Closing the Gap between Numerical Software Package and Spreadsheet Users in Process Computations

Mordechai Shacham, Michael B. Cutlip, and Michael Elly

Dept. of Chemical Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel /Dept. of Chemical Engineering, University of Connecticut, Storrs, CT 06269, USA / Intel Corp., Qiryat Gat, Israel and Portland, OR

Abstract

Recent surveys have found that most practicing chemical engineers do not take advantage of sophisticated numerical methods for their computational needs which may require the solution of systems of nonlinear algebraic equations and/or the integration of systems of ordinary differential equations. This is due in part to the use of spreadsheet programs which are most commonly used in the industry for problem solving. In the spreadsheets, the application of numerical methods often requires substantial modification and coding of the model equations which is difficult and challenging in the programming environment that is available. Additionally, there is no easily-used differential equation solver tool for use in spreadsheet programming.

The Polymath software package that we have authored has recently been upgraded so that automatic exporting of a problem to an Excel™ spreadsheet for solution in addition to problem solution within Polymath. An ordinary differential equation solver tool called ODE_Solver has also been developed as a separate Excel Add-In. In this paper we demonstrate the combined use of the tools that we have developed, and we then illustrate the use of Excel to solve fairly complex problems which include a three phase flash calculation (solution of a system of non-linear algebraic equations) and the simulation of a propylene oxide polymerization reactor (solution of a system of ordinary differential equations).

The automatic Excel export capability will be welcomed by practicing engineers for documenting their results and making them available as programs that run in Excel. In an industrial setting, Polymath can serve as a pseudo "compiler" that enables converting complex, realistic problems into Excel code while the results are obtained solely from within Excel. Since Polymath can also solve the same problems as those exported to Excel, there is the ability to verify the solutions found with these two separate software packages.

As this new version of Polymath has enhanced capabilities when compared to previous versions, it can also fulfill most of the numerical computational needs in undergraduate and graduate education. The main educational benefit of the Excel export capability is that the same techniques that are studied in the University can be applied in industry.

Introduction

A recent survey of practicing chemical engineers by Edgar[1] determined that about 99% use spreadsheet programs routinely; however, only about 25% use numerical or statistical software packages, and only about 40% use process simulation programs or do any programming. These findings can be interpreted to indicate that most chemical engineers do not take advantage of sophisticated numerical methods for their computational needs which may require solution of systems of nonlinear algebraic equations and/or the integration of systems of ordinary differential equations. The current use of such techniques with spreadsheets typically requires programming (with Visual Basic, for example) which is not done by most engineers according to the findings of the survey.

The main challenges, as we have identified them, to the construction and solution of mathematical models in spreadsheets by using advanced numerical techniques are the following:
1. The need to convert most variable names to cell addresses in the model equations.
2. The necessity to provide separate documentation to the model equations.
3. The lack of tools for easily solving differential equations.

Recently we have enhanced the Polymath (copyrighted by M. Shacham, M. B. Cutlip and M. Elly, http://www.polymath-software.com) software package so that it is able to automatically export a problem to an Excel (trademark of Microsoft Corporation, http://www.microsoft.com) spreadsheet. In contrast to Excel, Polymath accepts the model equations in a format that is very similar to their mathematical form with minimal change of the variable names. The Polymath equations, intrinsic functions, variable names, and comments are also exported to Excel so as to serve as documentation for the mathematical model. An ordinary differential equation solver tool called Ode_Solver has also been developed as a separate Excel Add-In. This separately functioning tool contains several stiff and non-stiff ODE solvers.

In this paper we demonstrate the combined use of the tools that we have developed, and we then illustrate the use of Excel to solve fairly complex problems which include solution of non-linear algebraic equations and ordinary differential equations.

Example 1 – Non-Ideal Phase Equilibrium Computation

This problem specification, which involves calculation of the bubble point with two liquid phases and one vapor phase, is shown in Appendix A. The equations, as entered into the Polymath program, are shown in Table 1. It is worth noting some of the "user friendly" features of Polymath which are helpful while entering and debugging the problem: The notation used in the equation entry is almost the same as in the problem definition (except that no Greek characters can be used). Polymath issues warnings for undefined variables so that errors such as using the number '1' in the variable name (like in x11) in one equation and the letter 'l' in another equation can be easily detected. The needed equations can be entered in the same order as they appear in the problem definition even if the calculation order must be different since Polymath reorders the equations when the calculations are made. For example, the necessary calculation for the vapor liquid equilibrium ratios (k_{ij}), first involves the vapor pressures and the activity coefficients. However Polymath allows direct entry of these equations. In addition, Polymath will not accept equations until the syntax is correct. Those features speed up considerably the technical tasks of entering and debugging the model equations.

The Polymath solution of the problem can be used to validate the correctness of the model equations and provide a test solution. After the validation step, the entire problem can be converted and exported to an Excel worksheet using a single command within Polymath. Part of the generated Excel worksheet is shown in Table 2. In the Excel worksheet, the variable names, the Polymath equations, and the comments are shown for documentation purposes. The Excel formulas are generated and placed in the 3rd column, marked as "Value". They are transparent to the user unless he/she asks explicitly to see them. The formula to calculate gamma11 for example is:

$= (10 \wedge ((C9 * C18 * C18) / (((C9 * C17) / C10 + C18) \wedge 2)))$

Direct Excel coding with formulas like this one is a very difficult and error prone process, and this can be done only for small-scale, simple problems. While in some cases the cell addresses can be replaced by variable names, this can limit the use of some convenient Excel options such as relative addressing. Thus, Polymath essentially serves as an efficient pseudo "compiler" that automatically converts an entire problem into equivalent Excel code.

This particular example involves the solution of six simultaneous nonlinear algebraic equations. The solution can be obtained within Excel using the "Solver" Add-In to minimize the "Sum of Squares" of the function values (see Table 2) while changing the values of the "Implicit Variables". The solution of this problem as obtained by the "Solver" is shown in Table 3.

Since the generated Excel solution involves only variable cell addresses, the initial solution may be easily copied to other locations in the spreadsheet for solution of additional cases with different parameter values.

Example 2 – Propylene Oxide Polymerization Reactor

Propylene oxide polymerization is a highly exothermic process that is carried out at high pressures. Nearly isothermal operation is required in order to prevent runaway conditions and the buildup of a pressure which is higher than the reactor's design pressure. Safety problems associated with the operation of such a reactor are described in Kneale and Foster[4]. Mathematical modeling and simulation of the reactor, which includes a bursting disk for pressure relieve in case of excessive pressure buildup, was carried out by Ingham et al[3].
They considered the manufacture of a polyol lubricant by step-wise condensation of propylene oxide with butanol:

$$C_4H_9OH + (n+1) \, C_3H_6O \rightarrow C_4H_9(OC_3H_6)nOCH_2CHOHCH_3 + \text{heat}$$
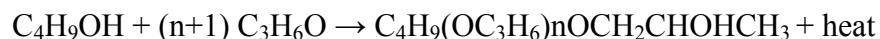
The catalyzed alcohol is initially charged into the reactor up to the "initial" level. The oxide is fed into the reactor at a constant rate until the batch is ready and the reactor is full. Excess heat of the reaction is removed via an external heat removal system. Economical considerations dictate that the reaction should be completed at the highest possible rate. The reaction rate is a function of the temperature, catalyst concentration, and the liquid phase oxide concentration (which is function of the pressure). The limits on the reactor temperature and catalyst concentration are set by considerations of thermal degradation and purification difficulties. To maximize the reaction rate, the pressure must be kept as high as possible for the entire duration of the batch. The higher limits on the pressure and reaction rate are dictated by the pressure suitability of the reactor system and the feasible heat removal rate.

The mathematical model of the reactor, the heat removal system, and the bursting-disk orifice as proposed by Ingham et al[3] were entered as a Polymath problem. The problem solution was first attained in Polymath, and the problem was then transferred to Excel, as shown in Table 4. The Excel worksheet represents a complete documentation of the mathematical model of the problem. The model equations, the constant values, the initial values of the variables and the output variable related to each of the model equations are defined and described. The model problem in this example includes four differential equations: mass balance in the reactor (yields the total mass, row/Eq. 31 in Table 4); component balance (yields the mass of the oxide component, Eq. 32); enthalpy balance (yields the temperature in the reactor, Eq. 33); and reaction rate (yields the mass of oxide reacted, Eq. 34). An additional differential equation was added to the model to represent the status of the bursting disk (open or closed, Eq. 35). Under normal operating conditions, reacting mass is being re-circulated through the external heat removal system at flow rate of F, and cooled to temperature T, (see Eqs. 1 and 10). The bursting disk is intact (*Open*= 0, see Eq. 35) and the vapor discharge rate through the orifice, *V*, is zero (see Eq. 2). If for some reason the pressure exceeds the limit of *Pburst*, the burst disk ruptures. In such a case, the variable '*Open*' becomes greater than zero (Eq. 35) and vapor discharge is initiated (Eq. 2) at either sonic (Eq. 4) or subsonic (Eq. 5) discharge rate. The latent heat of vaporization of the discharging oxide down the reactor (see Eqs. 3 and 13) and the reaction essentially stops. When the disk ruptures, the feed to the reactor is stopped (Eq. 1). Those equations are solved using the Polymath ODE Solver Add-In for Excel and shown in Figure 1.

The ODE Solver requires input of the following cell addresses and numerical values: 1) The range of the cells where the initial values are stored (see "ODE Initial values vector (Y) in Figure 1) ; 2) The range of the cells where the formulas of the differential equations are stored (see "ODE equations vector (Y')"); 3) The cell where the initial value for the independent variable (time, in this case) is stored (see "Differential variable cell"); 4) The final value of the independent variable (see "Diffr variable final value"); 5) The range of cells where formulas of additional variables for which the integration results should be stored (optional, see "Intermediate cells to store" in Figure 1). The default values for error tolerances and the number of data points to be stored and the integration algorithm used (default value: variable step-size 4th - 5th order Runge-Kutta) can also be changed.

After the equations are integrated, a table that includes the initial, minimal, maximal and final values of all the differential variables and the selected explicit variables is automatically displayed (see Table 5). Tabular display of detailed results for those variables is also provided. The results shown in Table 5 were obtained in normal operating conditions. It can be seen that the highest temperature in the reactor is 112 °C and the highest pressure is 6.35 bars, well below the safety disk rupture pressure, which is 8 bars. These results were verified with the original Polymath solution.

This problem has been used in a process safety course as an assignment in hazard and operability analysis (HAZOP). A typical student assignment includes verification of the model by carrying out the simulation in normal operating conditions and comparing results with the solution provided by the instructor. Then, checking the effects of reducing the cooling re-circulation (from 5000 kg/min to 4300 kg/min and to 4200 kg/min), of cooling water failure of a varying

duration at different stages of the batch, of cooling pipe blockage and of failure of the bursting disk to open at the specified rupture pressure (8 bar). If temperature runaway (reactor pressure exceeds the 8 bar limit) is due to occur in any of those cases, the students are also asked to suggest changes in the operating conditions that may enable a successful completion of the batch.

Additional reactor simulations for various conditions can be carried out in additional copies of the original worksheet that contain the problem specification and modified parameter values as desired. In Figure 2 the change of the pressure in the semi-batch reactor is shown for the case where the cooling recirculation rate was reduced to 2400 kg/min (from the normal value of 3300 kg/min). It can be seen that, under such conditions, a temperature runaway develops, and after about 7 hours and 30 minutes the pressure exceeds 8 bars which is the safety disk rupture limit. The disk opens, the oxide gases are released the pressure is reduced to the atmospheric level, and the reaction stops. Thus the model of the reaction can be presented in a clear and well-documented tabular or graphical format. The resulting Excel worksheet can be easily and effectively used for analyzing the reactors' behavior under different conditions.

Conclusions

It has been demonstrated, using two fairly complex real-life example problems, that the tools that have recently been developed: namely the Polymath equation export to Excel and the ODE_ Solver Add-In for Excel put advanced numerical problem solving within the reach of most engineers and scientists.

The model equations can be entered efficiently, easily, and with minimal technical effort into Polymath. The variable names, equations, comments and formulas that are exported to Excel yield a well documented worksheet of the mathematical model which can be solved either by the Excel "Solver" (for nonlinear algebraic equations) or the Polymath "ODE_Solver" Add-In for Excel (for differential equations). The Excel formulas may remain transparent for casual users or can serve as a basis for solving even more complex problems by sophisticated users.

The use of Excel is important to practicing engineers as it is usually the software tool of choice. Well-documented and verified problem calculations in Excel can be easily distributed and utilized by others. It is a convenient tool for ready documentation of engineering problem solving in technical reports. Polymath can enable the efficient solution of higher level problems within Excel while also providing separate Polymath solutions to the same problems. This allows independent verification of the problem solutions in both Excel and Polymath using these two different numerical software packages.

The Polymath/Excel capabilities can also be very important in educational settings in that students can be exposed to efficiently solving more detailed problems that involve numerical solution techniques with several software packages.

Bibliography

1. Edgar, T. F. "Computing Through the Curriculum: An Integrated Approach for Chemical Engineering," Technical Report, CACHE Corporation, 2003.
2. Henley, E. J.; Rosen, E. M. Material and Energy Balance Computations, Wiley: New York, 1969.
3. Ingham, J., Dunn, I. J., Heinzle, E. and J. E. Prenosil, *Chemical Engineering Dynamics*, VCH, Weinheim, 1994
4. Kneale, M. and G. M. Forster, "An Emergency Condensing System for a Large Propylene Oxide Polymerization Reactor", I. Chem. E. Symp. Series No. 25, 98 (1968)

Biography of the Authors

MORDECHAI SHACHAM is professor and a former chair of the Department of Chemical Engineering at the Ben-Gurion University of the Negev in Israel where he has been since 1974. He received his BSc and DSc degrees from the Technion, Israel Institute of Technology. His research interest includes analysis, modeling and regression of data, applied numerical method and prediction and consistency analysis of physical properties.

MICHAEL B. CUTLIP is professor emeritus of the Chemical Engineering Dept. at the University of Connecticut and has served as department head and director of the university's Honors Program. He has B. Ch. E. and M. S. degrees from Ohio State and a Ph. D. from the University of Colorado. His current interests include software for engineering problem solving and educational delivery of coursework by the Internet.

MICHAEL ELLY holds B. Sc. and MBA degrees from the Ben-Gurion University of the Negev. As a part of his graduation work he developed a software application for calculating Thermodynamic Vapor Liquid Equilibrium models for which he received the Dean's Prize of Excellence. He joined Intel Corporation in 1996, serving several IT positions in Israel and in the US. Currently he works at Intel in Oregon as a Senior Automation Engineer.

Appendix A

Non-ideal Phase Equilibrium Computation - Bubble Point with Two Liquid Phases and

One Vapor Phase

Phase equilibrium in a system, which is separated into two liquid phases and one vapor phase, at its bubble point, can be represented by the following equations.
Mole balance on component i yields:

$$z_i = x_{i,1}\left[\beta + (1-\beta)\frac{k_{i,1}}{k_{i,2}} \right] \qquad \text{(A-1)}$$

where z_i is the mol fraction of component i in the feed, $x_{i,1}$ is the mol fraction of component i in the first liquid phase, $k_{i,1}$ is the phase equilibrium constant of component i in the 1st liquid phase, $k_{i,2}$ is the phase equilibrium constant of component i in the 2nd liquid phase, $\beta = L_1/F$ where L_1 is the total amount (moles) of the 1st liquid phase and F is the total amount of the feed.
Phase equilibrium conditions:

$$y_i = x_{i,1}k_{i,1} = x_{i,2}k_{i,2} \qquad \text{(A-2)}$$

where y_i is the mol fraction of component i in the vapor phase.
Mole fraction summation:

$$\sum_i x_{i,1} - \sum_i y_i = 0 \qquad \text{(A-3)}$$

$$\sum_i x_{i,1} - \sum_i x_{i,2} = 0 \qquad (A\text{-}4)$$

The phase equilibrium coefficients of component i in liquid phase j can be calculated from the equation:

$$k_{i,j} = \frac{\gamma_{i,j} P_i}{P} \qquad (A\text{-}5)$$

where $\gamma_{i,j}$ is the activity coefficient of component i in liquid phase j, P_i is the vapor pressure of component i and P is the total pressure. The Antoine equation is used to calculate the vapor pressure:

$$\log P_i = Avp_i - \frac{Bvp_i}{Cvp_i + t} \qquad (A\text{-}6)$$

where P_i is the vapor pressure (mmHg), t is the temperature (°C) and Avp_i, Bvp_i and Cvp_i are Antoine equation constants of component i.

For a mixture of isobutanol (20%, component No. 1)) and water (80%) calculate the composition of the two liquid and the vapor phase and the temperature at the bubble point for total pressure of 760 mmHg. Use the following equations to calculate the activity coefficients of the isobutanol and the water (Henley and Rosen[2])

For isobutanol:

$$\log \gamma_{1,j} = \frac{1.7 x_{2,j}^2}{\left(\dfrac{1.7}{0.7} x_{1,j} + x_{2,j} \right)^2} \qquad (A\text{-}7)$$

For water:

$$\log \gamma_{2,j} = \frac{0.7 x_{1,j}^2}{\left(x_{1,j} + \dfrac{0.7}{1.7} x_{2,j} \right)^2} \qquad (A\text{-}8)$$

Use the following Antoine equation coefficients (Henley and Rosen[2]): $Avp_1 = 7.62231$, $Bvp_1 = 1417.09$ and $Cvp_1 = 191.15$ (for isobutanol) $Avp_2 = 8.10765$, $Bvp_2 = 1750.29$ and $Cvp_2 = 235.0$ (for water).

Table 1. Polymath input for Example 1.

Equations	Comments
f(x11) = x11 - 0.2 / (beta1 + (1 - beta1) * k11 / k12)	Mole fraction of comp. 1 in liquid phase 1
f(x21) = x21 - 0.8 / (beta1 + (1 - beta1) * k21 / k22)	Mole fraction of comp. 2 in liquid phase 1
f(x12) = x12 - x11 * k11 / k12	Mole fraction of comp. 1 in liquid phase 2
f(x22) = x22 - x21 * k21 / k22	Mole fraction of comp. 2 in liquid phase 2
f(t) = x11 * (1 - k11) + x21 * (1 - k21)	Bubble point temperature (deg. C)
f(beta1) = (x11 - x12) + (x21 - x22)	Liquid phase split ratio [L1/(L1+L2)]
k11 = gamma11 * p1 / 760	Vapor liquid equilibrium ratio of comp. 1 in liquid phase 1
k21 = gamma21 * p2 / 760	Vapor liquid equilibrium ratio of comp. 2 in liquid phase 1
k12 = gamma12 * p1 / 760	Vapor liquid equilibrium ratio of comp. 1 in liquid phase 2
k22 = gamma22 * p2 / 760	Vapor liquid equilibrium ratio of comp. 2 in liquid phase 2
p1 = 10 ^ (7.62231 - 1417.9 / (191.15 + t))	Vapor pressure of 1st component (mmHg)
p2 = 10 ^ (8.10765 - 1750.29 / (235 + t))	Vapor pressure of 2nd component (mmHg)
A = 1.7	Van Laar equations constant A
B = 0.7	Van Laar equations constant B
gamma11 = 10 ^ (A * x21 * x21 / ((A * x11 / B + x21) ^ 2))	Activity coefficient of comp. 1 in liquid phase 1
gamma21 = 10 ^ (B * x11 * x11 / ((x11 + B * x21 / A) ^ 2))	Activity coefficient of comp. 2 in liquid phase 1
gamma12 = 10 ^ (A * x22 * x22 / ((A * x12 / B + x22) ^ 2))	Activity coefficient of comp. 1 in liquid phase 2
gamma22 = 10 ^ (B * x12 * x12 / ((x12 + B * x22 / A) ^ 2))	Activity coefficient of comp. 2 in liquid phase 2
y1 = k11 * x11	Mole fraction of comp. 1 in the vapor phase
y2 = k21 * x21	Mole fraction of comp. 2 in the vapor phase
x11(0) = 0	Initial guesses for nonlinear equations variables
x21(0) = 1	
x12(0) = 1	
x22(0) = 0	
t(0) = 100	
beta1(0) = 0.8	

Table 2. Excel Worksheet Generated by Polymath for Example 1

POLYMATH NLE Migration Document

	Variable	Value	Polymath Equation
Explicit Eqs	k11	37.2819075	k11=gamma11 * p1 / 760
	k21	1.00482432	k21=gamma21 * p2 / 760
	k12	0.74387185	k12=gamma12 * p1 / 760
	k22	5.03605123	k22=gamma22 * p2 / 760
	p1	565.342607	p1=10 ^ (7.62231 - 1417.9 / (191.15 + t))
	p2	763.666485	p2=10 ^ (8.10765 - 1750.29 / (235 + t))
	A	1.7	A=1.7
	B	0.7	B=0.7
	gamma11	50.1187234	gamma11=10 ^ (A * x21 * x21 / ((A * x11 / B + x21) ^ 2))
	gamma21	1	gamma21=10 ^ (B * x11 * x11 / ((x11 + B * x21 / A) ^ 2))
	gamma12	1	gamma12=10 ^ (A * x22 * x22 / ((A * x12 / B + x22) ^ 2))
	gamma22	5.01187234	gamma22=10 ^ (B * x12 * x12 / ((x12 + B * x22 / A) ^ 2))
	y1	0	y1=k11 * x11
	y2	1.00482432	y2=k21 * x21
Implicit Vars	x11	0	x11(0)=0
	x21	1	x21(0)=1
	x12	1	x12(0)=1
	x22	0	x22(0)=0
	t	100	t(0)=100
	beta1	0.8	beta1(0)=.8
Implicit Eqs	f(x11)	-0.0184779	f(x11)=x11 - 0.2 / (beta1 + (1 - beta1) * k11 / k12)
	f(x21)	0.0475116	f(x21)=x21 - 0.8 / (beta1 + (1 - beta1) * k21 / k22)
	f(x12)	1	f(x12)=x12 - x11 * k11 / k12
	f(x22)	-0.1995262	f(x22)=x22 - x21 * k21 / k22
	f(t)	-0.0048243	f(t)=x11 * (1 - k11) + x21 * (1 - k21)
	f(beta1)	0	f(beta1)=(x11 - x12) + (x21 - x22)
Sum of Squares:		**1.04243278**	F = f(x11)^2+f(x21)^2+f(x12)^2+f(x22)^2+f(t)^2+f(beta1)^2

Table 3. Excel solution for Example 1.

	Variable	Value	Polymath Equation
Implicit Vars	x11	0.02269587	x11(0)=0.02
	x21	0.97720087	x21(0)=1
	x12	0.68667879	x12(0)=1
	x22	0.31321777	x22(0)=0
	t	88.5378342	t(0)=90
	beta1	0.7331214	beta1(0)=.7
Implicit Eqs	f(x11)	-1.152E-05	f(x11)=x11 - 0.2 / (beta1 + (1 - beta1) * k11 / k12)
	f(x21)	-2.084E-06	f(x21)=x21 - 0.8 / (beta1 + (1 - beta1) * k21 / k22)
	f(x12)	7.4323E-07	f(x12)=x12 - x11 * k11 / k12
	f(x22)	-1.126E-06	f(x22)=x22 - x21 * k21 / k22
	f(t)	-2.701E-07	f(t)=x11 * (1 - k11) + x21 * (1 - k21)
	f(beta1)	1.8364E-07	f(beta1)=(x11 - x12) + (x21 - x22)
Sum of Squares:		**1.3907E-10**	F = f(x11)^2+f(x21)^2+f(x12)^2+f(x22)^2+f(t)^2+f(beta1)^2

Table 4. Excel Worksheet Generated by Polymath for Example 2.

POLYMATH DEQ Migration Document

	Variable	Value	No	Polymath Equation	Comments
Explicit Eqs	F	100	1	F=If (Open > 0) Then (0) Else (100)	Oxide feed rate (kg/min)
	V	0	2	V=If ((P <= 1) Or (Open == 0)) Then (0) Else (V1)	Vapor discharge rate (kg/min)
	V1	7.52709929	3	V1=If (P < 1.9) Then (Vsub) Else (Vs)	Vapor discharge rate (kg/min)
	Vs	4.52409351	4	Vs=0.85 * Kv * P / sqrt(TR + 273)	Sonic vapor discharge rate (kg/min)
	Vsubs	7.52709929	5	Vsubs=Kv * P / sqrt((TR + 273)) * sqrt(1 + 1 / P ^ 2)	Sub-sonic - vapor discharge rate (kg/min)
	r	0	6	r=k * MC	Reaction rate (kg oxide/min)
	Hc	0	7	Hc=F * Cp * (T0 - TR)	Feed enthalpy change (kJ/min)
	Hv	0	8	Hv=V * Lamda	Latent heat of vapor discharge (kJ/min)
	Qg	0	9	Qg=r * HR	Heat of reaction (kJ/min)
	Qr	0	10	Qr=Fc * Cp * (TR - T0)	Heat removal (kJ/min)
	P	1	11	P=If (P1 < 1) Then (1) Else (P1)	Oxide vapor pressure (bar)
	P1	0	12	P1=(exp(-3430 / (TR + 273) + 11.7) + 1.45e-3 * MW) * C	Oxide vapor pressure (bar)
	k	0.00089458	13	k=9e9 * exp(-E / (R * (TR + 273)))	Reaction rate coefficient
	C	0	14	C=MC / M	Oxide concentration (kg/kg)
	MW	74	15	MW=(M0 + X) / (M0 / 74)	Molecular weight of the polymer (kg/mol)
	T0	80	16	T0=80	Feed temperature (deg. C)
	Lamda	670	17	Lamda=670	Heat of vaporization of the oxide (kj/kg)
	Cp	3.5	18	Cp=3.5	Spec. heat of feed and reacting mass (kJ/kg- deg C)
	HR	-1660	19	HR=-1660	Heat of reaction (kJ/ kg oxide)
	Fc	3300	20	Fc=3300	Recycle mass flow rate (kg/min)
	Pburst	8	21	Pburst=8	Disk rupture pressure (bar)
	R	1.987	22	R=1.987	Gas constant
	E	21000	23	E=21000	Activation energy
	M0	4400	24	M0=4400	Initial alcohol charge (kg)
	Kv	100	25	Kv=100	Valve discharge coefficient
Integration Vars	M	4400	26	M(0)=4400	
	MC	0	27	MC(0)=0	
	TR	80	28	TR(0)=80	
	X	0	29	X(0)=0	
	Open	0	30	Open(0)=0	
ODE Eqs	d(M)/d(t)	100	31	d(M)/d(t) = F - V	Total mass in the reactor (kg)
	d(MC)/d(t)	100	32	d(MC)/d(t) = F - V - r	Oxide component mass in the reactor (kg)
	d(TR)/d(t)	0	33	d(TR)/d(t) = (Hc - Hv - Qg - Qr) / (M * Cp)	Temperature in the reactor (deg C)
	d(X)/d(t)	0	34	d(X)/d(t) = r	The mass of oxide reacted (kg)
	d(Open)/d(t)	0	35	d(Open)/d(t) = If (P < Pburst) Then (0) Else (0.001)	Status of the bursting disk: 0 closed, >0 open
Indep Var	t	0	36	t(0)=0 ; t(f)=2000	

Table 5. Integration Results for Example 2

| | | Values of the Variables | | | |
	Variable	Initial	Minimal	Maximal	Final
1	t	0	0	2000	2000
2	M	4400	4400	2.04E+05	2.04E+05
3	MC	0	0	39923.32	32254.39
4	TR	80	80	112.0548	91.30254
5	X	0	0	1.68E+05	1.68E+05
6	Open	0	0	0	0
7	P	1	1	6.352159	2.212553
8	P1	0	0	6.352159	2.212553
9	k	0.000895	0.000895	0.010837	0.002265
10	C	0	0	0.744687	0.1578
11	MW	74	74	2895.176	2895.176

	A	B	C	D	E	
13		P	1		P=If (P1 < 1) Then (1) Else (P1)	Oxide vapor p
14		P1	0		P1=(exp(-3430 / (TR + 273) + 11.7) + 1.45e-3 * MW) * C	Oxide vapor p
15		k	0.000894577		k=9e9 * exp(-E / (R * (TR + 273)))	Reaction rate
16		C	0		C=MC / M	Oxide connce
17		MW	74		MW (M0 + X) / (M0 / 74)	Molecular we
18		T0	80			
19		Lamda	670			
20		Cp	3.5			
21		HR	-1660			
22		Fc	3300			
23		Pburst	8			
24		R	1.987			
25		E	21000			
26		M0	4400			
27		Kv	100			
28	Integration Vars	M	4400			
29		MC	0			
30		TR	80			
31		X	0			
32		Open	0			
33	ODE Eqs	d(M)/d(t)	100			
34		d(MC)/d(t)	100		d(MC)/d(t) = F - V - r	
35		d(TR)/d(t)	0		d(TR)/d(t) = (Hc - Hv - Qg - Qr) / (M * Cp)	
36		d(X)/d(t)	0		d(X)/d(t) = r	
37		d(Open)/d(t)	0		d(Open)/d(t) = If (P < Pburst) Then (0) Else (0.001)	
38	Indep Var	t	0		t(0)=0 ; t(f)=2000	
39						

Polymath DEQ

ODE initial values vector (Y): PL1!$C$28:$C$32

ODE equations vector (Y'): PL1!$C$33:$C$37

Differential variable cell: PL1!$C$38

Diffr variable final value: 2000

☑ Show Report

Intermediate Cells to Store: PL1!$C$13:$C$17

Data Points: 300

Exit | Advanced | Clear | Reload | Solve

Figure 1. Application of the Polymath ODE-Solver for Excel to Example 2
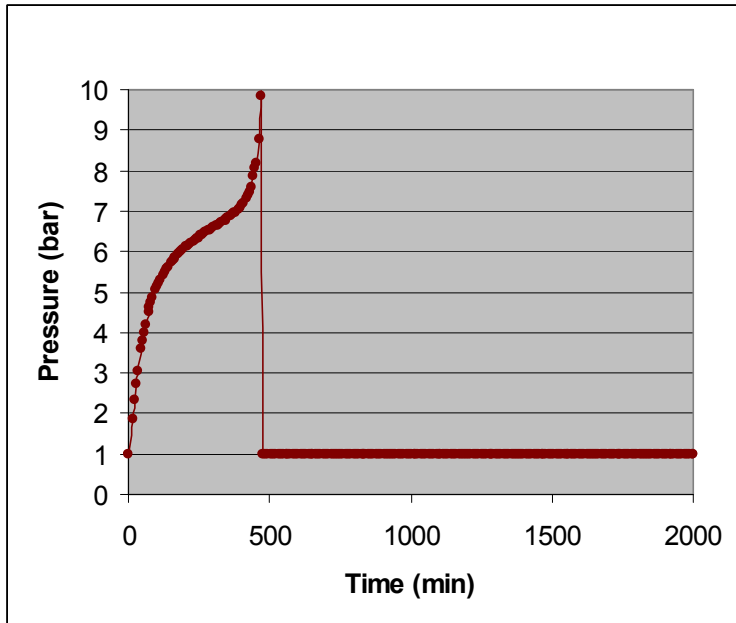
Figure 2. Change of the Pressure in the Semi-Batch Reactor after Reduction of the Cooling Recirculation Rate