Enhancing Problem Problem Solving in Excel and MATLAB with Polymath 6.1*

Michael B. Cutlip[a], Mordechai Shacham[b] and Michael Elly[c]

[a]Dept. Chem. Eng., University of Connecticut, Storrs, CT, USA

[b]Dept. Chem. Eng., Ben-Gurion University, Beer-Sheva, Israel

[c]Intel Corp., Qiryat Gat, Israel

*Paper 206g, presented at the 2006 AIChE Annual Meeting, San Francisco, November 12-17, 2006

Outline

- Assessing the *computing needs* in undergraduate and graduate ChE education*
- Selecting *a set of software packages* to fulfill the computing needs
- Demonstration of the *advantages and disadvantages* of the various software packages by solving a benchmark problem.
- *Combining* the various software *packages* for more effective and efficient problem solving
- Conclusions

*See also Edgar, T. F., “Enhancing the Undergraduate Computing Experience”, *Chem. Eng. Edu.*, 40 (3), 231(2006)

Assessing the computing needs in undergraduate and graduate ChE education

Ten Representative Problems from the ChE curriculum were prepared for the ASEE Chemical Engineering Summer School held in Snowbird, Utah on August 13, 1997.

The ten problems were solved using six software packages:

Excel - Edward M. Rosen
Maple - Ross Taylor
Mathematica - H. Eric Nuttall
Mathcad - John J. Hwalek, University of Maine
MATLAB - Joseph Brule, John Widmann, Tae Han, and Bruce Finlayson
POLYMATH - Michael B. Cutlip, and Mordechai Shacham

Assessing the computing needs in undergraduate and graduate ChE education

➢The ASEE summer school presentations identified advantages and disadvantages of the various software packages and lead to the conclusion (see: *Comput. Appl. Eng. Educ*. 6: 169-180, 1998) that there is a need to use a set of several packages in ChE education.

➢One possible set of packages with a CACHE* connection:

- POLYMATH* – Serves as a general, easy to use, user friendly *Problem Solving Environment* (PSE)

- Excel – Serves as a *spreadsheet based PSE* (the PSE most widely used by practicing engineers)

- MATLAB – Serves as a *programming language* and symbolic manipulation tool.

POLYMATH 6.1 Educational

➢A proven computational system (PSE, Problem Solving Environment) that supports advanced problem solving in chemical, biochemical and other engineering disciplines.

➢Capabilities include Linear Equations, Nonlinear Equations, Differential Equations, Data Analysis and Regression.

➢Low-cost site licenses are provided through the CACHE Corporation for academic departments. These licenses enable use in computer labs and distribution of individual copies to all student, faculty, and staff.

➢Individual use educational and professional versions are also available.

➢POLYMATH has a *minimal learning curve* and provides *extensive error checking* during problem entry that leads to great efficiency in problem solution.

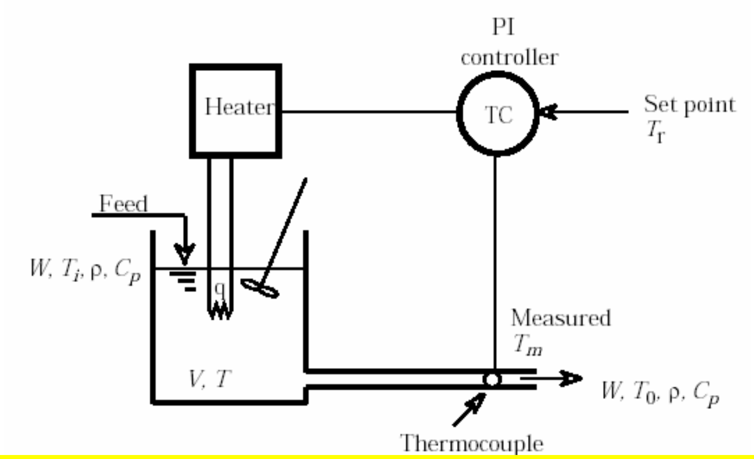
Limitations of the Individual Packages

➢In POLYMATH, the easiness of use and user friendliness dictate a *fixed set of capabilities and options*. When a particular problem does not fit into the options provided, repeated manual rerunning of the problem may be necessary.

➢In Excel, the PSE options were added in a late stage of development. Thus the *problem specification* (using cell addresses instead of variable names) *is difficult* and the documentation of the problem statement is hard to understand.

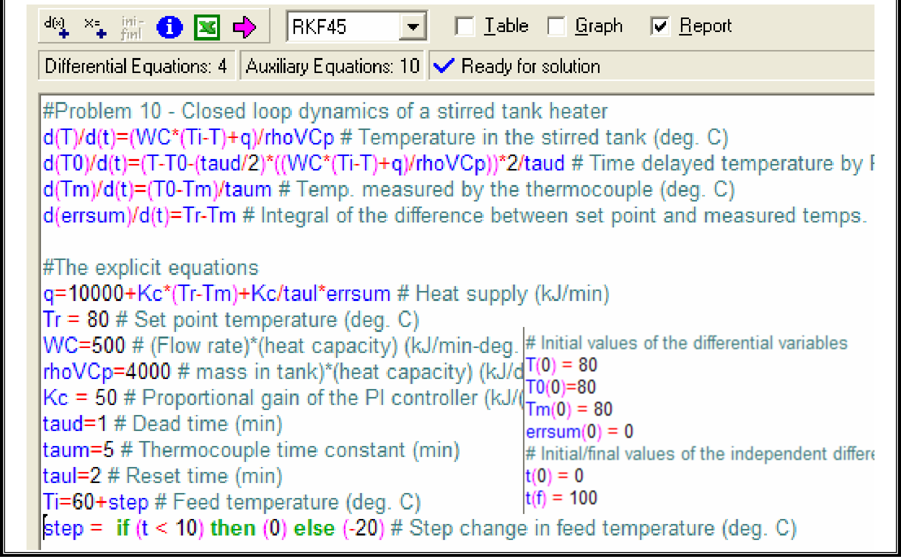
➢In MATLAB (as in other programming languages), it is the user's responsibility to *take care of many technical details* of the solution that can be more efficiently done by the computer (like arranging the equations in the calculation order). The error messages may not be clear enough for a novice user.

DYNAMICS OF A HEATED TANK WITH PI TEMPERATURE CONTROL* (DHT) – An Example



*Problem No. 10 presented in the session of “The Use of Mathematical Software Packages in ChE” in the ASEE ChE Summer School, Snowbird, Utah, 1997.

DHT Example – POLYMATH MODEL

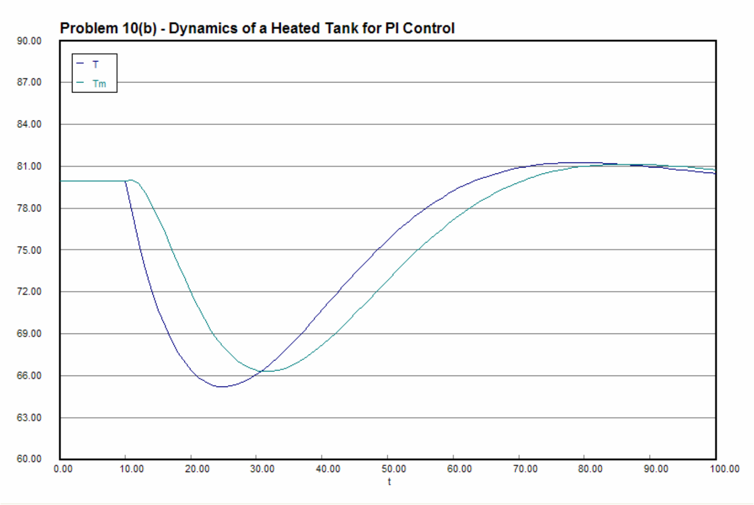


Differential Equations: 4 | Auxiliary Equations: 10 | ✓ Ready for solution

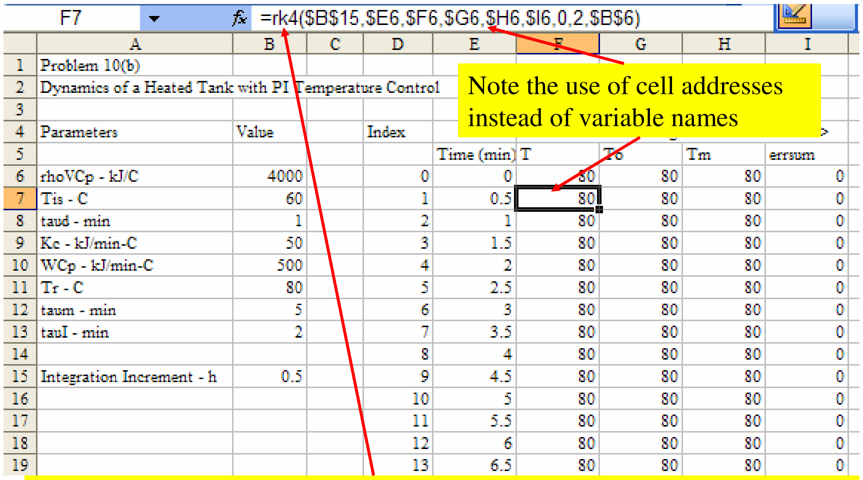
```
#Problem 10 - Closed loop dynamics of a stirred tank heater
d(T)/d(t)=(WC*(Ti-T)+q)/rhoVCp # Temperature in the stirred tank (deg. C)
d(T0)/d(t)=(T-T0-(taud/2)*((WC*(Ti-T)+q)/rhoVCp))*2/taud # Time delayed temperature by I
d(Tm)/d(t)=(T0-Tm)/taum # Temp. measured by the thermocouple (deg. C)
d(errsum)/d(t)=Tr-Tm # Integral of the difference between set point and measured temps.

#The explicit equations
q=10000+Kc*(Tr-Tm)+Kc/taul*errsum # Heat supply (kJ/min)
Tr = 80 # Set point temperature (deg. C)
WC=500 # (Flow rate)*(heat capacity) (kJ/min-deg.
rhoVCp=4000 # mass in tank)*(heat capacity) (kJ/d
Kc = 50 # Proportional gain of the PI controller (kJ/(
taud=1 # Dead time (min)
taum=5 # Thermocouple time constant (min)
taul=2 # Reset time (min)
Ti=60+step # Feed temperature (deg. C)
step =  if (t < 10) then (0) else (-20) # Step change in feed temperature (deg. C)
```

```
# Initial values of the differential variables
T(0) = 80
T0(0)=80
Tm(0) = 80
errsum(0) = 0
# Initial/final values of the independent differe
t(0) = 0
t(f) = 100
```

DHT Example – POLYMATH SOLUTION



DHT Example – Excel SOLUTION



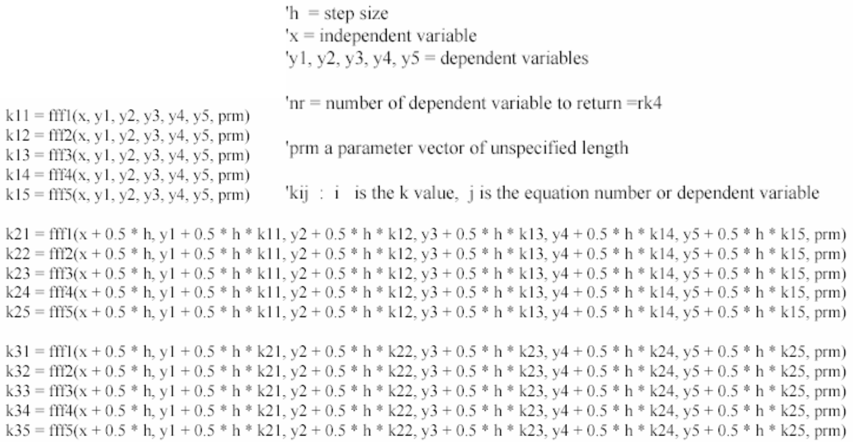
=rk4($B$15,$E6,$F6,$G6,$H6,$I6,0,2,$B$6)

	A	B	C	D	E	F	G	H	I
1	Problem 10(b)								
2	Dynamics of a Heated Tank with PI Temperature Control								
3									
4	Parameters	Value		Index					
5					Time (min)	T	To	Tm	errsum
6	rhoVCp - kJ/C	4000		0	0	80	80	80	0
7	Tis - C	60		1	0.5	80	80	80	0
8	taud - min	1		2	1	80	80	80	0
9	Kc - kJ/min-C	50		3	1.5	80	80	80	0
10	WCp - kJ/min-C	500		4	2	80	80	80	0
11	Tr - C	80		5	2.5	80	80	80	0
12	taum - min	5		6	3	80	80	80	0
13	tauI - min	2		7	3.5	80	80	80	0
14				8	4	80	80	80	0
15	Integration Increment - h	0.5		9	4.5	80	80	80	0
16				10	5	80	80	80	0
17				11	5.5	80	80	80	0
18				12	6	80	80	80	0
19				13	6.5	80	80	80	0

Note the use of cell addresses instead of variable names

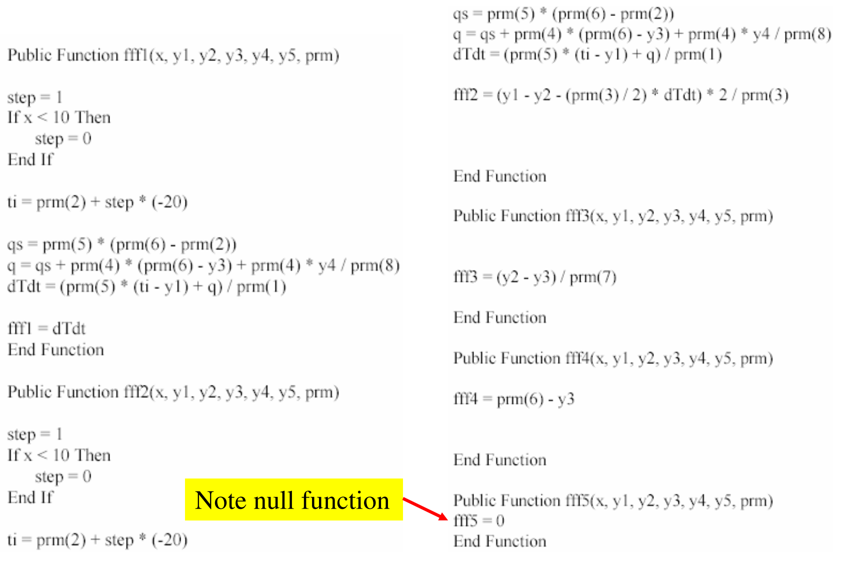
The worksheet does not provide enough information regarding the model used. It cannot serve as stand alone documentation

Visual Basic Program for Fixed Step Size RK Integration by E. M. Rosen



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

'nr = number of dependent variable to return =rk4

'prm a parameter vector of unspecified length

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

k11 = fff1(x, y1, y2, y3, y4, y5, prm)
k12 = fff2(x, y1, y2, y3, y4, y5, prm)
k13 = fff3(x, y1, y2, y3, y4, y5, prm)
k14 = fff4(x, y1, y2, y3, y4, y5, prm)
k15 = fff5(x, y1, y2, y3, y4, y5, prm)

k21 = fff1(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)
k22 = fff2(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)
k23 = fff3(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)
k24 = fff4(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)
k25 = fff5(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)

k31 = fff1(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)
k32 = fff2(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)
k33 = fff3(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)
k34 = fff4(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)
k35 = fff5(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)
```

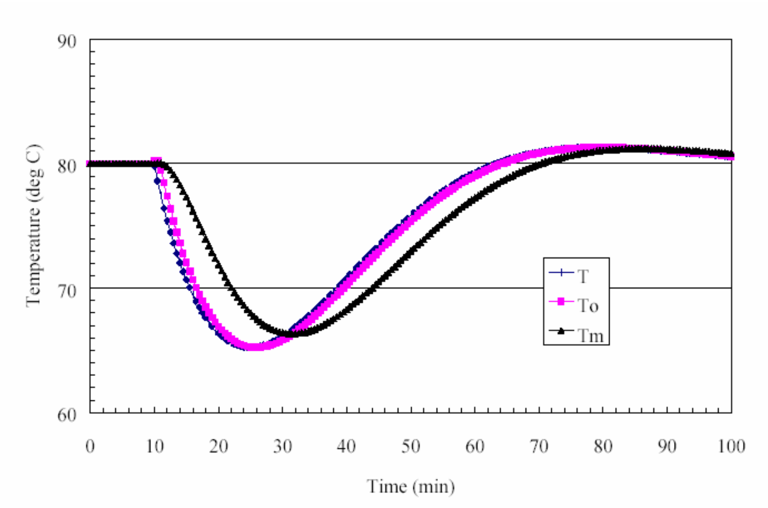
DHT Example – Visual Basic Model (functions)



```
Public Function fff1(x, y1, y2, y3, y4, y5, prm)

step = 1
If x < 10 Then
     step = 0
End If

ti = prm(2) + step * (-20)

qs = prm(5) * (prm(6) - prm(2))
q = qs + prm(4) * (prm(6) - y3) + prm(4) * y4 / prm(8)
dTdt = (prm(5) * (ti - y1) + q) / prm(1)

fff1 = dTdt
End Function

Public Function fff2(x, y1, y2, y3, y4, y5, prm)

step = 1
If x < 10 Then
     step = 0
End If

ti = prm(2) + step * (-20)
qs = prm(5) * (prm(6) - prm(2))
q = qs + prm(4) * (prm(6) - y3) + prm(4) * y4 / prm(8)
dTdt = (prm(5) * (ti - y1) + q) / prm(1)

fff2 = (y1 - y2 - (prm(3) / 2) * dTdt) * 2 / prm(3)


End Function

Public Function fff3(x, y1, y2, y3, y4, y5, prm)


fff3 = (y2 - y3) / prm(7)

End Function

Public Function fff4(x, y1, y2, y3, y4, y5, prm)

fff4 = prm(6) - y3


End Function

Public Function fff5(x, y1, y2, y3, y4, y5, prm)
fff5 = 0
End Function
```

Note null function

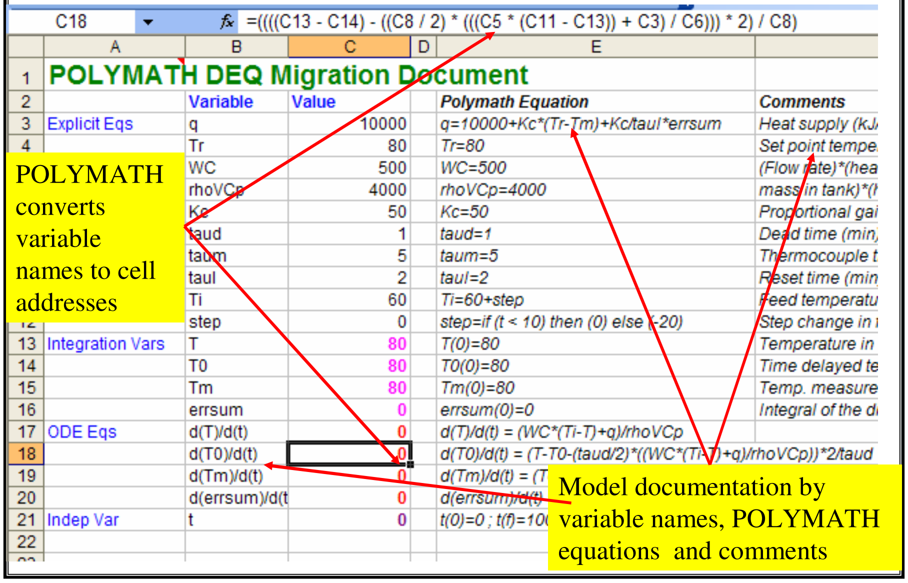
DHT Example – Excel Solution



Combining the Packages for Greater Efficiency in Effective Problem Solving – POLYMATH and Excel

1. The problem is entered and solved in its basic form using POLYMATH.
2. POLYMATH 6.1 automatically converts the problem into a well documented Excel spreadsheet, ready for solution.
3. The problem can be modified within Excel for parametric studies (using “two input data tables, for example)

DHT Example – Excel Model Generated by POLYMATH



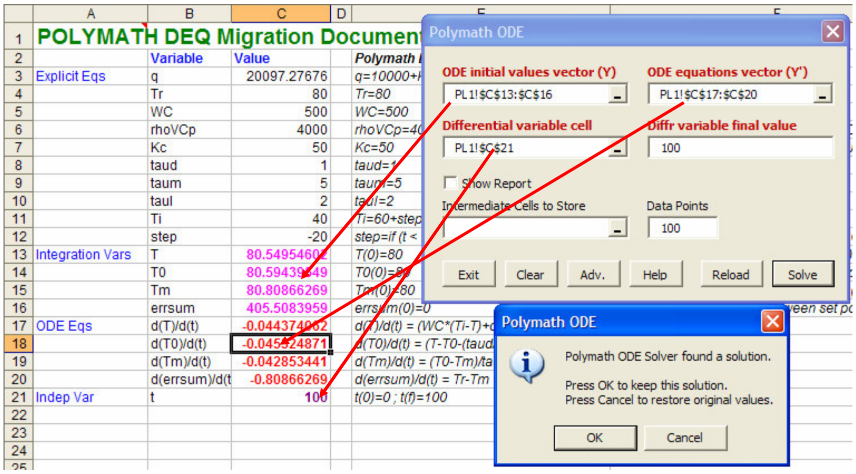
C18 =((((C13 - C14) - ((C8 / 2) * (((C5 * (C11 - C13)) + C3) / C6))) * 2) / C8)

	A	B	C	D	E	
1	POLYMATH DEQ Migration Document					
2		Variable	Value		*Polymath Equation*	*Comments*
3	Explicit Eqs	q	10000		*q=10000+Kc*(Tr-Tm)+Kc/taul*errsum*	*Heat supply (kJ*
4		Tr	80		*Tr=80*	*Set point tempe*
		WC	500		*WC=500*	*(Flow rate)*(hea*
		rhoVCp	4000		*rhoVCp=4000*	*mass in tank)*(*
		Kc	50		*Kc=50*	*Proportional gai*
		taud	1		*taud=1*	*Dead time (min,*
		taum	5		*taum=5*	*Thermocouple t*
		taul	2		*taul=2*	*Reset time (min*
		Ti	60		*Ti=60+step*	*Feed temperatu*
		step	0		*step=if (t < 10) then (0) else (-20)*	*Step change in*
13	Integration Vars	T	80		*T(0)=80*	*Temperature in*
14		T0	80		*T0(0)=80*	*Time delayed te*
15		Tm	80		*Tm(0)=80*	*Temp. measure*
16		errsum	0		*errsum(0)=0*	*Integral of the d*
17	ODE Eqs	d(T)/d(t)	0		*d(T)/d(t) = (WC*(Ti-T)+q)/rhoVCp*	
18		d(T0)/d(t)	0		*d(T0)/d(t) = (T-T0-(taud/2)*((WC*(Ti-T)+q)/rhoVCp))*2/taud*	
19		d(Tm)/d(t)	0		*d(Tm)/d(t) = (T*	
20		d(errsum)/d(t	0		*d(errsum)/d(t)*	
21	Indep Var	t	0		*t(0)=0 ; t(f)=10*	
22						

POLYMATH converts variable names to cell addresses

Model documentation by variable names, POLYMATH equations and comments

DHT Example – Integrating by the Unique POLYMATH ODE solver for Excel



	A	B	C	D	E
1	POLYMATH DEQ Migration Document				
2		Variable	Value		*Polymath*
3	Explicit Eqs	q	20097.27676		*q=10000+*
4		Tr	80		*Tr=80*
5		WC	500		*WC=500*
6		rhoVCp	4000		*rhoVCp=4*
7		Kc	50		*Kc=50*
8		taud	1		*taud=1*
9		taum	5		*taum=5*
10		taul	2		*taul=2*
11		Ti	40		*Ti=60+step*
12		step	-20		*step=if (t <*
13	Integration Vars	T	80.54954602		*T(0)=80*
14		T0	80.59439849		*T0(0)=80*
15		Tm	80.80866269		*Tm(0)=80*
16		errsum	405.5083959		*errsum(0)=0*
17	ODE Eqs	d(T)/d(t)	-0.044374062		*d(T)/d(t) = (WC*(Ti-T)+*
18		d(T0)/d(t)	-0.045524871		*d(T0)/d(t) = (T-T0-(taud*
19		d(Tm)/d(t)	-0.042853441		*d(Tm)/d(t) = (T0-Tm)/ta*
20		d(errsum)/d(t	-0.80866269		*d(errsum)/d(t) = Tr-Tm*
21	Indep Var	t	100		*t(0)=0 ; t(f)=100*
22					
23					
24					

Polymath ODE

ODE initial values vector (Y): PL1!$C$13:$C$16

ODE equations vector (Y'): PL1!$C$17:$C$20

Differential variable cell: PL1!$C$21

Diffr variable final value: 100

☐ Show Report

Intermediate Cells to Store:

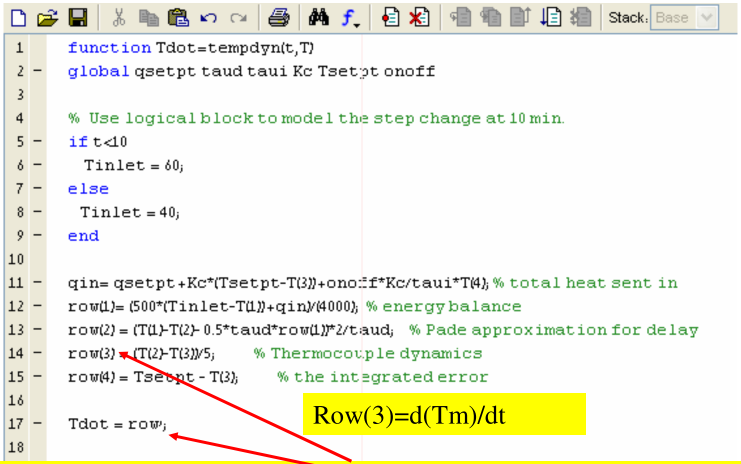
Data Points: 100

Polymath ODE

Polymath ODE Solver found a solution.

Press OK to keep this solution.
Press Cancel to restore original values.

DHT Example – MATLAB Model (function)

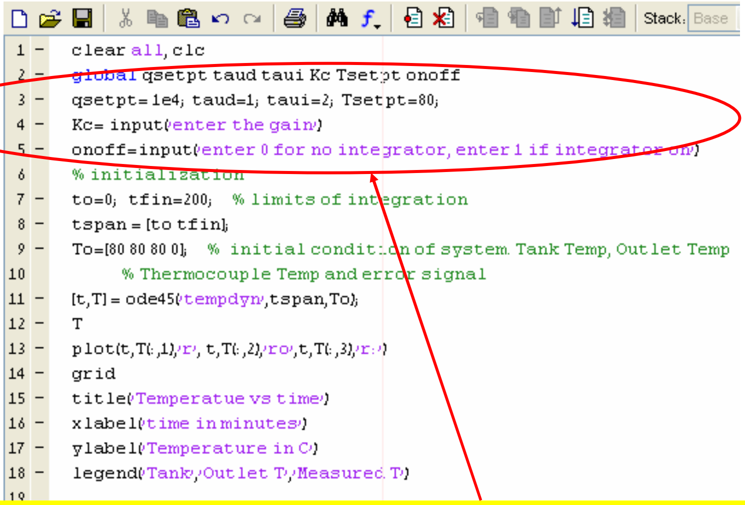


```
function Tdot=tempdyn(t,T)
global qsetpt taud taui Kc Tsetpt onoff

% Use logical block to model the step change at 10 min.
if t<10
  Tinlet = 60;
else
 Tinlet = 40;
end

qin= qsetpt+Kc*(Tsetpt-T(3))+onoff*Kc/taui*T(4); % total heat sent in
row(1)= (500*(Tinlet-T(1))+qin)/(4000); % energy balance
row(2) = (T(1)-T(2)- 0.5*taud*row(1))*2/taud;  % Pade approximation for delay
row(3) = (T(2)-T(3))/5;      % Thermocouple dynamics
row(4) = Tsetpt - T(3);      % the integrated error

Tdot = row;
```

Row(3)=d(Tm)/dt

Note the loss of clarity because of the need to define arrays of variables (instead of using variable names) and the special syntax requirements.

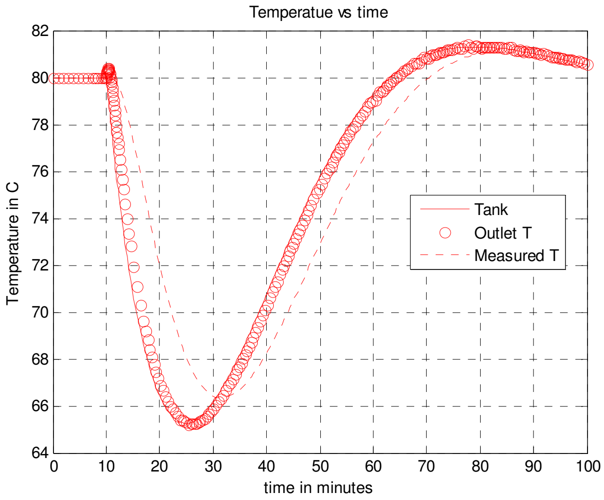
DHT Example – MATLAB "Main Program"



```
clear all, clc
global qsetpt taud taui Kc Tsetpt onoff
qsetpt=1e4; taud=1; taui=2; Tsetpt=80;
Kc= input('enter the gain')
onoff=input('enter 0 for no integrator, enter 1 if integrator on')
% initialization
to=0; tfin=200;  % limits of integration
tspan = [to tfin];
To=[80 80 80 0];   % initial condition of system. Tank Temp, Outlet Temp
       % Thermocouple Temp and error signal
[t,T] = ode45('tempdyn',tspan,To);
T
plot(t,T(:,1),'r', t,T(:,2),'ro',t,T(:,3),'r:')
grid
title('Temperatue vs time')
xlabel('time in minutes')
ylabel('Temperature in C')
legend('Tank','Outlet T','Measured T')
```

Note many problem specific components in the "Main Program"

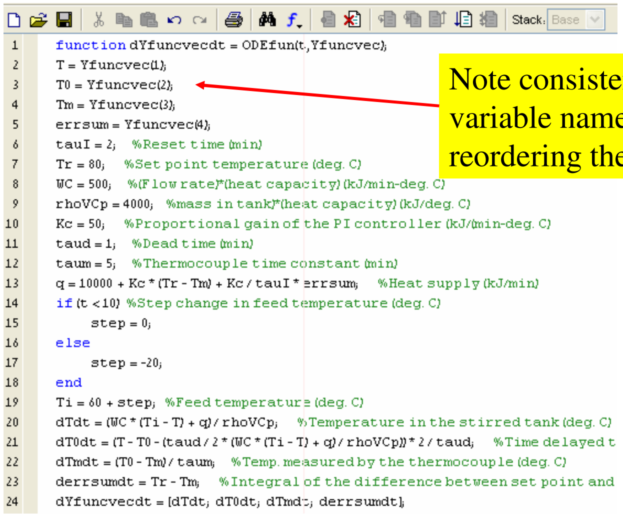
DHT Example – MATLAB Solution



Combining the Packages for Greater Efficiency in Effective Problem Solving – POLYMATH and MATLAB

1. The problem is entered and solved in its basic form using POLYMATH.
2. POLYMATH 6.1 automatically converts the problem into a MATLAB function and provides a template *m* file to run this function.
3. The problem can be modified within MATLAB for parametric studies, optimization, control etc. (using *for* and *while* statements, for example)

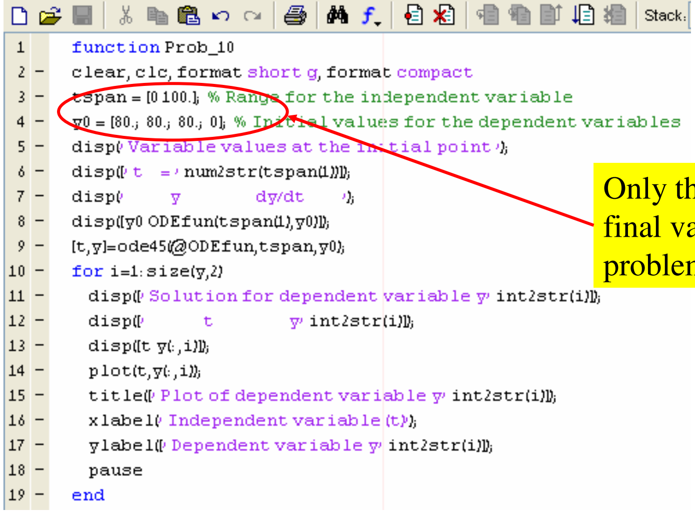
DHT Example – MATLAB Model (function) Generated by POLYMATH.



```
function dYfuncvecdt = ODEfun(t,Yfuncvec);
T = Yfuncvec(1);
T0 = Yfuncvec(2);
Tm = Yfuncvec(3);
errsum = Yfuncvec(4);
tauI = 2;   %Reset time (min)
Tr = 80;   %Set point temperature (deg. C)
WC = 500;   %(Flow rate)*(heat capacity) (kJ/min-deg. C)
rhoVCp = 4000;   %mass in tank)*(heat capacity) (kJ/deg. C)
Kc = 50;   %Proportional gain of the PI controller (kJ/(min-deg. C)
taud = 1;   %Dead time (min)
taum = 5;   %Thermocouple time constant (min)
q = 10000 + Kc * (Tr - Tm) + Kc / tauI * errsum;   %Heat supply (kJ/min)
if (t < 10) %Step change in feed temperature (deg. C)
    step = 0;
else
    step = -20;
end
Ti = 60 + step;  %Feed temperature (deg. C)
dTdt = (WC * (Ti - T) + q) / rhoVCp;   %Temperature in the stirred tank (deg. C)
dT0dt = (T - T0 - (taud / 2 * (WC * (Ti - T) + q) / rhoVCp)) * 2 / taud;   %Time delayed t
dTmdt = (T0 - Tm) / taum;   %Temp. measured by the thermocouple (deg. C)
derrsumdt = Tr - Tm;   %Integral of the difference between set point and
dYfuncvecdt = [dTdt; dT0dt; dTmdt; derrsumdt];
```

Note consistent use of variable names and reordering the equations

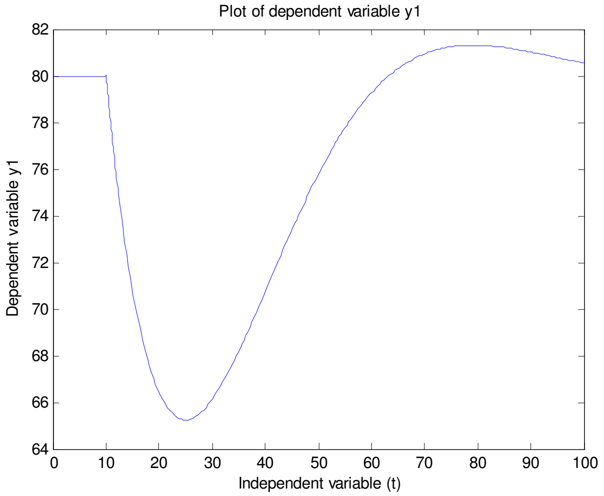
DHT Example – MATLAB "Main Program" Available in the POLYMATH "Help"



```
function Prob_10
clear, clc, format short g, format compact
tspan = [0 100.]; % Range for the independent variable
y0 = [80.; 80.; 80.; 0]; % Initial values for the dependent variables
disp('Variable values at the initial point');
disp([' t   = ' num2str(tspan(1))]);
disp('      y            dy/dt      ');
disp([y0 ODEfun(tspan(1),y0)]);
[t,y]=ode45(@ODEfun,tspan,y0);
for i=1:size(y,2)
  disp([' Solution for dependent variable y' int2str(i)]);
  disp(['       t          y' int2str(i)]);
  disp([t y(:,i)]);
  plot(t,y(:,i));
  title([' Plot of dependent variable y' int2str(i)]);
  xlabel(' Independent variable (t)');
  ylabel([' Dependent variable y' int2str(i)]);
  pause
end
```

Only the initial and final values are problem specific

DHT Example – MATLAB Solution Using the Program Generated by POLYMATH



Conclusions

The Excel worksheets and MATLAB functions generated by POLYMATH are *well documented problem models of uniform structure* obtained by minimal effort on behalf of the user.

These models can be easily modified to carry out *parametric studies, optimization, control*, etc., automatically. Thus the advantages of the various programs can be fully utilized while the required effort is reduced.

The POLYMATH conversion utility provides also considerable *educational benefits* by enabling learning of Excel and MATLAB by modifying existing spreadsheets or programs. *Learning by modification* may prevent frustration caused by repeated failures to obtain a working program when programming from scratch.

Inexpensive site licenses for POLYMATH 6.1 are available from the CACHE Corporation

Educational Site License Information
The site license allows an academic department to use POLYMATH in all computer labs and to make individual copies available to all students, faculty, and staff for installation and use on their computers. POLYMATH can be provided by computer networks.

The current site license costs for POLYMATH are $175 and $125 for each successive year thereafter. CACHE nonmember institution rates are an initial $200 and a $150 annual fee for subsequent years. These fees cover any updates or new versions.