

---

# POLYMATH SOLUTIONS TO THE CHEMICAL ENGINEERING PROBLEM SET

---

## Mathematical Software - Session 12\*

Michael B. Cutlip, Department of Chemical Engineering, Box U-222, University of Connecticut, Storrs, CT 06269-3222 (mcutlip@uconnvm.uconn.edu)

Mordechai Shacham, Department of Chemical Engineering, Ben-Gurion University of the Negev, Beer Sheva, Israel 84105 (shacham@bgumail.bgu.ac.il)

### INTRODUCTION

These solutions are for a set of numerical problems in chemical engineering developed for Session 12 at the ASEE Chemical Engineering Summer School held in Snowbird, Utah on August 13, 1997. The problems in this set are intended to utilize the basic numerical methods in problems which are appropriate to a variety of chemical engineering subject areas.

The package used to solve each problem is the POLYMATH Numerical Computation Package Version 4.0 which is widely used in Chemical Engineering. The complete POLYMATH package and all problem solutions are on a disk which is available from the authors (contact M. Cutlip). Details on POLYMATH are available from the WWW at <http://www.polymath-software.com>.

The POLYMATH Numerical Computation Package has four companion programs.

- SIMULTANEOUS DIFFERENTIAL EQUATIONS
- SIMULTANEOUS ALGEBRAIC EQUATIONS
- SIMULTANEOUS LINEAR EQUATIONS
- CURVE FITTING AND REGRESSION

POLYMATH is a proven computational system which has been specifically created for educational use by M. Shacham and M. B. Cutlip. The various POLYMATH programs allow the user to apply effective numerical analysis techniques during interactive problem solving on personal computers. Results are presented graphically for easy understanding and for incorporation into papers and reports. Students with a need to solve numerical problems will appreciate the efficiency and speed of problem solution. With POLYMATH, the user is able to focus complete attention to the problem rather than spending valuable time in learning how to use or reuse the programs.

INEXPENSIVE SITE LICENSES AND SINGLE COPIES ARE AVAILABLE FROM:

CACHE CORPORATION\*\*  
P. O. Box 7939  
AUSTIN, TX 78713-7939  
Phone: (512)471-4933 Fax: (512)295-4498  
E-mail: [cache@uts.cc.utexas.edu](mailto:cache@uts.cc.utexas.edu)  
Internet: <http://www.che.utexas.edu/cache/>

---

\*The Ch. E. Summer School was sponsored by the Chemical Engineering Division of the American Society for Engineering Education. This material is copyrighted by the authors, and permission must be obtained for duplication unless for educational use within departments of chemical engineering.

---

\*\*A non-profit educational corporation supported by most North American chemical engineering departments and many chemical corporation. CACHE stands for computer aides for chemical engineering.

### Polymath Problem 1 Solution

Equation (1) can not be rearranged into a form where  $V$  can be explicitly expressed as a function of  $T$  and  $P$ . However, it can easily be solved numerically using techniques for nonlinear equations. In order to solve Equation (1) using the POLYMATH *Simultaneous Algebraic Equation Solver*, it must be rewritten in the form

$$f(V) = \left( P + \frac{a}{V^2} \right) (V - b) - RT \quad \text{PM-(1)}$$

where the solution is obtained when the function is close to zero,  $f(V) \approx 0$ . Additional explicit equations and data can be entered into the POLYMATH program in direct algebraic form. The POLYMATH program will reorder these equations as necessary in order to allow sequential calculation.

The POLYMATH equation set for this problem are given by

```
Equations:
f(V)=(P+a/(V^2))*(V-b)-R*T
P=56
R=0.08206
T=450
Tc=405.5
Pc=111.3
Pr=P/Pc
a=27*(R^2*Tc^2/Pc)/64
b=R*Tc/(8*Pc)
Z=P*V/(R*T)
Search Range:
V(min)=0.4, V(max)=1
```

In order to solve a single nonlinear equation with POLYMATH, an interval for the expected solution variable,  $V$  in this case, must be entered into the program. This interval can usually be found by consideration of the physical nature of the problem.

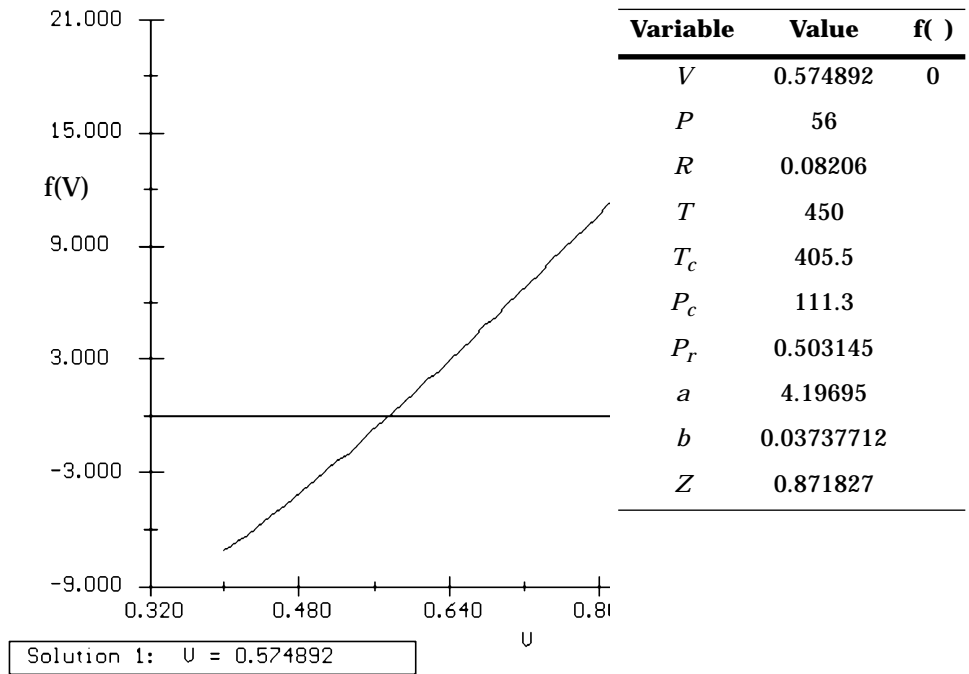
(a) For part (a) of this problem, the volume calculated from the ideal gas law as  $V = 0.66$  liter/g-mol can be a basis for specifying the required solution interval. An interval for the expected solution for  $V$  can be entered as between 0.4 as the lower limit and 1.0 as the higher limit. The POLYMATH solution, which is given in Figure PM-(1) for  $T = 450$  K and  $P = 56$  atm, yields  $V = 0.5749$  liter/gmol where the compressibility factor is  $Z = 0.8718$ .

(b) Solution for the additional pressure values can be accomplished by changing the equations in the POLYMATH program for  $P$  and  $P_r$  to

```
Pr=1
P=Pr*Pc
```

Additionally, the bounds on the molar volume  $V$  may need to be altered to obtain an interval where there is a solution. Subsequent program execution for the various  $P_r$ 's is required.

(c) The calculated molar volumes and compressibility factors are summarized in Table (1). These calculated results indicate that there is a minimum in the compressibility factor  $Z$  at approximately  $P_r = 2$ . The compressibility factor then starts to increase and reaches  $Z = 2.783$  for  $P_r = 20$ .



**Figure PM-1** Plot of  $f(V)$  versus  $V$  for van der Waals Equation

**Table PM-1** Compressibility Factor for Gaseous Ammonia at 450 K

<i>P(atm)</i>	<i>P<sub>r</sub></i>	<i>V</i>	<i>Z</i>
56	0.503	.574892	0.871827
111.3	1.0	.233509	0.703808
222.6	2.0	.0772676	0.465777
445.2	4.0	.0606543	0.731261
1113.0	10.0	.0508753	1.53341
2226.0	20.0	.046175	2.78348

### Polymath Problem 2 Solution

(a) The coefficients and the constants in the Equation Set Equation (6) can be directly introduced into the POLYMATH *Linear Equation Solver* in matrix form as shown

Name	x1	x2	x3	x4	b
1	0.07	0.18	0.15	0.24	10.5
2	0.04	0.24	0.1	0.65	17.5
3	0.54	0.42	0.54	0.1	28
4	0.35	0.16	0.21	0.01	14

The solution is

Variable	Value
x1	26.25
x2	17.5
x3	8.75
x4	17.5

which corresponds to the unknown flow rates of  $D_1 = 26.25$  mol/min,  $B_1 = 17.5$  mol/min,  $D_2 = 8.75$  mol/min, and  $B_2 = 17.5$  mol/min.

(b) The overall balances and individual component balances on column #2 given in Equation Set (7) can be solved algebraically to give  $X_{Dx} = 0.114$ ,  $X_{Ds} = 0.120$ ,  $X_{Dt} = 0.492$  and  $X_{Db} = 0.274$ . Similarly, overall balance and individual component balances on column #3 presented as Equation Set (8) yield  $X_{Bx} = 0.210$ ,  $X_{Bs} = 0.4667$ ,  $X_{Bt} = 0.2467$  and  $X_{Bb} = 0.0767$ .

### Polymath Problem 3 Solution

**(a) Data Regression with a Polynomial** The POLYMATH *Polynomial, Multiple Linear and Nonlinear Regression Program* can be used to solve this problem by first entering the data in a similar manner to using a spreadsheet. Let us denote the column of temperature data in °C as  $TC$  and the column of pressure data as  $P$ . This POLYMATH worksheet is reproduced in Figure PM-(2) where the first two columns are used in the polynomial regressions.

Name	TC	P	logP	TK	neginvTK	
1	-36.7	1	0	236.45	-0.004229	
2	-19.6	5	0.69897	253.55	-0.003944	
3	-11.5	10	1	261.65	-0.003822	
4	-2.6	20	1.30103	270.55	-0.003696	
5	7.6	40	1.60206	280.75	-0.003562	
6	15.4	60	1.77815	288.55	-0.003466	
7	26.1	100	2	299.25	-0.003342	
8	42.2	200	2.30103	315.35	-0.003171	
9	60.6	400	2.60206	333.75	-0.002996	
10	80.1	760	2.88081	353.25	-0.002831	
11						

**Figure PM-2** POLYMATH Entry for Regressions

A polynomial regression option within POLYMATH when the dependent variable column titled  $P$  is regressed with the independent variable  $TC$  corresponds directly to Equation (9). The results are summarized in Figure PM-(3) which also presents the value of the variance (var.) for each polynomial. The variance indicates that the polynomial which best represents the data in this case is the 4th degree.

**(b) Regression with Clausius-Clapeyron Equation** Data regression with the Clausius-Clapeyron expression, Equation (10), can be accomplished by three additional transformed variables (columns) in the POLYMATH program used for part (a). Columns can be defined by the relationships:  $\log P = \log(P)$ ,  $TK = T + 273.15$ , and  $\text{neginvTK} = -1/TK$  as indicated in Figure PM-(2). A request for linear regression when the first (and only) independent variable column is  $\text{neginvTK}$  and the dependent variable column is  $\log P$  yields the following plot and numerical results from POLYMATH as shown in Figure PM-(4).

**(c) Regression with the Antoine Equation** This expression, Equation (11), cannot be linearized and so it must be regressed with nonlinear regression option of the POLYMATH *Polynomial, Multiple Linear and Nonlinear Regression Program*. With this option, the user must supply initial estimates. In this case, it is helpful to use the initial estimates for  $A$  and  $B$  which were determined in part (b) and use the estimate for  $C$  as 273.15. Direct entry of Equation (11) with the initial estimates gives the converged results shown in Figure PM-(5).

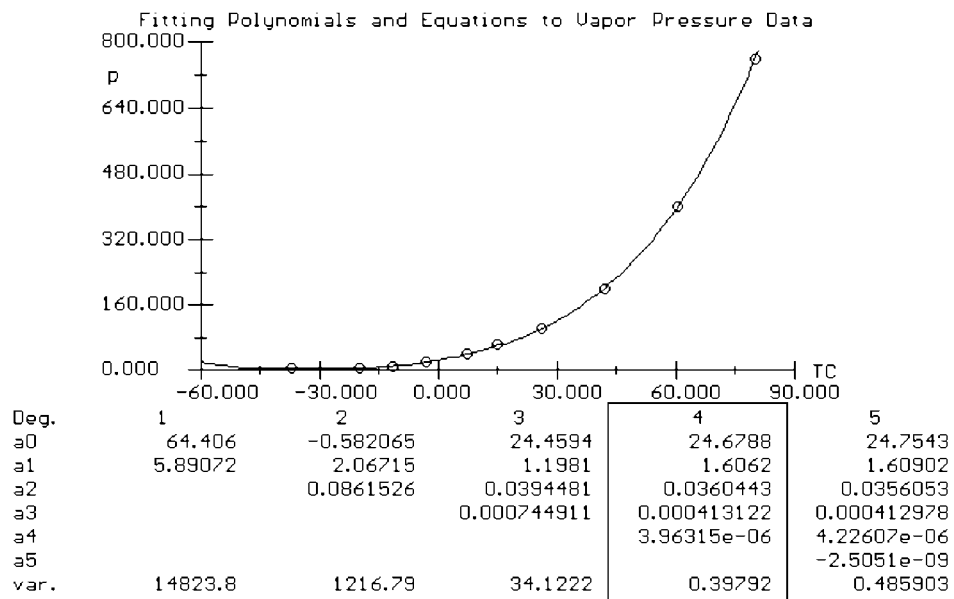


Figure PM-3 POLYMATH Results for Fitting Polynomials to Vapor Pressure Data

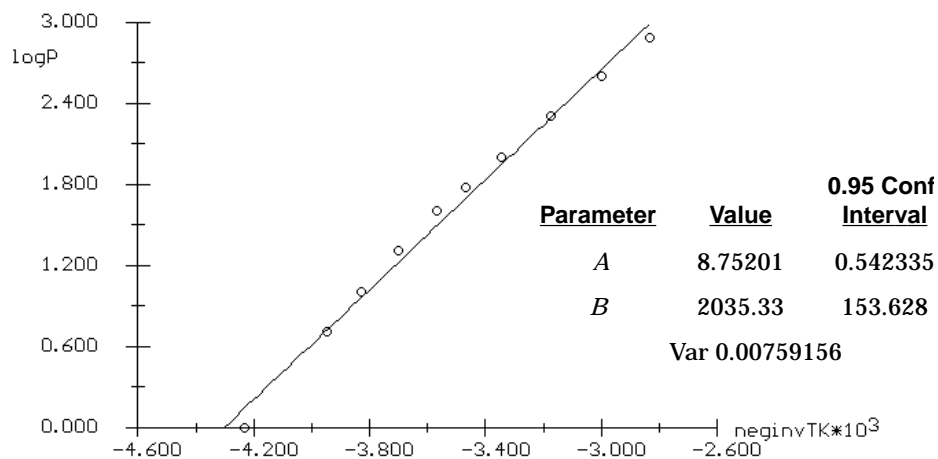
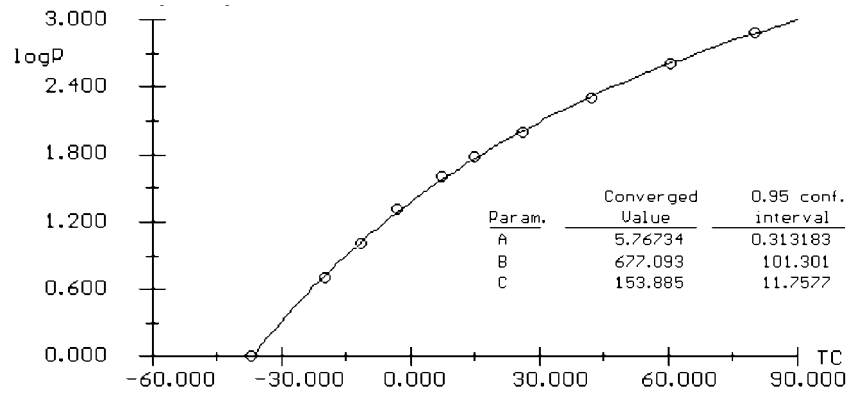


Figure PM-4 POLYMATH Results for Regression of Clausius-Clapeyron Equation



Model:  $\log P = A - B / (C + TC)$

A = 5.76734

C = 153.885

B = 677.093

5 positive residuals, 5 negative residuals. Sum of squares = 0.0022305

**Figure PM-5 POLYMATH Results for Nonlinear Regression of Antoine Equation**

**Polymath Problem 4 Solution**

The Equation Set (13) can be entered into the POLYMATH *Simultaneous Algebraic Equation Solver*; but the nonlinear equilibrium expressions must be written as functions which are equal to zero at the solution. A simple transformation of the equilibrium expressions of Equation Set Equation (12) to the required functional form yields

$$\begin{aligned} f(C_D) &= \frac{C_C C_D}{C_A C_B} - K_{C1} \\ f(C_X) &= \frac{C_X C_Y}{C_B C_C} - K_{C2} \\ f(C_Z) &= \frac{C_Z}{C_A C_X} - K_{C3} \end{aligned} \quad \text{PM-(2)}$$

The above equation set may be difficult to solve because the division by unknowns may make most solution algorithms diverge.

***Expediting the Solution of Nonlinear Equations***

An additional simple transformation of the nonlinear function can make many functions much less nonlinear and easier to solve by simply eliminating division by the unknowns. In this case, the Equation Set PM-(2) can be modified to

$$\begin{aligned} f(C_D) &= C_C C_D - K_{C1} C_A C_B \\ f(C_X) &= C_X C_Y - K_{C2} C_B C_C \\ f(C_Z) &= C_Z - K_{C3} C_A C_X \end{aligned} \quad \text{PM-(3)}$$

The POLYMATH equation set utilizing Equation Set PM-(3) with the initial conditions for part (a) is given below.

```
Equations:
f(CD)=CC*CD-KC1*CA*CB
f(CX)=CX*CY-KC2*CB*CC
f(CZ)=CZ-KC3*CA*CX
KC1=1.06
CY=CX+CZ
KC2=2.63
KC3=5
CA0=1.5
CB0=1.5
CC=CD-CY
CA=CA0-CD-CZ
CB=CB0-CD-CY
Initial Estimates:
CD(0)=0
CX(0)=0
CZ(0)=0
```

(a), (b) and (c) The POLYMATH solutions are summarized in Table PM-(2) for the three sets of initial conditions. Note that the initial conditions for problem part (a) converged to all positive concentrations. However the initial conditions for parts (b) and (c) converged to some negative values for some of the concentrations. Thus a "reality check" on Table PM-(2) for physical feasibility reveals that



the negative concentrations in parts (b) and (c) are the basis for rejecting these solutions as not representing a physically valid situation.

**Table PM-2** POLYMATH Solutions of the Chemical Equilibrium Problem

<b>Variable</b>	<b>Part (a)</b>	<b>Part (b)</b>	<b>Part (c)</b>
$C_D$	0.7053	0.05556	1.070
$C_X$	0.1778	0.5972	-0.3227
$C_Z$	0.3740	1.082	1.131
$C_A$	0.4207	0.3624	-0.7006
$C_B$	0.2429	-0.2348	-0.3779
$C_C$	0.1536	-1.624	0.2623
$C_Y$	0.5518	1.679	0.8078

### Polymath Problem 5 Solution

(a) For conditions similar to those of this problem, the Reynolds number will not exceed 1000 so that only Equations (16) and (17) need to be applied. The logic which selects the proper equation based on the value of  $Re$  can be employed using the “if.. then... else...” statement within the POLY-MATH *Simultaneous Algebraic Equation Solver*:

$$C_D = \text{if}( Re < 0.1) \text{ then } (24/ Re) \text{ else } (24 \times (1 + 0.14 Re^{0.7})) \quad \text{PM-(4)}$$

Equation (13) should be rearranged in order to avoid possible division by zero and negative square roots as it is entered into the form of a nonlinear equation for POLYMATH.

$$f(v_t) = v_t^2(3C_D\rho) - 4g(\rho_p - \rho)D_p \quad \text{PM-(5)}$$

The following equation set can be solved by POLYMATH.

Equations:

```
f(vt)=vt^2*(3*CD*rho)-4*g*(rhop-rho)*Dp
g=9.80665
rhop=1800
rho=994.6
Dp=0.208e-3
vis=8.931e-4
Re=Dp*vt*rho/vis
CD=if (Re<0.1) then (24/Re) else (24*(1+0.14*Re^0.7)/Re)
vt(min)=0.0001, vt(max)=0.05
```

Specifying  $v_{t, min} = 0.0001$  and  $v_{t, max} = 0.05$  leads to the results summarized in Table PM-(3).

**Table PM-3** Terminal Velocity Solution

Variable	Value	f( )
vt	0.0157816	-8.882e-16
rho	994.6	
g	9.80665	
rhop	1800	
Dp	0.000208	
vis	0.0008931	
Re	3.65564	
CD	8.84266	

(b) The terminal velocity in the centrifugal separator can be calculated by replacing the  $g$  in Equation PM-(5) by  $30g$ . Introduction of this change to the equation set gives the following results:

$$v_t = 0.2060 \text{ m/s} \quad Re = 47.72 \quad \text{and} \quad C_D = 1.5566$$

### Polymath Problem 6 Solution

Equations (20) to (22), together with the numerical data and initial values given in the problem statement, can be entered into the POLYMATH *Simultaneous Differential Equation Solver*. The initial startup is from a temperature of 20°C in all three tanks, thus this is the appropriate initial condition for each tank temperature. The final value or steady state value can be determined by solving the differential equations to steady state by giving a large time interval for the numerical solution. Alternatively one could set the time derivatives to zero, and solve the resulting algebraic equations. In this case, it is easiest just to numerically solve the differential equations to large value of  $t$  where steady state is achieved. The POLYMATH coding for this problem is shown below.

```

Equations:
d(T1)/d(t)=(W*Cp*(T0-T1)+UA*(Tsteam-T1))/(M*Cp)
d(T2)/d(t)=(W*Cp*(T1-T2)+UA*(Tsteam-T2))/(M*Cp)
d(T3)/d(t)=(W*Cp*(T2-T3)+UA*(Tsteam-T3))/(M*Cp)
W=100
Cp=2.0
T0=20
UA=10.
Tsteam=250
M=1000
Initial Conditions:
t(0)=0
T1(0)=20
T2(0)=20
T3(0)=20
Final Value:
t(f)=200

```

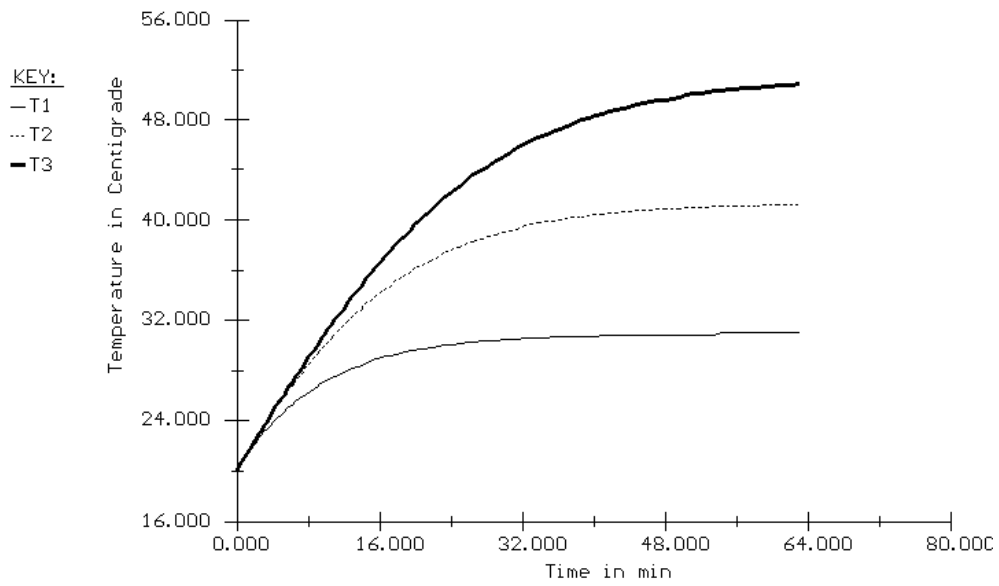
The time to reach steady state is usually considered to be the time to reach 99% of the final steady state value for the variable which is increasing and responds the most slowly. For this problem,  $T_3$  increases the most slowly, and the steady state value is found to be 51.317°C. In POLYMATH, this can be easily done by displaying the output in tabular form for  $T_1$ ,  $T_2$ , and  $T_3$  so that the approach to steady state can accurately be observed. Thus the time must be determined when  $T_3$  reaches 0.99(51.317) or 50.804 °C. Again the tabular form of the output is useful in determining this time as illustrated in Table 4 yielding the time to steady state as approximately 63.0 min. A plot of the three tank temperatures from POLYMATH is given in Figure PM-(6).

**Table PM-4** Tabular Output Option from POLYMATH

t	T3
60	50.662233
60.5	50.688128
61	50.713042
61.5	50.737011
62	50.760068
62.5	50.782246
<b>63</b>	<b>50.803577</b>
63.5	50.82409

**Table PM-4** Tabular Output Option from POLYMATH

t	T3
64	50.843817
64.5	50.862784
65	50.88102



**Figure PM-6** Dynamic Temperature Response in the Three Tanks

## Polymath Problem 7 Solution

### ***Solving Higher Order Ordinary Differential Equations***

Most mathematical software packages can solve only systems of first order ordinary differential equations (ODE's). Fortunately, the solution of an n-th order ODE can be accomplished by expressing the equation by a series of simultaneous first order differential equations each with a boundary condition. This is the approach that is typically used for the integration of higher order ODE's.

(a) Equation (23) is a second order ODE, but it can be converted into a system of first order equations by substituting new variables for the higher order derivatives. In this particular case, a new variable  $y$  can be defined which represent the first derivation of  $C_A$  with respect to  $z$ . Thus Equation (23) can be written as the equation set

$$\begin{aligned}\frac{dC_A}{dz} &= y \\ \frac{dy}{dz} &= \frac{k}{D_{AB}} C_A\end{aligned}\tag{PM-6}$$

This set of first order ODE's can be entered into the *POLYMATH Simultaneous Differential Equation Solver* for solution, but initial conditions for both  $C_A$  and  $y$  are needed. Since the initial condition of  $y$  is not known, an iterative method (also referred to as a shooting method) can be used to find the correct initial value for  $y$  which will yield the boundary condition given by Equation (25).

### ***Shooting Method-Trial and Error***

The shooting method is used to achieve the solution of a boundary value problem to one of an iterative solution of an initial value problem. Known initial values are utilized while unknown initial values are optimized to achieve the corresponding boundary conditions. Either "trial and error" or variable optimization techniques are used to achieve convergence on the boundary conditions.

For this problem, a first "trial and error" value for the initial condition of  $y$ , for example  $y_0 = -150$ , is used to carry out the integration and calculate the error for the boundary condition designated by  $\epsilon$ . Thus the difference between the calculated and desired final value of  $y$  at  $z = L$  is given by

$$\epsilon(y_0) = y_{f, calc} - y_{f, desired}\tag{PM-7}$$

Note that for this example,  $y_{f, desired} = 0$  and thus  $\epsilon(y_0) = y_{f, calc}$  only because this desired boundary condition is zero.

The equations as entered in the *POLYMATH Simultaneous Differential Equation Solver* for an initial "trial and error" solution are

```
Equations:
d(CA)/d(z)=y
d(y)/d(z)=k*CA/DAB
k=0.001
DAB=1.2E-9
err=y
Initial Conditions:
z(0)=0
CA(0)=0.2
y(0)=-150
Final Value:
z(f)=0.001
```

The calculation of  $\epsilon$  in the *POLYMATH* equation set which corresponds to Equation PM-(7) is only

valid at the end of the ODE solution. Repeated reruns of this POLYMATH equation set with different initial conditions for  $y$  can be used in a “trial and error” mode to converge upon the desired boundary condition for  $y_0$  where  $\varepsilon(y_0)$  or  $\text{err} \cong 0$ . Some results are summarized in Table PM-(5) for various values

**Table PM-5** Trial Boundary Conditions for Equation Set (6) in Problem 7 Part (a)

$y_0(z = 0)$	-120.	-130.	-140.	-150.
$y_{f,calc}(z = L)$	17.23	2.764	-11.70	-26.16
$\varepsilon(y_0)$	17.23	2.764	-11.70	-26.16

of  $y_0$ . The desired initial value for  $y_0$  lies between -130 and -140. This “trial and error” approach can be continued to obtain a more accurate value for  $y_0$ , or an optimization technique can be applied.

#### **Newton's Method for Boundary Condition Convergence**

A very useful method for optimizing the proper initial condition is to consider this determination to be a problem in finding the zero of a function. In the notation of this problem, the variable to be optimized is  $y_0$  and the objective function is  $\varepsilon(y_0)$  which is defined by Equation PM-(7).

Newton's method, an effective method for optimizing a single variable, can be applied here to minimize the above objective function. According to this method, an improved estimate for  $y_0$  can be calculated using the equation

$$y_{0, new} = y_0 - \varepsilon(y_0)/\varepsilon'(y_0) \quad \text{PM-(8)}$$

where  $\varepsilon'(y_0)$  is the derivative of  $\varepsilon$  at  $y = y_0$ . The derivative,  $\varepsilon'(y_0)$ , can be estimated using a finite difference approximation

$$\varepsilon'(y_0) \cong \frac{\varepsilon(y_0 + \delta y_0) - \varepsilon(y_0)}{\delta y_0} \quad \text{PM-(9)}$$

where  $\delta y_0$  is a small increment in the value of  $y_0$ . It is very convenient that  $\varepsilon(y_0 + \delta y_0)$  can be calculated *simultaneously with the numerical ODE solution* for  $\varepsilon(y_0)$  thereby allowing calculation of  $\varepsilon'(y_0)$  from Equation PM-(9) and a *new estimate* for  $y_0$  from Equation PM-(8).

Using  $\delta = 0.0001$  for this example, the POLYMATH equation set for carrying out the first step in Newton's method procedure is given by

```

Equations:
d(CA)/d(z)=y
d(y)/d(z)=k*CA/DAB
d(CA1)/d(z)=y1
d(y1)/d(z)=k*CA1/DAB
k=0.001
DAB=1.2E-9
err=y-0
err1=y1-0
y0=-130
L=.001
delta=0.0001
CAanal=0.2*cosh(L*(k/DAB)^.5*(1-z/L))/(cosh(L*(k/DAB)^.5))
derr=(err1-err)/(.0001*y0)
ynew=y0-err/derr
Initial Conditions:
z(0)=0

```

```

CA(0)=0.2
y(0)=-130
CA1(0)=0.2
y1(0)=-130.013
Final Value:
z(f)=0.001

```

This set of equations yields the results summarized in Table 6 where the new estimate for  $y_0$  is the final value of the POLYMATH variable  $y_{new}$  or -131.911. Another iteration of Newton's method can be obtained by starting with the new estimate and modifying the initial conditions for  $y$  and  $y_1$  and the value of  $y_0$  in the POLYMATH equation set. The second iteration indicates that the  $err$  is approximately  $3.e-4$  and that  $y_{new}$  is unchanged indicating that convergence has been obtained. For the value of  $y_0 = -131.911$ , the numerical and analytical solutions are equal to at least six significant digits.

**Table PM-6** Partial Results for Selected Variables during 1st Newton's Method Iteration

Variable	Initial Value	Maximum Value	Minimum Value	Final Value
z	0	0.001	0	0.001
y	-130	2.76438	-130	2.76438
CA	0.2	0.2	0.140428	0.140461
err	-130	2.76438	-130	2.76438
y1	-130.013	2.74558	-130.013	2.74558
CA1	0.2	0.2	0.140446	-0.142229
err1	-130.013	2.74558	-130.013	2.74558
derr	1	1.44642	1	1.44642
ynew	-5.22675e-11	-5.22675e-11	-131.911	-131.911

### Polymath Problem 8 Solution

This problem requires the simultaneous solution of Equation (27) while the temperature is calculated from the bubble point considerations implicit in Equation (29). A system of equations comprising of differential and implicit algebraic equations is called “differential algebraic” or a DAE system. There are several numerical methods for solving DAE systems. Most problem solving software packages including POLYMATH do not have the specific capability for DAE systems.

**Approach 1** The first approach will be to use the *controlled integration technique* proposed by Shacham, et al.<sup>4</sup>. Using this method, the nonlinear Equation (29) is rewritten with an error term given by

$$\varepsilon = 1 - k_1 x_1 - k_2 x_2 \quad \text{PM-(10)}$$

where the  $\varepsilon$  calculated from this equation provides the basis for keeping the temperature of the distillation at the bubble point. This is accomplished by changing the temperature in proportion to the error in an analogous manner to a proportion controller action. Thus this can be represented by another differential equation

$$\frac{dT}{dx_2} = K_c \varepsilon \quad \text{PM-(11)}$$

where a proper choice of the proportionality constant  $K_c$  will keep the error below a desired error tolerance.

The calculation of  $K_c$  is a simple trial and error procedure for most problems. At the beginning  $K_c$  is set to a small value (say  $K_c = 1$ ), and the system is integrated. If  $\varepsilon$  is too large, then  $K_c$  must be increased and the integration repeated. This trial and error procedure is continued until  $\varepsilon$  becomes smaller than a desired error tolerance throughout the entire integration interval.

The temperature at the initial point is not specified in the problem, but it is necessary to start the problem solution at the bubble point of the initial mixture. This separate calculation can be carried out on Equation (29) for  $x_1 = 0.6$  and  $x_2 = 0.4$  and the Antoine equations using the POLYMATH *Simultaneous Algebraic Equation Solver*. The solution equation set is given by

```
Equations:
f(Tbp)=xA*PA+xB*PB-760*1.2
xA=0.6
PA=10^(6.90565-1211.033/(Tbp+220.79))
PB=10^(6.95464-1344.8/(219.482+Tbp))
xB=1-xA
yA=xA*PA/(760*1.2)
yB=xB*PB/(760*1.2)
Search Range:
Tbp(min)=60, Tbp(max)=120
```

The resulting initial temperature is found to be  $T_0 = 95.5851$ .

The system of equations for the batch distillation as they are introduced into the POLYMATH *Simultaneous Differential Equation Solver* using  $K_c = 0.5 \times 10^6$  are

```
Equations:
d(L)/d(x2)=L/(k2*x2-x2)
d(T)/d(x2)=Kc*err
Kc=0.5e6
```



```

k2=10^(6.95464-1344.8/(T+219.482))/(760*1.2)
x1=1-x2
k1=10^(6.90565-1211.033/(T+220.79))/(760*1.2)
err=(1-k1*x1-k2*x2)
Initial Conditions:
x2(0)=0.4
L(0)=100
T(0)=95.5851
Final Value:
x2(f)=0.8

```

and the partial results from the solution are summarized in Table PM-(7)

**Table PM-7** Partial Results for DAE Binary Distillation Problem

Variable	Initial Value	Maximum Value	Minimum Value	Final Value
x2	0.4	0.8	0.4	0.8
L	100	100	14.0456	14.0456
T	95.5851	108.569	95.5851	108.569
k2	0.532535	0.785753	0.532535	0.785753
Kc	500000	500000	500000	500000
x1	0.6	0.6	0.2	0.2
k1	1.31164	1.8566	1.31164	1.8566
err	-3.64587e-07	7.75023e-05	-3.64587e-07	7.75023e-05

The final values from the table indicate that 14.05 mol of liquid remain in the column when the concentration of the toluene reaches 80%. During the distillation the temperature increases from 95.6°C to 108.6°C. The error calculated from Equation (10) increases from about  $-3.6 \times 10^{-7}$  to  $7.75 \times 10^{-5}$  during the numerical solution, but it is still small enough for the solution to be considered as accurate.

**Approach 2** A different approach for solving this problem can be used because Equation (29) can be differentiated with respect to  $x_2$  to yield

$$\frac{dT}{dx_2} = \frac{(k_2 - k_1)}{\ln(10) \left[ x_1 k_1 \frac{-B_1}{(C_1 + T)^2} + x_2 k_2 \frac{-B_2}{(C_2 + T)^2} \right]} \quad \text{PM-(12)}$$

Thus Equation PM-(12) can provide the bubble point temperature during the simultaneous integration with Equation (27). The equation set to be used with the POLYMATH *Simultaneous Differential Equation Solver* is given by

Equations:

```

d(L)/d(x2)=L/(k2*x2-x2)
d(T)/d(x2)=(k2-k1)/(ln(10)*(x1*k1*(-1211.033)/(220.79+T)^2+x2*k2*(-1344.8)/(219.482+T)^2))

```

```
k2=10^(6.95464-1344.8/(T+219.482))/(760*1.2)
k1=10^(6.90565-1211.033/(T+220.79))/(760*1.2)
x1=1-x2
Initial Conditions:
x2(0)=0.4
L(0)=100
T(0)=95.5851
Final Value:
x2(f)=0.8
```

The POLYMATH solution to this problem is essentially the same as that found in **Approach 1**.

### Polymath Problem 9 Solution

Introduction of the above equations including the numerical values of the parameter provided in the problem statement into the POLYMATH program yields

#### Equations

```

d(x)/d(W)=-rA/FA0
d(T)/d(W)=(.8*(Ta-T)+rA*delH)/(CPA*FA0)
d(y)/d(W)=-0.015*(1-.5*x)*(T/450)/(2*y)
Ta=500
delH=-40000
CPA=40
FA0=5
k=.5*exp((41800/8.314)*(1/450-1/T))
CA=.271*(1-x)*(450/T)/(1-.5*x)*y
CC=.271*.5*x*(450/T)/(1-.5*x)*y
Kc=25000*exp(delH/8.314*(1/450-1/T))
rA=-k*(CA^2-CC/Kc)

```

#### Initial Conditions:

```

W(0)=0
x(0)=0
T(0)=450
y(0)=1
W(f)=20

```

(a) The requested plot for part (a) is shown in Figure PM-(7) where there is a rapid increase in conversion and temperature within the reactor at approximately the midpoint of the catalyst bed. The bed pressure drop is enhanced by the increased temperature and reduced pressure even though the number of moles is decreasing.

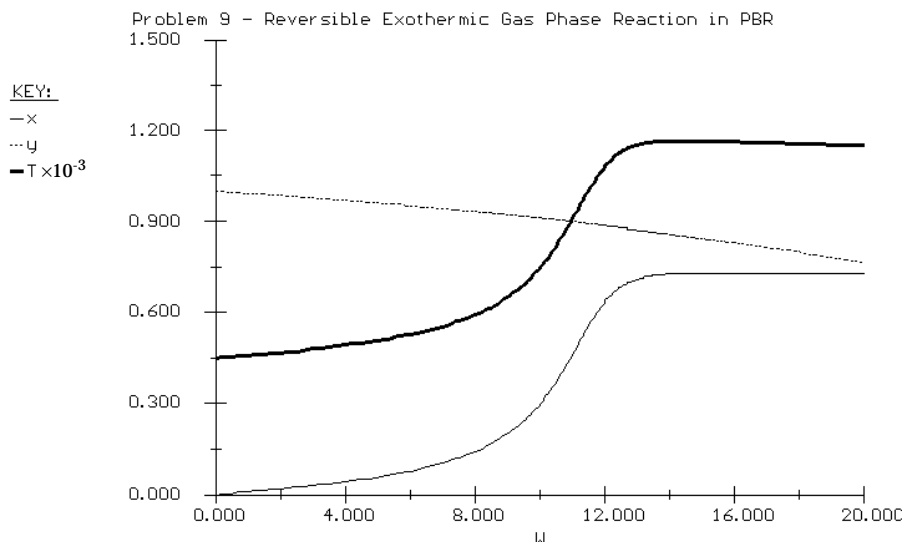


Figure PM-7 Conversion, Reduced Pressure, and Temperature Profiles in Catalytic Reactor

(b) This rapid increase is due to the exothermic reaction rapidly accelerating due to the increasing temperature even though the reactant concentration falling. Equilibrium is rapidly achieved after this hot spot is achieved with the temperature and conversion only reducing slightly due to the exter-

nal heat transfer which tends to slightly cool the reactor as the reacting mixture continues toward the reactor exit.

(c) The concentration profiles shown in Figure PM-(8) reflect the net effects of reaction rate and changes in temperature and pressure within the reactor.

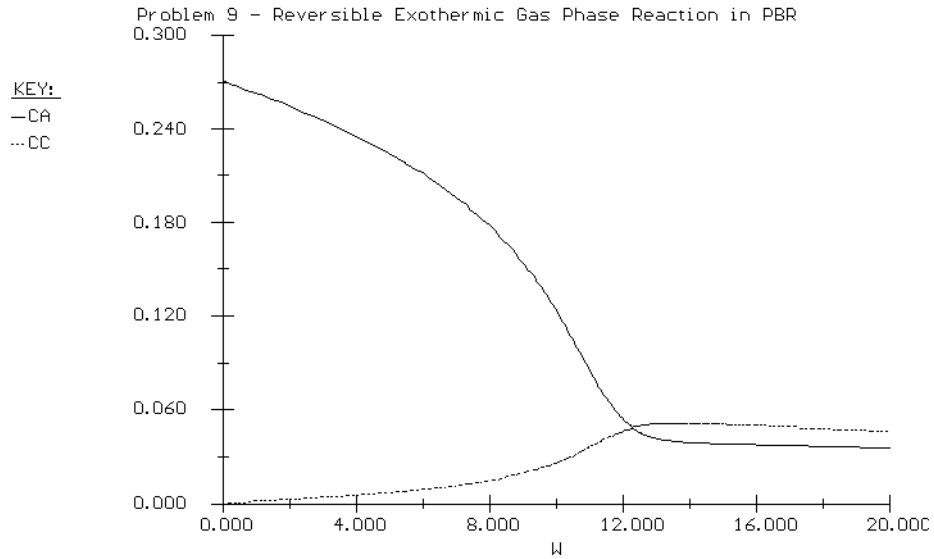


Figure PM-8 Concentration Profiles in Catalytic Reactor

### Polymath Problem 10 Solution

This problem requires the solution of Equations (40) and (42) through (47) which can be accomplished with the POLYMATH *Simultaneous Differential Equation Solver*. The step change in the inlet temperature can be introduced at  $t = 10$  by using the POLYMATH “if... then... else...” statement to provide the logic for a variable to change at a particular value of  $t$ . The generation of a step change at  $t = 10$ , for example, is accomplished by the POLYMATH program statement

```
step=if (t<10) then (0) else (1)
```

**(a) Open Loop Performance** The step down of 20°C in the inlet temperature at  $t = 10$  is implemented below in the equation set for the case where  $K_c = 0$  which gives the open loop response.

Equations:

```
d(T)/d(t)=(WC*(Ti-T)+q)/rhoVCp
d(T0)/d(t)=(T-T0-(taud/2)*dTdt)*2/taud
d(Tm)/d(t)=(T0-Tm)/taum
d(errsum)/d(t)=Tr-Tm
WC=500
rhoVCp=4000
taud=1
taum=5
Tr=80
Kc=0
tauI=2
```

```
step=if (t<10) then (0) else (1)
Ti=60+step*(-20)
q=10000+Kc*(Tr-Tm)+Kc/tauI*errsum
dTdt=(WC*(Ti-T)+q)/rhoVCp
```

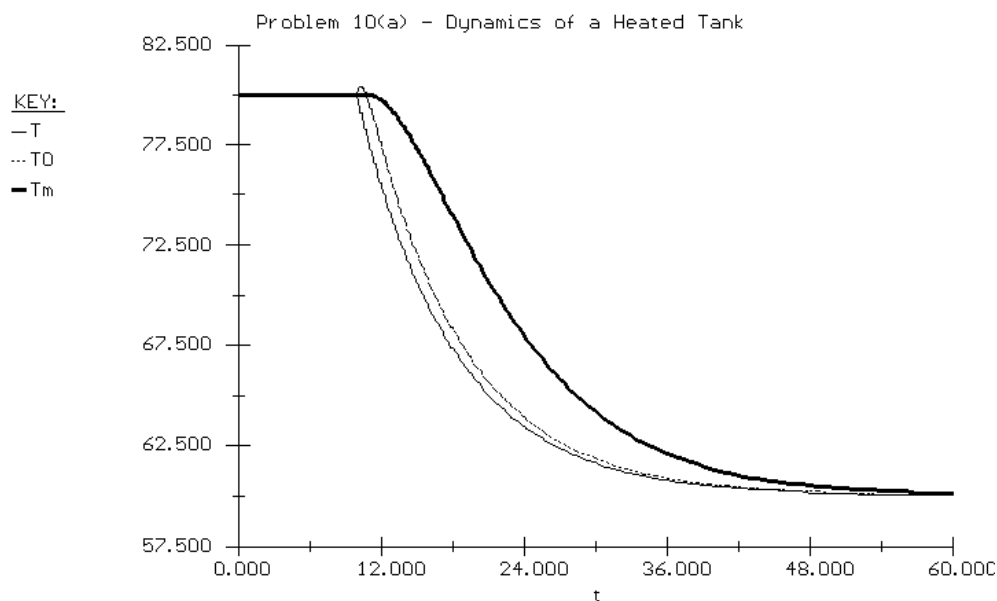
Initial Conditions:

```
t(0)=0
T(0)=80
T0(0)=80
Tm(0)=80
errsum(0)=0
t(f)=60
```

A plot of the temperatures  $T$ ,  $T_0$  and  $T_m$  as generated by POLYMATH is given in Figure PM-(9) which also verifies the steady state operation for  $t < 10$  min as there is no change in any of the temperature values. Since it is difficult to determine that the Padé approximation for a short time delay is working from a plot, the POLYMATH option to “output data to a file” has been used to prepare Table PM-(8). This table indicates that there is good agreement between  $T$  (at any  $t$ ) and  $T_0$  (one minute

**Table PM-8** Dead Time Generation by Padé Approximation

Time $t$ min	$T$ °C	$T_0$ °C	Time $t$ min	$T$ °C	$T_0$ °C
9	80	80	15	75.352614	76.066236
10	80	80	16	74.723801	75.353633
11	78.824973	79.821167	17	74.168787	74.724624
12	77.788008	78.801988	18	73.678794	74.1693
13	76.873024	77.786106	19	73.246627	73.679511
14	76.065326	76.873588	20	72.865268	73.247303



**Figure PM-9** Open Loop Response to Step Down in Inlet Feed Temperature at  $t = 10$  min

later) with more error at the initiation of the step change. This verifies that the Padé approximation for dead time is providing the one minute time delay.

**(b) Closed Loop Performance** The closed loop performance of the PI controller requires the change of  $K_c$  from zero in part (a) to the baseline proportional gain of 50. This simple change results in the temperature transients shown in Figure PM-(10).

**(c) Closed Loop Performance for  $K_c = 500$**  The increase of a factor of 10 in the proportional gain from the baseline case gives the unstable result plotted in Figure PM-(11). This is clearly an undesirable result.

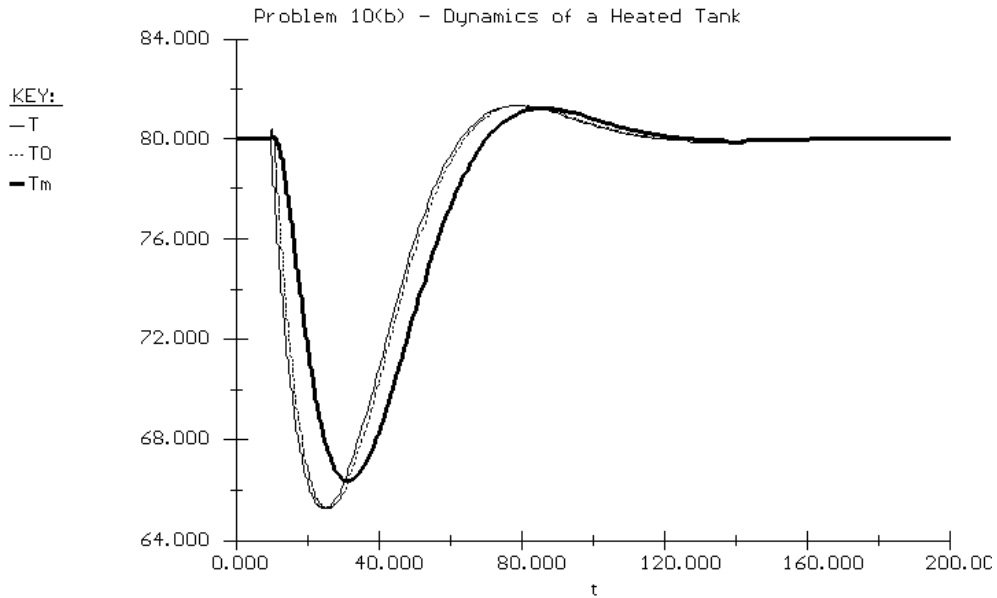


Figure PM-10 Closed Loop Response to Step Down in Inlet Feed Temperature at  $t = 10$  min.

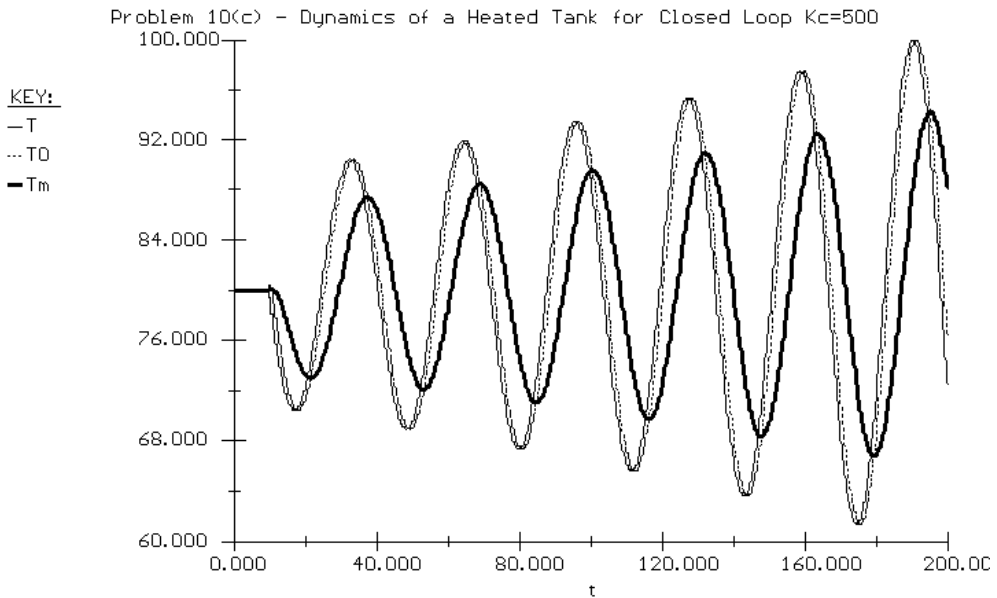
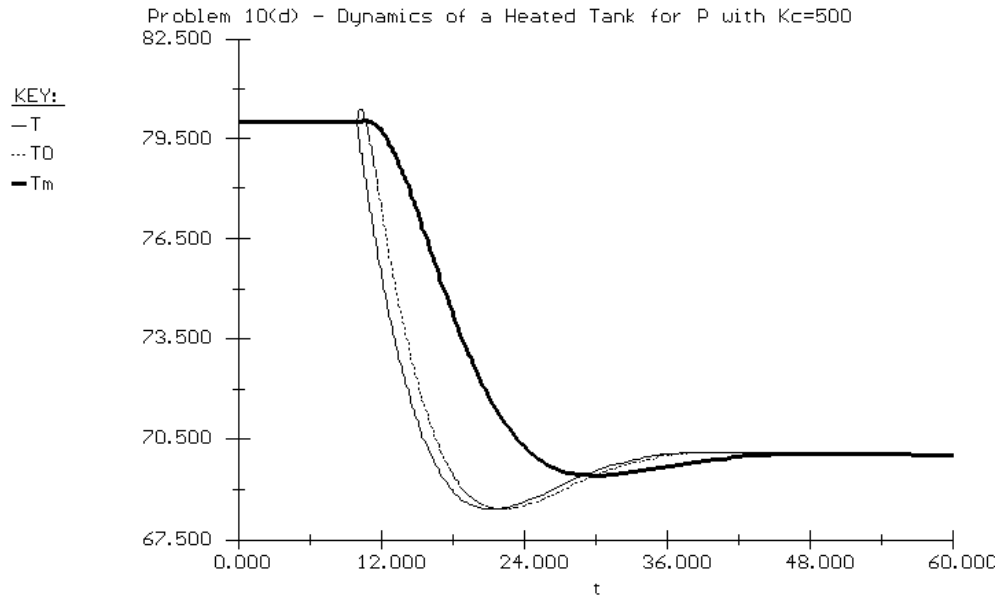


Figure PM-11 Closed Loop Response to Step Down in Inlet Feed Temperature at  $t = 10$  min for  $K_c = 500$ .

**(d) Closed Loop Performance for Only Proportional Control** The removal of the integral control action gives the stable result plotted in Figure PM-(12). Note that there is offset from the set point



**Figure PM-12** Closed Loop Response for only Proportional Control.

when the system returns to steady state operation. This is always the case for only proportional control, and the use of integral control allows the offset to be eliminated.



**(e) Closed Loop Performance with Limits on  $q$**  There are many times in control when limits must be established. In this example, the limits on  $q$  can be achieved by a POLYMATH “if... then... else...” statement which can be utilized as shown below

```
qlim=if(q<0)then(0)else(if(q>=2.6*10000)then(2.6*10000)else (q))
```

The complete POLYMATH equation set for part (e) of this problem is

Equations:

```
d(T)/d(t)=(WC*(Ti-T)+qlim)/rhoVCp
```

```
d(T0)/d(t)=(T-T0-(taud/2)*dTdt)*2/taud
```

```
d(Tm)/d(t)=(T0-Tm)/taum
```

```
d(errsum)/d(t)=Tr-Tm
```

```
WC=500
```

```
Ti=60
```

```
rhoVCp=4000
```

```
taud=1
```

```
taum=5
```

```
Kc=5000
```

```
tauI=2
```

```
step=if (t<10) then (0) else (1)
```

```
Tr=80+step*(10)
```

```
q=10000+Kc*(Tr-Tm)
```

```
qlim=if(q<0)then(0)else(if(q>=2.6*10000)then(2.6*10000)else (q))
```

```
dTdt=(WC*(Ti-T)+qlim)/rhoVCp
```

Initial Conditions:

```
t(0)=0
```

```
T(0)=80
```

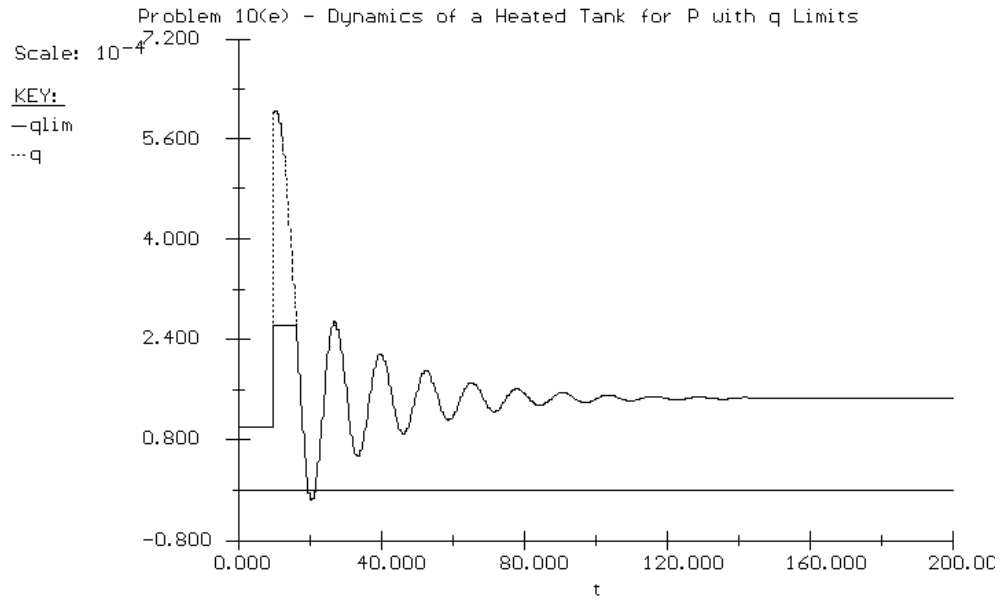
```
T0(0)=80
```

```
Tm(0)=80
```

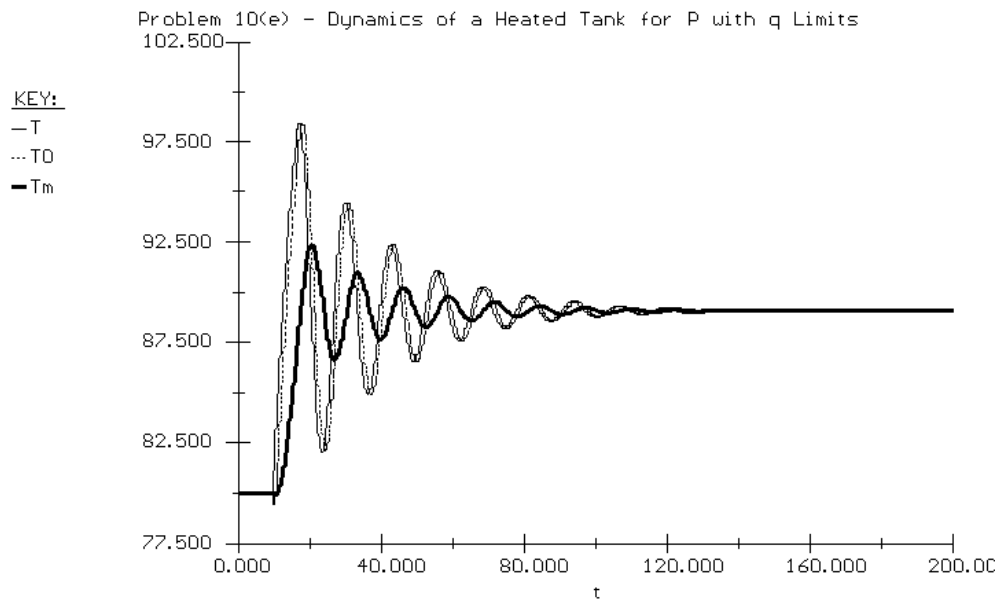
```
errsum(0)=0
```

```
t(f)=200
```

The values of  $q$  and  $q_{lim}$  plotted in Figure PM-(13) indicate that this proportional controller has wide oscillations before settling to a steady state, and the limits imposed on  $q_{lim}$  are evident. The corresponding plots of the system temperatures are presented in Figure PM-(14).



**Figure PM-13** Closed Loop Response for only Proportional Control.



**Figure PM-14** Closed Loop Response for only Proportional Control.



