On the Use of Excel 2000's Solver and a Differential Equation Routine

by

Edward M. Rosen

EMR Technology Group

Introduction

Excel 2000's Solver program can be a useful means for studying a system of differential equations. The program has the advantage of being embedded into Excel's spreadsheet, and as a result all the features of Excel may be exploited.

It is the purpose of this communication to illustrate the use of Solver (a nonlinear programming routine) on a simple kinetics system. Use is made of a 4^{th} order Runge Kutta routine (1) written in VBA (Visual Basic for Applications) to carry out the integration of the differential equations.

The Reaction System

Consider the classical consecutive reactions (2) in a batch reactor:

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C$$

and

$$-dA/dt = k_1 A$$

$$dB/dt = k_1 A_0 \exp(-k_1 t) - k_2 B$$

$$dC/dt = -dA/dt - dB/dt$$

A plot for $k_1 = 3$ and $k_2/k1 = 1/3$ with $A_0 = 1$, $B_0 = 0$ and $C_0 = 0$ is shown in Figure 1.

The VBA Coding

Figure 2 is a listing of the VBA coding (macro module) that can be used to carry out the integration and transfer back values that can be monitored and manipulated by the Solver program.
The Integ **array function** (3) has the following arguments:

h - The step size to be used by the integration routine
tf - The final time for the integration
y1, y2, y3 - The initial values of the dependent variables
prm - An array of parameters to be passed to the integration routine to construct the right hand side of the differential equations in function RHS.

The function returns the the values in the GGG array (as a row vector) to an area of the spreadsheet which has been highlighted. The function is typed into the formula bar and entered with **Cntrl+Shift+Enter**.

The actual integration is carried out one step at a time with the function Rk4y. Note that the new values for the arguments are returned after each step of the integration. In the Integ array function the values of each step are monitored so that the maximum value and time of the (B) dependent variable can be determined.

Use of Solver

Solver is an *add-in* program that's attached to Excel to give it additional functionality. To attach an *add-in* use Tools:Add-Ins command.

The Solver routine requires a target cell (the objective function). In addition constraints of various types can be specified by entering the cells to which the constraints apply. It is generally a good idea to examine the options. Using automatic scaling and showing iteration results is often good practice.

The Solver routine can be invoked by pulling it down from the tools menu. Figure 3 is a listing of the spreadsheet used to manipulate the cell containing the ratio of k1/k2 (cell B7) until the maximum value of B is equal to 0.8 (Cell G22 which is the target cell).. Note that the theoretical value of the maximum value of B_{max}/A_o is given by

$$B_{max}/A_o = (k1/k2)^{k2/(k2-k1)}$$

It occurs at the time

$$t_{max\ B} = \ln (k2/k1)/(k2-k1)$$

The spreadsheet indicates the theoretical value of the time at which Bmax/Ao occurs as well as the value of B_{max}/A_o. It can be compared with the values retuned from the integration.

	B_{max}/A_o	Corresponding Time
Theoretical	0.80	0.9082
Calculated	0.80	0.9075

Making the step size smaller than the 0.0075 which was used improves the comparison of the calculated values and the theoretical values.

Conclusions

The Solver routine found in Excel 2000's spreadsheet can used with an integration routine to perform optimization tasks based on the integration. The integration routine used in this study (Rk4y) could be replaced by one written in FORTRAN as well (4,5).

References

1. Rosen, Edward M. “Integrating Differential Equations Using Excel 7.0”
CACHE News Fall 1998

2. Carberry, James J. *Chemical and Reaction Engineering* McGraw-Hill
New York (1976)

3. Walkenbach, John *Excel 2000 Power Programming with VBA*
IDG Books, Foster City , CA (1999)

4. Rosen, Edward M. “Calling FORTRAN Subroutines from Excel 7.0”
CACHE News No 48 Spring 1999

5. Rosen, Edward M. “On the Use of IMSL Routines in Excel 7.0
CACHE News No 49 Fall 1999

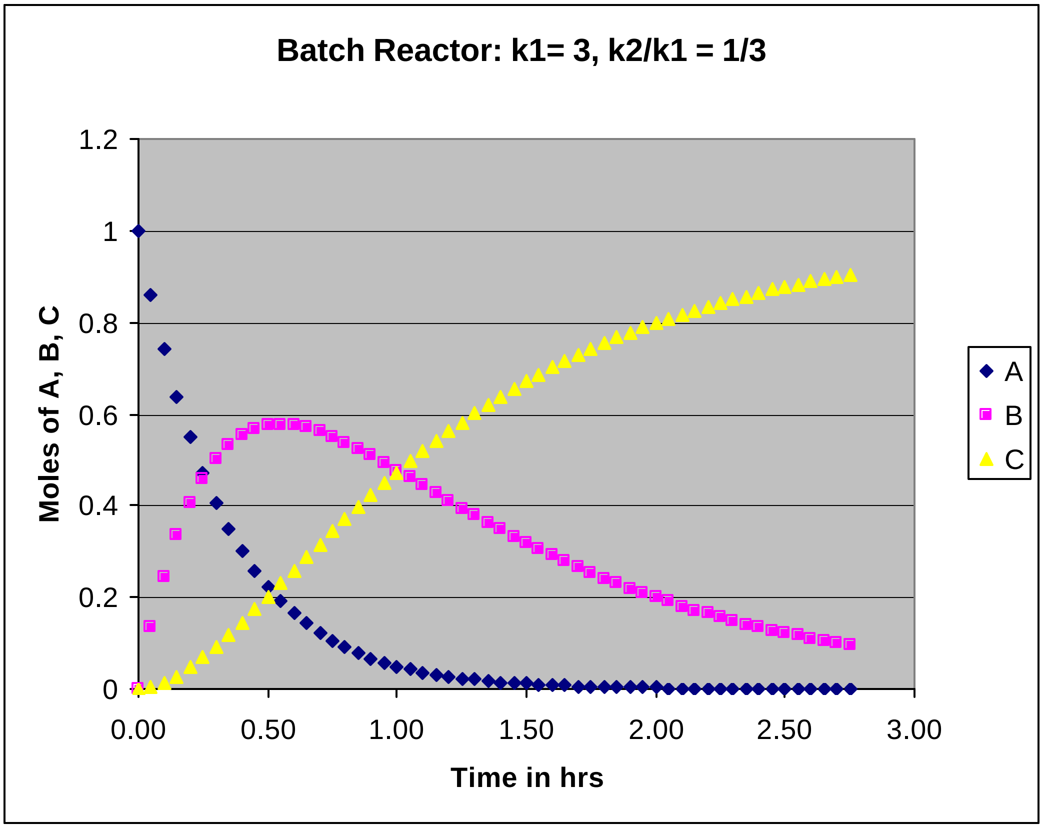

Figure 1 Plot of Moles vs. Time

```
Public Function Integ(h, tf, y1, y2, y3, prm)


Dim DDD(1 To 7)
Dim GGG(1 To 9)
Dim I As Integer
Dim Ntime As Integer
Dim Irtn As Integer
Dim hh As Single
Dim xx As Single
Dim yy1, yy2, yy3, yy4, yy5 As Single
Dim my1, my2, my3, tmy2 As Single

Ntime = tf / h + 0.00001

hh = h
xx = 0
yy1 = y1
yy2 = y2
yy3 = y3
yy4 = y4
yy5 = y5

'  Integrate and Obtain maximum Values of first three variables

my1 = y1
my2 = y2
my3 = y3
For I = 1 To Ntime
Irtn = Rk4y(hh, xx, yy1, yy2, yy3, yy4, yy5, prm)

If yy1 > my1 Then my1 = yy1
If yy2 > my2 Then tmy2 = xx
If yy2 > my2 Then my2 = yy2
If yy3 > my3 Then my3 = yy3

Next I

GGG(1) = hh
GGG(2) = xx
GGG(3) = yy1
GGG(4) = yy2
GGG(5) = yy3
GGG(6) = my1
GGG(7) = my2
GGG(8) = my3
GGG(9) = tmy2

Integ = GGG


End Function
```

```
Public Function Rk4y(h, x, y1, y2, y3, y4, y5, prm)

'Written by EMRosen   10/30/97
'Copyright (c)  EMR Technology Group


'h  = step size
'x = independent variable
'y1, y2, y3, y4, y5 = dependent variables

'prm a parameter vector of unspecied length
'prm(1) = Number of Equations Being Integrated


'kij  :  i   is the k value,  j is the equation number or dependent variable

'Output is the Seven Element Vector
'Actual Output must be pre-selected

Application.Volatile

Dim DDD(1 To 7)

Dim fff(1 To 5)

Neq = prm(1)


Rtrn1 = Rhs(x, y1, y2, y3, y4, y5, prm, fff)

k11 = fff(1)
k12 = fff(2)
k13 = fff(3)
k14 = fff(4)
k15 = fff(5)

Rtrn2 = Rhs(x + 0.5 * h, y1 + 0.5 * h * k11, y2 + 0.5 * h * k12, y3 + 0.5 * h * k13, y4 + 0.5 * h * k14, y5
+ 0.5 * h * k15, prm, fff)


k21 = fff(1)
k22 = fff(2)
k23 = fff(3)
k24 = fff(4)
k25 = fff(5)
```

```
Rtrn3 = Rhs(x + 0.5 * h, y1 + 0.5 * h * k21, y2 + 0.5 * h * k22, y3 + 0.5 * h * k23, y4 + 0.5 * h * k24, y5
+ 0.5 * h * k25, prm, fff)

k31 = fff(1)
k32 = fff(2)
k33 = fff(3)
k34 = fff(4)
k35 = fff(5)

Rtrn4 = Rhs(x + h, y1 + h * k31, y2 + h * k32, y3 + h * k33, y4 + h * k34, y5 + h * k35, prm, fff)

k41 = fff(1)
k42 = fff(2)
k43 = fff(3)
k44 = fff(4)
k45 = fff(5)

'Values of  new h and new x

    DDD(1) = h

    DDD(2) = x + h

'First Independent Variable

      DDD(3) = y1 + (h / 6) * (k11 + 2 * k21 + 2 * k31 + k41)

 'Second Independent Variable

      DDD(4) = y2 + (h / 6) * (k12 + 2 * k22 + 2 * k32 + k42)

 'Third Independent Variable

     DDD(5) = y3 + (h / 6) * (k13 + 2 * k23 + 2 * k33 + k43)

'Fourth Independent variable
      DDD(6) = y4 + (h / 6) * (k14 + 2 * k24 + 2 * k34 + k44)

 'Fifth Independent variable

      DDD(7) = y5 + (h / 6) * (k15 + 2 * k25 + 2 * k35 + k45)

  h = DDD(1)
  x = DDD(2)
  y1 = DDD(3)
  y2 = DDD(4)
  y3 = DDD(5)
  y4 = DDD(6)
  y5 = DDD(7)

  Rk4y = 0

  End Function
```

```
Public Function Rhs(x, y1, y2, y3, y4, y5, prm, fff)

Dim aa, rr, ss, k1, k2 As Single

' y1 is A , y2 is B ,  y3 is C
'
'prm (1) Neq
'prm (2) k1
'prm (3) Ratio k2/k1
'prm (4) Nao



 aa = y1
 bb = y2
 cc = y3
 xx = x
 k1 = prm(2)
 k2 = k1 * prm(3)

 fff(1) = -k1 * aa
 fff(2) = -k2 * bb + k1 * prm(4) * Exp(-k1 * xx)
 fff(3) = -fff(1) - fff(2)
 fff(4) = 0
 fff(5) = 0

  Rhs = 0

 End Function
```

Figure 2   VBA Coding of Functions Integ and Rk4y

Batch Reactor from Carberry - Chemical and Catalytic Reaction Engineering (1976)

prm array

prm(1)	Number of Equations	3	
prm(2)	k1	3	
prm(3)	Ratio k2/k1	0.081861359	<---Cell B7
prm(4)	Ao	1	

Intial Time	0
Time Increment (hrs)	0.0075
Final Time (hrs)	2.7
Intial A	1
Initial B	0
Initial C	0

Increment (hrs)	Time (hrs)	A	B	C	A max	B max	C max	t of Bmax/Ao
0.0075	2.70	0.000303539	0.560873921	0.43882254	1	0.8000	0.4388	0.9075
						Cell G22 - Up One		

Theoretical B max/Ao	0.8000
Theoretical Time of Bmax/Ao	0.9086

Figure 3 Spreadsheet Using Solver