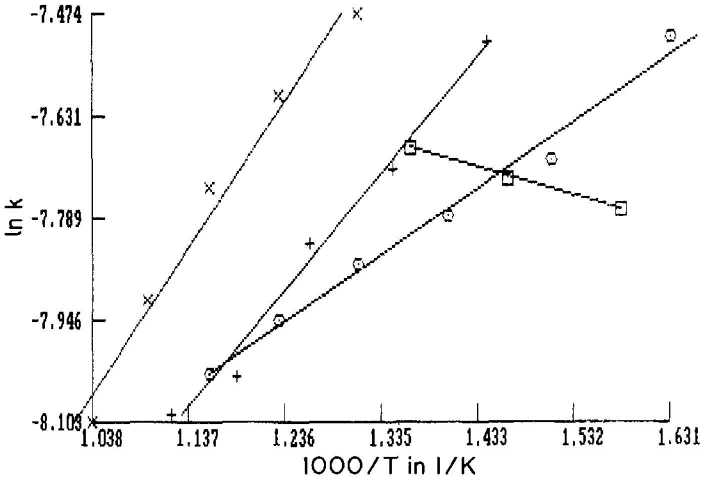

Fig. 8. Temperature derivative plots from the Sharp-Wentworth method for CHDMA complexes of: ○, Co(II); ×, Fe(II); +, Mn(II); and □, Ni(II).

TABLE 5

Non-isothermal parameters for some CHDMA complexes using the temperature derivative

No.	Metal ion in complex	Freeman–Carroll method		Sharp–Wentworth method	
		n	E_a/kJ mol^{-1}	E_a/kJ mol^{-1}	$A \times 10^4$/s^{-1}
1	Co(II)	0.06	5.59	8.69	97
2	Fe(II)	0.22	7.26	19.01	29
3	Mn(II)	−0.09	23.06	15.14	39
4	Ni(II)	0.08	5.37	3.70	8.5

present method for calculating the kinetic parameters. However, the results, as shown in Table 5, are not satisfactory.

To calculate the non-isothermal kinetic parameters using temperature derivatives, a computer program has been written in TURBO PASCAL. It works with version 5.0 or above which can be run until satisfactory results are obtained. The computer program is given in Appendix 1. For both the Freeman–Carroll and Sharp–Wentworth methods, the plots and values of the kinetic parameters can be readily obtained with this program. However the curves presented in Fig. 1 and the plots presented for the methods use different computer programs, for which the data from the above program had been fed into these programs to yield curves or plots on a single sheet.

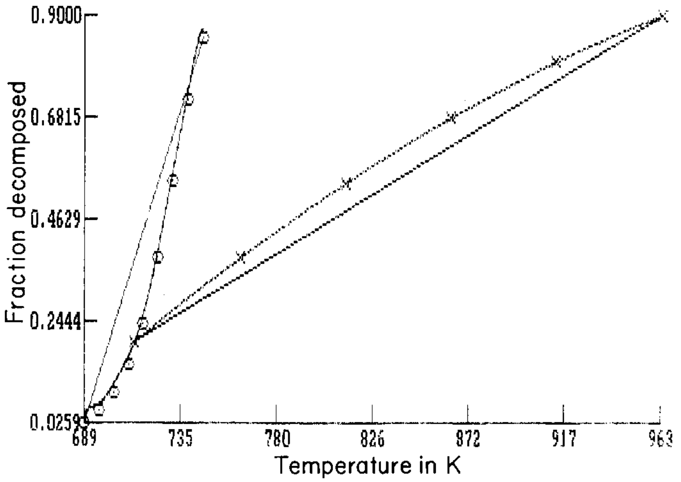

Fig. 9. Different type of TG curves in non-isothermal kinetics: ○, calcium oxalate; ×, Fe(II)CHDMA.

ACKNOWLEGEMENT

The authors thank Dr G.D. Nageswar, the Director, Laxminarayan Institute of Technology, Nagpur for providing computer facilities.

REFERENCES

1 E.S. Freeman and B. Carroll, J. Phys. Chem., 62 (1958) 294.
2 A.W. Coats and J.P. Redfern, Nature, 201 (1964) 68.
3 J.H. Sharp and S.A. Wentworth, Anal. Chem., 41 (1969) 2060.
4 E. Eftimie and E. Segal, Thermochim. Acta, 105 (1986) 247.
5 E. Eftimie and E. Segal, Bull. Inst. Politech. Bucuresti, Ser. Chim., 50 (1988) 51.
6 E. Segal and T. Coseac, Rev. Roum. Chim., 34 (1989) 287.
7 R. K. Sahoo, S. K. Mohanty, V. Chakravorthy and K.C. Dash, Thermochim. Acta, 130 (1988) 369.
8 J.M. Schempt, F.E. Freberg, D.J. Royer and F.M. Angelonic, Anal. Chem., 38 (1966) 520.
9 F. Carrasco and J. Costa, Ing. Quim. (Madrid), 21 (1989) 239.
10 J.L. Gonalez and J. Havel, React. Kinet. Catal. Lett., 39 (1989) 145.
11 J. Sestak, B. Brown, V. Rihak and G. Berggren, Therm. Anal. Proc. 2nd Int. Conf., 2 (1968) 1035, (Pub 1969).
12 H.R. Choudhari, Structural Studies of Metal Complexes, Ph.D Dissertation, Nagpur University, Nagpur, 1981.

APPENDIX

```
PROGRAM nonisoperameters;
uses crt,graph;
VAR
          gd,gm,h,i,n,n1,l,j,p,b:integer;
          a,g,m,k,im,rm,rh,cr,od,odd,er,mic,mac,mat,mit:double;
          f1,f,o,t,t1,c,x,y,y1,z:ARRAY[1..2Ø] OF double;
          d:ARRAY[1..2Ø,1..2Ø] OF double;
          sc,st:ARRAY[1..1Ø] OF string;
          cn,fn,xn,yn,crn:string[25];bo,hc,ch,ho:char;fl:text;
          flag:boolean;
PROCEDURE creatfile;
BEGIG
     write('File to be created (y/n)? : ');readln(ch);
IF upcase(ch)='Y' THEN BEGIN
   write('Enter file name to be created: ');readln(fn);
          assign(fl,fn);rewrite(fl);
write('Enter name of the compound used: ');readln(cn);writeln(fl,cn);
write('Enter initial mass : ');readln(im);writeln(fl,im);
write('Enter residual mass : ');readln(rm);writeln(fl,rm);
write('Enter time in min between two temperature intervals :');
readln(rh);writeln(fl,rh);
write('Enter no. of data points: ');readln(n);writeln(fl,n);i:=1;
REPEAT
          write(i,'. Enter temperature, mass of the  compound: ');
          readln(t[i],c[i]);writeln(fl,t[i],' ',c[i]);i:=i+1
UNTIL i>n END ELSE IF upcase(ch)='N' THEN BEGIN
          write('Enter file name:  ');readln(fn);
          assign(fl,fn);reset(fl);
  readln(fl,cn);readln(fl,im);readln(fl,rm);readln(fl,rh);readln(fl,n);i:=1;
REPEAT
          readln(fl,t[i],c[i]);i:=i+1
UNTIL i>n;
END;
close(fl);i:=1;
REPEAT
```

```
c[i]:=(im-c[i])/(im-rm);y1[i]:=1-c[i];
t[i]:=t[i]+273;t1[i]:=t[i];i:=i+1
UNTIL i>n;n1:=n;rh:=(t[2]-t[1])/(rh*6Ø);
writeln('Temp':1Ø,'Frac.decomp':22);
FOR i:=1 TO n DO writeln(t[i]:1Ø:Ø,c[i]:19:4);
END;
FUNCTION sign(po:double):double;
BEGIN
IF po=Ø THEN sign:=1 ELSE sign:=abs(po)/po
END;
FUNCTION pow(ii,po:double):double;
BEGIN
IF po=Ø THEN IF ii=Ø THEN pow:=1 ELSE pow:=Ø
ELSE pow:=sign(po)*exp(ii*ln(abs(po)))
END;
PROCEDURE standpol;
BEGIN
           g:=1;write('Enter the m value  :');readln(m);i:=1;
REPEAT
y[i]:=c[i];c[i]:=pow(m,c[i]);i:=i+1
UNTIL i>n;i:=1;
{Calculation of matrix elements}
REPEAT
FOR j:=1 TO n DO d[i,j]:=pow(j-1,t[i]);d[i,n+1]:=c[i];i:=i+1
UNTIL i>n;i:=1;
{Formation of diagonal matrix and solution of regression coefficents}
REPEAT
FOR l:=i TO n DO z[l]:=d[l,i];l:=i+1;
REPEAT
FOR j:=1 TO n+1 DO d[l,j]:=d[l,j]-d[i,j]*z[l]/z[i];l:=l+1;
UNTIL l>n;i:=i+1;
UNTIL i>n;i:=n;
REPEAT
FOR l:=i DOWNTO 1 DO z[l]:=d[l,i];l:=i-1;
REPEAT
FOR j:=n+1 DOWNTO 1 DO d[l,j]:=d[l,j]-d[i,j]*z[l]/z[i]; l:=l-1
UNTIL l<1;i:=i-1
UNTIL i<1;
FOR I:=1 TO n DO x[i]:=d[i,n+1]/d[i,i];
END;
FUNCTION set1(xn:double):integer;
BEGIN
set1:=7Ø+round(542*(xn-mit)/(mat-mit));
END;
FUNCTION set2(yn:double):integer;
BEGIN
set2:=179-round(16Ø*(mic-yn)/(mic-mac));
END;
FUNCTION valu(vn:double):double;
BEGIN
valu:=x[j]*pow((j-1),vn);
END;
PROCEDURE correl;
VAR
      a1,a2,a3,a4,b1,b2,c1,c2,c3:double;
BEGIN
i:=1;a2:=Ø;a3:=Ø;a4:=Ø;b1:=Ø;b2:=Ø;c1:=Ø;c2:=Ø;c3:=Ø;
REPEAT
a1:=Ø;FOR j:=1 TO n DO a1:=a1+valu(t[i]);
a2:=a2+a1*c[i];a3:=a3+c[i];a4:=a4+a1;
b1:=b1+a1*a1;b2:=b2+c[i]*c[i];i:=i+1
UNTIL i>n;
c1:=a2/n-(a3*a4)/(n*n);c2:=b1/n-sqr(a4/n);
c3:=b2/n-sqr(a3/n);cr:=c1/sqrt(c2*c3);
END;
PROCEDURE lines;
VAR
          a1,a2:double;
BEGIN
      gd:=detect;initgraph(gd,gm,'c:\tp\bgi');
      line(7Ø,19,7Ø,179);line(7Ø,179,61Ø,179);i:=Ø;
REPEAT
line(7Ø,19+i,82,19+i);i:=i+4Ø
```

```
UNTIL i>160;i:=90;
REPEAT
line(70+i,179,70+i,172);i:=i+90
UNTIL i>600;mic:=c[n];mac:=c[1];mat:=t[1];mit:=t[n];i:=1;
REPEAT
IF c[i]>mac THEN mac:=c[i];IF c[i]<mic THEN mic:=c[i];
IF t[i]>mat THEN mat:=t[i];IF t[i]<mit THEN mit:=t[i];i:=i+1;
UNTIL i>n;
fn:='temp.fil';assign(fl,fn);rewrite(fl);a1:=0;a2:=0;
a1:=(mac-mic)/4;a2:=mic;i:=1;
REPEAT
writeln(fl,a2:0:3);a2:=a2+a1;i:=i+1
UNTIL i>5;writeln(fl,m:0:2);close(fl);assign(fl,fn);reset(fl);i:=1;
REPEAT
readln(fl,sc[i]);i:=i+1
UNTIL i>5;readln(fl,crn);close(fl);
fn:='temp.fil';assign(fl,fn);rewrite(fl);a1:=0;a2:=0;
a1:=(mat-mit)/6;a2:=mit;i:=1;
REPEAT
IF g=1 THEN writeln(fl,a2:0:0);
IF (g=2) OR (g=3) THEN writeln(fl,a2:0:3);a2:=a2+a1;i:=i+1
UNTIL i>7;close(fl);assign(fl,fn);reset(fl);i:=1;
REPEAT
readln(fl,st[i]);i:=i+1
UNTIL i>7;close(fl);
FOR i:=1 TO n DO circle(set1(t[i]),set2(c[i]),5);i:=0;j:=5;
REPEAT
outtextxy(16,14+i,sc[j]);i:=i+40;j:=j-1;
UNTIL j<1;i:=0;j:=1;
REPEAT
outtextxy(60+i,182,st[j]);i:=i+90;j:=j+1;
UNTIL j>7;
END;
PROCEDURE curve;
VAR
          a1,a2,a3:double;
BEGIN
clrscr;lines;
a2:=mit;a3:=(mat-mit)/400;
REPEAT
a1:=0;FOR j:=1 TO n DO a1:=a1+valu(a2);
putpixel(set1(a2),set2(a1),1);a2:=a2+a3
UNTIL a2>mat;
settextstyle(smallfont,horizdir,5);outtextxy(260,188,'Temp.,(K)');
settextstyle(smallfont,vertdir,5);outtextxy(0,40,'(Frac. Decomp.)');
settextstyle(smallfont,vertdir,3);outtextxy(0,22,crn);
readln(ch) END;
FUNCTION der(dn:double):double;
BEGIN
der:=(j-1)*x[j]*pow((j-2),dn);
END;
PROCEDURE datcorr;
VAR
          a1,a2,a3:double;
BEGIN
closegraph;clrscr;writeln;
writeln('m value = ':10,m:2:2);
writeln('--------------------------------------------');
writeln('S.No.':2,'Temp.':8,'Org.mass':10,'Cal.mass':10,'Per.Err.':12)
writeln('--------------------------------------------');a3:=0;i:=1;
REPEAT
a1:=0;FOR j:=1 TO n DO a1:=a1+valu(t[i]);
a2:=(100*(a1-c[i])/c[i]);a3:=a3+abs(a2);
writeln(i:2,t1[i]:10:0,y[i]:10:4,pow(1/m,a1):10:4,a2:10:2);i:=i+1
UNTIL i>n;a3:=a3/n;FOR i:=1 TO n DO c[i]:=y[i];
writeln('--------------------------------------------');
writeln('correlation coefficent =  ':10,cr:2:4);
writeln('Average percent error = ':10,a3:2:3);flag:=false;
REPEAT
write('Like to change m value (y/n)? :');readln(ch);ch:=upcase(ch);
IF ch='Y' THEN BEGIN
standpol;correl;curve;closegraph;datcorr
END ELSE IF ch='N'THEN flag:=false
```

```
UNTIL flag=false;er:=a3;
END;
PROCEDURE differn;
{Determination of derivatives(dC/dT)}
BEGIN
datcorr;clrscr;i:=1;
REPEAT
a:=0;FOR j:=2 TO n DO a:=a+der(t[i]);a:=a*rh;
f[i]:=a/(m*pow(m-1,c[i]));f1[i]:=f[i];y[i]:=y1[i];i:=i+1
UNTIL i>n;n1:=n
END;
PROCEDURE select;
VAR
        a1,a2,a3:double;
BEGIN
clrscr;writeln('Starting FREEMAN-CARROLL method :':35);
writeln('S.no.':5,'dC/dt':8,'frc.left.':16);
FOR i:=1 TO n DO writeln(i:3,f[i]:12:6,y[i]:12:4);
write('Enter value to determine differences :');readln(b);j:=0;i:=1;
REPEAT
IF (b=i) OR (f[i]<0) THEN i:=i+1 ELSE IF (f[i]>0) THEN BEGIN
j:=j+1;a1:=ln(f[i]/f[b]);a2:=ln(y[i]/y[b]);
a3:=1000*(1/t1[i]-1/t1[b]);
c[j]:=a1/a2;t[j]:=a3/a2;z[j]:=a1/a2;o[j]:=a3/a2;i:=i+1 END
UNTIL i>n;n:=j;p:=n
END;
PROCEDURE order;
VAR
                 a1,a2,a3,a4,q1,q2,c1,c2,d1,d2:double;
BEGIN
i:=1;c1:=0;c2:=0;d1:=0;d2:=0;
REPEAT
c1:=c1+t[i];c2:=c2+sqr(t[i]);d1:=d1+c[i];
d2:=d2+c[i]*t[i];i:=i+1
UNTIL i>n;
od:=(c2*d1-c1*d2)/(n*c2-sqr(c1));k:=(d1-n*od)/c1;
clrscr;lines;
q2:=k*mat+od;q1:=k*mit+od;
line(70,set2(q1),610,set2(q2));
xn:=chr(127)+'(1000/T)/'+chr(127)+'ln(1-C)';
yn:=chr(127)+'(dC/dT)/'+chr(127)+'ln(1-C)';
settextstyle(smallfont,horizdir,5);outtextxy(260,188,xn);
settextstyle(smallfont,vertdir,5);outtextxy(0,40,yn);readln(ch);
END;
PROCEDURE acteng;
VAR
                 a1,a2,a3,a4,q1,q2,c1,c2,d1,d2:double;
BEGIN
write('Expected order of reaction :');readln(odd);
FOR i:=1 TO n DO o[i]:=f[i]/pow(odd,y[i]);
FOR i:=1 TO n DO t[i]:=1000/t[i];
i:=1;c1:=0;c2:=0;d1:=0;d2:=0;j:=0;
REPEAT
IF o[i]>0 THEN BEGIN
j:=j+1;c[i]:=ln(o[i]);c1:=c1+t[i];c2:=c2+sqr(t[i]);
d1:=d1+c[i];d2:=d2+c[i]*t[i] END ELSE c[i]:=0;i:=i+1
UNTIL i>n;
od:=(c2*d1-c1*d2)/(n*c2-sqr(c1));k:=(d1-n*od)/c1;
clrscr;lines;
q2:=k*mat+od;q1:=k*mit+od;
line(70,set2(q1),610,set2(q2));
settextstyle(smallfont,horizdir,5);outtextxy(260,188,'1000/T');
settextstyle(smallfont,vertdir,5);outtextxy(0,40,'ln(k)');readln(ch);
END;
PROCEDURE prinout;
BEGIN
IF g=2 THEN BEGIN
writeln('Name of the compound:',cn,'( m = ',m:2:2,')Point selected :',b);
writeln('------------------------------------------------------------');
writeln('S.No.':3,xn:25,yn:25);
writeln('------------------------------------------------------------');
i:=1;
REPEAT
```

```
writeln(i:3,t[i]:20:4,c[i]:28:4);i:=i+1;
UNTIL i>n;END;
IF g=3 THEN BEGIN
writeln('Name of the compound:',cn,'( Order = ',odd:2:2,' & m = ',m:2:2,')');
writeln('------------------------------------------------------------');
writeln('S.No.':3,'T(K)':7,'C':8,'dC/dT':14,'k':11,'ln(k)':14);
writeln('------------------------------------------------------------');
i:=1;
REPEAT
writeln(i:3,1000/t[i]-273:8:0,y[i]:12:4,f[i]:13:6,o[i]:13:6,c[i]:12:4);i:=i+1
UNTIL i>n;END;
writeln('------------------------------------------------------------');
writeln('Activation energy in kCal/mol = ',abs(1.987*k):3:3);
writeln('Activation energy in kJ/mol = ',abs(8.314*k):3:3);
IF g=2 THEN writeln('Order of the reaction = ',od:3:3);
IF g=3 THEN writeln('Pre-exponential factor = ',exp(od):0);
writeln('Regression coefficent =',cr:3:5);
writeln('Average % error =',er:5:3);
END;
PROCEDURE frecar;
BEGIN

g:=2;select;order;closegraph;prinout;flag:=true;
REPEAT
write('Like to deleate  few points (y/n)? :');readln(hc);hc:=upcase(hc);
IF hc='N' THEN BEGIN
write('Like to select new point (y/n)? :');readln(ho);ho:=upcase(ho);
IF (ho='N') AND (hc='N')THEN flag:=false ELSE IF ho='Y' THEN
BEGIN n:=n1;select;order;closegraph;prinout END;END;
IF hc='Y' THEN BEGIN
write('Enter no. of first points to be delated :');readln(i);
write('Enter no. of last points to be delated :');readln(l);j:=1;h:=1;
REPEAT
IF (j>i) AND (j<(p-l+1)) THEN BEGIN c[h]:=z[j];t[h]:=o[j];h:=h+1 END;j:=j+1;
UNTIL j>p;n:=h-1;order;closegraph;prinout END;
UNTIL flag=false;
END;
PROCEDURE shrwen;
BEGIN
clrscr;writeln('Starting SHARP-WENTWORTH method ':30);
i:=1;n:=n1;g:=3;
REPEAT
t[i]:=t1[i];y[i]:=y1[i];f[i]:=f1[i];i:=i+1
UNTIL i>n;flag:=true;
REPEAT
acteng;closegraph;prinout;
write('Like to deleate  few points (y/n)? :');readln(hc);hc:=upcase(hc);
IF hc='N' THEN flag:=false;
IF hc='Y' THEN BEGIN
write('Enter no. of first points to be delated :');readln(i);
write('Enter no. of last points to be delated :');readln(l);j:=1;h:=1;
REPEAT
IF (j>i) AND (j<(n1-l+1)) THEN BEGIN
y[h]:=y1[j];t[h]:=t1[j];f[h]:=f1[j];
h:=h+1 END;j:=j+1;
UNTIL j>n1;n:=h-1 END;
UNTIL flag=false;
END;
BEGIN
clrscr;creatfile;standpol;correl;curve;closegraph;differn;frecar;shrwen
END.
```