Thermochimica Acta, 180 (1991) 303–306
Elsevier Science Publishers B.V., Amsterdam

USE OF DATABASES IN THERMAL ANALYSIS

LEO REICH

Department of Chemistry and Chemical Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 (USA)

(Received 24 August 1990)

ABSTRACT

There have recently become available many commercial programs for microcomputers which can be utilized to manage databases (DBs). Such relational DB managers provide various mathematical, geometric, statistical, financial, string and error-trapping functions. Many of these functions are also available in spreadsheets which have been employed by the present author to determine successfully various kinetic parameters from thermogravimetric (TG) data, such as rate constants (k) and reaction order (n). In view of the preceding, the aim of the present study was to employ a DB script (just as macros were utilized in spreadsheets) to estimate kinetic parameters from TG data. Thus, a DB script was used which contains an algorithm that employs an iterative procedure along with isothermal TG data to determine values of k and n. The powerful Paradox 3 DB was utilized since it possesses the extensive paradox application language (PAL) which can be used to write scripts pertaining to the determination of kinetic parameters from TG data.

INTRODUCTION

In the past, I have utilized spreadsheets to analyse thermogravimetric (TG) data in the evaluation of kinetic parameters, i.e. activation energy (E) and reaction order (n), from non-isothermal TG data or rate constants (k) and n from isothermal TG data (ITG). Various macros were employed and the results obtained were found to be in very satisfactory agreement with previously reported values [1–6]. Thus, spreadsheet analysis was highly recommended for use in thermal analysis either as a primary or secondary (corroborative) method.

Along with the increasing number of spreadsheets that have become commercially available for various microcomputers during recent years, there have also become available commercially many programs for managing databases (DBs). Some of the more popular of these relational DB managers are: dBase IV from Ashton-Tate, R:Base 3.0 from Microrim, Paradox 3 from Borland International, FoxPro from Fox Software and DataEase for DataEase International. Although such DBs specialize in the treatment of DB information for business purposes, they can be considered to be flexible in that they provide many commands and functions (many of which can also be found in spreadsheets) which could conceivably be employed for other purposes. Thus, DBs can provide such mathematical functions as: ABS(), EXP(), LN(), POW(), SQR(), as well as graphics, sorting, etc. Other functions provided are: geometric, statistical, financial, string and error-trapping (however, a glaring omission is single and multiple regression analyses, which are provided by spreadsheets, and which are often utilized in algorithms used to determine kinetic parameters). As in the case of spreadsheets which allow the use of macros, DBs also permit the user to write scripts which allow for the automatic utilization of DB information by merely pressing a few keys (in this respect, it appears at present that spreadsheet macros are easier to handle than DB scripts for the estimation of kinetic parameters from TG).

In view of the preceding, the aim of the present study was to utilize a DB script (just as macros were employed in spreadsheets) to estimate kinetic parameters. Thus, a DB script was used, which contains an algorithm that employs an iterative method, along with ITG data to determine values of k and n. To this end, the powerful Paradox 3 (Borland International) was employed. This DB possesses the extensive paradox application language (PAL) which can be utilized to write scripts.

THEORETICAL ASPECTS

For an "n-order" unimolecular (or pseudo-unimolecular) decomposition which involves ITG data [7]

$$1-(1-\alpha)^{(1-n)}=k(1-n)t \tag{1}$$

where α is the degree of conversion and t is the reaction time.

Designating the left-hand side of eqn. (1) as LHS, then

$$\mathrm{LHS}/((1-n)t)=k \tag{2}$$

For various given values of α and t, an iterative procedure was employed as follows. For particular values of n, various values of k and its standard deviation (SD) could be obtained for various α–t pairs. When the "true" value of n is used or approached, the SD for the k values should attain or approach a minimum value (the true value of n should lead to an SD of zero). An algorithm based on the SD values was used to calculate values of n and k(avg) concurrently.

RESULTS AND DISCUSSION

The given theoretical data (α and t [8]) employed for the ITG analysis is listed in Tables 1 and 2. Tables 1 and 2 list values obtained for n and k

TABLE 1

Final table values of k and n (to 2 decimal places)

α	t (min)	n	k
0.203	10.00	0.67	0.02186
0.377	20.00		0.02191
0.523	30.00		0.02189
0.645	40.00		0.02193
0.743	50.00		0.02190
0.822	60.00		0.02193
0.883	70.00		0.02197
0.928	80.00		0.02198

when n was estimated to 2 and 3 decimal places respectively. Final values obtained for n and k(avg) were: 0.67 and 0.02192 (Table 1), 0.665 and 0.02186 (Table 2) (Ref. 7, 0.67 and 0.02197; 0.666 and 0.02187). From the preceding, the agreement between calculated and reported values (from spreadsheet analysis) was excellent. The PAL script that was employed to estimate n and k(avg) from the ITG data is listed in the Appendix.

The PAL script is depicted along with annotation. The user is initially given a choice of obtaining n to 2 or 3 decimal places (the latter to be chosen where highly accurate data are involved). Then an infinite "For" loop is used along with the ITG data in the DB table (named "thrmanal"). The initial value of n was arbitrarily selected to be 0.2. In order to obtain n to 3 decimal places for these theoretical ITG data, the value of n was ultimately incremented by $+0.1$, decremented by -0.2, incremented by $+0.01$, decremented by -0.02 and, finally, incremented by 0.001 (by means of the b variable). Various values of k(avg) and its SD and n were flashed on the PAL screen (as opposed to the Paradox screen). Final values of the kinetic parameters were indicated when the SD stopped decreasing and began to increase. At that point previous values of n and k(avg) were taken as the final values.

TABLE 2

Final table values of k and n (to 3 decimal places)

α	t (min)	n	k
0.203	10.00	0.665	0.02185
0.377	20.00		0.02188
0.523	30.00		0.02185
0.645	40.00		0.02188
0.743	50.00		0.02183
0.822	60.00		0.02185
0.883	70.00		0.02186
0.928	80.00		0.02186

REFERENCES

1 L. Reich and S.H. Patel, Am. Lab., 19 (1987) 23.
2 L. Reich, Thermochim. Acta, 138 (1989) 177.
3 L. Reich, Thermochim. Acta, 143 (1989) 311.
4 L. Reich, Thermochim. Acta, 164 (1990) 1.
5 L. Reich, Thermochim. Acta, 164 (1990) 7.
6 L. Reich, Thermochim. Acta, 173 (1990) 253.
7 L. Reich, L.Z. Pollara and S.S. Stivala, Thermochim. Acta, 88 (1985) 485.
8 M.E. Brown and C.A.R. Phillpotts, J. Chem. Educ., 55 (1978) 556.

APPENDIX

```
! estimate rate const. and reactn order (to 2 OR 3 dec. places)/isothermal TG
; reaction order can undergo changes of: +0.1, -0.2, +0.01, -0.02, +0.001

Clear  ?" Do you want 'n' to 2 or 3 dec. places (enter 2 or 3): "
Accept "S" To ans
Clear  a=100  b=0.1 mrkr=0    ; initial variable vals.
For counter From 1     ; infinite loop
Edit "thrmanal"
z=[Order]
moveto [Const]
Scan                   ; scan the rate const column
    x=[Alpha] y=[Time]
    []=(1-pow((1-x),(1-z)))/((1-z)*y)   ; calc rate constants
    if z> 2 then canceledit quit        ; arbitrary max for reactn order
    endif
endscan
Do_It!                           ; stop editing
l=cstd("thrmanal","Const")       ; calc std. dev. for rate constants

;------------- display intermediate vals. of k, STD(k), n
    k=caverage("thrmanal","Const")
    @8,20 ?" Avg. value of k = ",k
    @10,20 ?" STD for k = ",l
    @12,20 ?" For a val. of n of ",z
    sleep 1000

;------------ continue with pgm.
if l>=a and mrkr=0 then mrkr=1 a=100
Home editkey
[Order]=[Order]-0.2 Do_It!     ; subtract 0.2 from Order
b=0.01                         ; add 0.01 to Order
endif
if l<a and mrkr=0 then a=l
endif

;------------ if 'n' chosen to 2 dec. places then print final vals./stop
if l>=a and mrkr=1 and ans=2  then Clearall Clear
? "Final vals: n= ",zz," and k(avg.) =",Format("W8.5",kk) sleep 10000 quit
endif

if l>=a and mrkr=1 then mrkr=2 a=100
Home editkey
[Order]=[Order]-0.02           ; subtract 0.02 from Order
b=0.001                        ; add 0.001 to Order
endif
if l<a and mrkr=1 then a=l
endif
if l>=a and mrkr=2 then Clearall Clear
? "Final vals: n= ",zz," and k(avg.) =",Format("W8.5",kk) sleep 10000 quit
endif
if l<a and mrkr=2 then a=l
endif
zz=z kk=k Home
Editkey [Order]=[Order]+b Do_It!   ; continue to change n-values
Clear
Endfor
```