REFERENCES

Analla, V., Humberto, E.A. and Williams, J.J.A. Ind. Eng.Chem.Prod,Res.Dev. 23(1984) : 375-379.

Aranguren, M.I., Borrajo, J. and Williams, R.J.J. Ind. Eng.Chem.Prod.Res.Dev. 23(1984) : 370-374.

Arthur, A.F. and Ralph, G.P. Kinetics and Mechanism. New York: Jonh Wiley & Sons, Inc., 1953.

Constantinides Alkis. Applied Numerical Methods with Personal Computers. Singapore: McGraw-Hill,1987.

Debing, L.M., Murray, G.E. and Schatz, R.J. Ind.Eng. Chem. 44(1952) : 356-359.

DeJong, J.I. and DeJonge, J. Rec.Trav.Chim. 72(1952) : 597.

________. Rec.Trav.Chim. 75(1956) : 1289-1301.

De Jonge, J. and Dijkstra, R. Rec.Trav.Chim. 76(1957) : 93-100.

Dijkstra, R., De Jonge, J. and Lammers, M.F. Rec.Trav. Chim. 81(1962) : 285-296.

Edgar, T.F. and Himmelblau, D.M. Optimization of Chemical Process: Chapter 5-7, McGraw-Hill, 1989.

Fraser, D.A., Hall, R.w. and Raum, A.L.J. J.Appl.Chem. 7 December (1957):

Freeman, J.H. and Lewis, C.W. J.Am.Chem.Soc. 76(1954) : 2080-2087.

Higginbottom, H.P., Culbertson, H.M. and Woodbrey, J.C. Analytical Chemistry. 37(1965): 102-1206.

Himmelblau, D.M. and Bischoff, K.B. Process Analysis and Simulation. Willey, New York, 1947.

________. Process Analysis by Statistical Methods. Newyork, John Wiley&Sons, Inc., 1970.

Hughes, E.D. and Ingold, R.C. J.Chem.Soc. 244(1935).

Iliceto, A., Bezzi, S., Dallaporta, N. and Giacommetti, G. Gazz. Chim, Ital. 81(1951):915

Kirk, R.E. and Othmer, D.F. Encyclopedia of Chemical Technology. Vol. 17. New York: THE Interscience Encyclopedia, Inc., 1984.

Knop, L.A. and Pilato, A. Phenolic Resin. New Jersy: Temecon Group International, Inc., 1985.

Kumar, A., Kulshreshtra, A.K. and Gupta, S.K. Polymer. 21(1980) : 2461.

________. Polymer. 23(1982) : 215.

________. J. Appl. Polym. Sci. 27(1982) : 3393-3405.

Laidler, K.J. and Eyring, H. Ann.Ny.Acad.Sci. 39(1940): 303.

Lindfield, G. and Penny, J. Numerical methods Using MATLAB. Ellis Horwood Limited, 1955.

Malhotra, H.C. and Avinash. J. Appl. Polym. Sci. 20 (1986): 2461.

Martin, R.W. The Chemistry of Phenolic Resins. New York: J. Wiley, 1956.

Pal, P.K., Kumar, A. and Gupta, S.K. Polymer. 22(1981): 1669-1704.

Peer, H.G. Rec.Trav.Chim. 78(1959) : 851-863.

Peter et.al, Encyclopedia of Polymer Science and Engineering. vol 11, John Wiley&Sons,N.Y.,1988.

Sprengling, G.R. and Lewis, C.W. J. Am. Chem. Soc. 75 (1953):5709.

Scatchard, G. Chem. Revs. 8(1931):321.

The Math Works, MATLAB-Reference Guide. The Math Works, Inc, 1992.

Walker, J.F. J. Phys. Chem. 35(1934): 1104.

William, H.P., Saul, A.T., William, T.V. and Brian, P.F. Numerical Recipes.The art of scientific Computing

Fortran version). Cambridge University Press,1992

Yeddanapalli, L.M. and Gopalakrishna, V.V. Makromol. Chem. 32(1959) : 112-123.

Zavitsas, A.A. J. Am. Chem. Soc: Div. Org. Coatings and Plastic. 26(1966):93-106.

________. J. Am. Chem. Soc: Div. Org. Coatings and plastic, 27(1967): 100-113.

________. Journal of Polymer Science: Part A-1., 6(1968): 2533-2540.

________. Journal of Polymer Science: Part A-1., 6(1968): 2541-2559.

APPENDIX A

REVIEW OF PARAMETER ESTIMATION METHODS FOR NONLINEAR MODEL

Introduction

Parameter estimation arises in fitting models containing several unknown parameters to experimental data through adjustment of these parameters. Model formaulation is not a unique process; many different formulations may be used to fit the data and optimize model parameters. Of particular concern are formulations that are sufficiently accurate to represent physical or chemical phenomena.

Model formulation can be grouped into linear and nonlinear. The term"nonlinear" as applied to models means the model is nonlinear in the parameters (coefficients) to be estimated. The nonlinear estimation problem appears as simply an optimization problem in parameter space.

Numerical Methods for Nonlinear Model

Suppose N' data points(X_i, Y_i) are being fitted with a nonlinear model,

$$y=y(x;b) \tag{A-1}$$

where b is vector of parameters to be estimated.

The objective is to minimize the sum of squares function,f(b) to obtain the desired parameter estimates.

$$f(b)=\sum_{i=1}^{n}(Y_i-y_i(x_i,b))^2 \tag{A-2}$$

The nonlinear estimation problem apppears as simply an optimization problem in parameter space. The optimization techniques fall into two broad calsses:(1) derivative-free methods (direct method) and (2) derivative methods (indirect method). If in the search for a minimum of f(b),the partial derivatives of f(b) must be calculated, then the method is derivative type;otherwise, the method will be termed a derivative-free method of estimation. There are five more effective optimization techniques, namely:

1. **Direct search method** Direct method do not require the use of derivatives in determining the search

direction. A direct search method simply selects a starting vector b^0, evaluates $f(b^0)$. Each b^0 is changed by $+\Delta b^0$ and, if $f(b)$ is improved, $b^0+\Delta b^0$ is adopted as a new estimate of b^1. If f is not improved, $b^0-\Delta b^0$ is tested. If no improvement is experienced for either $\pm\Delta b^0$,$b^1=b^0$. This process is continued to complete an "exploratory move". The new estimated parameters define a vector in parameter space that represents a successful direction to reduce f(b). A series of accelerating steps is made along this vector as long as f(b) is reduced. The magnitude of the pattern move in each coordinate derection is proportional to the number of prior successful moves in that direction. If f(b) is not improved by one of these pattern moves, a new exploratory move is made in order to define a new duccessful derection, the Δb are reduced gradually until either a new successful direction can be defined or each Δb becomes smaller than some predetermined tolerance. Failure to improve f(b) for a very small Δb indicates that a local optimum has been reached. Two basic tests have been employed to determine when the search should terminate. One test is made on the fractional change in the individual estimated parameters on the step sizes. Minimum desirable values of the fraction change in the variables are read into the computer program, and the test is conducted after each exploratory search failure. Another test occurs after each exploratory search or pattern move; the change in the value of f(b) is compared to a specified fraction read into the computer program. If the value of f(b) has not decreased from the value on the previous move by an amount greater than the specified fraction, and exploratory search or pattern move is considered a failure. The calculations terminate when both tests are satisfied on a specified number of cycles. The disadvantage of direct search method is that they are slow in comparison with derivative of simplex methods, especially as the number of parameters becomes large.

2. **Simplex method** The second derivative-free method of minimization of a nonlinear objective function is by using of regular geometric patterns of search involving simplexes. In the search for a minimum of the sum of the squares of the deviations,f(b),trial values of the model parameters can be selected at points in parameter space located at the vertices of the simplex. The sum of the squares of the deviations is evaluated at each of the vertices of the simplex: a projection is made from the point yielding the highest value of the objective function, point A in Figure A-1 through the centroid of the simplex. Point A is deleted and a new simplex, termed a reflection,is formed composed of the remaining old points and one new point, B, located along the projected line at the proper distance from the centroid. Continuation of this procedure, always deleting

the vertex that yields the highest value of the objective function, plus rules for reducing the size of the simplex and rules to prevent cycling in the vicinity of the extremum, permits a derivative-free search in which the step size is, in essence, fixed at successively reduced levels but the direction of search is permitted to change.

Nelder and Mead (1965) described a more efficient version of the simplex method called 'Downhill simplex method' that permitted the geometric figures to expand and contract continuously during the search. This method minimized a function of n variables using (n+1) vertices of a flexible polyhedron. The algorithm and step of calculation are summarized in Figure A-2.

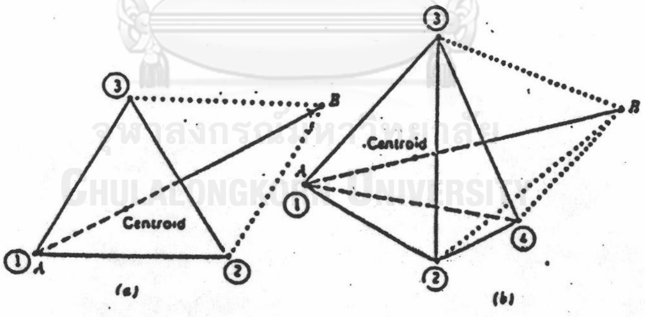

Figure A-1 Regular simplexes of two and three independent parameters.

(a) two variable simplex

(b) three variable simplex

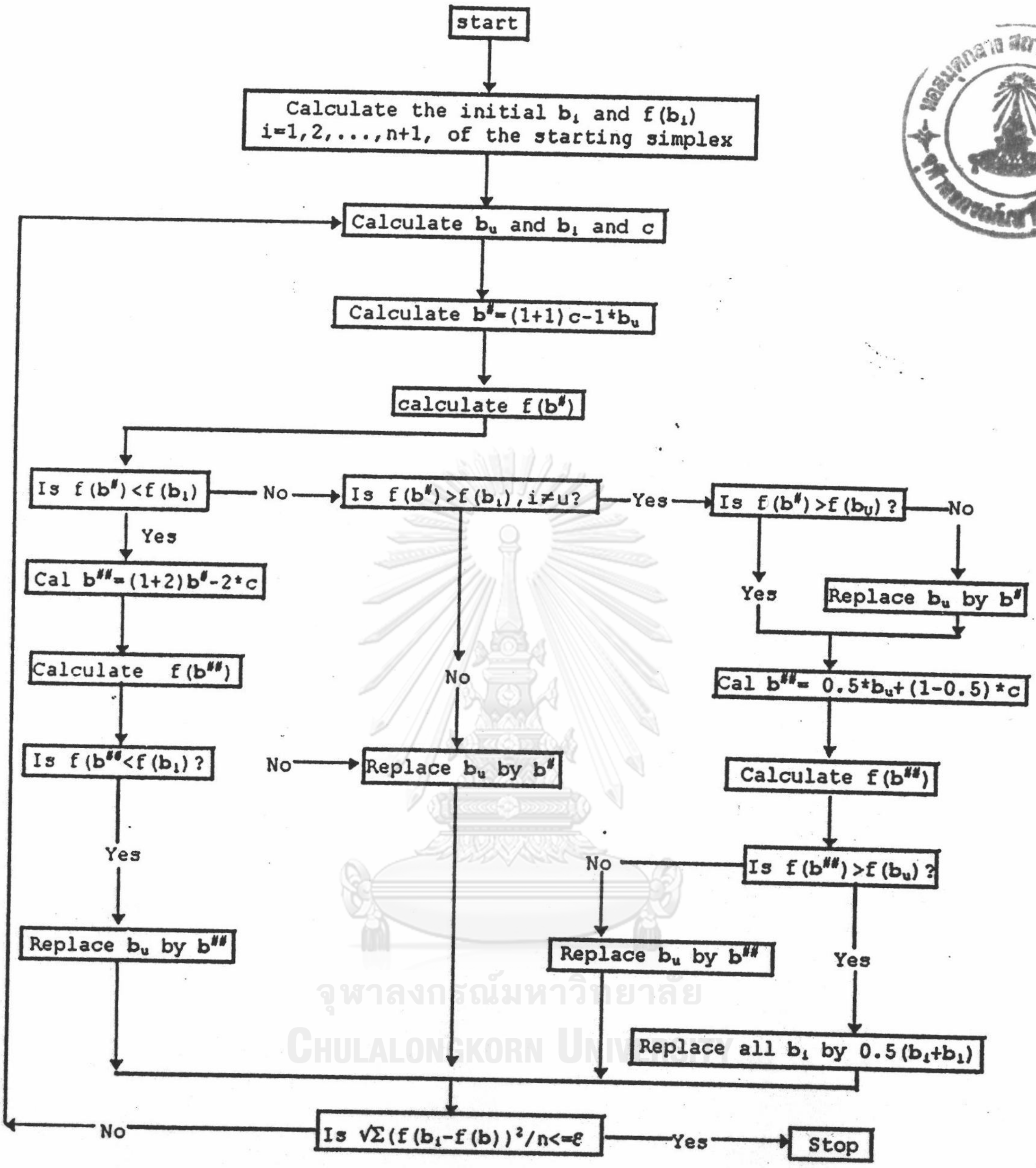
start
Calculate the initial b_i and $f(b_i)$ $i=1,2,...,n+1$, of the starting simplex
Calculate b_u and b_l and c
Calculate $b^{\#}=(1+1)c-1*b_u$
calculate $f(b^{\#})$
Is $f(b^{\#})<f(b_l)$
No
Is $f(b^{\#})>f(b_i), i\neq u$?
Yes
Is $f(b^{\#})>f(b_U)$?
No
Yes
Cal $b^{\#\#}=(1+2)b^{\#}-2*c$
Yes
Replace b_u by $b^{\#}$
Calculate $f(b^{\#\#})$
No
Cal $b^{\#\#}= 0.5*b_u+(1-0.5)*c$
Is $f(b^{\#\#}<f(b_l)$?
No
Replace b_u by $b^{\#}$
Calculate $f(b^{\#\#})$
Yes
No
Is $f(b^{\#\#})>f(b_u)$?
Replace b_u by $b^{\#\#}$
Replace b_u by $b^{\#\#}$
Yes
Replace all b_i by $0.5(b_i+b_l)$
No
Is $\sqrt{\Sigma(f(b_i-f(b))^2/n}<=\varepsilon$
Yes
Stop

Figure A-2 Flow chart for downnhill simplex method

This techniques have proved very successful in finding an extremum of and unconstrained objective function, as well as a constrained extremum, and are especially effective as the number of model parameters increases and also very effective and easily implemented on a digital computer.

3. **Newton's methods.** Newton's method makes use of the second-order approximation of f(b) at b^k, and thus employs second-order information about f(b), that is, information obtained from the second partial derivatives of f(b) with respect to the independent variables.

The minimum of f(b) in the direction of b^k is obtained by differentiating the approximation of f(b) with respect to each of the components of b and equating the resulting expressions to zero to give

$$\nabla f(b^k) = \nabla f(b^k) + \mathbf{H}(b^k)\Delta b^k = 0 \qquad \text{(A-3)}$$

or

$$b^{k+1} - b^k = \Delta b^k = -[\mathbf{H}(b^k)]^{-1}\nabla f(b^k) \qquad \text{(A-4)}$$

where $[\mathbf{H}(b^k)]^{-1}$ is the inverse of the Hessian matrix $\mathbf{H}(b^k)$.

Note that both the direction and step length are specified as a result of Eq.(A-3). If f(b) is actually quadratic, only one step is required to reach the minimum of f(b). However, for a general nonlinear objective function, the minimum of f(b) will not be reached in one step, so that Eq.(A-4) can be modified to

$$b^{k+1} - b^k = -\lambda^k[\mathbf{H}(b^k)]^{-1}\nabla f(b^k) \qquad \text{(A-5)}$$

where λ^k is the step length. The search direction **s** is given (for minimization) by

$$\mathbf{s}^k = -[\mathbf{H}(b^k)]^{-1}\nabla f(b^k) \qquad \text{(A-6)}$$

For the initial estimates of the parameters far from the final estimates, it is a characteristically slow method but converges rapidly near termination (in contrast to the gradient method which converges very slowly).

4. **Gradient Method** This gradient(steepest-descent /ascent) method uses only the first derivatives of the f(b) in the calculations.

The gradient is the vector at a point b that gives the (local) direction of the greatest increase in f(b) and is orthogonal to the contour of f(b) at b. For maximization, the search direction is simply the gradient (when used the algorithm is called steepest ascent); for minimization, the search direction is the negative of the gradient(steepest descent).

$$\mathbf{s}^k = \nabla f(b^k) \qquad \text{(A-7)}$$

In steepest descent at the kth stage, the transition from point b^k to another point b^{k+1} can be viewed as given by the following expression:

$$b^{k+1} = b^k+\Delta b^k = b^k + \lambda^k\mathbf{s}^k = b^k - \lambda^k\nabla f(b^k) \qquad \text{(A-8)}$$

where Δb^k = vector from b^k to b^{k+1}

$\mathbf{s}^k$ = search direction, the direction of steepest descent

λ^k = scalar that determines the step length in direction $\mathbf{s}^k$

The negative of the gradient gives the direction for minimization but not the magnitude of the step to be taken, so that various steepest descent procedures are possible, depending upon the choice of λ^k.

5. **Marquardt's method** Marquardt, Levenberg, and others have suggested that the Hessian matrix of f(b) be modified on each stage of the search as needed to ensure that the modified **H**(**b**), **H**(**b**), is positive definite and well-conditioned. The procedure adds elements to the diagonal element of H(x)

$$\mathbf{H}(\mathbf{b}) = [\mathbf{H}(\mathbf{b}) + \beta\mathbf{I}] \qquad \text{(A-9)}$$

where β is a positive constant large enough to make **H**(**b**) positive definite when **H**(**b**) is not. Also it is possible to useγ

$$[\mathbf{H}(\mathbf{b})]^{-1} = [\mathbf{H}^{-1}(\mathbf{b}) + \gamma\mathbf{I}] \qquad \text{(A-9)}$$

Marquardt's method has been recommended as being quite effective. It is definitely superior to either the

Newton's method or the method of steepest descent. Because either analytical or numerical derivatives at the minimum of f(b) are availabl, it is superior to the simplex method in that subsequent estimates of the precision of the parameters are easy to make.On the other hand, the simplex method has the advantage that the partial derivatives of f(b) need not be caculated at all, thus saving considerable computer time in estimation. Moreover, it can more easily treat problems involving functions with discontinuities, points of inflection, and end points. For very complex models, the simplex method has proved the more effective in estimating the parameters in simulation studies.

The model in this study is too complex to use Marquard's method, because the model in this study are compose of 7 simultaneous differential equations, 6 simultanious algebriac equations and an nonlinear-algebraic equation to find the root of equation at every step of the calculation. Thus the optimization method to be used• in this study is the downhill simplex method (William et.al (1992)).

Step of Parameter Estimation

The models in this study consisting of 7 simultaneous differential equations, 6 simultaneous algebraic equations and a nonlinear-algebraic equation as can be shown in the general form:

$$d[A_i]/dt = f(m,k,A_i^-,t) \quad \text{(A-10)}$$

$$A_i^- = f(A_i,K_i,A_i^-) \quad \text{(A-11)}$$

$$f(m) = 0 \quad \text{(A-12)}$$

where $d[A_i]/dt$ = vector of derivatives of A_i

f = vector of functions

t = independent variable

A_i = vector of dependent variables

k = vector of parameters

K_i = vector of known constants

let A_i^* = vector of experimental observations of the dependent variable

A_i = vector of calculated values of the dependent variables obtained from integrating Eq.(A-10)

(1) Assume initial guesses for the parameter k

(2) Use the vector k and the initial condition to integrate the differential equations by 4th order Rung-Kutta and solve algebraic equations by Newton-Raptson (Constantinides,(1988)) to obtain the profiles of A_i.

(3) Let err is error between observation with calculated value.

$$err_i^2 = (A_i - A_i^*)^2 \qquad \text{(A-13)}$$

(4) Let E is total error

$$E_i^2 = \sum err_i^2 \qquad \text{(A-14)}$$

(5) The procedure from step (1) to (4) are called objective function to calculate the sum of square error. The main procedure to optimize the E value (objective function, f(k)) is downhill simplex method as the algorithm described in the preceding section. All problems are solved by writing program in MATLAB program.

APPENDIX B

Table B-1 Initial condition input used in the simulation

Case	Phenol (mole/l)	Formaldehyde (mole/l)	NaOH (mole/l)	Temperature (°C)
1	1.003	2.119	0.03138	30
2	4.71	9.189	0.09369	30
3	0.9583	2.030	0.01250	57
4	4.68	9.456	0.09615	57
5	5.375	5.375	0.1000	40
6	5.375	5.375	0.1000	50
7	5.375	5.375	0.0500	60
8	5.375	5.375	0.0500	70
9	0.600	0.100	0.0045	90
10	0.200	0.0500	0.0500	30
11	0.400	0.09602	0.200	31
12	4.770	8.4700	0.2860	30
13	4.694	7.2700	0.04388	57
14	4.680	9.4600	0.0953	57
15	4.800	8.5300	0.0462	57
16	4.950	7.0100	0.0489	57
17	2.000	0.2000	0.1300	30
18	2.000	0.2000	0.6000	30
19	1.804	5.9400	1.8040	30
20	4.420	9.3000	0.0937	45

Table B-2 References of Each Case

Case	Reference
1,12,13,20	Zavitsas et.al(1968)
2,4	Zavitsas (1966)
3,14,15,16	Zavitsas et.al(1967)
5,6,7,8	Debing et.al (1952)
9	DeJong et.al(1956)
10	Dijkstra et.al (1962)
11	Dijkstra et.al (1957)
17,18	Peer et.al (1959)
19	Freeman et.al (1954)

APPENDIX C

EQUILIBRIUM CONSTANT

Appendix C-1 Equilibrium constant of phenolic species

The equilibrium constant of all phenolic species at each temperature can be calculated from the relationship between temperature and equilibrium constant which were determined by Sprengling et.al(1965) and Zavitsas et.al(1966) individually. The results were in good agreement that pK_i of each species were linear with temperature as the following equations:

$pK_1 = -0.0108*temp+10.125$ (c-1)

$pK_2 = -0.0117*temp+10.1275$ (c-2)

$pK_3 = -0.0105*temp+9.9925$ (c-3)

$pK_4 = -0.0107*temp+9.8375$ (c-4)

$pK_5 = -0.0108*temp+9.95$ c-5)

$pK_6 = -0.0105*temp+9.7125$ (c-6)

Appendix C-2 The equilibrium constant of formaldehyde equilibrium

The constant value of Q_1 = 45 and Q_2 = 400 are used to calculate the methylene glycol fraction at all studied temperature as the average value found by Zavitsas et.al (1968). Because from the studies of Iliceto et.al (1951) and Zavitsas (1968) found that the fraction of formaldehyde in methylene glycol form were not affect by temperature from 20-100 °C and pH of the solution as shown in **Figure** C-1

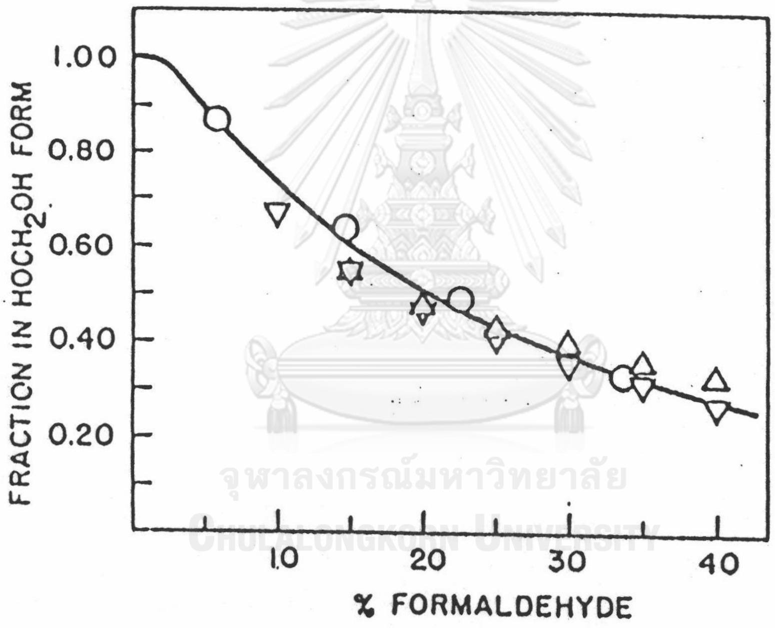

Figure C-1 Fraction of formaldehyde vs. wt.% formaldehyde in water: (-) curve calculated according to equation of formaldehyde equilibrium ;
(o) experiment at 20 °C (Iliceto et.al(1951))
(Δ) experimental at 35 °C.(Zavitsas (1967))
(∇) experimental at 100 °C. (Zavitsas (1967))

Table C-1 Rate Constants Obtained from Zavitsas' Model (Zavitsas et.al (1968))

Temp., (°C)	30	57	30	57	30
$[P]_o$, (M.)	1.003	0.9583	4.71	4.68	1.804
$[F]_o$, (M.)	2.119	2.030	9.189	9.456	5.94
$[NaOH]_o$, (M.)	0.03138	0.0125	0.09369	0.09615	1.804
k_1 , (l/mole.min)x10^3	2.322	30.6	5.01	87.78	3
k_2, (l/mole.min)x10^3	1.65	20.34	2.847	46.68	1.98
k_3, (l/mole.min)x10^3	2.892	29.7	5.088	81.06	3.3
k_4, (l/mole.min)x10^3	2.094	21.6	4.656	61.32	3.18
k_5, (l/mole.min)x10^3	2.49	33.48	4.608	80.7	3
k_6, (l/mole.min)x10^3	6.174	64.68	9.318	128.04	7.5
k_7, (l/mole.min)x10^3	1.476	15.96	3.714	50.58	2.7

Table C-2 Activation Parameters for the NaOH-Catalyzed Hydroxymethylation from Zavitsas' Model

	Dilute Systems		Concentrate Systems	
Rate Constant	E_a, (kcal)	ln A	E_a, (kcal)	ln A
k_1	19	25.45	21.1	29.7
k_2	18.	2429	20.6	28.25
k_3	17.	22.62	20.4	28.55
k_4	17.	22.35	19.0	26.14
k_5	19.	25.77	21.0	29.61
k_6	17.	23.62	19.3	27.35
k_7	17.	22.58	19.2	26.32

APPENDIX D

COMPUTER PROGRAM WRITTEN BY USING MATLAB

```
function [s]=downhill(p)
% The routine for finding the parameters which make the
% minimum of the objective function by downhill simplex
% method.  The objective function named "object(p)"
% where p is the matrix of initial guess parameters.

mp=8;np=7;  % mp are numbers of vectors which are the
            % vertics of the starting simplex.
            % np are numbers of parameter in a vector
            % of vertic.
p=zeros(mp,np);
y=zeros(mp,1);
nmax=20;alpha=1;beta=0.5;gamma=2;itmax=500;ftol=0.01;
pbar=zeros(1,np);
pr=zeros(1,np);
prr=zeros(1,np);
mpts=mp;
iter=0;
for i=1:mp
  q=p(i,:);
  [a]=object(q);
  y(i)=a;  % y is the vector of objective value
end
y
ilo=1;
if y(1)>y(2),
  ihi=1;
  inhi=2;
  else
  ihi=2;
  inhi=1;
end
for j=1:mp
  if y(j)<y(ilo),ilo=j;end
  if y(j)>y(ihi),inhi=ihi;ihi=j;
   elseif y(j)>y(inhi)
     if j~=ihi,inhi=j;end
  end
end

%Compute the fractional range from highest to lowest
%and return if satisfactory.

%rtol=2.*abs(y(ihi)-y(ilo))/(abs(y(ihi))+abs(y(ilo)));
rtol=abs(y(ihi)-y(ilo));

while  (rtol>ftol) ¦ (iter==itmax)
  iter=iter+1
  pbar=(sum(p)-p(ihi,:))/np;
  pr=(1+alpha)*pbar-alpha*p(ihi,:);
  [ypr]=object(pr);
  if ypr<=y(ilo)
    prr=gamma*pr+(1-gamma)*pbar;
    [yprr]=object(prr);
    if yprr<y(ilo)
       p(ihi,:)=prr;
       y(ihi)=yprr;
    else
       p(ihi,:)=pr;
       y(ihi)=ypr;
    end
  elseif ypr>=y(inhi)
     if ypr<y(ihi)
       p(ihi,:)=pr;
       y(ihi)=ypr;
     end
     prr=beta*p(ihi,:)+(1-beta)*pbar;
     [yprr]=object(prr);
     if yprr<y(ihi)
        p(ihi,:)=prr;
        y(ihi)=yprr;
     else
        for i=1:mp
          if i~=ilo
             pr=0.5*(p(i,:)+p(ilo,:));
             p(i,:)=pr;
             [c]=object(pr) ;
             y(i)=c;
          end
        end
     end
  else
     p(ihi,:)=pr;
     y(ihi)=ypr;
  end

  ilo=1;
if y(1)>y(2),
  ihi=1;
  inhi=2;
  else
  ihi=2;
  inhi=1;
end
for j=1:mp
  if y(j)<y(ilo),ilo=j;end
  if y(j)>y(ihi),inhi=ihi;ihi=j;
   elseif y(j)>y(inhi)
     if j~=ihi,inhi=j;end
  end
end

%rtol=2.*abs(y(ihi)-y(ilo))/(abs(y(ihi))+abs(y(ilo)))
rtol=abs(y(ihi)-y(ilo))
end
s=p(ilo,:);
```

```
function [sumerr]=object302(p)

% funtion to calculate the objective function
% by using Rung-Kutta algorithm to solve differential
% equation and Newton-Raptson to find the root of
% m fraction in the model
% This is the example of calculating objective function
% of Simple model with Zavitsas data at 30 C at
% concentrate run


[texp,yexp]=dat302;
steps=1;y0=[4.71 0 0 0 0 0 9.189]';
least=0;unit=ones(size(y0'));
i=length(texp);
y=y0;
t0=0;
count=0;tm=max(texp);

for t=t0:steps:tm
  yold=y; count=count+1;
  t1=t;
  temp=feval('model1',t1,yold,p);
  b1=temp;
  y=yold+0.5*steps*temp;
  t2=t1+0.5*steps;
  temp=feval('model1',t2,y,p);
  b2=temp;
  y=yold+0.5*steps*temp;
  temp=feval('model1',t2,y,p);
  b3=temp;
  y=yold+steps*temp;
  t3=t1+steps;
  temp=feval('model1',t3,y,p);
  y=yold+steps*(b1+temp+2*(b2+b3))/6;
    for j=1:i
        %disp('t');disp(t);disp(texp(j));
        if t3==texp(j)
                %disp(t3);disp(texp(j));
        %a=(yexp(j,:)./y');
        %b=(unit-a).^2;
            b=(yexp(j,:)-y').^2;
              err=sum(b);
              least=least+err;
        end
    end
  %if count==100
   %fprintf('%8.2f%8.2f%8.2f\n',t3,y(1),y(2));
    %count=0;
  %end
  end
sumerr=least;
```

```
function [tout,yout]=simnew(phe,f,oh,temp,time)
% Function for simulating the simple model which
% concern water concentration.
% phe is initial phenol concentration
% f is initial fomaldehyde concentration
% oh is initial base concentration
% temp is reaction temperature
% time is a vector of reaction time

steps=1;y0=[phe 0 0 0 0 0 f]';
least=0;unit=ones(size(y0'));
i=length(time);ho=h2o(phe,f,oh);
tout=zeros(i+1,1);
yout=zeros(i+1,length(y0));
y=y0;yout(1,:)=y0';
t0=0.00;tout(1)=t0;
count=0;tm=max(time);

% Calculate rate constant p which relate to
% water concentration as p=A*exp(b/[H2O])
c1=46.9;c2=15;
k1=energyd(temp);
k2=energyc(temp);

ln=k1./k2;lnk=log(ln);
ee1=1/c1;ee2=1/c2;
ee=ee1-ee2;
slop=lnk/ee;
k01=k1./(exp(slop/c1));
lnk01=log(k1./(exp(slop/c1)));
p=exp(lnk01+slop/ho)


for t=t0:steps:tm
  yold=y; count=count+1;
  t1=t;
  temp=feval('model1',t1,yold,p);
  b1=temp;
  y=yold+0.5*steps*temp;
  t2=t1+0.5*steps;
  temp=feval('model1',t2,y,p);
  b2=temp;
  y=yold+0.5*steps*temp;
  temp=feval('model1',t2,y,p);
  b3=temp;
  y=yold+steps*temp;
  t3=t1+steps;
  temp=feval('model1',t3,y,p);
  y=yold+steps*(b1+temp+2*(b2+b3))/6;
      for j=1:i
        if t3==time(j)
           tout(j+1)=time(j);
           yout(j+1,:)=y';
           end
      end
  end
```

```
function [tout,yout]=zavnew(phe,f,NaOH,Temp,time)
% Function for simulating Zavitsas model which
% concern water concentration.
% phe is initial phenol concentration
% f is initial fomaldehyde concentration
% oh is initial base concentration
% temp is reaction temperature
% time is a vector of reaction time

steps=1;y0=[phe 0 0 0 0 0 f]';
yion=[NaOH 0 0 0 0 0 f]';oh=NaOH;
ho=h2o(phe,f,NaOH);
mf=raptson(f,f,ho,1);

% Calculate equilibrium constant
aa=[-.0108 -.0117 -.0105 -.0107 -.0108 -.0105];
cc=[10.125 10.1275 9.9925 9.8375 9.95 9.7125];
dd=-1*(Temp*aa+cc);k=10.^dd;
a=zeros(1,6);
tm=max(time);
i=length(time);
tout=zeros(i+1,1);
yout=zeros(i+1,length(y0));
y=y0;yout(1,:)=y0';
t0=0.00;tout(1)=t0;
count=0;d=1;option=2;

% Calculate rate constant p which relate to
% water concentration as p=A*exp(B/[H2O])
  c1=47;c2=21;
        k1=parazd(Temp);
        k2=parazc(Temp);
ln=k1./k2;lnk=log(ln);
ee1=1/c1;ee2=1/c2;
ee=ee1-ee2;
slop=lnk/ee;
k01=k1./(exp(slop/c1));
lnk01=log(k1./(exp(slop/c1)));
p=exp(lnk01+slop/ho)

%slope=(k2-k1)/(c2-c1);inter=k2-slope*c2;
%p=slope.*ho+inter
%ratio=p(1)/(p(1)+p(2))
for t=t0:steps:tm
  yold=y; count=count+1;
  t1=t;
  temp=feval('model3',t1,yion,p,mf);
  b1=temp;
  y=yion+0.5*steps*temp;
  t2=t1+0.5*steps;
  temp=feval('model3',t2,y,p,mf);
  b2=temp;
  y=yion+0.5*steps*temp;
  temp=feval('model3',t2,y,p,mf);
  b3=temp;
  y=yion+steps*temp;
  t3=t1+steps;
  temp=feval('model3',t3,y,p,mf);
  y=yold+steps*(b1+temp+2*(b2+b3))/6;

        if option==1
           for n=1:6
               a(n)=[k(n)*y(n)]/[k(1)*y(1)];
           end
           yion(1)=oh/sum(a);
           for m=1:6
               yion(m)=[k(m)*y(m)*yion(1)]/[k(1)*y(1)];
           end
        end
        if option==2
           n=y(1:6);
           yi=rapt(1,n,oh,k);
           for l=1:6
               yion(l)=(k(l)/k(1))*y(l)*yi/[(y(l)-yi)+yi*(
k(l)/k(1))];
           end
        end
   yion(7)=y(7);
   mf=raptson(y0(7),y(7),ho,1);

      for j=1:i
        if t3==time(j)
           tout(j+1)=time(j);
           yout(j+1,:)=y';
           end
      end
  end
```

```
function [tout,yout]=zav(phe,f,NaOH,Temp,time)
% Function for simulating Zavitsas model which
% concern water concentration.
% phe is initial phenol concentration
% f is initial fomaldehyde concentration
% oh is initial base concentration
% temp is reaction temperature
% time is a vector of reaction time

steps=1;y0=[phe 0 0 0 0 0 f]';
yion=[NaOH 0 0 0 0 0 f]';oh=NaOH;
ho=h2o(phe,f,NaOH);
mf=raptson(f,f,ho,1);

% Calculate equilibrium constant
aa=[-.0108 -.0117 -.0105 -.0107 -.0108 -.0105];
cc=[10.125 10.1275 9.9925 9.8375 9.95 9.7125];
dd=-1*(Temp*aa+cc);k=10.^dd;
a=zeros(1,6);
tm=max(time);
i=length(time);
tout=zeros(i+1,1);
yout=zeros(i+1,length(y0));
y=y0;yout(1,:)=y0';
t0=0.00;tout(1)=t0;
count=0;d=1;option=2;

% Calculate rate constant p which relate to
% water concentration as p=A+B*[H2O]
   c1=47;c2=21;
        k1=parazd(Temp);
        k2=parazc(Temp);
slope=(k2-k1)/(c2-c1);inter=k2-slope*c2;
p=slope.*ho+inter
%ratio=p(1)/(p(1)+p(2))
for t=t0:steps:tm
  yold=y; count=count+1;
  t1=t;
  temp=feval('model3',t1,yion,p,mf);
  b1=temp;
  y=yion+0.5*steps*temp;
  t2=t1+0.5*steps;
  temp=feval('model3',t2,y,p,mf);
  b2=temp;
  y=yion+0.5*steps*temp;
  temp=feval('model3',t2,y,p,mf);
  b3=temp;
  y=yion+steps*temp;
  t3=t1+steps;
  temp=feval('model3',t3,y,p,mf);
  y=yold+steps*(b1+temp+2*(b2+b3))/6;

        if option==1
           for n=1:6
               a(n)=[k(n)*y(n)]/[k(1)*y(1)];
           end
           yion(1)=oh/sum(a);
           for m=1:6
               yion(m)=[k(m)*y(m)*yion(1)]/[k(1)*y(1)];
           end
        end
        if option==2
           n=y(1:6);
           yi=rapt(1,n,oh,k);
           for l=1:6
               yion(l)=(k(l)/k(l))*y(l)*yi/[(y(l)-yi)+y
i*(k(l)/k(l))];
           end
        end
   yion(7)=y(7);
   mf=raptson(y0(7),y(7),ho,1);

      for j=1:i
        if t3==time(j)
           tout(j+1)=time(j);
           yout(j+1,:)=y';
           end
      end
  end
```

```
function [tout,yout]=gennew(phe,f,NaOH,Temp,time,option)
% Main program for simulating all datas by inputting the
% initial condition: phe is the input concentration of
% phenol in mole/1;  f is the input concentration of
% formaldehyde in mole/1; NaOH is the input concentration
% of base in mole/1; Temp is reaction temperature;
% time is the vector of time in minute;
steps=1;y0=[phe 0 0 0 0 0 f]';
yion=[NaOH 0 0 0 0 0 f]';oh=NaOH;
ho=h2o(phe,f,NaOH);
% h2o is the function for calculating  water concetration
mf=raptson(f,f,ho,1);
% raptson is the function for calculating fraction of
% formaldehyde in methylene glycol form.

% calculate equilibrium constant
aa=[-.0108 -.0117 -.0105 -.0107 -.0108 -.0105];
cc=[10.125 10.1275 9.9925 9.8375 9.95 9.7125];
dd=-1*(Temp*aa+cc);k=10.^dd;

a=zeros(1,6);
tm=max(time);
i=length(time);
tout=zeros(i+1,1);
yout=zeros(i+1,length(y0));
y=y0;yout(1,:)=y0';
t0=0.00;tout(1)=t0;
count=0;d=1;

% calculate rate constant which relate to water
% concentration as p=A*exp(B/[H2O])
c1=47;c2=15;
k1=energyzd(Temp); %calculate k at dilute concentration
k2=energyzc(Temp); %calculate k at high  concentration
ln=k1./k2;lnk=log(ln);
ee1=1/c1;ee2=1/c2;
ee=ee1-ee2;
slop=lnk/ee;
k01=k1./(exp(slop/c1));
lnk01=log(k1./(exp(slop/c1)));
p=exp(lnk01+slop/ho)

% routine for calculating the profile of output
% concentration
for t=t0:steps:tm
  yold=y; count=count+1;
  t1=t;
  temp=feval('model3',t1,yion,p,mf);
  b1=temp;
  y=yion+0.5*steps*temp;
  t2=t1+0.5*steps;
  temp=feval('model3',t2,y,p,mf);
  b2=temp;
  y=yion+0.5*steps*temp;
  temp=feval('model3',t2,y,p,mf);
```

```
   b3=temp;
   y=yion+steps*temp;
   t3=t1+steps;
   temp=feval('model3',t3,y,p,mf);
   y=yold+steps*(b1+temp+2*(b2+b3))/6;

         if option==1
            for n=1:6
                a(n)=[k(n)*y(n)]/[k(1)*y(1)];
            end
            yion(1)=oh/sum(a);
            for m=1:6
                yion(m)=[k(m)*y(m)*yion(1)]/[k(1)*y(1)];
           end
        end
        if option==2
           n=y(1:6);
           yi=rapt(1,n,oh,k);
           for l=1:6
               yion(l)=(k(l)/k(1))*y(l)*yi/[(y(l)-yi)+yi*(
k(l)/k(1))];
           end
        end
   yion(7)=y(7);
   mf=raptson(y0(7),y(7),ho,1);

      for j=1:i
        if t3==time(j)
           tout(j+1)=time(j);
           yout(j+1,:)=y';
           end
      end
  end
```

APPENDIX E

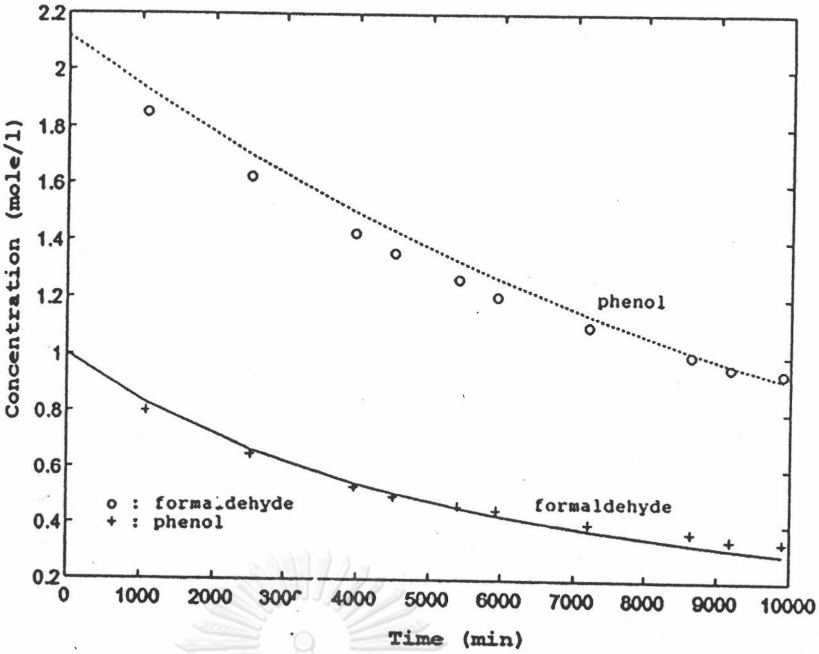

Figure E-1 Reactant concentration vs. time at 30°C, $[P]_o$=1.003 M. $[F]_o$ = 2.119 M. $[NaOH]_o$=0.03138 M.: Curve calculated with rate constants from Table 5-1 by Simple model. Point: experimental data (Zavitsas (1966)).

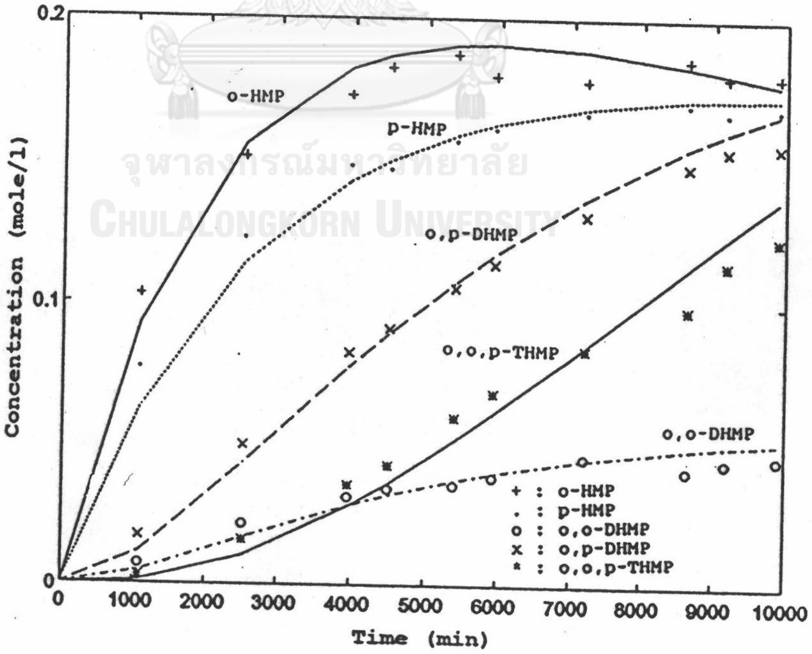

Figure E-2 Product concentration vs. time at 30°C, $[P]_o$=1.003 M. $[F]_o$ = 2.119 M. $[NaOH]_o$=0.03138 M.: Curve calculated with rate constants from Table 5-1 by Simple model. Point: experimental data (Zavitsas (1966)).

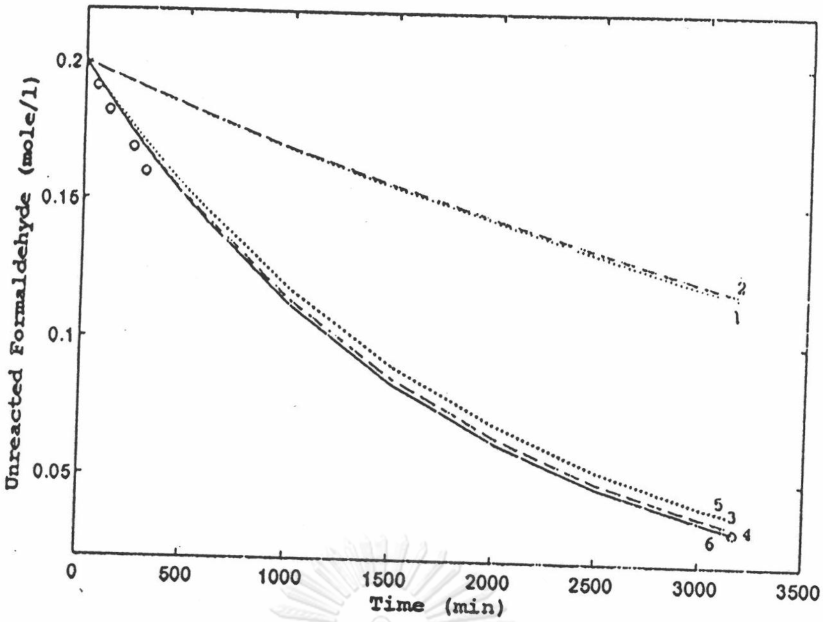

Figure E-3 Formaldehyde concentration vs. time at 30°C, $[P]_0$= 2 M.$[F]_o$ = 0.2 M. [NaOH]=0.13 M.: Curve calculated compare between 6 models described in Page 58 . Point: experimental data (Peer et.al (1959)).

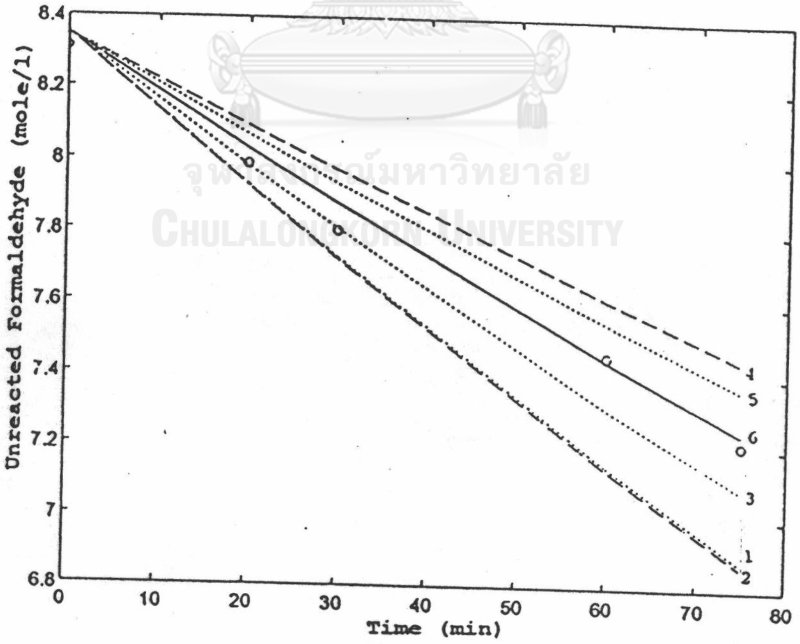

Figure E-4 Formaldehyde concentration vs. time at 57°C, $[P]_0$= 4.8 M.$[F]_o$ = 8.53 M. [NaOH]= 0.0462 M.: Curve calculated compare between 6 models described in Page 58 . Point: experimental data (Zavitsas et.al (1967)).

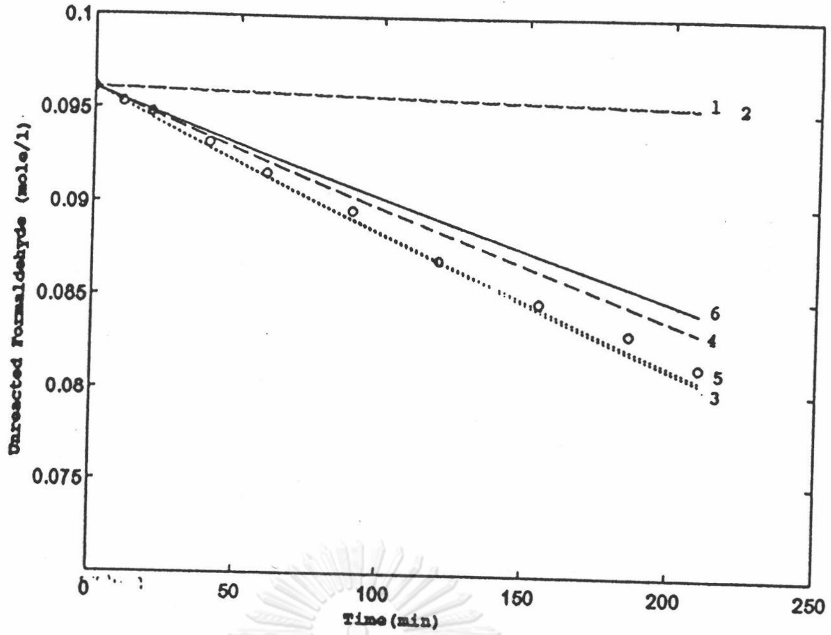

Figure E-5 Formaldehyde concentration vs. time at 31°C, $[P]_0$= 0.4 M. $[F]_o$ = 0.09602 M. [NaOH]=0.2 M.: Curve calculated compare between 6 models described in Page 58. Point: experimental data (Dijkstra et.al (1959)).

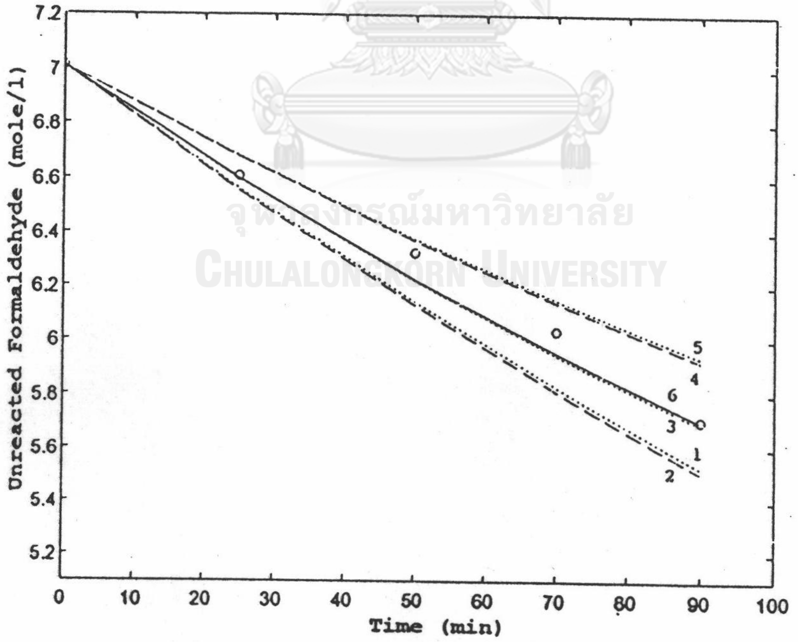

Figure E-6 Formaldehyde concentration vs. time at 57°C, $[P]_0$ = 4.95 M. $[F]_o$ = 7.01 M. [NaOH]=0.0489 M.: Curve calculated compare between 6 models described in Page 58. Point: experimental data (Zavitsas et.al (1967)).

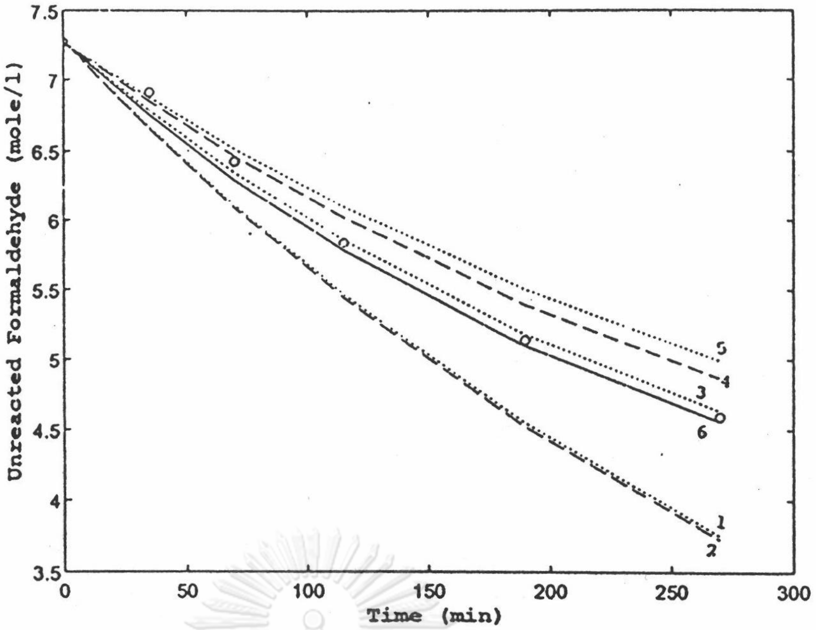

Figure E-7 Formaldehyde concentration vs. time at 57°C, $[P]_0$= 4.694 M. $[F]_o$ = 7.27 M. [NaOH]=0.04388 M.: Curve calculated compare between 6 models described in Page 58 . Point: experimental data (Zavitsas et.al (1968)).

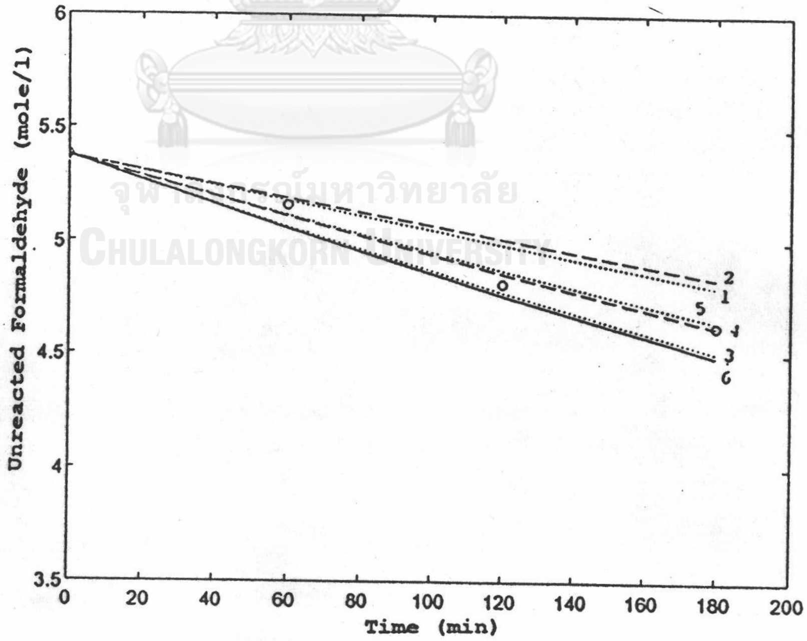

Figure E-8 Formaldehyde concentration vs. time at 40°C, $[P]_0$= 5.375 M. $[F]_o$ = 5.375 M. [NaOH]=0.1 M.: Curve calculated compare between 6 models described in Page 58 . Point: experimental data (Dejong et.al (1952)).

VITA

Miss Surat Atthajariyakul was born on December 6, 1968 in Bangkok, Thailand. She finished her primary school from Singfha and secondary school from Sainampheung. She graduated her B.Sc (Agro-Industrial Product Development) with first class honours from the Faculty of Agro-Industry, Kasetsart University, Bangkok. She then pursues her post-graduate study at the Department of Chemical Engineering, Faculty of Engineering, Chulalongkorn University.