International Journal of Pharmaceutics, 98 (1993) 209–218

IJP 03268

Evaluation of reaction order. Software in pharmaceutics: III

Sverre A. Sande and Jan Karlsen

Department of Pharmaceutics, Institute of Pharmacy, University of Oslo, P.O. Box 1068 Blindern, 0316 Oslo 3 (Norway)

(Received 10 February 1993)
(Accepted 5 April 1993)

Key words: Stability testing; Reaction order; Sampling; Simulation; Computer software

Summary

A computer program for the simulation of experiments intended to determine the order of a reaction is presented along with a discussion and evaluation of the important parameters for the reaction order analysis. The source code is written in C. The program calculates the maximum acceptable amount of reactant remaining at the last sampling point for the estimation of the correct reaction order with a level of significance of 0.95. Based upon the simulations, the precision of the analytical method was shown to be more important than the number of samples in determining the correct order of the reaction at an early stage. The simulations also showed that even with good analytical methods (SD = 1.0%), the reaction had to be followed for about one half-life ($t_{1/2}$) of the reactant. A sampling scheme with equal times between sampling was found to be as good as a scheme with equal differences between the measured concentrations. This is due to the lack of correlation between the rate of reaction and differences between the course of reaction predicted by the two reaction orders. When the experimental errors are expected to be normally distributed, the use of integer orders is preferred over a model employing decimal orders due to the increase in speed of the simulations.

Introduction

Mathematical models describing the kinetics of chemical reactions employ the concept of reaction orders for the calculation. Basically the term is supposed to indicate the number of molecules taking part in a reaction, and is thus considered to be an integer. However, for complex reactions and reactions of higher orders where only one of several reacting species is determined, orders given as integers do not suffice. In a previous paper, a mathematical model for this purpose was presented (Sande and Karlsen, 1991):

$$C_s = \left(C_0^{1-x} - kt \cdot (1-x)\right)^{1/(1-x)},$$

with the half-life

$$T_{1/2} = C_0^{1-x} \cdot (2^{x-1} - 1)/k(x-1)$$

C_s is here the concentration of unreacted substance at time t, C_0 denotes the initial concentration, k is a rate constant and x represents the order of the reaction. The model is valid with some restrictions for all orders, except $x = 1$. For first order reactions ($x = 1$) the usual equations:

$$C_s = C_0 \cdot e^{-kt}$$

Correspondence to: S.A. Sande, Department of Pharmaceutics, Institute of Pharmacy, University of Oslo, P.O.Box 1068 Blindern, 0316 Oslo 3, Norway.

with the half-life

$$T_{1/2} = \ln(2)/k$$

apply.

This model may also be employed for elementary reactions where the order is known to be an integer, but where the actual order is unknown. However, due to analytical errors, fitting the model to experimental data will usually not produce an integer order as best fit. The purpose of this paper is to examine the factors influencing the determination of the correct order of a reaction through simulation, thereby enabling better experimental design and differentiation between competing orders.

Theory

The main parameter affecting the calculation of the order of a given reaction is the extent of time the reaction is followed. As stated by several authors (e.g., Taylor et al., 1987; and Yang, 1981), following the reaction for too short a period (e.g., until less than 25% decomposed) makes the differentiation between reaction orders impossible, and therefore the calculation of the reaction order ambiguous. The reason is that the differences between the course of the reaction predicted by the different orders are too small. All parameters of the model will influence the speed of degradation. As a general discussion on this subject it is thus preferable to employ the amount of substance remaining (or amount reacted) instead of the time after the start of the reaction, thereby eliminating the need to specify values for C_0 and k.

Other factors which may influence the differentiation between different reaction orders are:

The order of the reaction itself (x): The difference between the degradation profiles predicted by second and third order is less than that between zero and first order. Consequently, it is more difficult to differentiate between second and third order reactions than between zero and first order reactions.

The magnitude of difference in the reaction order one wishes to determine: For elementary reactions this would be 1. For complex reactions other figures could be relevant.

The time scheme of sampling: The sampling scheme most frequently used in kinetic studies is a scheme providing equal spacing between the measured concentrations during the experiment.

This means frequent sampling at the start of the experiment, becoming less frequent as degradation proceeds. An alternative is sampling at equally spaced times. We have investigated both types of sampling schemes.

The number of samples drawn during the reaction.

The precision of the analytical method employed.

We found it most reasonable to investigate the differentiation between a first order reaction and zero and second order reaction, respectively. Based on this supposition it is possible to construct tables for a given number of samples vs analytical precision showing the maximum acceptable amount of reactant remaining in the last sample in order to determine the correct reaction order with a satisfactory degree of significance.

Calculations

The calculations are outlined as follows: Starting out with the generation of a perfect first order course of reaction from 100 to 1% of reactant with a given number of samples at equal time intervals (e.g., 6), normally distributed random errors were imposed on the samples, thereby simulating a real analysis. The standard deviation of the random errors corresponded to the precision of our fictitious analytical method. The general model was then fitted to the generated set of data producing a calculated order of the reaction. This procedure was then repeated a predetermined number of times, thereby determining the expected distribution of the calculated reaction orders.

Since we followed the entire course of the reaction, we would expect the distribution to be

narrow and close to the correct value of the reaction order. The procedure was therefore repeated for a course of the reaction from 100 to 2% remaining reactant, thus producing a somewhat broader distribution, then from 100 to 3%, etc. Sooner or later a minimum concentration C_{mn} is reached where the distribution of the calculated reaction orders is so broad that it is not significantly different from a zero or second order reaction. As a condition for the level of significance we chose: the estimated reaction orders should equal the correct reaction order (i.e., 1) in 95% of the calculations. The previous value of C_{mn} is thus the concentration limit to which an analysis has to be carried out in order to determine the correct reaction order with the given level of significance. A flow chart for the evaluation is given in Fig. 1.

A question that must be addressed is the criterion for an excessively broad distribution, i.e., what are the results from the curve fitting that would lead to erroneous conclusions?

Two different approaches are possible. In both cases, we have to follow the reaction long enough so that the results are within the limits in 95% of the simulations:

(1) A reaction is considered a first order reaction if the order with minimum squared sum of residuals (SSR) for fitting of the general equation is greater than or equal to 0.5 and less than 1.5.

(2) Fitting of the model is performed only for integer orders and the order with the least squared sum of residuals is chosen as the assumed correct order.

Method 2 corresponds to the traditionally employed procedure for fitting the data to the possible models for the integer orders and choosing the one showing the best fit either using graphical techniques (linear regression) or other methods.

When the distribution of SSR vs reaction order for a data set is symmetrical around the minimum, these two methods produce equal results. For other distributions it is conceivable that, e.g., the best fit decimal order is 0.4 and the SSR for a first order reaction is less than the SSR for a zero order reaction. In this case, determination of which order to choose would have to be

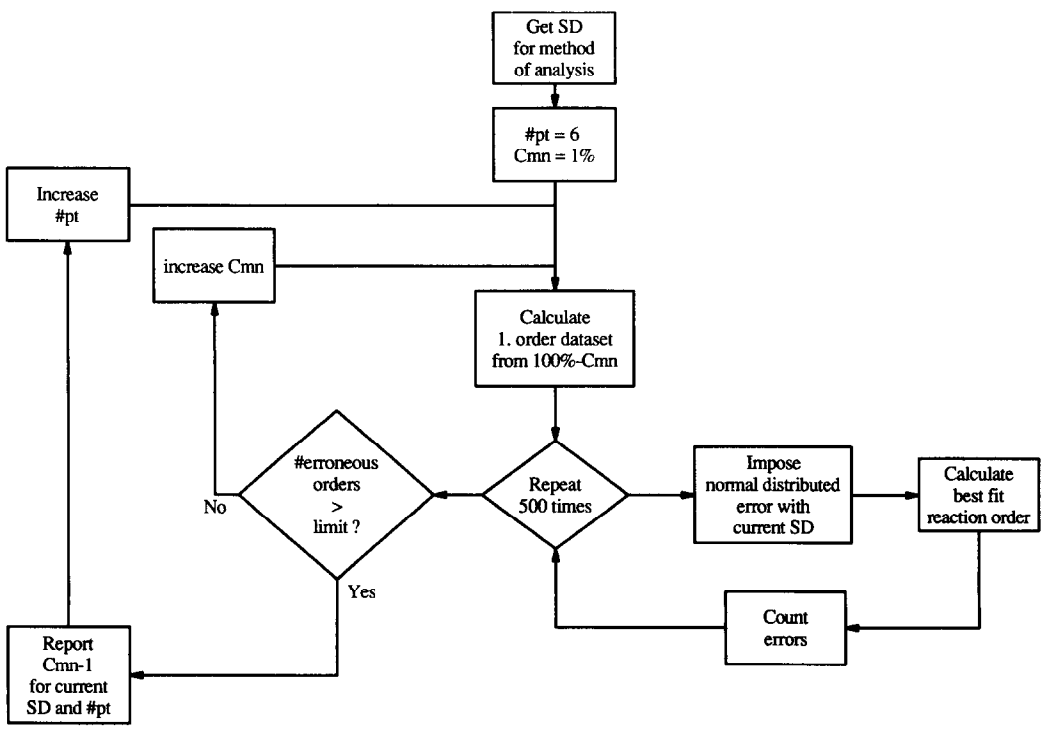

Fig. 1. Flow-chart describing estimation of C_{mn} by simulation of a degradation reaction.

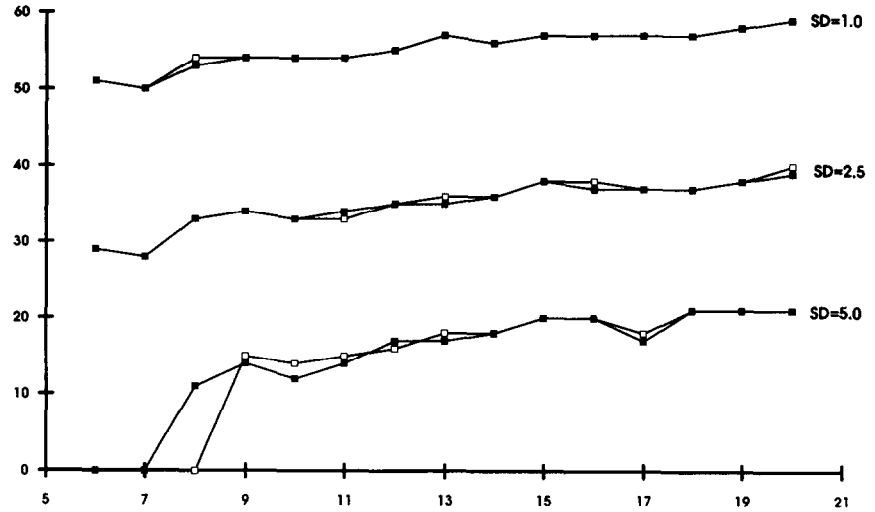

Fig. 2. C_{mn} vs number of samples for methods of analysis with 1, 2.5 and 5% SD, and a sampling scheme with equal difference between sample time. (□) Using integer orders; (■) using decimal orders.

based on additional information (e.g., chemical structure). Method 1 is thus a more general approach yielding more consistent results. In our case we decided to try both methods to evaluate any possible differences.

The next question that has to be considered is the number of simulated data sets necessary to obtain a good picture of the limits. 100 data sets give a rough estimate, and 1000 give a good estimate. The only upper limit for the calcula-

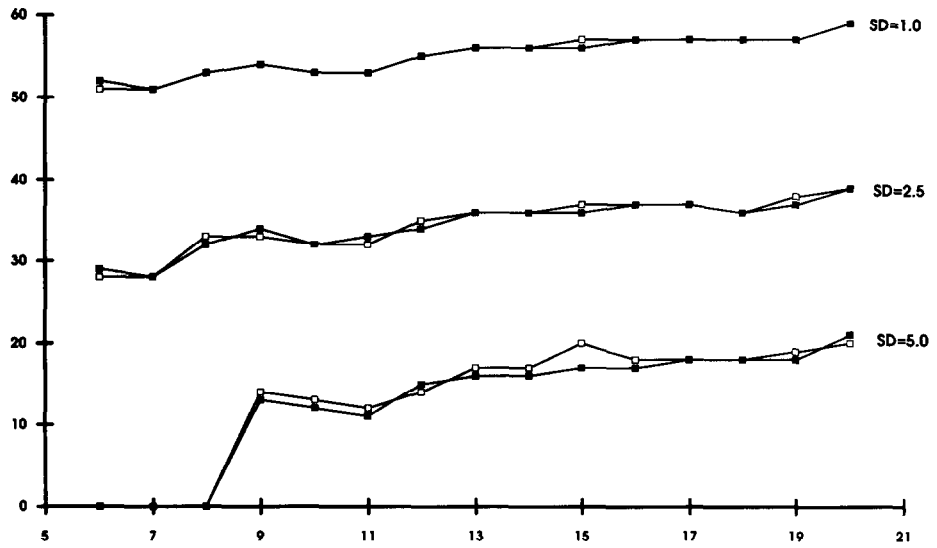

Fig. 3. C_{mn} vs number of samples for methods of analysis with 1, 2.5 and 5% SD, and a sampling scheme with equal difference between sample concentration. (□) Using integer orders; (■) using decimal orders.

tions is the time required to complete the simulation, and is therefore dependent on the computer capacity available. We chose the simulation of 500 data sets as a reasonable compromise.

The outline of the source code for a computer program in C performing the necessary calculations is given in the Appendix.

A caveat is required regarding the random error generator. With 500 data sets and 20 points per data set the number of random errors required is 10000. If the numbers generated are correlated too much an optimum random distribution of the errors is not acquired. An in-depth discussion on the subject along with proper source code for an error generator has been described by Press et al. (1988)

Simulation

Based upon the considerations described above the following simulations were performed:

For all simulations the number of data sets employed was 500. The precisions of the fictive analytical method were 1, 2.5 and 5%. The number of generated points for each simulated analysis ranged from 6 to 20.

Simulation 1: Using the general model for curve fitting, and a sampling scheme with equal time spacing.

Simulation 2: Using integer orders for curve fitting, and a sampling scheme with equal time spacing.

Simulation 3: Using the general model for curve fitting, and a sampling scheme producing equal differences between the sample concentrations.

Simulation 4: Using integer orders for curve fitting, and a sampling scheme producing equal differences between the sample concentrations.

Results and Discussion

As can be seen from Figs 2 and 3, the results from the four different simulations showed only insignificant differences. In summary, the reactions had to be followed until 50–60% remaining using a method of analysis with SD of 1%, until 30–40% remaining using a method of analysis with SD of 2.5% and until 10–20% remaining using a method of analysis with SD of 5%. Using less than nine points renders the order of the reaction indeterminant with the least precise method of analysis.

Increasing the number of samples from 6 to 20 had a marked influence on the results by a near linear increase in C_{mn} by 10 units, however, the most important parameter was the precision of the method of analysis by an increase in C_{mn} by 20 units by halving the SD. This emphasizes the importance of optimizing the method of analysis (i.e., choosing the best analytical method), since compensating for a poor analytical method by increasing the number of samples analyzed is only possible to a limited extent.

Due to the fewer number of curve fittings required for integer orders (simulations 2 and 4) this method used only 2/3 of the time required for simulation 1 and 3. The time required for the simulation of 500 data sets with an error SD of 1 and number of generated points from 6 to 20 on a Intel 386 SX, 16 MHz computer equipped with a mathematical coprocessor is 50 min.

The lack of difference between the results from the two methods for fitting the model to the data (decimal vs integer orders) indicates a symmetric distribution of SSR vs order of reaction for the type of errors used in this simulations. Due to the higher speed of calculation, employing integer orders is preferred for this type of simulations. For the estimation of reaction orders from real analysis we recommend the general model for the reasons described above (see Calculation).

The simulations demonstrate that a sampling scheme with more frequent sampling at the start of the reaction does not provide any advantages with respect to the calculation of correct reaction order over a scheme with samples taken at constant time intervals. This is probably due to the fact that the differences between the functions predicted by the two reaction orders are evenly distributed over the sampling period, i.e., there is no correlation between the rate of the reaction

and the differences between the reaction courses. Duc to the lack of differences we may conclude that the points of time for the measurements have less influence on the calculated reaction order than the number of points measured, as long as the points are reasonably spread out in time.

It is necessary to emphasize that this conclusion is only valid with respect to the calculation of reaction order. When the reaction order is known, and the purpose of curve fitting is to determine the kinetic parameters (C_0 and k), more frequent sampling at the initial stages is recommended.

The simulations employ normally distributed random errors. We have thereby assumed the errors introduced to be purely random. However, in analytical laboratories there are also other types of errors, e.g., systematic and outliers (errors which should not occur according to the normal distribution). Systematic errors will only affect the determination of reaction order to a limited extent, whereas outliers may distort a curve completely. It would therefore be wise to interpret these results as guidelines to the concentrations to which extent the reaction at least has to be followed in order to determine the order of the reaction.

If the reaction orders calculated using the general model were normally distributed, the standard deviation could have been used as a better measure for the broadness of the distribution. The SD would have to be less than 0.25 ($2 \cdot SD < 0.5$). However, the actual distribution of the estimated orders can not be determined a priori. Evaluation of 1000 calculated reaction orders showed a slightly negative skewness of the distribution (mean, 1.02; skewness, -0.19; curtosis, -0.04). We therefore had to count 'erroneous' estimates corresponding to the method using integer orders.

It is interesting to note that also with very precise methods of analysis, the reaction has to be followed for about one half-life ($t_{1/2}$). This is a problem if the products of a reaction react with the primary substance or alter the original reaction, which is often the case with photochemical reactions (Moore, 1987). In such reactions it is probably better to use methods other than the reaction order to determine the mechanism and kinetics of the degradation.

We have chosen to differentiate between first order reactions and zero and second order, but the principles described are also generally applicable with other orders of reaction and differences. Minor alterations in the source code will allow simulation of other differentiations desired.

Conclusion

The main important parameter in determining the order of a reaction is the precision of the method of analysis.

When the experimental errors may be expected to be normally distributed, the use of integer orders is preferred due to the increase in speed of the simulations (simulation 2). It should be emphasized, however, that in the fitting of the model to real experimental values, decimal orders of reaction should be used due to the increased information gained by the display of the distribution of the SSR over the range of reaction orders studied.

The design of the sampling scheme has little influence on the estimated reaction order.

Appendix

Outline of the source code for a computer program in C estimating C_{mn} for a given analytical precision, and number of samples from 6 to 20

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <float.h>
```

```
#define MAX - PKT       21
#define NUMSIM          500     /*      Simulate 500 data sets for each
                                        nopt and Cmn*/

/*GLOBAL DEFINITIONS*/
struct parset { double order,ratek,Co,SSR; };
struct linpars { double alfa,beta; };

/*FUNCTIONS*/
/*************************************************************************/
double time_calc(struct parset *pars,double conc)
      /*    The function calculates the time corresponding to a
            given concentration for the parameters given in pars*/
  {   /*   See Sande and Karlsen, 1991*/
  }
/*************************************************************************/
double conc_calc(struct parset *pars,double tim)
      /*    The function calculates the concentration
            corresponding to a given time (tim)for the parameters given in pars */
  {   /*   See Sande and Karlsen, 1991*/
  }
/*************************************************************************/
struct linpars *wlinreg(double *x,double *y,double *w,int np)
      /*    The function calculates the slope (beta) and intercept (alfa) from the np points
            in the arrays x,y and w (x-value, y-value and weight) using general weighted
            linear regression. Return value is a pointer to a structure containing alfa and beta */
  {   /*   See Sande and Karlsen, 1991*/
  }
/*************************************************************************/
struct parset *linfit(double *tval, double *cval, double ordr, int nopt)
      /*    The function calculates the best fit parameters and
            Squared Sum of Residuals for the data and order */
  {   /*   See Sande and Karlsen, 1991*/
  }
/*************************************************************************/
double ordcalc(double *timeval,double *conc,int no_of_points)
      /*    The function calculates the best fit order of the
            reaction */
  {   double tmpdb;
      int i;
      struct parset ordarr[5];

/******* Calculating integer orders *******/
      tmpdb = 0.0;
      ordarr[0] = *linfit(timeval,conc,tmpdb,no_of_points);
      for (i = 1;i < 3;i++)
       {      tmpdb = i;              /* converting i to type double*/
              ordarr[i] = *linfit(timeval,conc,tmpdb,no_of_points);
```

```
            if (ordarr[i].SSR > ordarr[i-1].SSR)
                                    /*No need for comparing orders*/
            return (ordarr[i-1].order);
                                    /*i.e. return when bottom found*/
      };
      return (ordarr[i].order); /*return 2 if no convergence*/
/*******************/
/**For calculation of decimal orders,                                    *\
*     replace the source above with :                                    *
*                                                                        *
**Since we are not interested in the ACTUAL calculated order,           *
*         we only check the borders and return inside or outside         *
**        Lower border                                                  **
*         tmpdb = 0.499;                                                 *
*         ordarr[0] = *linfit(timeval,conc,tmpdb,no_of_points);          *
*         tmpdb = 0.5;                                                   *
*         ordarr[1] = *linfit(timeval,conc,tmpdb,no_of_points);          *
*         if (ordarr[0].SSR < ordarr[1].SSR                              *
*                                    return (ordarr[0].order);           *
*                                                                        *
* *       Upper border *                                                 *
*         tmpdb = 1.499;                                                 *
*         ordarr[2] = *linfit(timeval,conc,tmpdb,no_of_points);          *
*         tmpdb = 1.5;                                                   *
*         ordarr[3] = *linfit(timeval,conc,tmpdb,no_of_points);          *
*         if (ordarr[3].SSR < ordarr[2].SSR                              *
*                                    return (ordarr[3].order);           *
*                                                                        *
* *       If not returned, best-fit value is OK*                         *
*         tmpdb = 1.0;return(tmpdb);                                     *
*                                                                        *
\**************************************************************************/
  }



/**************************************************************************/
float gausdev(int *idum)
      /*   The function returns a random, normally distributed number with a standard deviation = 1
           When idum < 0 the number-generator is reset */
  {        /* See Press et al., 1988 */
  }
/**************************************************************************/
main()
  {  int i,j,nopt,seed,orgseed,NoERR;
     double deltaorder,bestorder[NUMSIM];
     double SDanal,Cmn,minTIME;
```

```
double timeval[MAX - PKT],conc[MAX - PKT],errconc[MAX - PKT];
struct parset midl;

/*INITIALIZING*/
deltaorder = 1.0;
/*   Co fixed to 100 (%) and ratek to 1 */
midl.ratek = 1.0;
midl.Co = 100.0;
midl.order = 1.0;midl.SSR = 0.0;

printf("\nGive SD for the analysis in");putchar('%');
printf(" of Co:");scanf("%lf",&SDanal);
SDanal = SDanal*midl.Co/100.0;
printf("\nGive a SEED:");scanf("%d",&orgseed);
if (orgseed > 0) orgseed* = -1; if (orgseed = = 0) orgseed = -1;
/*original seed must be negative in order to restart the random series*/

printf("\nNo of sim:500,seed:%d",orgseed);
printf("\nStandard deviation for analysis:%f",SDanal);
printf("\n\n The reaction must be followed");
printf(" to the given percent REMAINING");
printf("\n to determine %2.0f.order",midl.order);
printf("reactions with +/- %3.1f accuracy:",deltaorder);
printf("\nno of points          Cmn          ");

Cmn = 0.0;
for (nopt = 6;nopt < MAX - PKT;nopt + + )
  {   NoERR = 0;
      if (Cmn > 5)Cmn- = 5;
      /*To reduce the effect of erroneous high Cmn levels*/
      while (NoERR < NUMSIM/20)          /*Max. 5% ERRORS*/
        {   Cmn + = midl.Co/100.0;
/*************************************************************\
\* Equally CONCENTRATION-spaced samples:                      */
            for(i = 0;i < nopt;i + + )
                        /*Initialize conc and timeval*/
             {   conc[i] = midl.Co-((midl.Co-Cmn)/(nopt-1)*i);
                 timeval[i] = time - calc(&midl,conc[i]);
             }
/*************************************************************\
** For equally TIME-spaced samples                           *
*                        replace the source above with:      **
*            minTIME = time - calc(&midl,Cmn);               *
*                        **Calculate time for last sample**  *
*            for(i = 0;i < nopt;i + + )*Initialize conc and timevalue *
*              { timeval[i] = minTIME/(nopt-1)*i;            *
*                conc[i] = conc - calc(&midl,timeval[i]);    *
*              }                                             *
```

```
\ ********************************************************************/
                NoERR = 0;
                for(j = 0;j < NUMSIM;j + + )
                  { if (j = = 0) seed = orgseed;
                                  /*Initialize random number*/
                    /* Introduce a normally distributed error in the calculated conc.*/
                    for (i = 0;i < nopt;i + + )
                      {errconc[i] = conc[i] + (gausdev(&seed)*SDanal);
                        if (errconc[i] < = 0.0) errconc[i] = DBL - MIN;
                      }
                    bestorder[j] = ordcalc(timeval,errconc,nopt);
                    if((bestorder[j] < 0.5)||(bestorder[j] > = 1.5))
                        NoERR + +;                              /*Count errors*/
                  }
                printf(".");
            }; /*end while NoERR < 5%*/
        Cmn- = 1.0;
            /*NoERR too large, report previous value for Cmn*/
        printf("\n %2i %7.0f",nopt,Cmn);
    }; /*Cmn evaluated for nopt from 6..MAX - PKT*/
    exit(0);
}
```

References

Moore, D., Principles and practice of drug photodegradation studies *J. Pharm. Biomed. Anal.*, 5 (1987) 441–453.

Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T., *Numerical Recipes in C*, Cambridge University Press, Cambridge, 1988, pp. 204–228.

Sande, S.A. and Karlsen, J., Curve fitting of stability data by personal computer. Software in pharmaceutics: II. *Int. J. Pharm.*, 73 (1991) 147–156.

Taylor, R.B. and Shivji, A.S.H., A critical appraisal of drug stability testing methods. *Pharm. Res.*, 4 (1987) 177–180.

Yang, W., Errors in the estimation of the activation energy and the projected shelf life in employing an incorrect kinetic order in an accelerated stability test. *Drug. Dev. Ind. Pharm.*, 7 (1981) 717–738.