



Combining simulation to nonlinear fitting for the determination of rate constants of elementary reactions by using Excel spreadsheets

Ammar M. Tighezza*, Daifallah M. Aldhayan and Nouir A. Aldawsari

Chemistry Department, College of Sciences, Riyadh, Kingdom of Saudi Arabia.

Received 20 November 2011, Revised 20 June 2012, Accepted 27 June 2012, Available online 29 August 2012

ABSTRACT

A common problem in chemistry is to determine parameters (constants) in an equation used to represent experimental data. Examples are fitting a set of data to a model equation (straight line or curve) to obtain unknown parameters. In chemical kinetics, a set of data is usually a number of concentrations versus time, but the model equation is not well defined! Instead of a well defined model equation we have a set of coupled ODE's (ordinary differential equations) which represent rate equations for reactants and products. The analytical integration of these ODE's is rarely possible. The numerical integration is the alternative. In this work are combined the simulation of chemical reactions, by using numerical integration, and nonlinear regression (curve fitting) by using "Solver add-in" of Microsoft Excel to find rate constants of elementary reactions from experimental data. This method is illustrated on three complex mechanisms. The simulation of chemical reactions in Excel spreadsheets is illustrated with/without VBA programming. The automation (automatic obtaining of rate equations from mechanism: no need of chemical kinetics knowledge from the end user!) of mechanism simulation is demonstrated on many example.

| Numerical integration | Coupled ODE's | Nonlinear regression | Complex mechanisms | Chemical kinetics |

© 2012 Ibnu Sina Institute. All rights reserved.
<http://dx.doi.org/10.11113/mjfas.v8n4.165>

1. INTRODUCTION

As scientists we are frequently fitting experimental data to models by using regression analysis. Linear regression is familiar to us. An equation of the form $y = mx + b$ is to be fitted to raw data and the parameters m (slope of the line) and b (the intercept) are to be deduced. The drawn line is chosen to minimise the sum of the squares of vertical distances of the points from that line (least squares method). Linear regression assumes Gaussian distribution of the points and same standard deviation at any x value [1]. These assumption are usually not valid when performing linear regression on transformed data[2]. Non linear data must not be linearized for the regression analysis purpose. Many data are best analysed by using nonlinear regression and many software are available to do it [3]. Solver add-in function of Microsoft Excel is readily available for each scientist to do this nonlinear analysis.

The regression analysis consists in modelling relationship between a dependent variable and one or more independent variables. In chemical kinetics the dependent variable is usually the concentration of any reactant or product of the reaction. The independent variable is the time. The rate constants of elementary steps constituting the mechanism of a complex reaction are the parameters to be determined.

A direct relationship between concentration and rate constants is rarely known for complex reactions. Only concentration derivatives (rates) are given as function of rate constants. Most often, the analytical integration of these ordinary differential equations (ODEs) is complicated or even intractable. Obtaining concentration goes through numerical integration, but for some simple mechanisms, analytical solutions could be found in the work of Andraos [4]. Most of chemical integration methods fall in two categories: deterministic [5] or stochastic [6]. In this work, we illustrate the use of Microsoft Excel sheet to simulate chemical reactions with a deterministic method and without writing any code, just by using standard functions of Excel. The use of stochastic methods in simulating chemical kinetics could be found in our earlier work [7,8]. The performance of simulation is, of course, greatly improved by using VBA (Visual Basic for Application) code as will be shown. "Solver" is a powerful tool in the Microsoft Excel spreadsheet that is increasingly used in fitting experimental data to nonlinear functions [9]. The estimation of chemical kinetic parameters, namely the rate constants, by combining simulation with the minimizing function of solver add-in is illustrated. A friendly user interface is used for the mechanism input and a set of ODEs is automatically generated by analysing it. The big advantage of this Excel application is that final user must not be neither genius mathematician nor luminous chemist to use it, and therefore it is suitable for beginners and education purpose.

Corresponding author at:

E-mail addresses: ammr@ksu.edu.sa (Ammr Tighezza)

2. MODELING EXPERIMENTAL DATA

2.1 Defining the model

The main point in fitting experimental data to a given model is well defining this model. The model could be either a linear or nonlinear relationship between recorded data. As example of linear equation, we can give the relationship between absorbance and concentration (Beer's law) from which we can deduce the molar absorptivity ϵ of a studied substance: $A = \epsilon b c$. Where A is the absorbance, c the concentration and b the cell length. A nonlinear example could be the Arrhenius equation relating rate constant, k , to temperature T : $k = A \cdot \exp(-Ea/RT)$. The parameters to determine are the pre-exponential factor, A , and the activation energy, Ea , and R is the universal constant of gases. This relationship is usually linearized by using the natural logarithm but should not as we said in the introduction. The change in retention time t_r (in gas chromatography) versus the number of carbon Z in a series of organic compounds, namely $t_r = t_m + A \exp(BZ)$, is an example of non linearizable relationship in which t_m , A and B are parameters to determine.

2.2 The model in chemical kinetics

In chemical kinetics, the model to fit is a set of N elementary reactions with S species. A set of S ordinary differential equations (ODEs) is obtained by writing the global rate for each species:

$$dy_i/dt = f_i(t, y_1, y_2, y_3, \dots, y_S, k_1, k_2, k_3, \dots, k_N) \quad (1)$$

y_i ($i = 1$ to S) is the concentration of the species i and k_j ($j = 1$ to N) is the rate constant of the step j and t is the time. The experimental data collected in chemical kinetics are usually concentrations versus time. Then we need to solve the set of ODEs to get "calculated concentrations" that could be fitted to the experimental ones. The simplest solver for these ODEs is the Euler based algorithm. Many more complicated and more accurate solvers are known [9]. For clarity purpose, the Euler method is used in this work. In this method the instantaneous rate at some instant t_1 is considered as equal to the average rate between this instant and a close instant t_2 :

$$dy_i/dt \approx \Delta y_i / \Delta t = [(y_i)_2 - (y_i)_1] / [t_2 - t_1] \quad (2)$$

Obviously, the accuracy of this numerical integration is related to the difference ($t_2 - t_1$): more t_2 is close to t_1 more is accurate the integration. This difference is said "step of integration" and is usually given a particular symbol h .

$$(y_i)_2 = (y_i)_1 + h f_i(t_1) \quad (3)$$

The use of this equation (3) is illustrated on the two steps, three species mechanism (M1) which can be

easily identified with many chemical and biochemical reactions:

From equation (2) we can find $(y_i)_2$ as function of $(y_i)_1$



The deployment of equation (3) for the species A , B , and C gives (k_{-1} is the rate constant for the backward reaction in step 1):

$$[A]_2 = [A]_1 + h(k_{-1}[B]_1 - k_1[A]_1) \quad (3.a)$$

$$[B]_2 = [B]_1 + h(k_1[A]_1 - k_{-1}[B]_1 - k_2[B]_1) \quad (3.b)$$

$$[C]_2 = [C]_1 + h(k_2[B]_1) \quad (3.c)$$

These three equations allow us to simulate the given mechanism starting with some given initial conditions. In the next section we illustrate how to do it easily in Microsoft Excel spreadsheet.

3. USING EXCEL IN SIMULATION

3.1 Without using VBA code

Figure (1) shows the chosen values of rate constants (rows 1 to 3), the step of integration (row 4) and the initial conditions (row 8) for the simulation of the above mechanism. For clarity purpose, cells containing the values of k_1 , k_{-1} , k_2 and h are **named** respectively *k1f*, *k1r*, *k_2* and *step*. In cells A7, B7, C7 and D7 are entered the labels: "time", [A], [B] and [C] respectively. In row 9 are entered the following formulas:

in cell A9: $= A8 + step$

in cell B9: $= B8 + step * (k1r * C8 - k1f * B8)$

in cell C9: $= C8 + step * (k1f * C8 - k1r * B8 - k_2 * C8)$

in cell D9: $= D8 + step * k_2 * C8$

The first formula (in cell A9) is the formula for increasing time as function of step of integration ($step = h$) and the three others are the translation of equations (3.a), (3.b) and (3.c) in an Excel acceptable language.

Once formulas entered in cells A9 to D9, these cells are selected and a "fill handle" (a small black square in the lower-right corner of the selection) is dragged down until the needed time is reached. The simulated concentrations appear instantaneously after releasing the "fill handle".

As we can see in figure 2, the time is increased by a small steps equal to the chosen step of integration and we need to drag down the "fill handle" more than 100 rows to reach the time 1.0 s. If we need to reach this time (1.0 s) after 10 rows only we can change the value of h (the step of integration) from 0.01 s to 0.1 s, but this change will reduce the accuracy of the simulation. The other way to get the same result without reducing the accuracy is to write a

small program (in VBA language) and we assign it to some *user defined* function or start it via a command button that

step				f_x
	A	B	C	D
1	$k_1 =$	0.5	s^{-1}	
2	$k_{-1} =$	0.6	s^{-1}	
3	$k_2 =$	0.7	s^{-1}	
4	$h =$	0.01	s	
5				
6				
7	time	A	B	C
8	0	1	0	0

Fig. 1 Initial conditions and rate constants for the simulation of the mechanism (M1).

we insert on the spreadsheet as explained in the next section.

6				
7	time	A	B	C
8	0	1	0	0
9	0.01	0.9950	0.0050	0.0000
10	0.02	0.9901	0.0099	0.0000
11	0.03	0.9852	0.0147	0.0001
12	0.04	0.9803	0.0195	0.0002
13	0.05	0.9755	0.0241	0.0003
14	0.06	0.9708	0.0287	0.0005
15	0.07	0.9661	0.0332	0.0007
16	0.08	0.9615	0.0376	0.0009
17	0.09	0.9569	0.0419	0.0012
18	0.1	0.9524	0.0461	0.0015

Fig. 2 Simulation results of the mechanism (M1) without using VBA.

3.2 Using VBA code in simulation

The listing of this small program is showed in figure 3. It uses two subroutines, the first one (*read_data*) to read data from the spreadsheet and the second one (*write_results*) to write the simulation results on the spreadsheet. In data section, as we can see on figure 4, are added two new variables: n and t_f , the first one is the number of loops to do before writing results (10 in this case) and the second is the final time (7 is chosen here) at which will be stopped the simulation. We can see also the "Start Simulation" command button on which we have to click to start the simulation. The time is now increasing by a 0.1 s step, although the step of integration is 0.01 s and we get 1.0 s in the 10th row and the final time, 7.0 s, is reached after 70 rows. The limited number of rows in simulation make the graph presentation of results easiest. By comparing concentrations obtained at time 0.1 s (row 9) in figure 4 to those obtained at the same time in figure 2 (row 18), we can see that they are identical and this means that the accuracy is the same.

3.3 Automation of the simulation

An Excel application is developed by defining a menu sheet, mechanism sheet and results sheet. From the menu sheet we can choose to enter a new mechanism or simulate the entered one. The matrix sheet is used for visualizing the content of the stoichiometry matrix and other related matrices. The stoichiometry matrix is automatically generated by the analysis of the entered mechanism. The simulation is facilitated by the use of the matrix multiplication function (MMULT). The user interface for the mechanism input, shown in figure 5, give an easy way to enter a mechanism without constraints on the format: any symbol can be used for reactants and products and should be preceded by its stoichiometric coefficient if not equal to

one. The number of steps and the number of species included in the mechanism are deduced automatically when we click on the "Finish" button. The mechanism is presented on the sheet labelled "Mechanism" for reviewing purpose. The results of simulation are presented in the sheet labelled "Simulation by VBA" and are compared to the manually obtained results in the "Simulation in sheet" sheet. This last sheet is used only for validation purpose. From the "menu" sheet we can start the simulation by clicking on the "simulation" button. The step of integration, the final time, and the initial concentrations are entered via "input boxes". The simulation program is constituted by three subroutines: the *main* sub, the *initialization* sub and the *simulation* sub. In the *initialization* sub are created the necessary matrices needed in *simulation* sub. The VBA source code of this application is available upon request.

4. FITTING CHEMICAL KINETICS EXPERIMENTAL DATA

The main goal of chemical kinetics experiments is to establish a mechanism of the studied reaction. Deducing the unknown rate constants of some (or all) elementary steps is an important task in establishing a mechanism. This task is facilitated by the availability of software allowing nonlinear regression. "Solver" is a powerful tool in the Microsoft Excel spreadsheet that make this task very easy as will be shown in the next sections.

4.1 Deducing rate constants of mechanism

We modified the simulation results obtained in section 3.2 by adding a random noise at 1% of the maximum concentration to make them like experimental data. We retained only 18 points at variable time intervals (as in real experiments) from the 70 points simulated. Then

we supposed that rate constants (k_1 , k_{-1} and k_2) are unknown and must be determined. We defined three "user defined" functions with 8 arguments among k_1 , k_{-1} and k_2 that will be varied by the solver in the fitting process. The listing of one of these functions is given in figure 6. We used these 3 functions to calculate the concentration of A, B, and C in mechanism M1 with some guessed values of the rate constants ($k_1 = 2 \text{ s}^{-1}$, $k_{-1} = 3 \text{ s}^{-1}$, and $k_2 = 4 \text{ s}^{-1}$). The standard function of Excel "SUMXMY2" is used to calculate the residuals (sum of squared differences between simulated and "experimental" concentrations) of A, B and C. In cell **J28** is inserted the sum these 3 residuals ($A_{\text{resid}} + B_{\text{resid}} + C_{\text{resid}}$) and will be the **target cell** for Solver. In the solver

options window, we entered as target cell J28 and in **cells to modify** the cells containing the values of k_1 , k_{-1} and k_2 , namely \$F\$1, \$F\$2 and \$F\$3, as shown in figure 7. We selected the "Min" option and started the minimization process by clicking on "Solve". Figure 8 shows the profiles of simulated and "experimental" concentrations: (a) before executing solver; (b) after executing solver. The new values of k_1 , k_{-1} and k_2 after the fitting process are respectively 0.4866 s^{-1} , 0.5676 s^{-1} and 0.6857 s^{-1} . As we can see, the average of relative differences between true values (0.5 s^{-1} , 0.6 s^{-1} and 0.7 s^{-1}) and the calculated values is less than 4%.

```

Dim A, B, C
Dim k1f, k1r, k2, h, tf
Dim n, counter
Dim t, A0
Sub read_data()
    k1f = Sheet2.Cells(1, 2)
    k1r = Sheet2.Cells(2, 2)
    k2 = Sheet2.Cells(3, 2).Value
    h = Sheet2.Cells(4, 2)
    n = Sheet2.Cells(5, 2)
    tf = Sheet2.Cells(6, 2)
    A0 = Sheet2.Cells(8, 2)
End Sub
Sub Simulation()
    read_data
    A = A0: t = 0
    While t < tf
        For i = 1 To n
            A = A0 + h * (k1f * B0 - k1r * A0)
            B = B0 + h * (k1f * A0 - k1r * B0 - k2 * B0)
            C = C0 + h * k2 * B0
            A0 = A: B0 = B: C0 = C
        Next i
        counter = counter + 1
        t = t + n * h
        write_results
    Wend
End Sub
Sub write_results()
    Sheet2.Cells(8 + counter, 1) = t
    Sheet2.Cells(8 + counter, 2) = A
    Sheet2.Cells(8 + counter, 3) = B
    Sheet2.Cells(8 + counter, 4) = C
End Sub

```

Fig. 3 VBA code of the program used to improve the simulation.

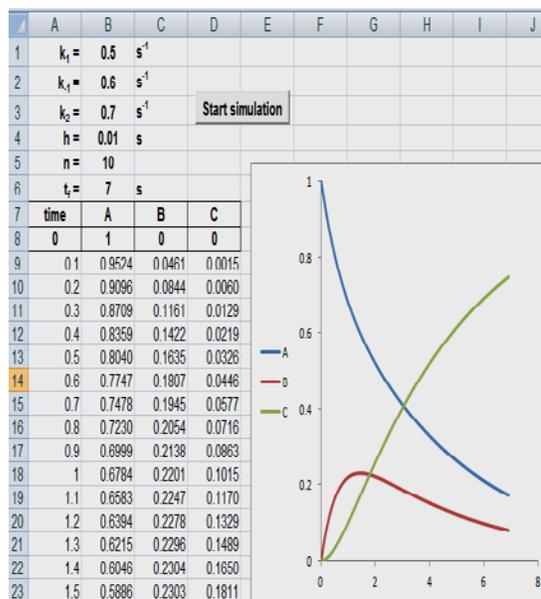


Fig. 4 Simulation results (table and graph) for the mechanism M1 when using VBA code.

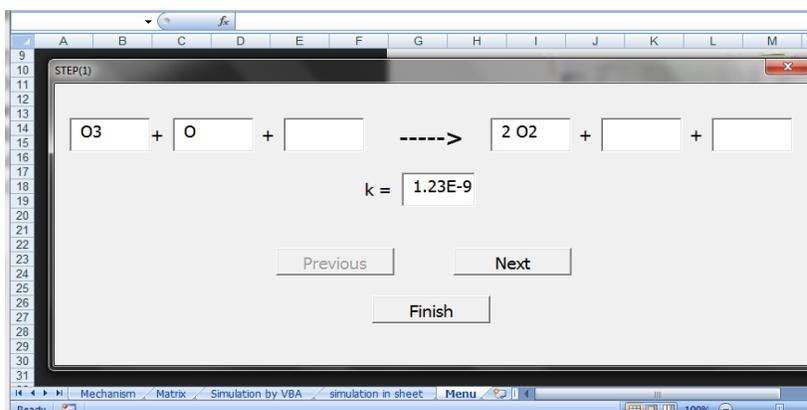


Fig. 5 Mechanism input window in the Excel spreadsheet application.

```

Function small_Steps_A(t1, t2, A1, B1, C1, k1f, k1r, k2)
  h = Sheet3.Cells(4, 2)
  A = A1: B = B1: C = C1: t = t1

  While t < t2
    A = A1 + h * (k1r * B1 - k1f * A1)
    B = B1 + h * (k1f * A1 - k1r * B1 - k2 * B1)
    C = C1 + h * k2 * B1
    A1 = A: B1 = B: C1 = C
    t = t + h
  Wend

  small_Steps_A = A
End Function

```

Fig. 6 The function calculating the concentration of A.

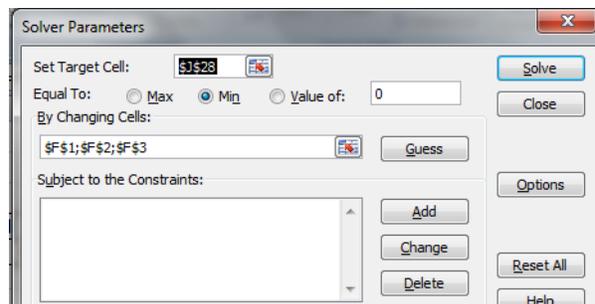


Fig. 7 The solver options window.

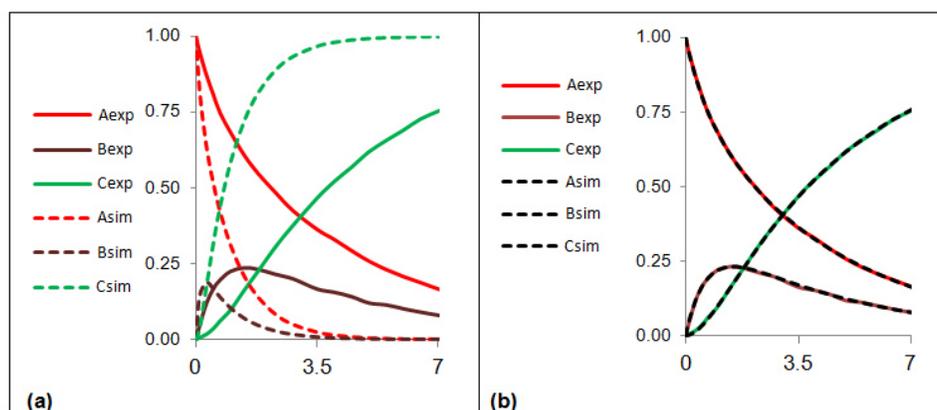


Fig. 8 Profiles of simulated and experimental concentrations: (a) before executing solver; (b) after executing solver.

4.2 More complex mechanisms

The fitting procedure is tested on more complex mechanisms with more than 5 steps and 7 species with success. In some cases, some of rate constants are considered as known and only the unknown ones are changed by the solver. For reversible steps with known equilibrium constant, we can add in the solver options window a constraint relating the forward constant to the reverse constant via the equilibrium constant ($K = k_f / k_r$).

5. CONCLUSION

We showed in this work the importance of the MS Excel Solver add-in in non-linear regression analysis. It's readily available for each scientist and easy to use. In case of chemical kinetics, we showed that we can easily simulate any complex reaction in Excel spreadsheet by using Euler method of integration. This simulation is combined to the minimization function of Solver to deduce unknown rate constants. An Excel spreadsheet application for automation of the simulation of chemical kinetics is illustrated. Our next objective is to automate the fitting process by invoking the Solver from simulation code.

ACKNOWLEDGEMENT

The authors greatly acknowledge the support of the College of Science Research Center, Deanship of Scientific Research and King Saud University for the unlimited support for project.

REFERENCES

- [1] Xin Yan and Xiao Gang Su, Linear Regression Analysis: Theory and Computing, World Scientific Publishing, 2009.
- [2] Motulsky H. J.; Ransnas L. A., FASAB J, 1 (1987), 365-374.
- [3] Zmostni P. and Belohlav Z., Computers & Chemistry, 23 (1999), 479-485.
- [4] Andraos J., Journal of Chemical Education, 76 (1999), 1578-83.
- [5] Olcese L., Toselli B., International Journal of Chemical Kinetics, 30 (1998), 349-58.
- [6] Gillespie D., Annual review of physical chemistry, 58 (2007), 35-55.
- [7] Tighezza, A., Aldhayan, D., Rezgui, Y., Alarifi, A., Asian J. Chem. 17 (2005), 840-846.
- [8] Tighezza, A., Aldhayan and Almthar, A., Journal of Saudi Chemical Society, 15 (2011), 351-355.
- [9] Harris, D. C., Journal of Chemical Education, 75 (1998), 119-121.
- [10] Noskov, M and Smooke, M, Journal of Computational Physics, 203 (2005), 700-730.