LIBRARY AND INFORMATION CENTER
MAHIDOL UNIVERSITY

23 AUG 1994

MATHEMATICAL MODELLING AND COMPUTER SIMULATION OF POLYMERIZATION REACTIONS: THE FREE-RADICAL POLYMERIZATION OF STYRENE

อภินันทนาการ
จาก
บัณฑิตวิทยาลัย มหิดล

WUTTIPONG RUNGSEESANTIVANON

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
(POLYMER SCIENCE)

IN

FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
1994

Copyright by Mahidol University

27350

Thesis

entitled

MATHEMATICAL MODELLING AND COMPUTER SIMULATION OF POLYMERIZATION REACTIONS: THE FREE-RADICAL POLYMERIZATION OF STYRENE

W. Rungseesantivanon

Wuttipong Rungseesantivanon
Candidate

A. Euapermkiati

Anucha Euapermkiati, Ph.D.
Major Advisor

Pranee Phinyocheep

Pranee Phinyocheep,
Doctorat de l'Université du Maine
Co-advisor

Preedeeporn Limcharoen

Preedeeporn Limcharoen, Ph.D.
Co-advisor

M. Chulasamaya
B. Riewkhan

Monthree Chulasamaya,
M.D., Ph.D.
Dean
Faculty of Graduate Studies

Pranee Phinyocheep

Pranee Phinyocheep,
Doctorat de l'Université du Maine
Chairman
Master of Science Program in
Polymer Science
Faculty of Science

Copyright by Mahidol University

Thesis

entitled

MATHEMATICAL MODELLING AND COMPUTER SIMULATION OF POLYMERIZATION REACTIONS: THE FREE-RADICAL POLYMERIZATION OF STYRENE

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Master of Science (Polymer Science)
on
May, 27, 1994

W. Rungseesantivanon

Wuttipong Rungseesantivanon
Candidate

Anucha Euapermkiati, Ph.D.
Chairman

Pranee Phinyocheep

Pranee Phinyocheep,
Doctorat de l'Université du Maine
Member

Preedeeporn Limcharoen

Preedeeporn Limcharoen, Ph.D.
Member

M. Chulasamaya
B. Riewkgan.

Monthree Chulasamaya,
M.D., Ph.D.
Dean
Faculty of Graduate Studies

Pornchai Matangkasombut
M.D., Ph.D.
Dean
Faculty of Science

Copyright by Mahidol University

BIOGRAPHY

NAME	Wuttipong Rungseesantivanon
DATE OF BIRTH	December 6, 1968
PLACE OF BIRTH	Surin
INSTITUTIONS ATTENED	Khon Kaen University, 1987-1990: Bachelor of Science (Chemistry)
	Mahidol University, 1991-1993: Master of Science (Polymer Science)

Copyright by Mahidol University

ACKNOWLEDGEMENTS

I would like to express my appreciation to people who contributed a share in this thesis. The first person is, my supervisor, Dr. Anucha Euapermkiati, who gave valuable suggestions and help during this study. Dr. Pranee Phinyocheep and Dr. Preedeeporn Limcharoen for their guidance and encouragement throughout the preparation of the thesis.

In addition, my sincere thanks are given to Dr. Krisda Suchiva and all of the lecturers of the polymer group, to Dr. Prapin Wilairat and to Dr. Saran Poshyachinda (MTEC). And those people who gave help as follows,

National Metal and Materials Technology Center (MTEC), National Science and Technology Development Agency (NSTDA), supporting this project.

Department of Chemistry, who providing the laboratory and other facilities.

Siam fine Chemical Co.,Ltd., donating commercial styrene monomer. Brighten Polytrading Co.,Ltd., donating AIBN initiator.

Finally, I wish to thank all my friends for their helpfulness during this study.

Wuttipong Rungseesantivanon

Copyright by Mahidol University

ชื่อวิทยานิพนธ์ การจำลองแบบทางคณิตศาสตร์โดยคอมพิวเตอร์ของปฏิกิริยาพอลิเมอร์ไรเซชั่นแบบอนุมูลอิสระของสไตรีน

ผู้วิจัย วุฒิพงษ์ รังษีสันติวานนท์

ปริญญา วิทยาศาสตรมหาบัณฑิต (วิทยาศาสตร์พอลิเมอร์)

คณะกรรมการควบคุมวิทยานิพนธ์

อนุชา เอื้อเพิ่มเกียรติ, Ph.D.

ปราณี ภิญโญชีพ, Doctorat de l'Université du Maine

ปรีดีพร ลิ้มเจริญ, Ph.D.

วันที่สำเร็จการศึกษา 27 พฤษภาคม พ.ศ. 2537

บทคัดย่อ

ได้ศึกษาจลนศาสตร์ของปฏิกิริยาพอลิเมอร์ไรเซชั่นแบบอนุมูลอิสระของสไตรีน (ST) โดยใช้คาร์บอนเตตระคลอไรด์ (CCl_4) และ1-บิวเทนไทออล (1-BuSH) เป็นสารย้าย-สายโซ่ (Chain transfer agent (CTA)) โดยการทดลองและโดยวิธีการคำนวณใช้คอมพิวเตอร์ ได้สร้างแบบจำลองการทำนายแบบพลศาสตร์ของจลนศาสตร์ดังกล่าวขึ้น และได้เขียนโปรแกรมคอมพิวเตอร์ใช้ภาษา Microsoft QuickBASIC เพื่อคำนวณสมการเชิงพีชคณิตและเชิงอนุพันธ์ของแบบจำลองโดยอาศัยเทคนิค 4^{th} Order Runge-Kutta และ Newton-Ralphson ได้สร้างสมการของเมโยแบบปรับปรุงจากกลไกปฏิกิริยาของจลนศาสตร์ที่เสนอ ได้ใช้เทคนิคฟูเรียทรานฟอร์มรามานสเปคโตรสโกปีเพื่อวัดปริมาณสไตรีน คาร์บอนเตตระคลอไรด์ และ1-บิวเทนไทออล ที่เวลาในการทำปฏิกิริยาต่างๆกัน ได้ใช้เทคนิคเจลเพอร์มีเอชั่นโครมาโตกราฟี (GPC) หาน้ำหนักโมเลกุล (MW) และการกระจายของน้ำหนักโมเลกุล (MWD) ของพอลิเมอร์สังเคราะห์ อัตราการเกิดปฏิกิริยารวมของปฏิกิริยาพอลิเมอร์ไรเซชั่น ($R_{ST,CTA}$) และ อัตราการหายไปรวมของสารย้ายสายโซ่ (R_{CTA}) มีความสัมพันธ์กับ $k'_{ST,CTA}[ST]^p[CTA]^q$ และ $k'_{CTA}[ST]^s[CTA]^t$ ตามลำดับ โดยที่ $k'_{ST,CTA}$ และ k'_{CTA} คือค่าคงที่ปรากฏของ

Copyright by Mahidol University

ปฏิกิริยา และ p, q, s และ t คืออันดับของปฏิกิริยา ในกรณีที่ใช้คาร์บอนเตตระคลอไรด์เป็นสารย้ายสายโซ่ ค่า p, q, s, t, $k'_{ST,CTA}$ และ k'_{CTA} จากการทดลองมีค่าเท่ากับ 0.95, -0.07, 0.99, 0.91, 4.88×10^{-5} l/mol.s และ 8.40×10^{-7} l/mol.s ตามลำดับ ในกรณีที่ใช้ 1-บิวเทนไทออลเป็นสารย้ายสายโซ่ ค่า p, q, s, t, $k'_{ST,CTA}$ และ k'_{CTA} จากการทดลองมีค่าเท่ากับ 0.97, 0.24, 1.51, 0.38, 1.10×10^{-4} l/mol.s และ 5.31×10^{-5} l/mol.s ตามลำดับ ได้คำนวณค่าคงที่ของปฏิกิริยาขั้นแผ่ขยาย (Propagation reaction (k_p)) และค่าคงที่ของปฏิกิริยาย้ายสายโซ่ (Transfer reaction (k_{tr})) โดยวิธีซิมมูเลชั่น และได้ใช้ค่าคงที่เหล่านี้ในการทำนายน้ำหนักโมเลกุลเฉลี่ยโดยจำนวน (M_n) และโดยน้ำหนัก (M_w) และการกระจายของน้ำหนักโมเลกุล (MWD) ของพอลิเมอร์ ในกรณีที่ใช้คาร์บอนเตตระคลอไรด์เป็นสารย้ายสายโซ่ ผลจากการทำซิมมูเลชั่นได้ใกล้เคียงกับผลการทดลอง ซึ่งตรงกันข้ามกับในกรณีที่ใช้1-บิวเทนไทออลเป็นสารย้ายสายโซ่ ได้คำนวณหาค่าคงที่ย้ายสายโซ่ (Chain transfer constant (C_x)) จากสมการของเมโย และสมการของเมโยแบบปรับปรุง ในกรณีที่ใช้คาร์บอนเตตระคลอไรด์เป็นสารย้ายสายโซ่ ค่า C_x ที่คำนวณได้จากสมการของเมโยมีค่าเท่ากับ 1.06×10^{-2} จากผลการทดลอง และ 6.7×10^{-3} จากการทำซิมมูเลชั่น ค่า C_x ที่คำนวณได้จากสมการของเมโยแบบปรับปรุงมีค่าเท่ากับ 1.07×10^{-2} จากผลการทดลอง และ 6.9×10^{-3} จากการทำซิมมูเลชั่น ในกรณีที่ใช้1-บิวเทนไทออล์เป็นสารย้ายสายโซ่ ค่า C_x ที่คำนวณได้จากสมการของเมโย มีค่าเท่ากับ 0.422 จากผลการทดลอง และ 0.761 จากการทำซิมมูเลชั่น ค่า C_x ที่คำนวณได้จากสมการของเมโยแบบปรับปรุง มีค่าเท่ากับ 1.632 จากผลการทดลอง และ 1.243 จากการทำซิมมูเลชั่น

Copyright by Mahidol University

Thesis Title Mathematical Modelling and Computer Simulation of Polymerization Reactions: The Free-Radical Polymerization of Styrene.
Name Wuttipong Rungseesantivanon
Degree Master of Science (Polymer Science)
Thesis Supervisory Committee
Anucha Euapermkiati, Ph.D.
Pranee Phinyocheep, Doctorat de l'Université du Maine
Preedeeporn Limcharoen, Ph.D.
Date of Graduation 27 May B.E. 2537 (1994)

ABSTRACT

The kinetics of thermal-initiated free radical polymerization of styrene (ST) in the presence of carbon tetrachloride (CCl_4) or 1-butanethiol (1-BuSH) chain transfer agents (CTA) were studied experimentally and computationally in details. The deterministic dynamic predictive model of the kinetics was established. Computer programmes were written in Microsoft® Quick BASIC computer language to solve the algebraic and differential equations using the 4^{th} order Runge-Kutta and Newton-Ralphson numerical integration methods. The modified Mayo Equation derived from the proposed kinetics was also established. Fourier-transfrom Raman spectroscopic technique was used to evaluate styrene, carbon tetrachloride and 1-butanethiol consumptions at specific reaction time. Gel Permeation Chromatography (GPC) was employed to obtain molecular weight (MW) and MW distribution (MWD) of the polymers synthesized. The overall rates of polymerization ($R_{ST,CTA}$) and of loss of CTA (R_{CTA}) had the relation to $k'_{ST,CTA}[ST]^p[CTA]^q$ and to $k'_{CTA}[ST]^s[CTA]^t$, respectively, where $k'_{ST,CTA}$ and k'_{CTA} are the apparent rate constants and p, q, s and t are the exponent order. When

Copyright by Mahidol University

CTA is CCl_4, the values of p, q, s, t, $k'_{ST,CTA}$, and k'_{CTA} were evaluated equal to 0.95, -0.07, 0.99, 0.91, 4.88 $x10^{-5}$ l/mol.s, and $8.40x10^{-7}$ l/mol.s, respectively. When CTA is 1-BuSH, the values of p, q, s, t, $k'_{ST,CTA}$, and k'_{CTA} were determined equal to 0.97, 0.24, 1.51, 0.38, $1.10x10^{-4}$ l/mol.s., and $5.31x10^{-5}$ l/mol.s, respectively. The simulation values of rate constant for the propagation reaction (k_p) and that for the transfer reaction (k_{tr}) were evaluated, and the values were used for computer simulation to predict M_n, M_w and MWD. The experimental and simulation results of M_n and M_w are in good agreement, while those of MWD were not when CTA is 1-BuSH. The experimental and simulation chain transfer constant (C_x) were determined from the original and modified Mayo Equations. For CTA is CCl_4, C_x as obtained from the original Mayo Equation was $1.06x10^{-2}$ and $6.7x10^{-3}$ and as obtained from the modified Mayo Equation was $1.07x10^{-2}$ and $6.9x10^{-3}$. For CTA is 1-BuSH, C_x obtained from the original Mayo Equation was $4.42x10^{-1}$ and $7.61x10^{-1}$ whereas that obtained from the modified Mayo Equation was 1.63 and 1.24, respectively.

Copyright by Mahidol University

Table of Contents

		Page
Abstract in Thai		i
Abstract in English		iii
Table of Contents		v
List of Tables		viii
List of Figures		ix
Glossary		xiv
Aims and Objectives of the Research		xvi
Chapter I	Free Radical Polymerization Reactions	1
1.1	Introduction	1
1.2	Classical Free Radical Polymerization	2
1.3	Non-Classical Free Radical Polymerization	3
Chapter II	Mathematical Modelling of the Free Radical Polymerization Reactions of a Vinyl Monomer in the Presence of a Chain Transfer Agent.	15
2.1	Introduction	15
2.2	Mathematical Modelling Techniques	16
	2.2.1 Deterministic Techniques	16
	2.2.2 Stochastic Techniques	18
2.3	Mathematical Modelling and Computer Simulation of the Free Radical Polymerization Reactions in the Presence of a Chain Transfer Agent.	19
2.4	Stiffness and Quasi-Steady-State Assumption (QSSA)	19
Chapter III	Experimental	20
3.1	Materials	20
3.2	Instrumentation	22
3.3	Experimental	24
	3.3.1 Preparation of Stock Reagents	24
	3.3.2 The Kinetic Studies of Thermal-initiated Free Radical Polymerization Reaction of Styrene in the Presence of Carbon tetrachloride or 1-Butanethiol	25

Copyright by Mahidol University

	3.3.3 Determination of Styrene, Carbon tetrachloride, and 1-Butanethiol Conversions	26
3.4	Determination of Molecular Weight and Molecular Weight Distribution of Polymers Synthesized using Gel Permeation Chromatographic (GPC) Technique	35
3.5	Computer Programming	35
3.6	Computer simulation experiment	40
Chapter IV	Results and Discussion	42
4.1	The Thermal-Initiated Free Radically Bulk and Solution Polymerization Reactions of Styrene at 70 °C.	42
4.2	The Thermal-Initiated Free Radically Solution Polymerization of Styrene in the Presence of a Chain Transfer Agent at 70 °C.	47
	4.2.1 Carbon tetrachloride (CCl_4) as the Chain Transfer Agent.	47
	4.2.2. 1-Butylmercaptan as the Chain Transfer Agent.	54
4.3	Mathematical Modelling and Computer Simulation of the Non-Classical Thermal-initiated Free Radical Solution Polymerization of a Vinyl Monomer in the Presence of a Chain Transfer Agent.	59
	4.3.1 The Mathematical Modelling	59
	4.3.2 The Computer Simulations	64
4.4	Comparisons of the Results obtained from the Experiments and from the Computer Simulations	73
	4.4.1 Molecular Weight and Molecular Weight Distribution of Polymers	73
	4.4.2 The Chain Transfer Constants (C_x) of CCl_4 and 1-BuSH	75
	4.4.2.1 C_x value for CCl_4	77
	4.4.2.2 C_x value for 1-BuSH	77
Chapter V	Conclusions and Suggestions for Further Work	92
5.1	Conclusions	92
5.2	Suggestions for Further Work	95
Bibliography		98

Copyright by Mahidol University

Appendix I	Mathematical Modelling of the Kinetics of Thermal-Initiated Free Radical Polymerization Reaction of a Vinyl monomer in the Presence of a Chain Transfer Agent.	105
Appendix II	Overall Velocity Coefficient for the AIBN-initiated Polymerization Reactions (k) and Evaluation of the Exponent Orders.	120
Appendix III	Computer Programmes For the Dynamic Predictive Thermal-Initiated Free Radical Polymerization of a Vinyl monomer in the Presence of a Chain Transfer Agent.	122
Appendix IV	Derivation of Calculation of Chain Transfer Agent Conversion with the Effect of Volume of Reaction Mixture Change	198
Appendix V	The Modified Mayo Equation	200

Copyright by Mahidol University

List of Tables

Table		Page
1-1	Examples of chain transfer agents having ability to give active CTA*	5
2-1	Summary of previous works on mathematical modelling and computer simulation of the free radical polymerization reactions.	17
3-1	Amount of substances in each polymerization batch.	26
3-2	Rate coefficients for computer simulation of the kinetic model of free radical polymerization of styrene in the presence of chain transfer agents	41
4-1	Values of $R_{ST,CTA,0}$, $R_{ST,CTA,0}/[CTA]_0$, $k'_{ST,CTA,0}$, $R_{CTA,0}$, $R_{CTA,0}/[CTA]_0$, and $k'_{CTA,0}$ for the thermal-initiated free radical solution polymerization of styrene using AIBN initiator at 70 °C	49
4-2	$R_{ST,CTA}$ and $k'_{ST,CTA}$ at 0.0, 2.5 and 5.0 % styrene conversions.	51
4-3	Values of R_{CTA} and k'_{CTA} at 0.0, 2.5 and 5.0 % styrene conversions.	53
4-4	The simulation values of k_p and k_{tr} at 70 °C.	71
4-5	Values of [CTA]/[M] and degree of polymerization (DP_n) at 2.5 and 5.0 % styrene conversion	77
4-6	Values of chain transfer constant (C_x) of carbon tetrachloride and 1-butanethiol at 2.5 and 5.0 % styrene conversion.	78

Copyright by Mahidol University

List of Figures

Figure		Page
1-1	Descriptive kinetic flowchart of the thermal-initiated free radical polymerization of a vinyl monomer in the presence of a chain transfer agent.	11
1-2	Descriptive diagram of chain transfer reactions of growing radicals to dead-end polymers leading to crosslinked and branched polymers.	12
1-3	Descriptive diagram of chain transfer reactions of active radical species R*, CTA*, RM*, and CTAM* to dead-end polymers leading to crossliked and branched polymers.	13
1-4	Descriptive diagram of addition reactions of radical species and growing radical species with unsaturated penultimate units of dead-polymers to form linear, braced and crossliked polymers.	14
3-1	Model for vacuum distillation setup	21
3-2	Typical HPLC chromatograph of pure styrene.	21
3-3	Reaction tube.	24
3-4	Typical Raman spectrum (4000 - 200 cm^{-1}) of the stock solution and of the polymerized samples when CCl_4 was the chain transfer agents.	27
3-5	Typical Raman spectrum (4000 - 200 cm^{-1}) of the stock solution and of the polymerized samples when 1-BuSH was the chain transfer agents.	28

Copyright by Mahidol University

Figure		Page
3-6	Typical Raman spectra of the polymerized samples : (A) Internal standard band, (B) Original spectrum, (C) the Curve-fitted spectrum.	31
3-7	Typical Raman spectrum (2650 - 2490 cm^{-1}) of Batches I-M, showing ν_s(-SH) $\approx$ 2577 cm^{-1}	33
3-8	Typical Raman spectrum (370 - 250 cm^{-1}) of Batches C-H, showing γ(C-Cl) $\approx$ 314 cm^{-1}.	34
3-9	GPC chromatogram of Batch C at reaction time at 20, 40, 60, 80, 100, and 120 min.	36
3-10	GPC chromatogram of Batch I at reaction time at 20, 40, 60, 80, 100, 120, 140, 160, and 180 min.	37
3-11	Programming flow chart of the first routine.	39
3-12	Programming flow chart of the second routine.	40
4-1	The plot of styrene conversion against time (Experiment: (□) Batch A and (◊) Batch B).	43
4-2	The plot of styrene conversion against time (Experiment: (□) Batch C, (◊) Batch D, (Δ) Batch E, (x) Batch G, and (o) Batch H).	45
4-3	The plot of CCl_4 conversion against time (Experiment: (□) Batch C, (◊) Batch D, (Δ) Batch E, (x) Batch G, and (o) Batch H).	46

Copyright by Mahidol University

Figure		Page
4-4	The plot of styrene conversion against time (Experiment: (□) Batch I, (◊) Batch J, (Δ) Batch K, (x) Batch L, and (o) Batch M).	55
4-5	The plot of 1-BuSH conversion against time (Experiment: (◊) Batch J, (Δ) Batch K, (x) Batch L, and (o) Batch M).	56
4-6	The kinetic mechanism of thermal-initiated free radically solution polymerization of a vinyl monomer in the presence of a chain transfer agent.	60
4-7	The plot of styrene conversion against time (Continuous line (———) for the simulation results, Experiment: (□) Batch A (k_p=320 l/mol.s) and (◊) Batch B (k_p=220 l/mol s)).	66
4-8	The plot of styrene conversion against time (Continuous line (———) for the simulation results, Experiment: (□) Batch C (k_p=245 l/mol.s), (◊) Batch D (k_p=230 l/mol.s), (Δ) Batch E (k_p=220 l/mol.s), (x) Batch G (k_p=200 l/mol.s)and (o) Batch H (k_p=180 l/mol.s)).	67
4-9	The plot of styrene conversion against time (Continuous line (———) for the simulation results, Experiment: (□) Batch J (k_{p1}=250, k_{p2}=170 l/mol.s), (Δ) Batch K(k_{p1}=275, k_{p2}=200 l/mol.s), (x) Batch L (k_{p1}=280, k_{p2}=225 l/mol.s), and (o) Batch M (k_{p1}=285 l/mol.s)).	68
4-10	The plot of CCl_4 conversion against time (Continuous line (———) for the simulation results, Experiment: (□) Batch C (k_{tr}=4.5x10^{-4} l/mol.s), (◊) Batch D (k_{tr}=1.2 l/mol.s), (Δ) Batch E (k_{tr}=4.5 l/mol.s), (x) Batch G (k_{tr}=4.8 l/mol.s),and (o) Batch H (k_{tr}=5.0 l/mol.s)).	69

Copyright© by Mahidol University

Figure		Page
4-11	The plot of 1-BuSH conversion against time (Continuous line (———) for the simulation results, Experiment: (□) Batch J (k_{tr}=950 l/mol.s), (Δ) Batch K (k_{tr}=1020 l/mol.s), (x) Batch L (k_{tr}=1050 l/mol.s), and (o) Batch M (k_{tr}=1060 l/mol.s)).	70
4-12	Overlaid MWD of the polymerization Batch A (Experiment (——), Computer simulation (——)).	71
4-13	Overlaid MWD of the polymerization Batch B (Experiment (——), Computer simulation (——)).	72
4-14	Overlaid MWD of the polymerization Batch C (Experiment (——), Computer simulation (——)).	73
4-15	Overlaid MWD of the polymerization Batch D (Experiment (——), Computer simulation (——)).	74
4-16	Overlaid MWD of the polymerization Batch E (Experiment (——), Computer simulation (——)).	75
4-17	Overlaid MWD of the polymerization Batch G (Experiment (——), Computer simulation (——)).	76
4-18	Overlaid MWD of the polymerization Batch H (Experiment (——), Computer simulation (——)).	77
4-19	Overlaid MWD of the polymerization Batch I (Experiment (——), Computer simulation (——)).	78
4-20	Overlaid MWD of the polymerization Batch J (Experiment (——), Computer simulation (——)).	79

Copyright by Mahidol University

Figure		Page
4-21	Overlaid MWD of the polymerization Batch K (Experiment (——), Computer simulation (——)).	80
4-22	Overlaid MWD of the polymerization Batch L (Experiment (——), Computer simulation (——)).	81
4-23	Overlaid MWD of the polymerization Batch M (Experiment (——), Computer simulation (——)).	82
4-24	Molecular Weight Distribution of Polymers (Batch N): Comparison between separated polymers and unseparated polymers	83
4-25	The plot of reciprocal degree of polymerization against [CTA]/[M], (CTA is carbon tetrachloride and M is styrene monomer).	84
4-26	The plot of reciprocal degree of polymerization against [CTA]/[M], (CTA is 1-butanethiol and M is styrene monomer)	85

Copyright by Mahidol University

Glossary

Symbol	Nomenclature
CTA	Chain transfer agent.
M	Monomer.
I	Initiator.
R*	Primary initiator radical.
CTA*	Primary chain transfer agent radical
M_n*	Growing polymer radical .
RM_n*	Growing polymer radicals having fragment R derived from the initiator.
$CTAM_n$*	Growing polymer radicals having molecular fragment derived from CTA.
M_n	Dead-end polymer.
M_nX	Dead-end polymer having fragment X derived from CTA at one entity.
RM_n	Dead-end polymer having fragment R derived from the initiator at one entity.
RM_nX	Dead-end polymer having fragment R and X derived from the initiator and CTA, respectively, at each entity.
RM_nR	Dead-end polymer having fragments R derived from the initiator at both entities.
$CTAM_n$	Dead-end polymer having molecular fragment derived from CTA at one entity.
$CTAM_nX$	Dead-end polymer having molecular fragments or atom derived from CTA at both entities.
$CTAM_nCTA$	Dead-end polymer having molecular fragments derived from CTA at both entities.
$CTAM_nR$	Dead-end polymer having molecular fragments derived from the initiator and CTA at each entity.
k_d	Rate coefficient for the initiator dissociation reaction.
k_{sr}	Rate coefficient for the reaction between R* with the chain transfer agent.
$k_{i,CTA}$	Rate coefficient for the reaction between CTA* with the monomer.

©Copyright by Mahidol University

k_i	Rate coefficient for the initiation reaction
k_p	Rate coefficient for the propagation reaction.
k_{tc}	Rate coefficient for the termination by combination reaction.
k_{td}	Rate coefficient for the termination by disproportionation reaction.
k_{tcc}	Rate coefficient for the reaction between CTA* with the growing polymer radical.
k_{tci}	Rate coefficient for the reaction between R* with the growing polymer radical.
k_{tr}	Rate coefficient for the chain transfer reaction of active growing radicals to the chain transfer agent.
k_{trm}	Rate coefficient for the chain transfer reaction of active growing radicals to the monomer.
f	Initiator efficiency.
Δ	The thermal energy.
C_x	Chain transfer constant.
Q_r	The r-th moment.
W_n	Weight fraction of dead-end polymers.
P(x)	The growing radical concentration of species x.
P_x	The total growing radical concentration of species x.
R_p	Rate of polymerization.
MW	Monomer molecular weight.
M_n	The number-average molecular weight of polymers.
M_w	The weight-average molecular weight of polymers.
PDI	Polydispersity index.

Copyright by Mahidol University

Aims and Objectives of the Research

Overall aim

To study the kinetics of the thermal-initiated free radical polymerization reaction in the presence of a chain transfer agent.

Specific objectives

Specific objectives include the following :-

a) To study experimentally and computationally the non-classical kinetics of thermal-initiated free radical polymerization reaction in the presence of various types of chain transfer agents

b) To establish the deterministic dynamic predictive model representing such the kinetics.

c) To establish the Modified Mayo Equation.

Scope of the Research

1) To establish the non-classical kinetic mechanism of thermal-labile free radical polymerization reactions of vinyl monomers in the presence of two types of chain transfer agents, i.e. carbon tetrachloride and 1-butylmercaptan.

Copyright by Mahidol University

2) To establish the deterministic dynamic predictive mathematical model of the kinetics in 3.3 1 and to simulate the model using a computer.

3) To compare results obtained from the computer simulations and from experiments.

4) To derive the Modified Mayo Equation.

Research Methodology

Raman spectroscopic technique was employed to follow conversions of styrene monomer and the chain transfer agents at specific reaction time Gel Permeation Chromatography (GPC) was used to obtain various average molecular weights and molecular weight distribution of polymers. Deterministic mathematical modelling techniques in particular the generating function and the numerical integration (4^{th} Order Runge-Kutta and Newton-Ralphson) methods were used to solve the sets of differential equations of the kinetic model by using computer programmes written in the Microsoft Quick BASIC computer language and a IBM/PC computer.

Copyright by Mahidol University

Chapter I
Free Radical Polymerization Reactions

1.1 Introduction

Free-radical chain reactions have been known almost a quarter of century prior to establishment of details of free radical polymerization reaction in the mid-1930s [1]. Since then relevant research activities were conducted in development of industrial polymerization processes for production of a large quality of polymeric materials. There were also some research activities on other chain reactions apart from the free radical polymerization such as ionic and insertion polymerization reactions. In the ionic polymerization, molecular weight and molecular weight distribution of polymers can be controlled by controlling a concentration ratio of an initiator to a monomer. In contrast, it is rather impossible to control such the said characters via the free radical polymerization without the presence of a chain transfer agent (CTA). For the insertion polymerization (also called Ziegler-Natta Polymerization), stereoregular polymers can be synthesized [43] while the others are rather impossible.

The free radical polymerization has been employed to prepare polymers commercially up to now because of it simplifies synthesis methodology. Related researches on the free radical polymerization have been reviewed [36, 63, 69]. A number of researches have also studied the kinetics of free radical polymerization reactions and have been reviewed in many papers [18, 33, 36, 48, 63]. Mathematics and computer programming have been used as tools for studing such kinetics in details [18, 25, 32, 70, 72, 74].

Copyright by Mahidol University

Results obtained from computer simulations base on the mathematical modelling, which is depending on the kinetic mechanism of polymerization reactions or processes.

1.2 Classical Free Radical Polymerization

It is well-known that the typical free radical reaction comprises three elementary steps, namely, initiation, propagation and termination reactions. These also appear in the ideal kinetic mechanism of the free radical polymerization. Initiator radicals may be obtained from homolytic or heterolytic cleavage of a labile chemical bonding such as C-N, C-S, O-O, depending on energy used for dissociation of those bonds [36]. Thermal, and low- and high-energy irradiation are sources of energies causing the bond cleavages.

In the free radical polymerization reactions substances which can be used as initiators may be organic compounds, such as 2,2'-azo-bis-2-methylpropionitrile (AIBN), benzoyl peroxide, or inorganic compounds, such as potassium persulphate, ferric hydroxide (extensive reviews on this topic can be found in a monograph written by Herman (1985)). I is hereafter abbreviation for an initiator. When I is exposed to irradiation or subjected to thermal treatment, initiator radicals (R*, where R is a molecular fragment of the initiator, and * represents hereafter as the radical symbol) are formed. This is commonly known as the initiator dissociation reaction (R-1), Figure 1-1. In the case of radiation-initiated polymerization, it is not be necessary to employ an initiator since a monomer can be induced to be a monomer radical itself. In this work only

© Copyright by Mahidol University

the thermal-labile initiator is focused.

In the initiation (R-2), a monomer (M) adds to R* to form an active radical RM*. The propagation reaction (R-3) then takes place subsequently by addition of the monomer molecules one to another to RM* to form growing active radicals of the type RM_n*, where n is the degree of polymerization or the number of repeating unit, and n must be an integer greater than 1). When RM_n* has long chain enough causing higher viscosity in the reaction medium, termination of these radicals occurred either by disproportionation (R-5) or by combination (R-4) mechanisms.

1.3 Non-Classical Free Radical Polymerization

In fact the free radical polymerization reaction is not simply composed of the three elementary steps as aforementioned. Chain transfer reaction is another one which always exists during polymerization. By definition, chain transfer reaction is that in which a labile atom of a molecule presented in a reaction medium is abstracted by an active radical. A molecule containing a labile atom(s), which may be H and halogen atoms, is called a transfer agent. In polymerization reaction, chain transfer reaction (R-6) usually occurs between RM_n* and the transfer agent (in this case it is preferably called chain transfer agent (CTA)) to give dead-end polymer of the type RM_nX, where X is the fragment derived from CTA, and primary CTA radical (CTA*). CTA may be any substances presented in the reaction medium provided that they contain the labile atom(s) or molecule(s) in the molecule. Initiator, monomer, dead-end polymer, the

Copyright by Mahidol University

reaction medium reagent (i.e., solvent), dedicated CTA, or foreign substances [16, 36] in the polymerization medium may be act as CTA.

Ideally CTA* should not be active to react with the monomer molecule so that CTA could be used to control the polymer average molecular weight (MW) [2]. Besides, It was shown by Ray and Rudin [18, 49] that CTA could be used to control the average molecular weight distribution (MWD) to be the normal distribution.

In the case that CTA* is active and reacts with M (C-2) the other primary active radical of the type CTAM* is produced. CTAM* propagates by reacting with the monomer molecules to give the active growing radicals of the type $CTAM_n$* (C-3). Destruction of $CTAM_n$* either by the combination (C-4) or by disproportionation (C-5) reactions gives dead-end polymers of the type $CTAM_{x+y}CTA$, $CTAM_x$, and $CTAM_y$,where x and y are the degree of polymerization, and they must be integer greater than 1. $CTAM_n$* may proceed the chain transfer reaction (C-6) with CTA to give dead-end polymer of the type $CTAM_nX$ and CTA*. $CTAM_nX$ has end-functional groups CTA- and X-, which are the fragment of CTA. Low molecular weight polymers having functional groups at both chain ends are named "telechelic or end-functionalized polymers". In the case that the low molecular weight polymers have only one functional group at the end, they are called "semi-telechelic or semi-end-functionalized polymers".

Copyright by Mahidol University

CTA* produced may reinitiate to M (reaction C-2) and then the propagation and the termination reactions are proceeded subsequently in the same manner as mentioned earlier. This type of reaction has been

named as Degradative Chain Transfer Reaction [22, 29]. Examples of CTA having ability to give active CTA* are shown in Table 1-1. It can be observed that those CTAs have quite high chain transfer constant (C_x), defined as the ratio of the rate coefficient for the destruction of active growing radicals by chain transfer reaction to that for the propagation.

Table 1-1 Examples of chain transfer agents having ability to give active CTA*.

Substances	C_x
Carbon tetrabromide	1.8
1-Dodecanethiol	14.8
1-Octanethiol	19.0

(From Seymour R.B. and Carraher C.E. *Polymer Chemistry*,1987)

It has also been reported that R* can abstract an electron from a labile atom of CTA (C-1) to give CTA* and inactive initiator derivative (RX) [55]. In this case, CTA must have high C_x values, such as carbon tetrabromide [52]. Then CTA* reacts with M to yield CTAM* (C-2), and further addition of the monomers to CTAM* yields $CTAM_n$* (C-3). Further reactions are the termination reactions of $CTAM_n$* via combination (C-4), disproportionation (C-5), or the chain transfer reaction (C-6) to give series of dead-end end-functionalized and semi-end-functionalized polymers.

Copyright by Mahidol University

Competition reactions between R* and CTA* with M should occur during the free radical polymerization reaction in the presence of a chain transfer

agent. If CTA has high value of C_x, it will be very active to react with R* immediately to give CTA*. Then further reaction between CTA* and M proceeds in parallel to that of R* with M. The growing radicals $CTAM_n$* will of course prefer the chain transfer reaction rather than the combination or disproportionation reactions. These cause CTA to be exhausted much more quickly, and the dead-end polymers will have very low degree of polymerization (DP). A polymer with low DP is called an oligomer, which is defined as a macromolecule having MW in the range of 2-20 daltons [35]. Oligomers prepared via the free radical reaction in the presence of a chain transfer agent are called telomers. The terms telomerization, taxogen and telogen are used instead of the terms polymerization, monomer and CTA, respectively [3].

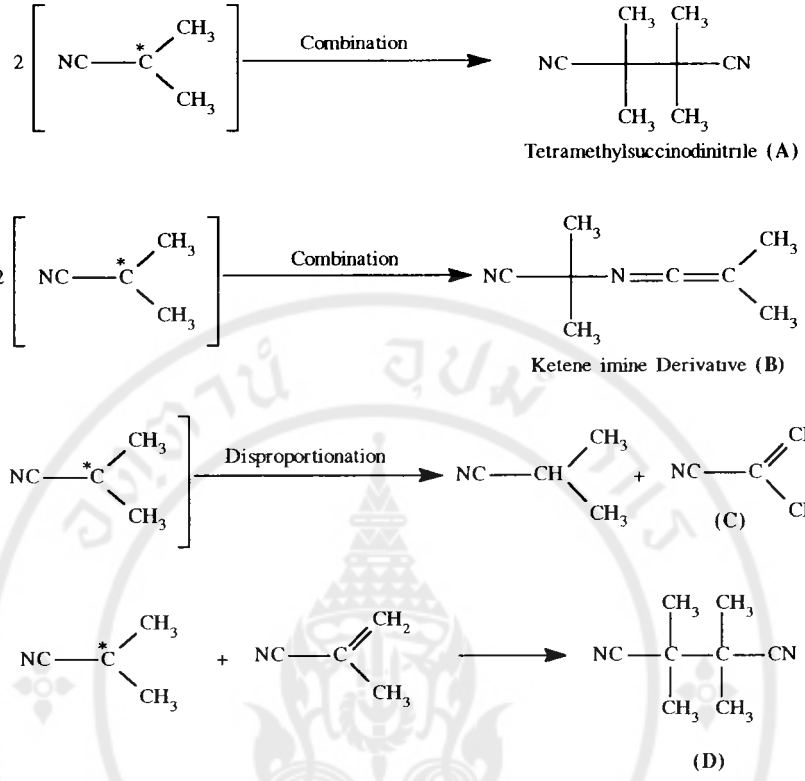
Even though CTA has been exhausted from the reaction medium, the reaction of R* and the monomer left in the reaction medium proceeds continuously. The growing radicals RM_n* produced while CTA being presented in the reaction is certainly preferred the chain transfer reaction to CTA. While RM_n* produced in the absence of CTA should proceed the termination either by combination or disproportionation reactions to obtain dead-end polymers in the same manner as mentioned in Section 1.2, the mechanism described earlier has not yet been studied by any workers and it has been attempted to study here for the first time. Bearing in mind that destruction of RM_n* may also be the chain transfer reaction to the telomers produced in the early stage of the reaction. This is possible since each telomer has labile end groups at both chain ends. Products obtained in this case are the mixture of telomers ($CTAM_{n+m}CTA$, $CTAM_{n+m}X$) and polymers of the types $RM_{n+m}R$, $RM_{n+m}CTA$, RM_nX, $CTAM_n$.

Copyright by Mahidol University

In the case that any types of dead-end polymers produced act as the chain transfer agents, the growing active radicals and active radicals (of the types R*, CTA*, RM*, and CTAM*) can then abstract the labile atoms of such polymers yielding the new dead-end polymers and inactive species (of the type R, CTA, RM, and CTAM) and the new polymer radicals. The latter should have in general the active radical sites on the polymer backbones rather than at the chain ends. The new active polymer radicals may propagate by reacting with the monomers and with polymers having unsaturated molecules at the chain ends yielding branched unit with active radical sites. The polymer radicals and the branched polymer radicals may be terminated via the combination or the disproportionation reactions with other growing active radicals, or via the chain transfer reactions with CTA. The combination of these radicals gives polymers having crosslinking structure while the termination via the disproportionation and the chain transfer reactions obtain branched polymers. These are depicted in details in Figures 1-2, 1-3 and 1-4.

The non-classical kinetic mechanism discussed above has not yet been mentioned concerning possible reactions of R* itself. Barbe and Rüchardt (1983) reported that AIBN radical, i.e., $(CH_3)_2(CN)C^*$, could be destroyed either by combination or disproportionation reactions to give **(A)**, **(B)** or **(C),** respectively. Furthermore $(CH_3)_2(CN)C^*$ was also possibly added to **(C)** yielding **(D)**.

Copyright© by Mahidol University



The term initiator efficiency (f) has been used in general in order not to account for these reactions in the free radical polymerization kinetics. f can be defined in many ways, and in this work f is defined following Herman et al (1985) as :

$$f = \frac{R_i}{nk_d[I]} \qquad (1\text{-}1)$$

where

[I] is the initiator concentration at reaction time t,

n is the number of R* dissociate from each initiator molecule,

k_d is the rate coefficient for the initiator dissociation reaction,

R_i is the rate of initiation reaction.

Copyright by Mahidol University

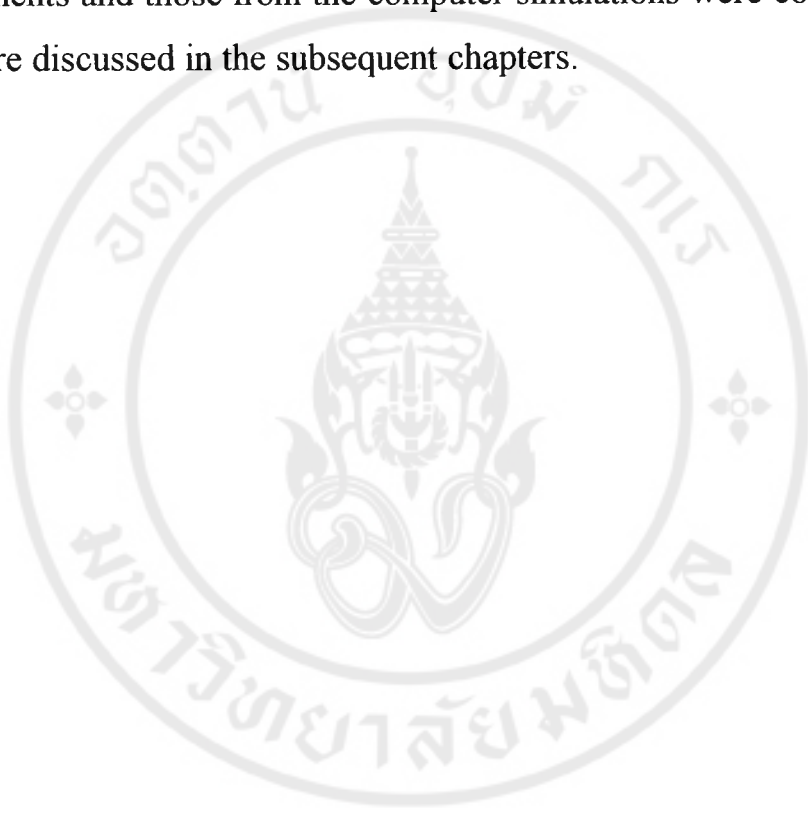
f may be defined as the probability factor for the primary radical R* to undergo reaction with monomer rather than to combine with other growing

radicals to form dead-end polymers [38].

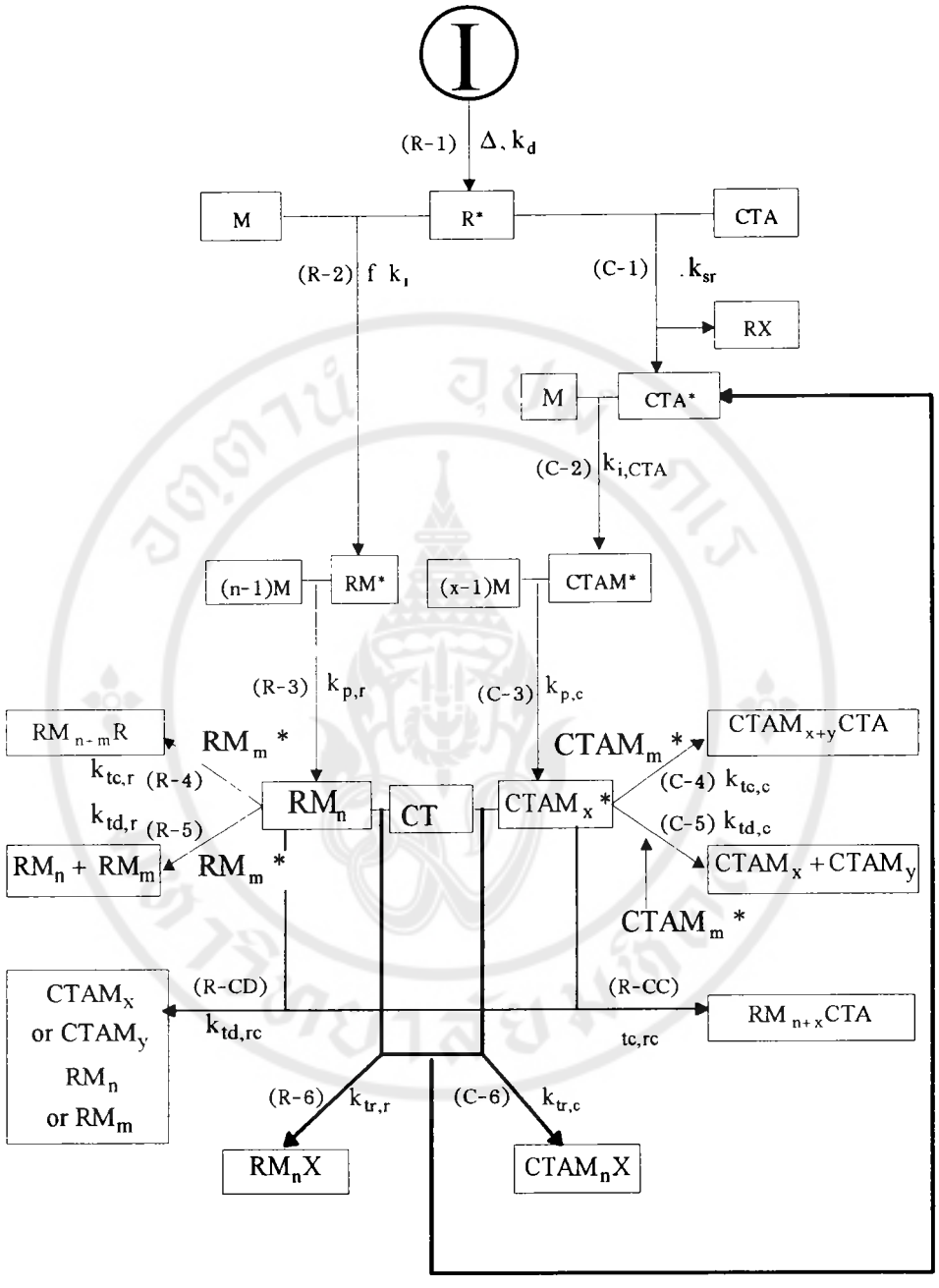
As discussed above the non-classical kinetics of free radical polymerization reactions in the presence of chain transfer agents are very complicated. In particular, the chain transfer agents have various C_x values. C_x of a chain transfer agent indicates its effectiveness in the chain transfer reaction with the growing radicals, which affect on MW and types of dead-end polymers obtained. The amount of CTA affects the same manner as the value of C_x. Thus CTA is not only used as polymer chain length controller, it can also be used to provide functional groups at the polymer chain end(s). The Mayo Equation[2] is generally used for evaluation a value of C_x of a chain transfer agent or for prediction and controlling of a polymer MW synthesized via the free radical polymerization in the presence of CTA. The Mayo Equation was derived from the classical free radical polymerization kinetics with the additional of the chain transfer reaction of CTA to the growing active radicals [2]. Many assumptions were made on the classical kinetics, such as *(1) rate coefficient for the propagation and termination reactions are independent of the polymer kinetic chain length, (2) auto acceleration reaction (Trommsdroff effect or gel effect) is not accounted.* Therefore, the Mayo Equation has the same assumptions mentioned earlier. In addition to this, another assumption is made that the CTA* is not active to react with monomers, which has been later found contrary to many works [18, 22, 25 29, 32, 55, 57, 58, 68, 70, 71, 75, 76]. As mentioned earlier, the competition reactions of R* with M and R* with CTA are possible. If CTA is very reactive, R* trends to abstract a labile atom from CTA rather than reacts with M. This evidence was also not accounted for the Mayo Equation.

Copyright by Mahidol University

Therefore, in this work the non-classical free radical polymerization kinetics were studied aiming to derive the Modified Mayo Equation. Mathematical modelling and computer simulation techniques were also employed as a tool for studing the kinetics in detail. Results obtained from experiments and those from the computer simulations were compared, and these are discussed in the subsequent chapters.



Copyright by Mahidol University



Where RM_n and $CTAM_x$ are the dead-end polymers having saturated penultimate units at the chain ends. RM_m and $CTAM_y$ are the dead-end polymers having unsaturated penultimate units at the chain ends.

Copyright by Mahidol University

Figure 1-1 Descriptive kinetic flowchart of the thermal-initiated free radical polymerization of a vinyl monomer in the presence of a chain transfer agent.

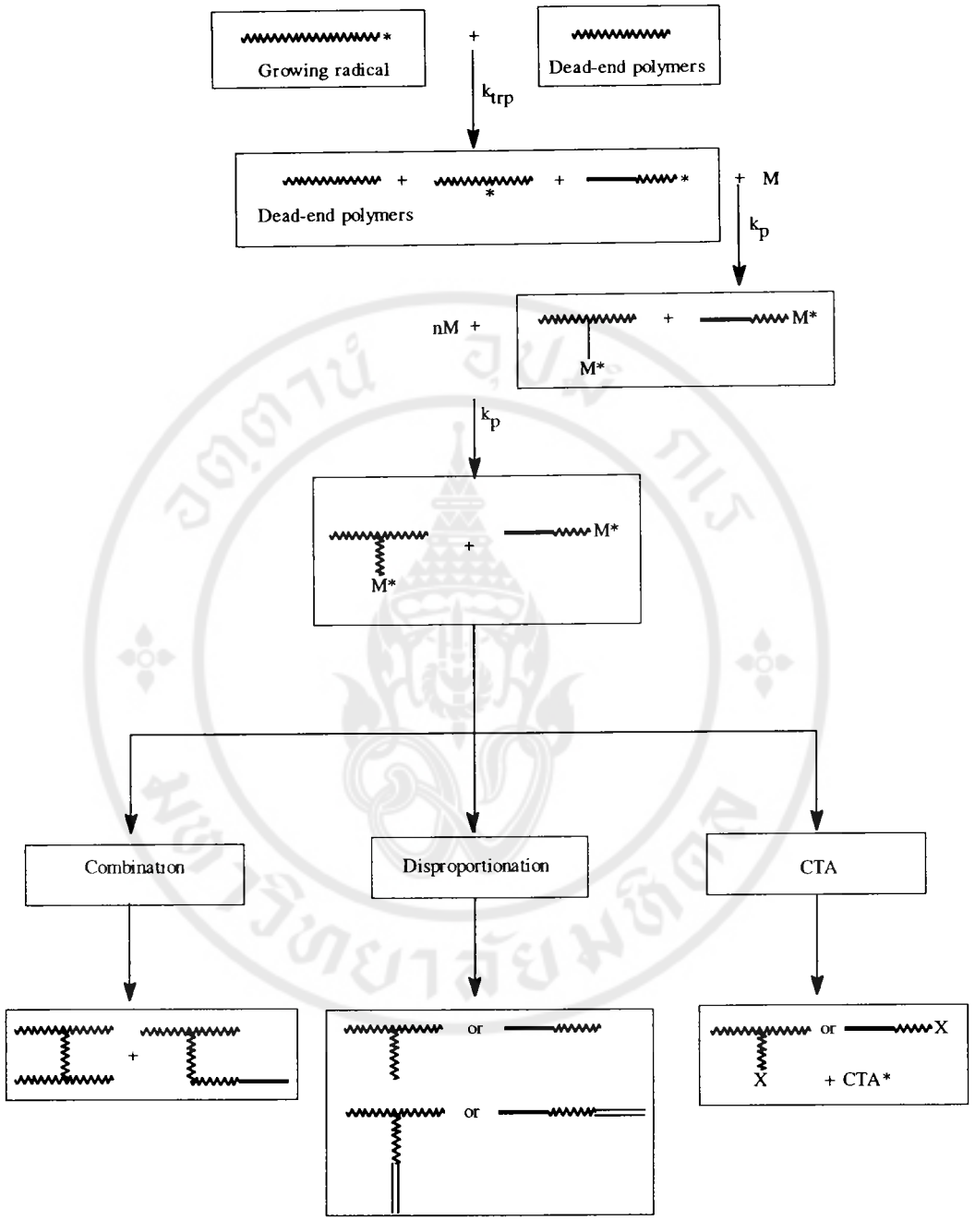


Figure 1-2 Descriptive diagram of chain transfer reactions of growing radicals to dead-end polymers leading to crosslinked and branched polymers.

Copyright by Mahidol University

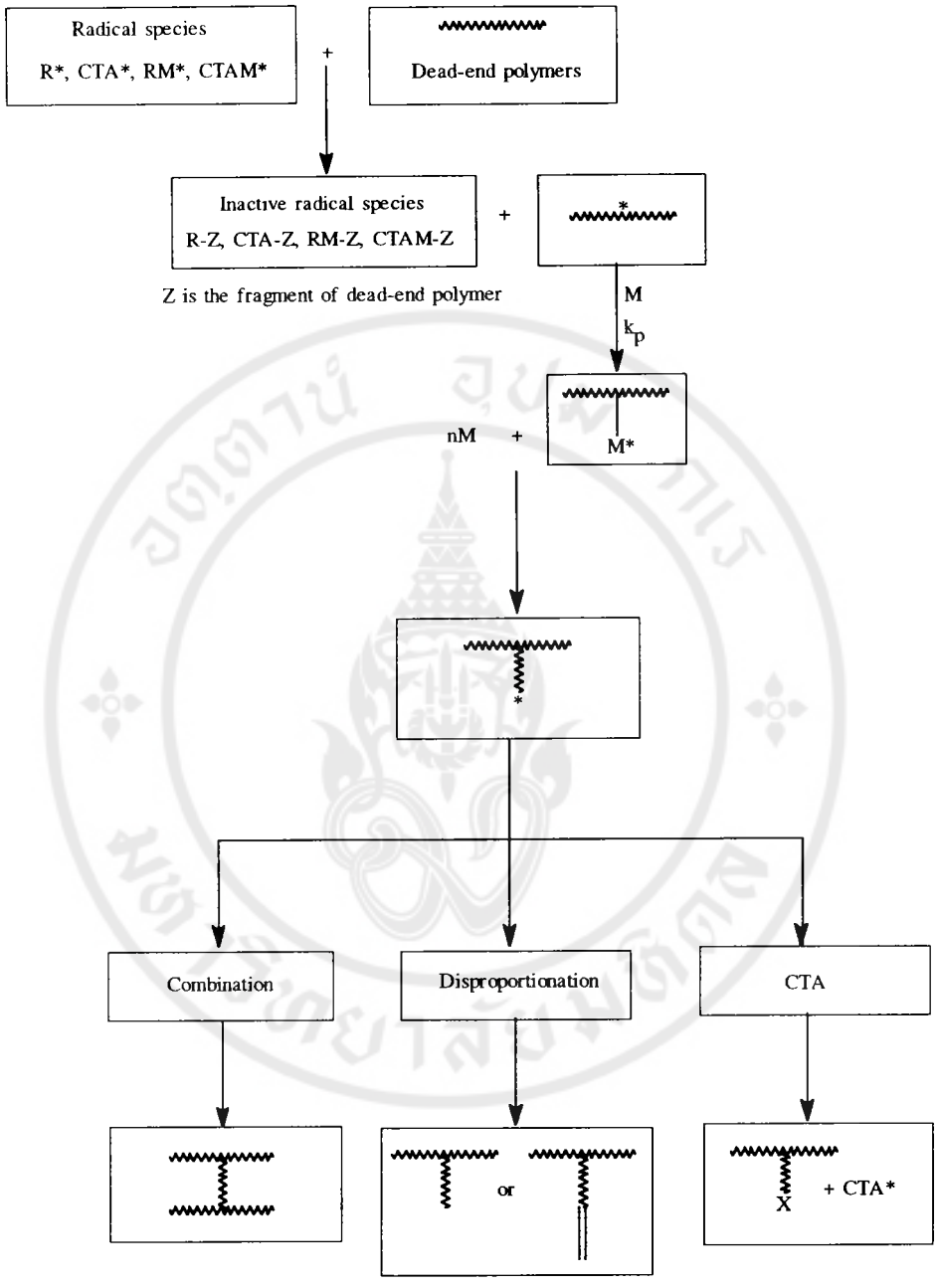


Figure 1-3 Descriptive diagram of chain transfer reactions of active radical species R*, CTA*, RM*, and CTAM* to dead-end polymers leading to crossliked and branched polymers.

Copyright by Mahidol University

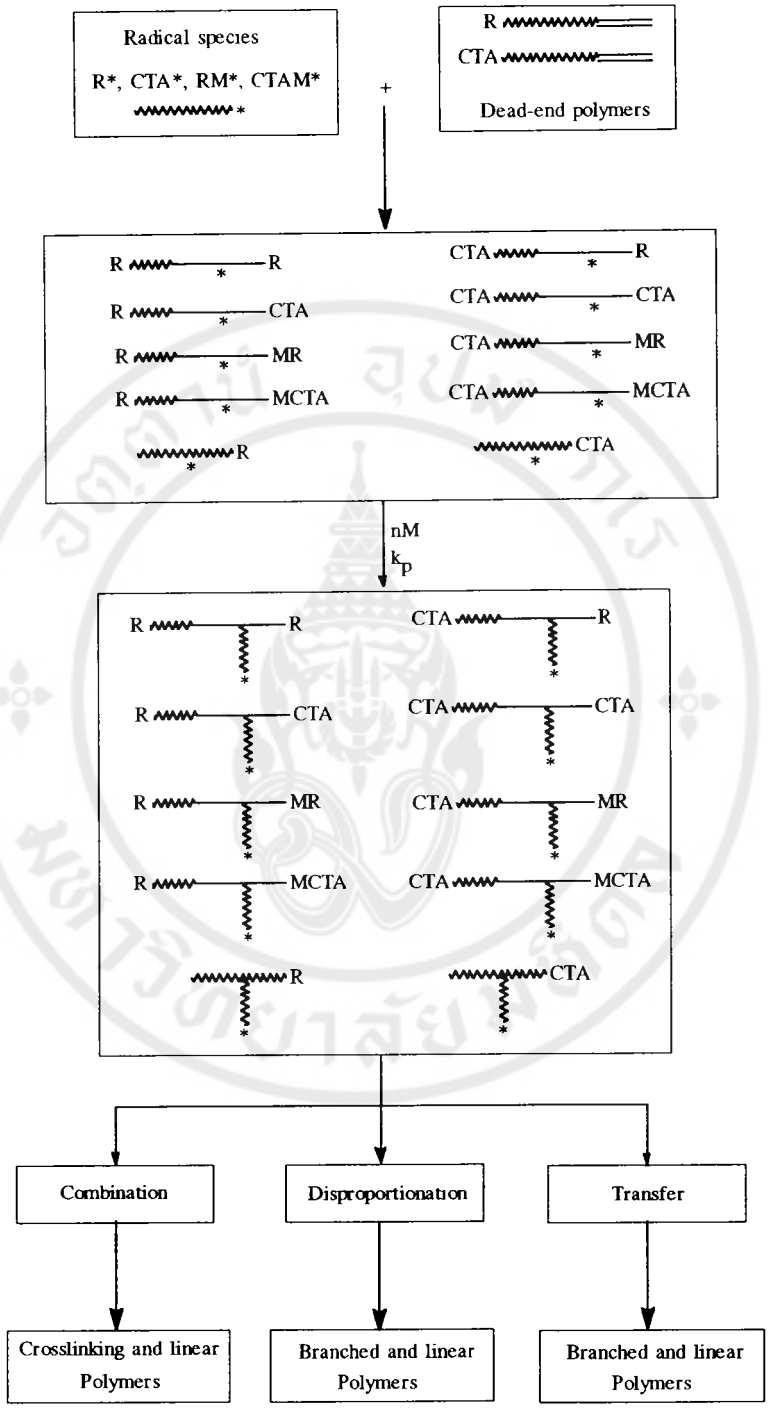


Figure 1-4 Descriptive diagram of addition reactions of radical species and growing radical species with unsaturated penultimate units of dead-polymers to form linear, branched and crossliked polymers.

Copyright by Mahidol University

Chapter II

Mathematical Modelling of the Thermal-Initiated Free Radical Polymerization Reaction of a Vinyl Monomer in the Presence of a Chain Transfer Agent

2.1 Introduction

Mathematical modelling is a technique used to represent a mathematical model of a case study. In Polymer Science and Technology, the mathematical models have been employed to gain fuller understanding of kinetic mechanisms of polymerization reactions and processes, structures and properties of polymers [31, 60, 61]. Discussions hereafter are confined to the kinetics of free radical polymerization reactions of vinyl monomers in the presence of chain transfer agents which are relevant to the work presented here.

Polymerization reactions present particular problems distinct from those found in the modelling of other chemical reactions because the former involves a very large number of distinct growing species. Careful use of mathematical modelling techniques is extremely essential to avoid a problem of having to explicitly derive equations for each particular species in the polymerization reactions. There are two major mathematical techniques, i.e., Deterministic method and the Probabilistic (Statistical) technique (sometimes called Stochastic technique). They have been used for studying the kinetic mechanisms of the free polymerization reactions, and summary of a few previous works are in Table 2-1. The rigorous

Copyright by Mahidol University

degree of mathematics involved in deriving a model, which in this case is the kinetic mechanism, is of little consequence, should the model be based on poor mechanistic assumptions. Therefore, an understanding of the fundamental kinetic mechanism of the system studied is of extreme importance.

2.2 Mathematical Modelling Techniques

2.2.1 Deterministic Techniques

Generating functions, Z-transforms, the continuous variable approximation and numerical integration methods are examples of the deterministic techniques [4, 6, 25, 32, 70, 72]. The generating functions and the numerical integration methods, in particular the fourth order Runge-Kutta and the Newton-Ralphson numerical methods, were used in this work. Theories of these methods can be found in standard monographs [14, 24, 47].

The generating function approach has been widely used for the prediction of differential MW distribution of a polymer, since the moment of the distribution is readily related to the generating functions. Scanlan (1956) was the first to use the technique and many others followed such as Bamford and Jenkins (1960), Liu and Amundson (1962), Ray (1972), Armitage (1983).

Copyright by Mahidol University

Table 2-1 Summary of previous works on mathematical modelling and computer simulation of the free radical polymerisation reactions.

Mathematical technique	Study case	References
Continuous Variable Approximation -Laplace Transforms and Laguerre Polynomials	Vinyl polymerization: molecular weight distributions in branched polymers.	Bamford and Tompa (1954)
Z-Transforms	Irreversible condensation polymerization.	Abraham (1966)
Generating Functions	Vinyl polymerization: combination of vinyl polymers.	Bamford and Jenkins (1960)
	Free radical polymerization in batch reactor.	Liu and Amundson (1962)
	Free radical polymerization of styrene in batch reactor.	Ray (1972)
Monte-Carlo	Free radical copolymerization in the presence of chain transfer agents (thiols).	Galbraith et al (1987)
Numerical Integrations -Runge-Kutta-Gill	Free radical polymerization in batch reactor.	Liu and Amundson (1962)
Numerical Integrations -Fourth order Runge-Kutta	Free radical polymerization of styrene in benzene.	Hamielec and Hodgins (1967)
Numerical Integrations	Thermal-initiated free radical polymerization of styrene in ethylbenzene.	Cutter and Drexler (1982)
	Bulk polymerization of styrene in cyclohexane.	Villermaux et al (1983)
	Bulk polymerization of vinyl acetate: transfer to chain transfer agent and terminal double bond polymerization.	Villermaux et al (1983)
Numerical Integrations -Fourth order Runge-Kutta and Bessel Functions.	Free radical polymerization of methyl mathacrylate in batch reactor.	Venkateshwaran and Kumar (1992)
Numerical Integrations -Statistical Moments	Non-steady-state kinetics of thermal-initiated free radical polymerization.	Yan et al (1992)
Numerical Integrations -Fourth order Runge-Kutta and Newton-Ralphson.	Kinetic of thermal-initiated free radical polymerization of a vinyl monomer in the presence of a chain transfer agent.	This work

Copyright by Mahidol University

This method is inappropriate for an application to the kinetics of free radical telomerization reactions, where the telomers have very low number of repeating units. It is because the MW distribution of telomers cannot be approximated to a continuous function, and therefore cannot be meaningfully described by the moments [52].

The numerical integration methods have been widely used, and many computer software packages of a range of numerical algorithms have been available in various computer languages [37, 41]. For solving the particular numerical integration of model equations of polymerization reactions, REMECH package was developed by Detar (1966 and 1967), Pryor and Coco (1970). FACSIMILE developed by Chance (1977) was designed especially for use in the modelling of chemical processes. The computer programs written in BBC BASIC and the Microsoft© QuickBASIC (version 4.0) for studies the kinetics of thermal-initiated free radical telomerization reactions were developed by Euapermkiati (1990).

2.2.2 Stochastic Techniques

An example of these techniques is Monte-Carlo methods, which can be employed to approximate integrals as well as to solve differential equations with the usage of random numbers [23, 40, 77]. The excellent monographs by Lowry (1970), Bruns et al (1981), and Binder et al (1987) described the techniques and applications in not only Polymer Science but also in Physics.

Copyright by Mahidol University

2.3 Mathematical Modelling and Computer Simulation of the Free Radical Polymerization Reactions in the Presence of a Chain Transfer Agent.

The kinetics of free radical polymerization reactions in the presence of chain transfer agents are very complicated as discussed in Chapter I. In turn this may cause scant attention to researchers in the field. Much more attention has been made to researches on fundamental kinetic studies, such as evaluations of the rate coefficients for the propagation and termination reactions [44, 62] and of C_x for various chain transfer agents [2, 58, 75].

For the last two decades, the mathematical modelling and computer simulation of the free radical polymerization reactions have been paid more attention on, and this has been reviewed by Ray (1972), Villermaux et al (1983), Rintelen et al (1983), Herman et al (1985), Kumar and Khandelwal (1989), Yuan et al (1991).

2.4 Stiffness and Quasi-Steady-State Assumption (QSSA)

In the free radical polymerization, stiffness might be encountered when using numerical methods for the simulation of the process models because of the fast rate of initiator dissociation to initiator radicals relatively to those of the slow rate of propagation and termination reactions. Therefore, the differential expressions take a very wide range of numerical values, requiring a very small integration step size with most integration algorithms if numerical instability is to be avoided. The problem can be overcome using the special Gear method, which uses a backward

Copyright by Mahidol University

differential technique [42]. In fact the fastest reaction rates decrease after a short time making the stiffness of the system less severe. Hence, the differential equations will be integrated efficiently over periods of several hours by selecting suitable step length in the region of acceptable stability of the solution. An alternative approach is to use the quasi-steady-state-assumption (QSSA).

In general QSSA is usually applied in modelling of free radical polymerization kinetics by which the rate of formation of radicals is assumed to be equal to that of the loss. The concentrations of initiator and monomer change slowly so as to the equilibrium radical concentrations. In effect, the rates of change of all radical species with respect to time can be approximately equal to zero, which allows algebraic expressions to be derived relating the radical concentrations to those of non-radical species. The kinetic differential equations of the non-species remain to be solved by a numerical integration method. QSSA technique was also applied in this work.

Copyright by Mahidol University

Chapter III

Experimental

3.1 Materials

Styrene monomer donated by Siam Fine Chemical Co., Ltd. was purified by modification of the method described by Collins et al (1973). The inhibitor, hydroquinone, was removed by washing twice with 10% NaOH (aq.) solution in a separating funnel. The aqueous phase was drained off. The monomer phase was then washed with distilled water until pH 7 was obtained. Then anhydrous sodium sulphate was added to absorb small amount of water in the monomer phase for 1-2 hours. Later the monomer was filtered and then distilled under vacuum. The vacuum distillation setup was shown in Figure 3-1. It is noted that to a litre of monomer was added 1 g $CuCl_2$ stabilizer during the distillation process. The pure styrene was distilled and obtained at 40-45 °C (the pressure cannot be measured as a pressure gauge was not available). The purity of styrene was determined using a high performance liquid chromatography (HPLC: model 510 Waters pump, model 484 Waters UV absorbance detector at 254 nm, a Nova-Pak C-18 column (30 cm length, and 0.75 cm diameter)). HPLC grade acetonitrile was the mobile phase. Typical HPLC chromatograph is shown in Figure 3-2; peak retention time of pure styrene is at 18 minutes. The purified styrene was stored in a brown capped bottle sealed with parafin film in a refrigerator at -5 °C, and it was used within 2 days.

Copyright by Mahidol University

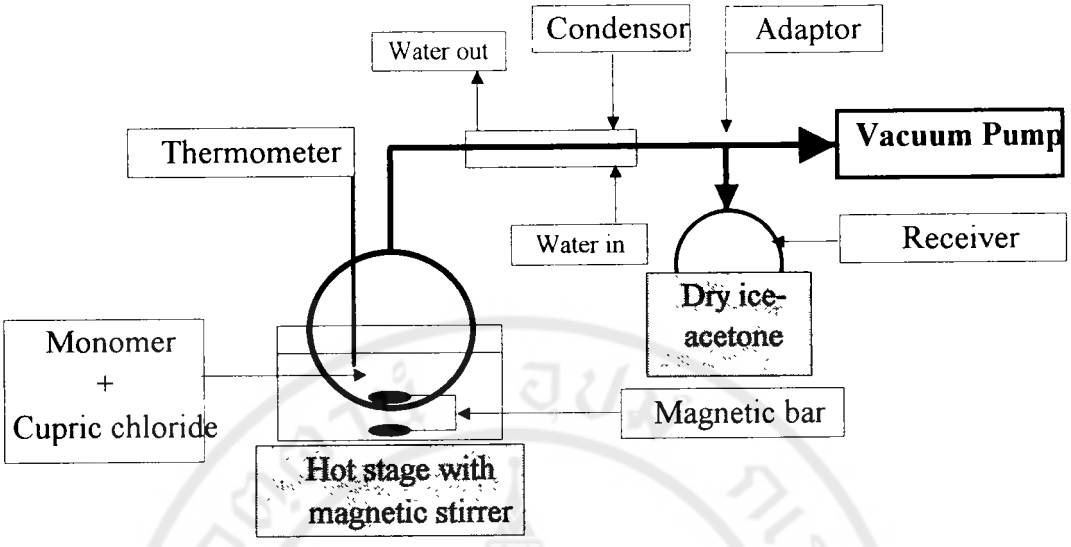


Figure 3-1 Model for vacuum distillation setup.

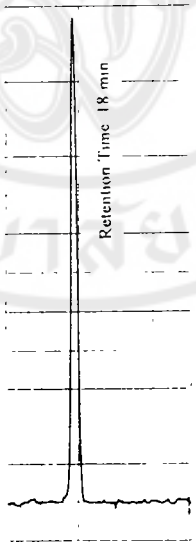


Figure 3-2 Typical HPLC chromatograph of pure styrene (injection volume = 20 μl of 0.08 M styrene solution (acetonitrile solvent), chart speed 300 mm/hr)

Copyright by Mahidol University

AIBN (Brighten Polytrading Co., Ltd.) was purified twice by recrystallization in acetone. Purified AIBN was stored in a tube in the absence of air. The tube was covered by aluminuim foil to protect radiation and stored in a freezer at -20 °C.

Carbon tetrachloride (CCl_4, Merck) was purified by distillation at normal atmosphere. A Raman spectrometer (system 2000 Perkin Elmer) was used to confirm the purity of CCl_4 by comparison with standard CCl_4 spectra.

1-Butanethiol (1-BuSH, Fluka) was purified under normal pressure. The middle portion of distillate was collected and stored in a refrigerator at -5 °C.

Laboratory grade sodium hydroxide (Eka Nobel AB, Swenden) and cupric chloride (BDH), anhydrous sodium sulphate (Na_2SO_4, Fluka), analar tetrahydrofuran (BDH) and cyclohexane (Fluka), HPLC grade acetonitrile (Merck) were used as obtained from suppliers.

Crude acetone and methanol were purified by distillation at normal atmosphere. Distilled water (3A HPLC grade) was supplied by Chemistry Department, Faculty of Science, Mahidol University.

3.2 Instrumentation

Copyright by Mahidol University

A near infra-red fourier transform Raman spectrometer (system 2000 Perkin Elmer) was used for following the kinetics of thermal-initiated free radical polymerization of styrene in the presence of CTA (in this study,

CTA is CCl_4 and 1-Butanethiol). A power source of the spectrometer is Nd YAg laser (5 W maximum), with typical working power of 2 W-100 mW. A detector used is ultra-InGaAs (Indium-Gallium-Arsenide) operated at the ambient temperature. Acceptable wave number for obtaining a spectra ranges from 200-4000 cm^{-1}.

A Gel permeation chromatography (GPC, 150CV Waters) was employed for determination of MW and MWD of synthesized polymers. Two mixed bead columns (Polymer Laboratories, 10 μm particle sizes, length 30 cm., and diameter 7.5 mm.) were used. A detector used was a refractrometer which was the instrument compartment.

The EX-600 water shaking bath (NESLAB) was employed; the bath temperature can be controlled with the tolerance ± 0.1 °C.

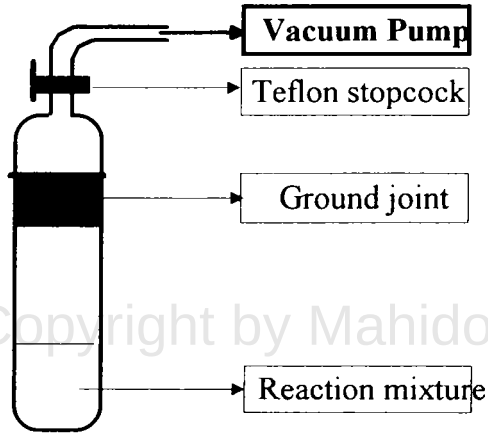
A IBM PC/AT compatible computer (Wearnes® Intel® *486* DX-2) was employed for computer simulation experiments. The computer CPU unit has a Intel® 80486, 50 MHz main processor, built-in math co-processor, DOS operating system with 640 KB RAM extended to 3328 KB, installed 207 MB hard diskette, built-in 2 disk drives (3.5 and 5.25 inches), and an Inport mouse Version 8. A terminal is of the type video graphics array.

Copyright by Mahidol University

3.3 Experimental

3.3.1 Preparation of Stock Reagents

Accurate quantity (in g.) of AIBN (W_{AIBN}) was placed in a dry 50 ml volumetric flask. Accurate amount (in ml.) of Cyclohexane ($V_{cyclohexane}$), CCl_4 (V_{CCl4}), or 1-BuSH ($V_{butanethiol}$) were also charged into the flask. Swirl until mix well and then the purified styrene was used to make up to the scale. The mixtures (stock reagents) were again mixed well and 3 ml. of the stock reagents were introduced into a designed reaction tube (Figure 3-3) using a needle syringe. A Teflon stopcock was introduced to the top of the tube, and then the stopcock was turned off. The tube was subsequently cooled down using the acetone-dried ice mixture for approximately 5 minutes until the stock solution was freezed. The acetone-dry ice mixture was removed, and a vacuum oil pump was applied to vacuate air presented in the stock solution for a few minutes. This process was repeated thrice. The stock solution was then ready to pursue polymerization reaction in a shaking bath.



Copyright by Mahidol University

Figure 3-3 Reaction tube.

3.3.2 The Kinetic Studies of Thermal-initiated Free Radical Polymerization of Styrene in the Presence of Carbon tetrachloride or 1-Butanethiol Chain Transfer Agent

The polymerization was performed in a shaking water bath with constant temperature at 70 °C. For a limited period, the reaction mixture was removed from the bath and quenched immediately in acetone-dried ice mixture.

In general, a polymerization batch had 6-9 stock solution tubes. The collection periods were every 20 minutes interval. For the polymerization batch N, the collection periods were every 1 hour interval. Table 3-1 shows the quantities of starting reagents used in 14 batches. For Batches A-H and L-M, each had 6 reaction tubes. For Batches I-K, each had 9 reaction tubes. Batch N had 7 reaction tubes.

Reaction mixtures in the reaction tubes for the specific collecting time were removed from the shaking bath and quenched immediately using acetone-dried ice for one minute. Subsequently remove the reaction tubes and stand in dark at room temperature. Then the mixtures were transferred from the reaction tubes to test tubes (8 mm x 8 cm). The mixtures were stored in dark in a freezer prior to further analysis and characterization.

Copyright by Mahidol University

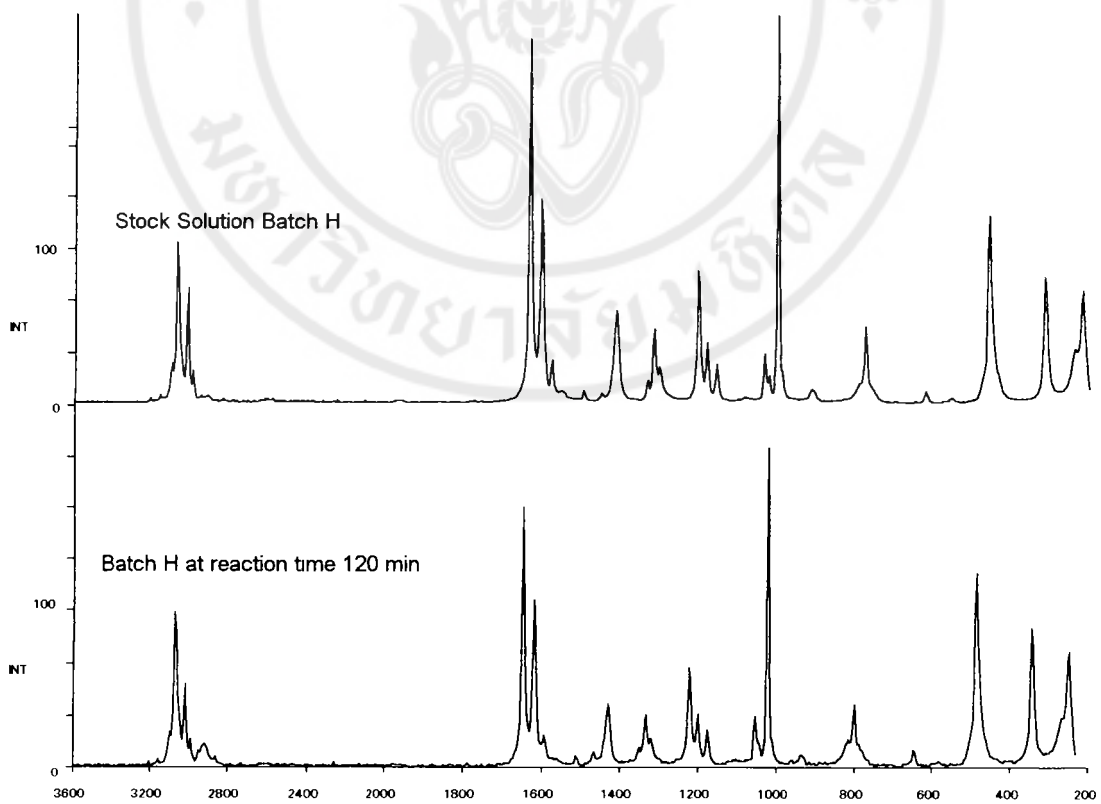
Table 3-1 Amount of substances in each polymerization batch.

Batch ID	Substances	Conc. (mol/l)	Weight (g)	Volume (ml)	%W/W of monomer
A	Styrene	8.70	44.30	49	-
	AIBN	0.12	1.00	-	2.26
B	Styrene	6.96	35.23	39	-
	AIBN	0.12	1.00	-	2 84
	Cyclohexane	-	-	10	-
C	Styrene	6.96	35.23	39	-
	AIBN	0.12	1.00	-	2.84
	Cyclohexane	-	-	8	-
	CCl_4	0.40	-	2	0.35
D	Styrene	6.96	35.23	39	-
	AIBN	0.12	1.00	-	2.85
	Cyclohexane	-	-	6	-
	CCl_4	0.81	-	4	1.43
E	Styrene	6.96	35.23	39	-
	AIBN	0.12	1.00	-	2.84
	Cyclohexane	-	-	4	-
	CCl_4	1.22	-	6	3.21
G	Styrene	6.96	35.23	39	-
	AIBN	0.12	1.00	-	2.85
	CCl_4	2.04	-	10	8.91
H	Styrene	1.74	8.05	9	-
	AIBN	0.12	1.00	-	12.45
	CCl_4	8.16	-	40	623.69
I	Styrene	5.22	26.17	29	-
	AIBN	0.12	1.00	-	3.84
	Cyclohexane	-	-	18	-
	1-Butanethiol	0.36	-	2	0.25
J	Styrene	5.22	26.17	29	-
	AIBN	0.12	1.00	-	3.83
	Cyclohexane	-	-	16	-
	1-Butanethiol	0.72	-	4	0.99
K	Styrene	5.22	26.17	29	-
	AIBN	0.12	1.00	-	3.82
	Cyclohexane	-	-	14	-
	1-Butanethiol	1.08	-	6	2.24
L	Styrene	5.22	26.17	29	-
	AIBN	0.12	1.00	-	3.82
	Cyclohexane	-	-	12	-
	1-Butanethiol	1.44	-	8	3.98
M	Styrene	5.22	26.17	29	-
	AIBN	0.12	1.00	-	3.82
	Cyclohexane	-	-	10	-
	1-Butanethiol	1.80	-	10	6.25
N	Styrene	5.22	26.17	29	-
	AIBN	0.12	1.00	-	3.84
	Cyclohexane	-	-	18	-
	1-Butanethiol	0.36	-	2	0.25

Copyright by Mahidol University

3.3.3 Determination of styrene, CCl_4, and 1-Butanethiol Conversions using Raman Spectroscopic Technique

Approximately 1 ml of the polymerized samples were put into a 4-ml glass test tube, and then the tube was introduced into the sample holder of a fourier transform Raman spectrometer. Then the spectrometry was performed at room temperature. Typical scan ranged from 10- 20 times. Typical Raman spectra (wave number 4000-200 cm^{-1}) of the stock solution and of the polymerized samples when CCl_4 and 1-BuSH were the chain transfer agents are shown in Figures 3-4, and 3-5, respectively.



Copyright by Mahidol University

Figure 3-4 Typical Raman spectrum (4000 - 200 cm^{-1}) of the stock solution and of the polymerized samples when CCl_4 was the chain transfer agents.

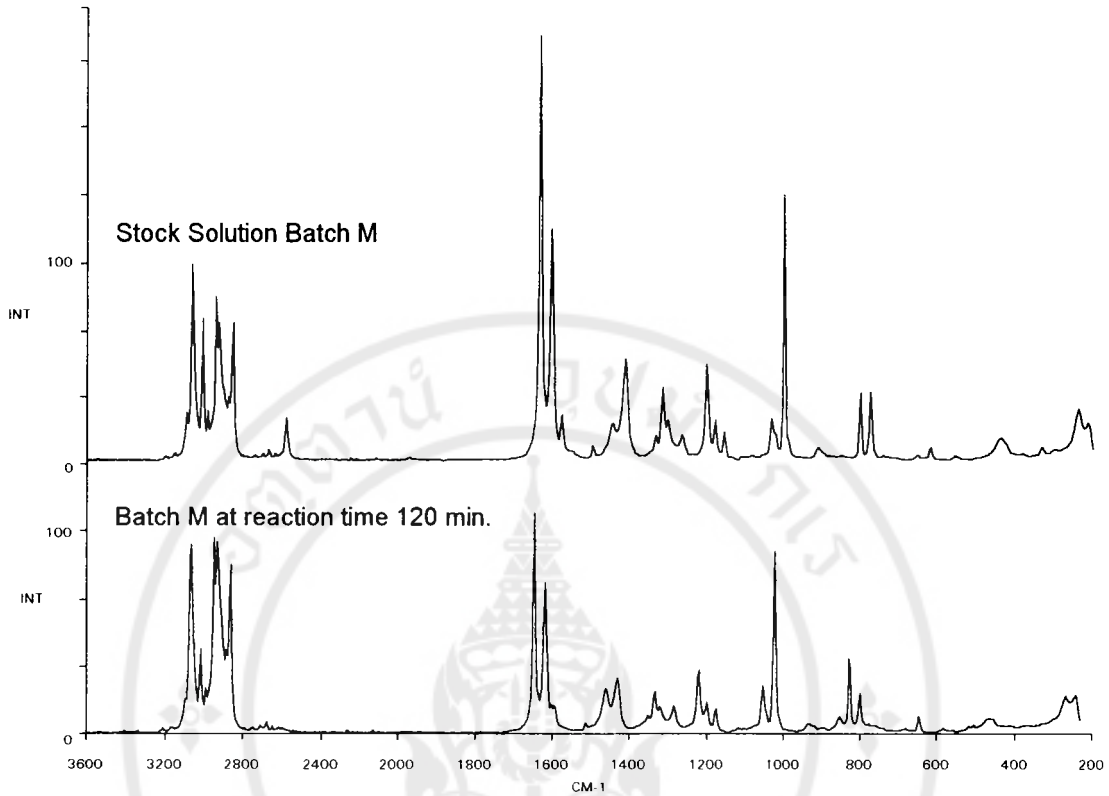


Figure 3-5 Typical Raman spectrum (4000 - 200 cm^{-1}) of the stock solution and of the polymerized samples when 1-BuSH was the chain transfer agents.

Patrick et al (1991) found that intensity of Raman spectrum was linear proportional to concentration of the sample. Therefore, concentration of the sample can be obtained by those method.

A) Calculation of Styrene Conversion

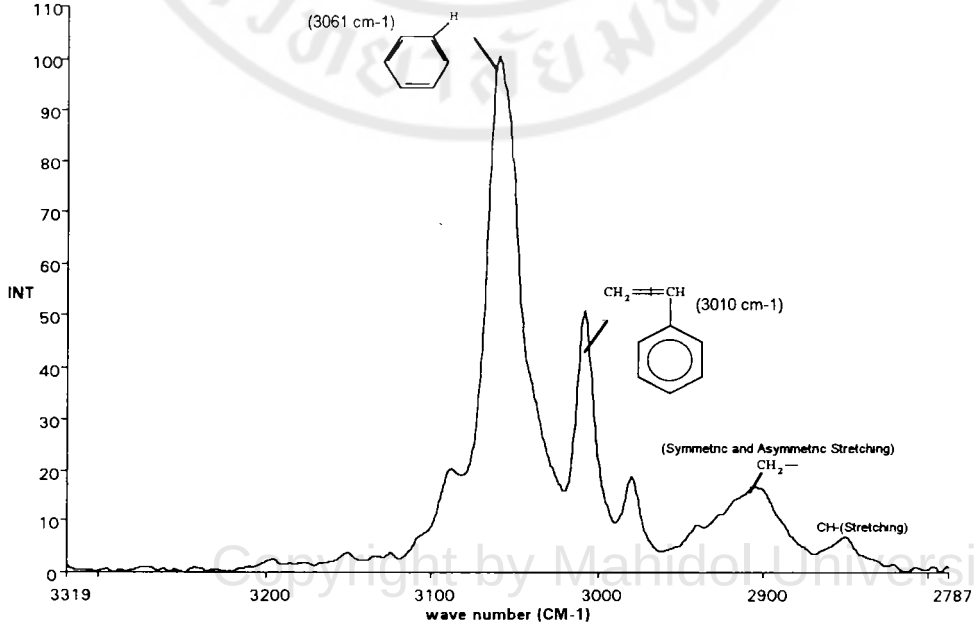
At the 1631 cm^{-1} wave number (ν) is the characteristic band of C=C stretching (ν_S(C=C)) of the vinyl group of styrene which is shown in Figure 3-6(B). The band of ν_S(C=C) was interfered by neighboring bands causing the obtained absolute value of band intensity of ν_S(C=C). The

Copyright by Mahidol University

neighboring bands are at ν = 1617.5 cm^{-1} (possible one of ring stretching modes), ν = 1601 cm^{-1} (symmetric ring stretching), ν = 1575 cm^{-1} (one of ring stretching modes), and ν = 1582 cm^{-1} (overtone of the symmetric out-of-plane bending of benzene ring). Therefore, it is necessary to fit the spectrum (from ν = 1700 cm^{-1} to 1500 cm^{-1}) using Spectra cal® curve-fitting software packages for accurate measurement of ν_S(C=C). Figure 3-6(C) shows typical curve-fitted spectrum. The band at ν = 3061 cm^{-1} was used as internal reference by which this band was always normalized to 100 %. Then the band height at 1631 cm^{-1} was measured and used for calculation of the percentage of styrene conversion using Equation (3-1).

$$\text{Styrene conversion (\%)} = 100.\frac{H_0 - H_t}{H_0} \qquad (3\text{-}1)$$

where H_0 and H_t are peak heights of ν_S(C=C) at the initial time and at the reaction time t.



(A)

Copyright by Mahidol University

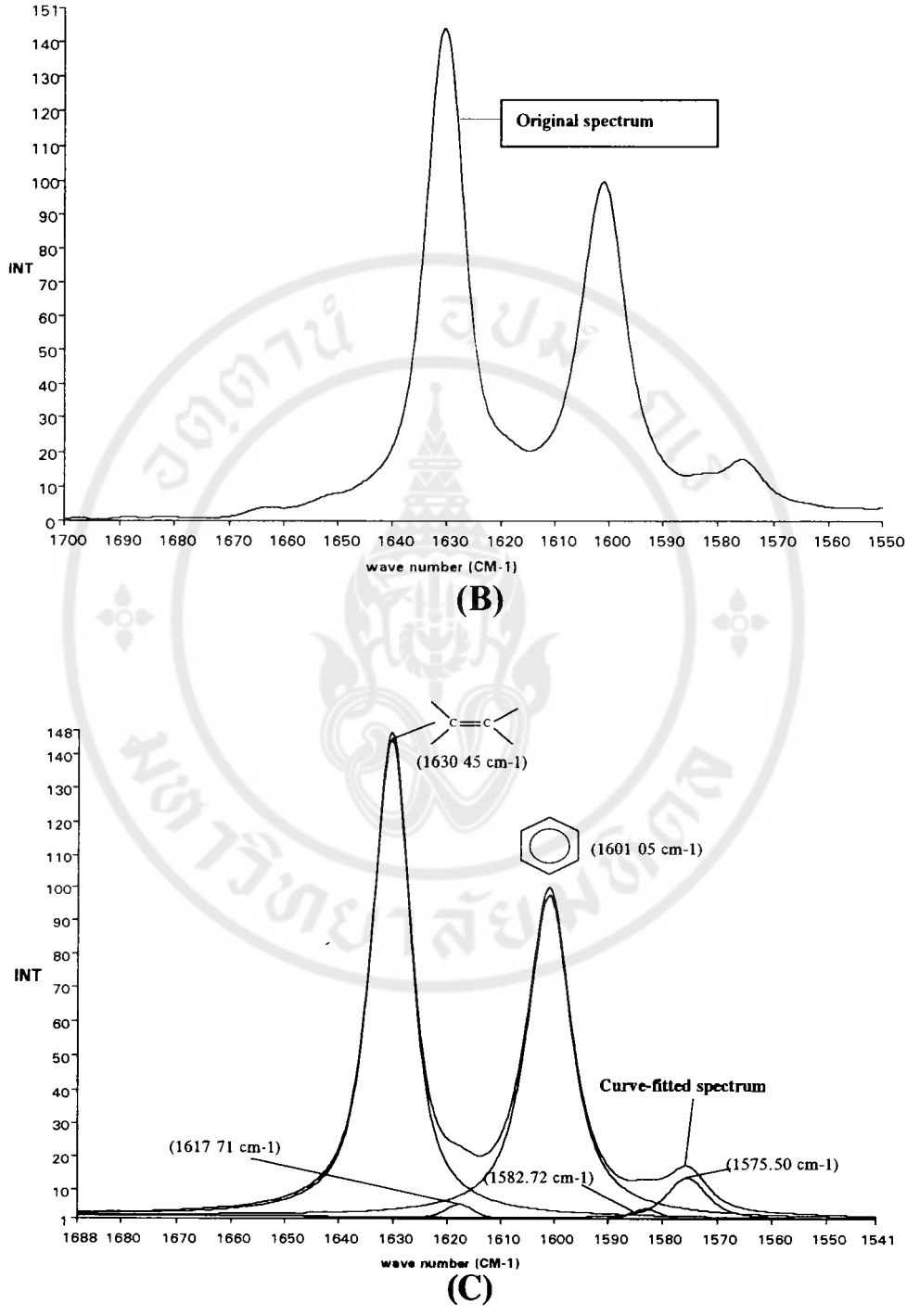


Figure 3-6 Typical Raman spectra of the polymerized samples : (A) Internal standard band, (B) Original spectrum, (C) the curve-fitted spectrum.

Copyright© by Mahidol University

B) Calculation of 1-butanethiol conversion

The method for calculation of 1-butanethiol conversion is similar to that of styrene conversion except for the band area at $\nu \approx 2577$ cm^{-1} (ν_S(-SH)) was used for the evaluation. It is noted that ν_S(-SH) is singlet and not disturbed by any neighboring bands. Therefore, it was unnecessary to perform curve fitting. The band was also compared to that at $\nu = 3061$ cm^{-1}, which was always normalized to 100 %, prior to the calculation using Equation (3-2).

$$\text{1-Butanethiol conversion (\%)} = 100.\frac{A_0 - A_t}{A_0} \qquad (3\text{-}2)$$

where A_0 and A_t are the band areas of ν_S(-SH) at the initial time and at the reaction time t.

C) Calculation of carbon tetrachloride conversion

In the case that carbon tetrachloride was employed as CTA, increase in styrene consumption decreases the volume of reaction mixture, resulting in abnormal changes in the intensity of C-Cl deformation band (γ(C-Cl)) at $\approx$ 314 cm^{-1}. Figure 3-8 shows that at the reaction time 120 minutes the band height of γ(C-Cl) is higher than that at the initial time. Therefore, it is essential to account the effect of volume changes in the calculation. The percentage of CCl_4 conversion was calculated using Equation (3-3).

Copyright by Mahidol University

LIBRARY AND INFORMATION CENTER
MAHIDOL UNIVERSITY

$$CCl_4 \text{ conversion } (\%) = 100.\frac{H_0 - H_t.\frac{V_t}{V_0}}{H_0} \qquad (3\text{-}3)$$

where H_0 and H_t are the band heights of γ(C-Cl) at the initial time and at reaction time t. V_0 and V_t are the total volume of reaction mixture at the initial time and at reaction time t (for the detail of derivation of V_t refers to Appendix IV).

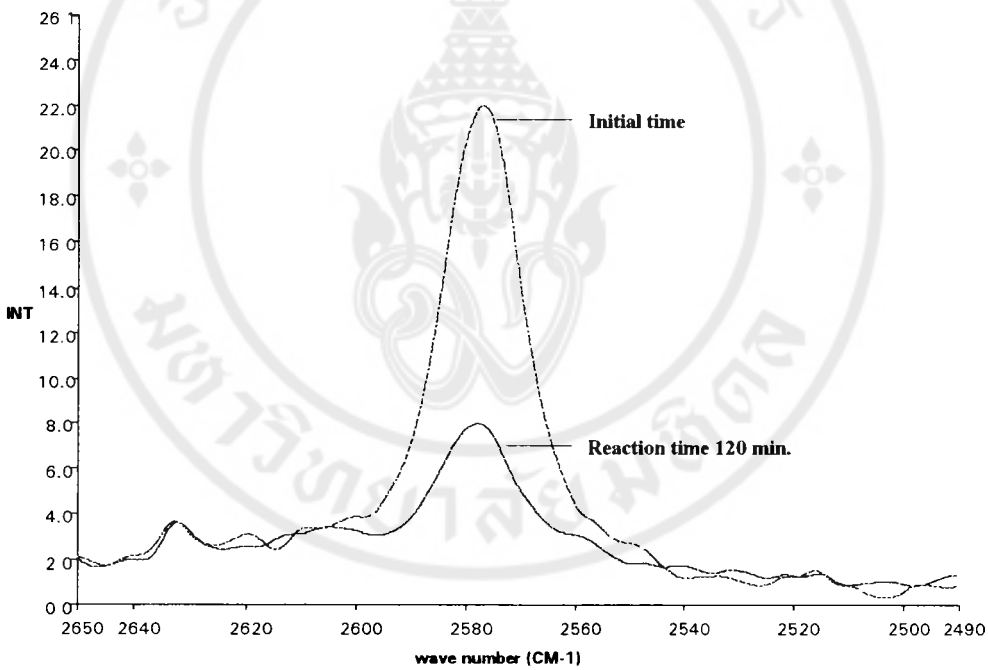


Figure 3-7 Typical Raman spectrum (2650 - 2490 cm^{-1}) of Batches I-M, showing ν_S(-SH) $\approx$ 2577 cm^{-1}

Copyright by Mahidol University

27350

1102682 48

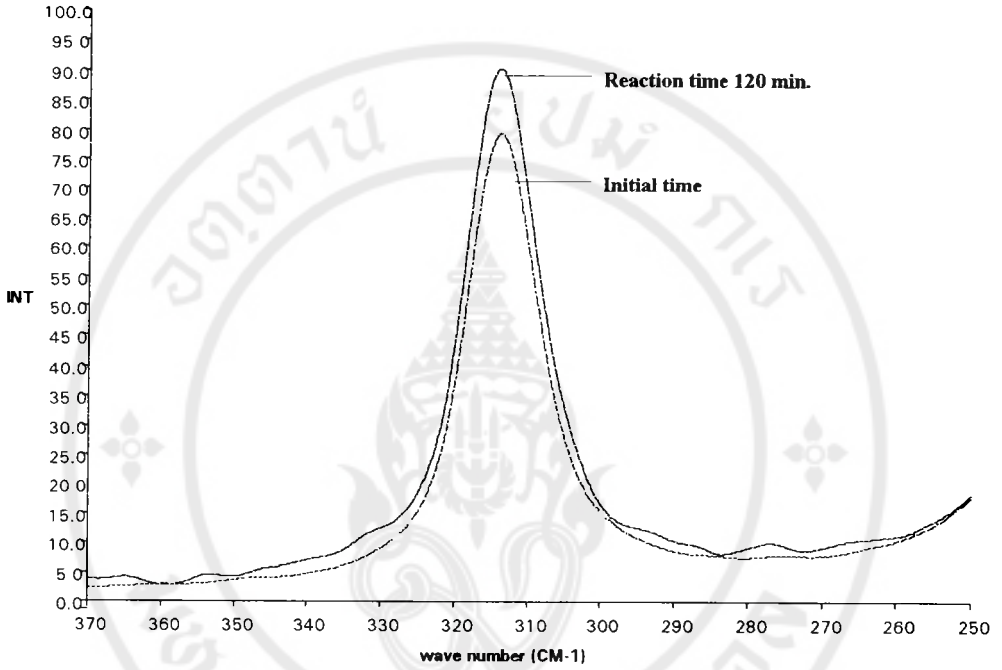


Figure 3-8 Typical Raman spectrum (370 - 250 cm^{-1}) of Batches C-H, showing γ(C-Cl) $\approx$ 314 cm^{-1}.

Copyright by Mahidol University

3.4 Determination of molecular weight and molecular weight distribution of polymers synthesized using Gel Permeation Chromatographic (GPC) Technique

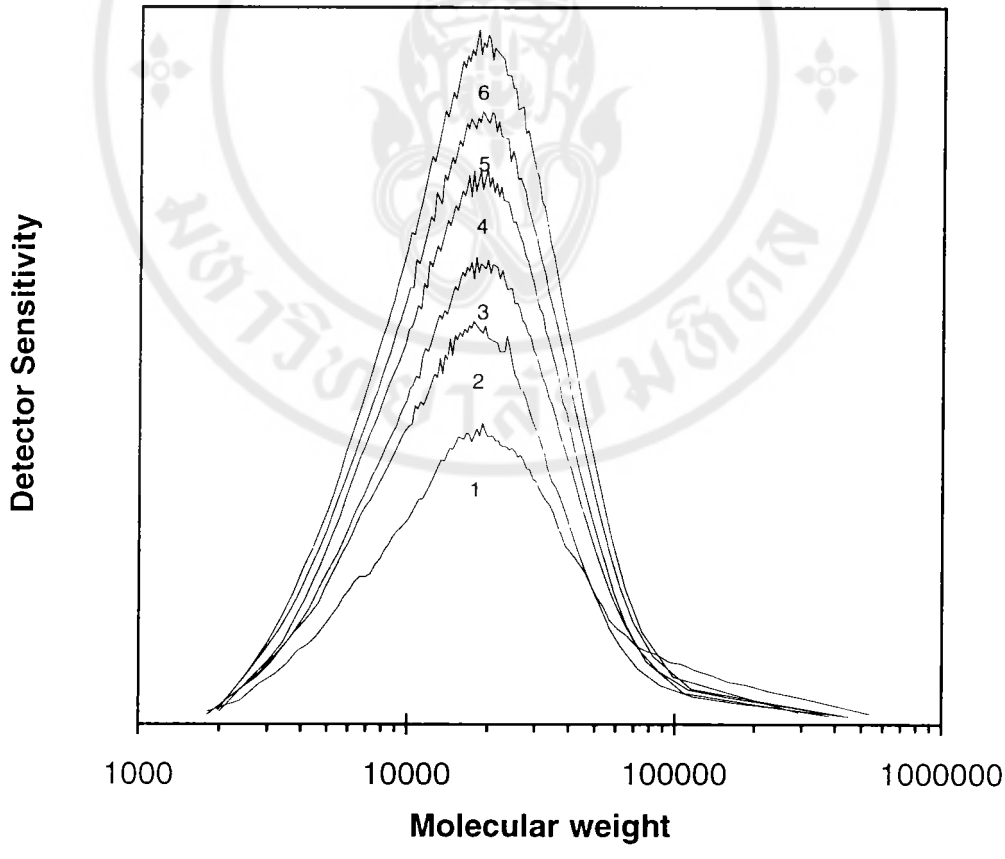
The modified GPC technique of Beoit et al (1967), Schröder and Karl-Friedrich Arndt (1988) was employed in this study; only brief description of the technique is given. The polymerized sample (0.5 ml) was dissolved in HPLC grade tetrahydrofuran (THF, 10 ml). Then the sample solution was filtered through nylon membrane (0.22 μm pore size), and the filtered sample solution (100 μl) was introduced to the automatic injection GPC (model 150CV Waters). THF was used as eluent; the flow rate was 1 ml/min. Typical GPC chromatographs of the polymerized samples are shown in Figures 3-9 and 3-10.

MW of synthesized polymers was measured relative to MW of standard polystyrene using graphical method, i.e., the calculation was based on the stansard polymer carlibration curve. The calibration curve of standard polystyrene, having MW range from $5.1x10^2$ to $3.12x10^6$ g/mol. with very narrow polydispersity index of less than 1.1, was established by ploting of MW against elution time or retention volume. Various types of average-molecular weights including M_p, M_n, M_v, M_w, M_z and molecular weight distribution were calculated using GPC multidetector software packages, supplied together with the instrument.

Copyright by Mahidol University

3.5 Computer programming

Many computer programmes were written in Microsoft© QuickBASIC (Version 4.5) for solving the derived non-linear algebraic and simultaneous differential equations. In particular the fourth order Runge-Kutta and Newton-Raphson numerical integration algorithms were written to solve the differential equations. The incremental time, which affects on the numerical solutions, could be varied.



Copyright by Mahidol University

Figure 3-9 GPC chromatogram of Batch C at reaction time. (1) 20, (2) 40, (3) 60, (4) 80, (5) 100, and (6) 120 min, respectively.

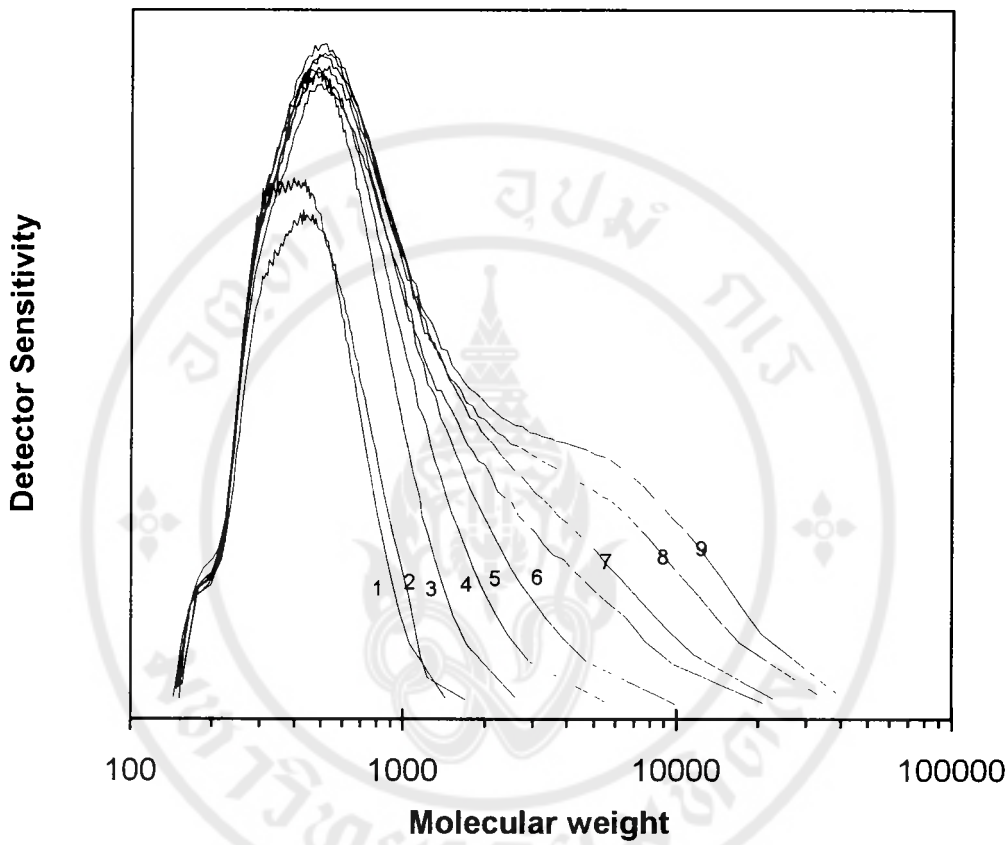


Figure 3-10 GPC chromatogram of Batch I at reaction time. (1) 20, (2) 40, (3) 60, (4) 80, (5) 100, (6) 120, (7) 140, (8) 160, and (9) 180 min, respectively.

Copyright by Mahidol University

The smaller the incremental time is, the more accurate solutions should be obtained. The fourth order Runge-Kutta algorithms were used for determination of concentrations of monomer, chain transfer agent, and various types of dead-end polymer at reaction time t. The algorithms of Newton-Raphson numerical technique were written for estimation of the concentration of propagating radicals. In fact the fourth-order Runge-Kutta algorithms could also be used to evaluate the propagating radical concentrations; however, Newton-Raphson numerical technique was employed in this case because of its simplicity in comparison to the Runge-Kutta method.

The computer software written in this work contains three-main routines. The first routine is for calculations of monomer and CTA conversions at the specific reaction time as well as MW and MWD of polymers. The second one is written to evaluate C_X at the specific monomer conversion. These two routines provide users to print out the calculated values and to show graphic displays. Printers' setting programmes are wriiten in the third routine. Programming flow charts of the first and the second routines is shown in Figures 3-11 and 3-12, respectively.

Programming limitations are the followings.

(a) Long computational time was used for the calculation of the exact solution of the simultaneous differential equations because the numerical integration algorithms are the iterative programming.

Copyright by Mahidol University

(b) The assumption was made on that the rate coefficients for the propagation, termination, and chain transfer reactions are independent on the kinetics chain length of the growing active radicals. This assumption is not true for the polymerisation reactions leading to low molecular weight polymers.

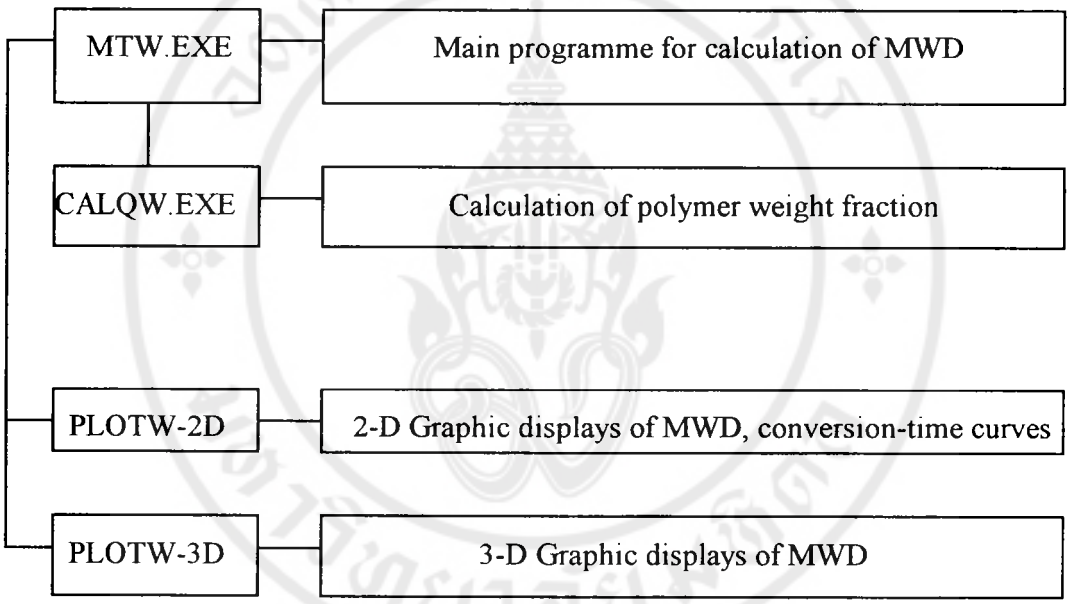


Figure 3-11 Programming flow chart of the first routine.

Copyright by Mahidol University

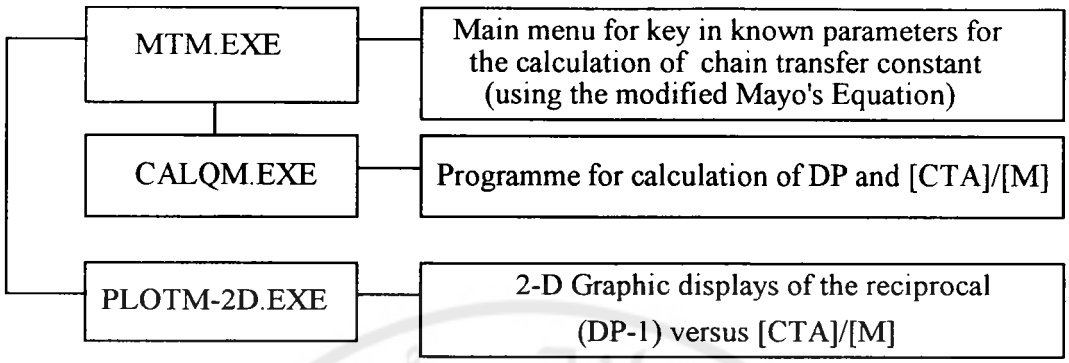


Figure 3-12 Programming flow chart of the second routine.

3.6 Computer simulation experiment

Values of rate coefficients appeared in the kinetic mechanism proposed in this study are listed in Table 3-2. These values except for those of k_p and k_{tr} are used for the computer simulation experiments.

Copyright by Mahidol University

Table 3-2 Values of rate coefficients for computer simulation experiments.

Substances	Rate coefficient for	Symbol	Value at 70 °C	Literature
Styrene	Propagation	k_p	240 $l.mol^{-1}s^{-1}$	Villermaux et al (1983)
	Termination by disproportionation	k_{td}	1.0×10^8 $l.mol^{-1}s^{-1}$	Model value
	Termination by combination	k_{tc}	1.0×10^7 $l.mol^{-1}s^{-1}$	Hamielec and Hodgins (1967), Brandrup and Immergut (1975 and 1989), Cutter and Drexler (1982), Herman et al (1985)
	Chain transfer reaction to monomer	k_{trm}	0.152 $l.mol^{-1}s^{-1}$	Hamielec and Hodgins (1967), Cutter and Drexler (1982), Villermaux et al (1983)
Carbon tetrachloride	Chain transfer reaction to CTA	k_{tr}	2.43 $l.mol^{-1}s^{-1}$	Brandrup and Immergut (1975 and 1989)
	The reaction of R* with CTA	k_{sr}	$\approx k_i$ or k_{tr}	Model value
Butanethiol	Chain transfer reaction to CTA	k_{tr}	3600 $l.mol^{-1}s^{-1}$	Brandrup and Immergut (1975)
	The reaction of R* with CTA	k_{sr}	$\approx k_i$ or k_{tr}	Model value
AIBN	Initiator decomposition	k_d	3.26×10^{-5} s^{-1}	Brandrup and Immergut (1975 and 1989), Villermaux et al (1983), Herman et al (1985)
	Initiation	k_i	$\approx 2fk_d$	Brandrup and Immergut (1975)
	Initiator efficiency	f	0.60	Herman et al (1985)

Copyright by Mahidol University

Chapter IV
Results and Discussion

4.1 The Thermal-Initiated Free Radically Bulk and Solution Polymerization Reactions of Styrene at 70 °C.

The free radically bulk polymerization of styrene using AIBN thermal labile initiator at 70 °C (Batch A) gives the styrene conversion much higher than the solution polymerization (Batch B; cyclohexane was the solvent) as shown in Figure 4-1. The overall rate of free radical polymerization is equal to the rate of propagation reaction (R_p), which can be determined from Equation (4-1).

$$R_p = -\frac{d[M]}{dt} = [M]_0\frac{dCon}{dt} \qquad (4\text{-}1)$$

where Con is abbreviated for the styrene conversion, and

$$Con = \frac{[M]_0 - [M]_t}{[M]_0} \qquad (4\text{-}2).$$

$[M]$ and $[M]_0$ are the concentrations of styrene at the reaction time t and at the initial time, respectively.

The initial R_p ($R_{p,0}$) was determined to have the value of 6.17×10^{-4} mol/l.s for the bulk polymerization, while that for the solution polymerization was of 2.83×10^{-4} mol/l.s.

Copyright by Mahidol University

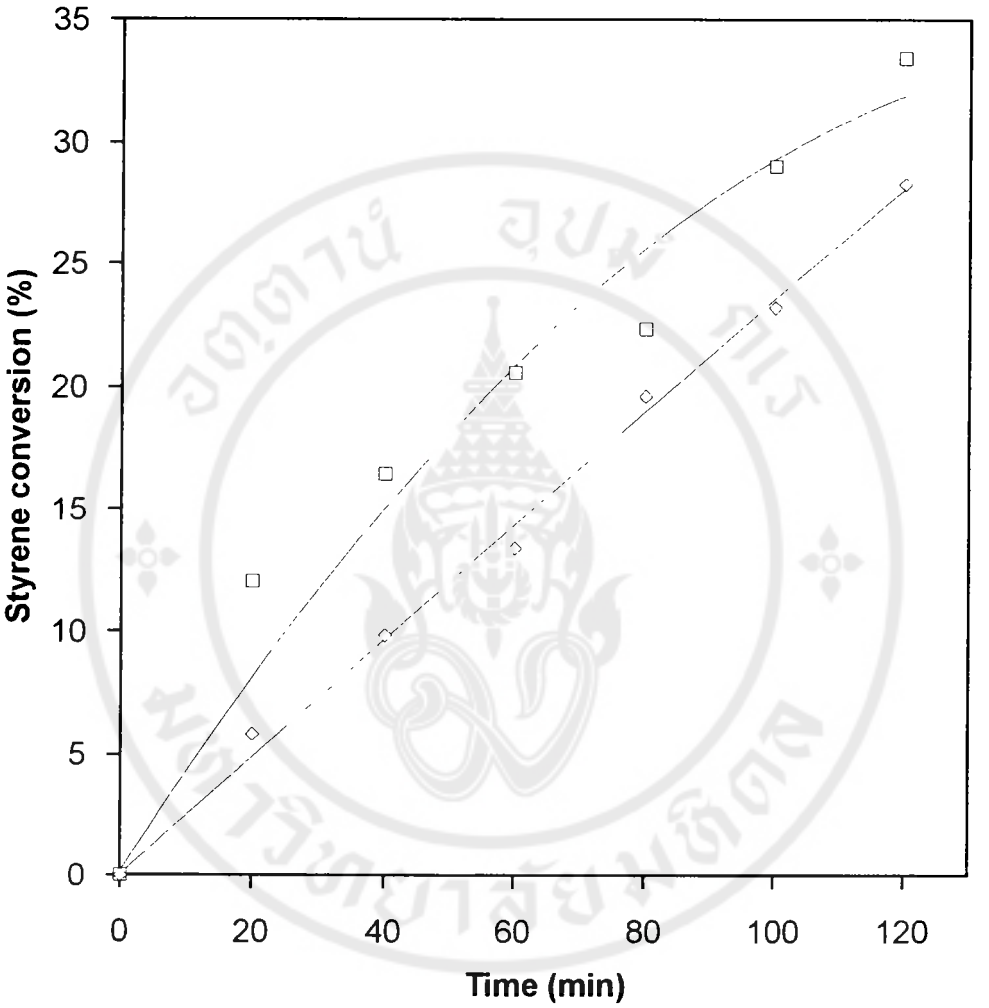


Figure 4-1 The plot of styrene conversion against time (Experiment: (□) Batch A and (◊) Batch B).

Copyright by Mahidol University

The initial overall rate coefficient ($k_{p,0}$) for the free radical polymerization can be calculated from Equation (4-3) to be 340.36 l/mol.s and 195.07 l/mol.s for the bulk and solution polymerization, respectively.

$$R_{p,0} = k_{p,0}[ST]_0[AIBN]_0^{0.5} \quad (4\text{-}3)$$

where $k_{p,0}$ is the initial overall rate coefficient for the thermal-initiated free radical polymerization reaction. $[ST]_0$ and $[AIBN]_0$ is the initial concentrations of styrene (ST) monomer and 2,2'-azo-bis-(2-methylpropionitrile) (AIBN) initiator, respectively.

In the free radically bulk polymerization of styrene, the Trommsdorff effect has been reported occurring when the polymerization time was approximately 18 hours. Therefore, it is unnecessary to account for the effect in this study.

Copyright by Mahidol University

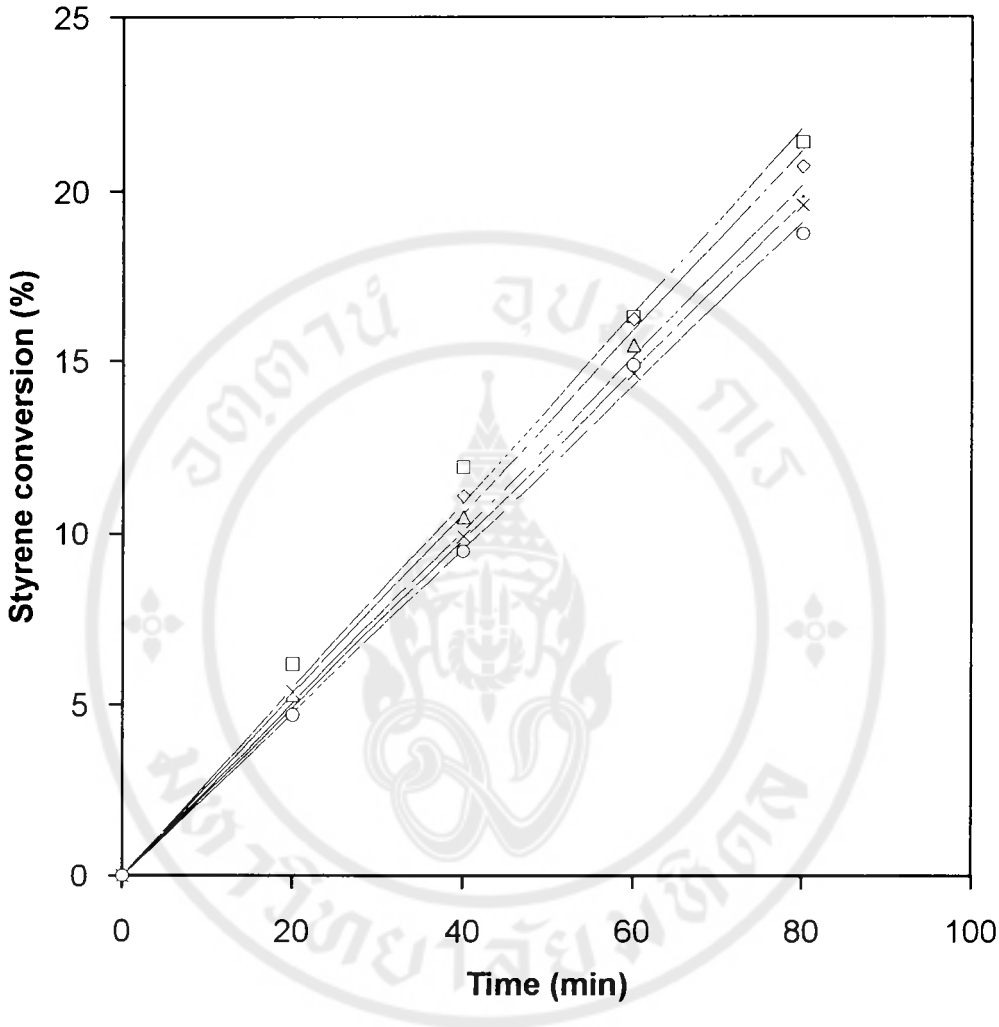


Figure 4-2 The plot of styrene conversion against time (Experiment: (□) Batch C, (◊) Batch D, (Δ) Batch E, (x) Batch G, and (o) Batch H).

Copyright by Mahidol University

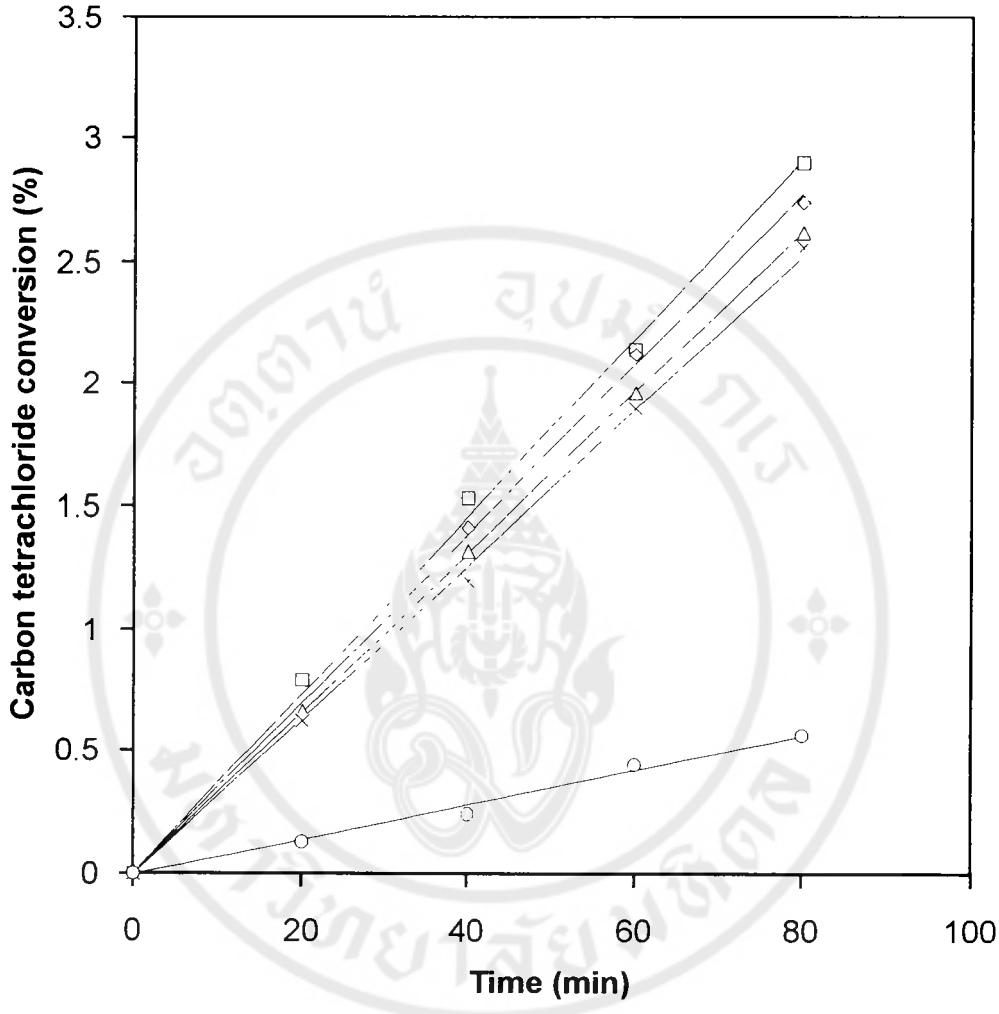


Figure 4-3 The plot of CCl_4 conversion against time (Experiment: (□) Batch C, (◊) Batch D, (Δ) Batch E, (x) Batch G, and (o) Batch H).

Copyright by Mahidol University

4.2 The Thermal-Initiated Free Radically Solution Polymerization of Styrene in the Presence of a Chain Transfer Agent at 70 °C

4.2.1 Carbon tetrachloride (CCl_4) as the chain transfer agent

The plots of percentages of styrene and CCl_4 conversions against reaction time of the thermal labile solution free radical polymerization of styrene using AIBN initiator in the presence of various amount of CCl_4 chain transfer agent (Batches C-H) at 70 °C are shown in Figures 4-2 and 4-3, respectively. The polymerization Batch C had the smallest amount of CCl_4 and Batch H had the largest amount. The larger amount of CCl_4 resulted in lesser styrene and CCl_4 conversions at the same reaction time.

Euapermkiati (1990) reported that the overall rate of polymerization of styrene (R_{ST}) using AIBN initiator and CCl_4 chain transfer agent had the relation to the concentrations of those substances, i.e.,

$$R_{ST} \quad \alpha[\mathrm{AIBN}]^{0.4}[\mathrm{ST}]^{0.7}[\mathrm{CCl_4}]^{-0.1} \qquad (4\text{-}4).$$

Starks (1974) reported that the rate of polymerization of ethylene (R_{et}) using benzoyl peroxide initiator and CCl_4 chain transfer agent was also dependent upon the concentrations of these substances, i.e.,

$$R_{et} \quad \alpha[(C_2H_5CO_2)_2]^{0.5}[C_2H_4]^{1}[CCl_4]^{1} \qquad (4\text{-}5).$$

Copyright by Mahidol University

Therefore, it can be concluded that the overall rate of polymerization should be dependent upon the concentrations of monomer, initiator and

CTA, which differs from Equation (4-3). In this study, the overall rate of styrene polymerization ($R_{ST,CCl4}$) in the presence of CCl_4 chain transfer agent is defined in Equation (4-6).

$$R_{ST,CCl_4} = k_{ST,CCl_4}[ST]^p[CCl_4]^q[AIBN]^r \qquad (4\text{-}6)$$

where p, q and r are the individual exponent order for styrene, CCl_4, and AIBN, respectively. The total reaction order with respect to $R_{ST,CCl4}$ is equal to p+q+r. k_{ST} is the overall rate constant for the polymerization reaction.

The initial $R_{ST,CCl4}$ ($R_{ST,CCl4,0}$) were determined using Equation (4-1); the values for the polymerization Batches C-H are shown in Table 4-1. The values ranges from $0.69x10^{-4}$-$3.16x10^{-4}$ mol/l.s. In the Batches C-G, the concentrations of styrene and AIBN were almost the same, therefore the difference of $[CCl_4]_0$ were the only factor governed on the different values of $R_{ST,CCl4,0}$. If the order q is equal to 1 so that the values of $\frac{R_{ST,CCl4,0}}{[CTA]_0}$ in each batch should be equal to one another. However, Table 4-1 shows that the calculated values are different from one another, indicating firmly that q is not equal to 1.

Since the experimental data on AIBN conversions at the particular reaction time could not be obtained in this work, Equation (4-6) can therefore be rewritten as :-

Copyright by Mahidol University

Table 4-1 Values of $R_{ST,CTA,0}$, $\frac{R_{ST,CTA,0}}{[CTA]_0}$, $k'_{ST,CTA,0}$, $R_{CTA,0}$, $\frac{R_{CTA,0}}{[CTA]_0}$, and $k'_{CTA,0}$ for the thermal-initiated free radical solution polymerization of styrene using AIBN initiator at 70 °C (Batches A and B are in the absence of CTA. Batches C-H, CCl_4 is CTA. Batches I-M, 1-BuSH is CTA).

Batch	$R_{ST,CTA,0}$ x10^4 mol/l.s	$\frac{R_{ST,CTA,0}}{[CTA]_0}$ x10^4 s^{-1}	$k'_{ST,CTA,0}$ x10^5 l/mol.s	$R_{CTA,0}$ x10^6 mol/l.s	$\frac{R_{CTA,0}}{[CTA]_0}$ x$10^6 s^{-1}$	$k'_{CTA,0}$ x10^7 l/mol.s
A	6.2	-	2.03	-	-	-
B	2.8	-	1.16	-	-	-
C	3.16	7.75	4.73	2.48	6.07	8.24
D	3.07	3.76	4.82	4.72	5.78	8.09
E	2.93	2.39	4.73	6.70	5.47	8.17
G	2.85	1.40	4.78	10.8	5.27	8.15
H	0.69	0.08	4.76	9.49	1.16	8.15
I	3.0	8.3	-	-	-	-
J	4.2	5.7	8.83	317	438	433
K	5.3	4.9	10 3	397	366	477
L	5.6	3.9	10.2	485	336	526
M	5.7	3.1	9.64	607	336	615

Copyright by Mahidol University

$$R_{ST,CCl_4} = k'_{ST,CCl_4}[ST]^p[CCl_4]^q \quad (4\text{-}7)$$

where

$$k'_{ST,CCl_4} = k_{ST,CCl_4}[AIBN]^r \quad (4\text{-}8)$$

The exponent orders p and q were determined equal to 0.95±0.01 and -0.07±0.04. The method used for evaluation of the exponent orders is described in detail in Appendix II. Substitution of p and q into Equation (4-7) to give

$$R_{ST,CCl_4} = k'_{ST,CCl_4}[ST]^{0.95\pm0.01}[CCl_4]^{-0.07\pm0.04} \quad (4\text{-}9)$$

k'_{ST,CCl_4} at 0.0, 2.5 and 5.0 % styrene conversions of the polymerization Batches C-H were calculated from Equation (4-9), and the values are shown in Table 4-2. The average k'_{ST,CCl_4} is $4.88\times10^{-5} \pm 4.18\times10^{-7}$ l/mol.s.

The exponent order r has been found to be equal to 0.5 [7,20,52] and if this value is employed, k_{ST,CCl_4} can be evaluated from Equation (4-8) equal to $1.41\times10^{-4} \pm 1.2\times10^{-6}$ l/mol.s.

The overall rate of loss of CCl_4 (R_{CCl_4}) for the thermal-initiated free radically solution polymerization of styrene using AIBN initiator and CCl_4 chain transfer agent can be expressed in the similar manner as for R_{ST,CCl_4}, i.e.,

Copyright by Mahidol University

$$R_{CCl_4} = k_{CCl_4}[ST]^s[CCl_4]^t[AIBN]^u \quad (4\text{-}10)$$

Table 4-2 $R_{ST,CTA}$ and $k'_{ST,CTA}$ at 0.0, 2.5 and 5.0 % styrene conversions.

Batch	0.0% Conversion		2.5% Conversion		5.0% Conversion	
	$R_{ST,CTA}$ x10^4 mol/l.s	$k'_{ST,CTA}$ x10^5 l/mol.s	$R_{ST,CTA}$ x10^4 mol/l.s	$k'_{ST,CTA}$ x10^5 l/mol.s	$R_{ST,CTA}$ x10^4 mol/l.s	$k'_{ST,CTA}$ x10^5 l/mol.s
CTA is CCl_4						
C	3.16	4.73	3.16	4.84	3.16	4.96
D	3.07	4.82	3.07	4.94	3.07	5.06
E	2.93	4.73	2.93	4.84	2.93	4.96
G	2.85	4.78	2.85	4.89	2.85	5.01
H	0.69	4.76	0.69	4.88	0.69	5.00
Average	2.54	4.77	2.54	4.88	2.54	5.00
Average $k'_{ST,CCl4} = 4.88x10^{-5} \pm 4.18x10^{-7}$ l/mol.s						
CTA is 1-BuSH						
J	4.15	8.83	3.99	9.30	3.80	9.41
K	5.31	10.3	5.16	10.80	4.99	10.94
L	5.63	10.2	5.49	10.70	5.33	10.86
M	5.65	9.64	5.50	10.14	5.34	10.26
Average	5.19	9.72	5.04	10.23	4.86	10.37
Average $k'_{ST,1\text{-}BuSH} = 1.01x10^{-4} \pm 6.81x10^{-6}$ l/mol.s						

Copyright by Mahidol University

where s, t and u are the individual exponent order for styrene, CCl_4, and AIBN, respectively. The total reaction order with respect to R_{CCl4} is equal to s+t+u.

The values of initial overall rate of loss of CCl_4 ($R_{CCl4,0}$) of the polymerization Batches C-H were calculated using the same method as for the calculation of $R_{ST,CCl4,0}$; the results are also shown in Table 4-1.

Rewritten Equation (4-10) gives

$$R_{CCl_4} = k'_{CCl_4}[ST]^s[CCl_4]^t \qquad (4\text{-}11)$$

where

$$k'_{CCl_4} = k_{CCl_4}[AIBN]^u \qquad (4\text{-}12)$$

Using the same method to determine p and q, the exponent order s and t were evaluated equal to 0.99±0.07 and 0.91±0.03, respectively. Equation (4-11) is therefore rewritten to give

$$R_{CCl_4} = k'_{CCl_4}[ST]^{0.99\pm0.07}[CCl_4]^{0.91\pm0.03} \qquad (4\text{-}13)$$

k'_{CCl4} at 0.0, 2.5 and 5.0 % styrene conversions of the polymerization Batches C-H were calculated from Equation (4-13), and the values are shown in Table 4-3. The average k'_{CCl4} is $8.40\times10^{-7} \pm 6.28\times10^{-9}$ l/mol.s.

Copyright by Mahidol University

Table 4-3 Values of R_{CTA} and k'_{CTA} at 0.0, 2.5 and 5.0 % styrene conversions.

Batch	0.0% Conversion		2.5% Conversion		5.0% Conversion	
	R_{CTA} x10^6 mol/l.s	k'_{CTA} x10^7 l/mol.s	R_{CTA} x10^6 mol/l.s	k'_{CTA} x10^7 l/mol.s	R_{CTA} x10^6 mol/l.s	k'_{CTA} x10^7 l/mol.s
CTA is CCl_4						
C	2.48	8.24	2.48	8.33	2.48	8.60
D	4.72	8.09	4.72	8.47	4.72	8.75
E	6.70	8.17	6.70	8.32	6.70	8.60
G	10 8	8.15	10.8	8.42	10.8	8.68
H	9.49	8.15	9.49	8.38	9.49	8.67
Average	6.84	8 15	6.84	8.38	6.84	8.67
Average $k'_{CCl_4} = 8.40x10^{-7} \pm 6.28x10^{-9}$ l/mol.s						
CTA is 1-BuSH						
J	317	433	299	456	278	460
K	397	477	382	502	365	510
L	485	526	465	553	443	559
M	607	610	582	642	556	649
Average	452	511	432	538	411	545
Average $k'_{1\text{-BuSH}} = 5.31x10^{-5} \pm 7.88 x10^{-6}$ l/mol.s						

Copyright by Mahidol University

The value of u was found to be approximately equal to 0.0 for the thermal-labile free radical solution telomerisation of styrene taxogen using CBr_4 telogen and AIBN initiator [52], to 0.5 for that of ethylene taxogen using CCl_4 telogen and benzoyl peroxide initiator [20], and to 1.0 for that of styrene taxogen using CCl_4 taxogen and AIBN initiator [7, 52]. It is likely to assume that the exponent order u is equal to 1.0 so that the average k_{CCl4} can be calculated from Equation (4-12) equal to $2.43 \times 10^{-6} \pm 1.84 \times 10^{-8}$ l/mol.s.

4.2.2. 1-Butylmercaptan as the chain transfer agent

1-Butylmercaptan (1-BuSH) was also employed as the chain transfer agent in the thermal-initiated free radically solution polymerization of styrene at 70 °C using AIBN initiator. The plots of percentages of styrene and 1-BuSH conversions against reaction time (Batch J-M) are respectively shown in Figures 4-4 and 4-5. Batch I had the smallest amount of 1-BuSH, while Batch M had the largest amount. The larger amount of 1-BuSH resulted in lesser 1-BuSH conversion but higher styrene conversion.

The overall rate of polymerization reaction ($R_{ST,1\text{-}BuSH}$) and that of loss of 1-BuSH ($R_{1\text{-}BuSH}$) are defined and shown in Equations (4-14) and (4-15), respectively.

$$R_{ST,1-BuSH} = k_{ST,1-BuSH}[ST]^a[CCl_4]^b[AIBN]^c \qquad (4\text{-}14)$$

Copyright by Mahidol University

$$R_{1-BuSH} = k_{1-BuSH}[ST]^d[CCl_4]^e[AIBN]^f \qquad (4\text{-}15)$$

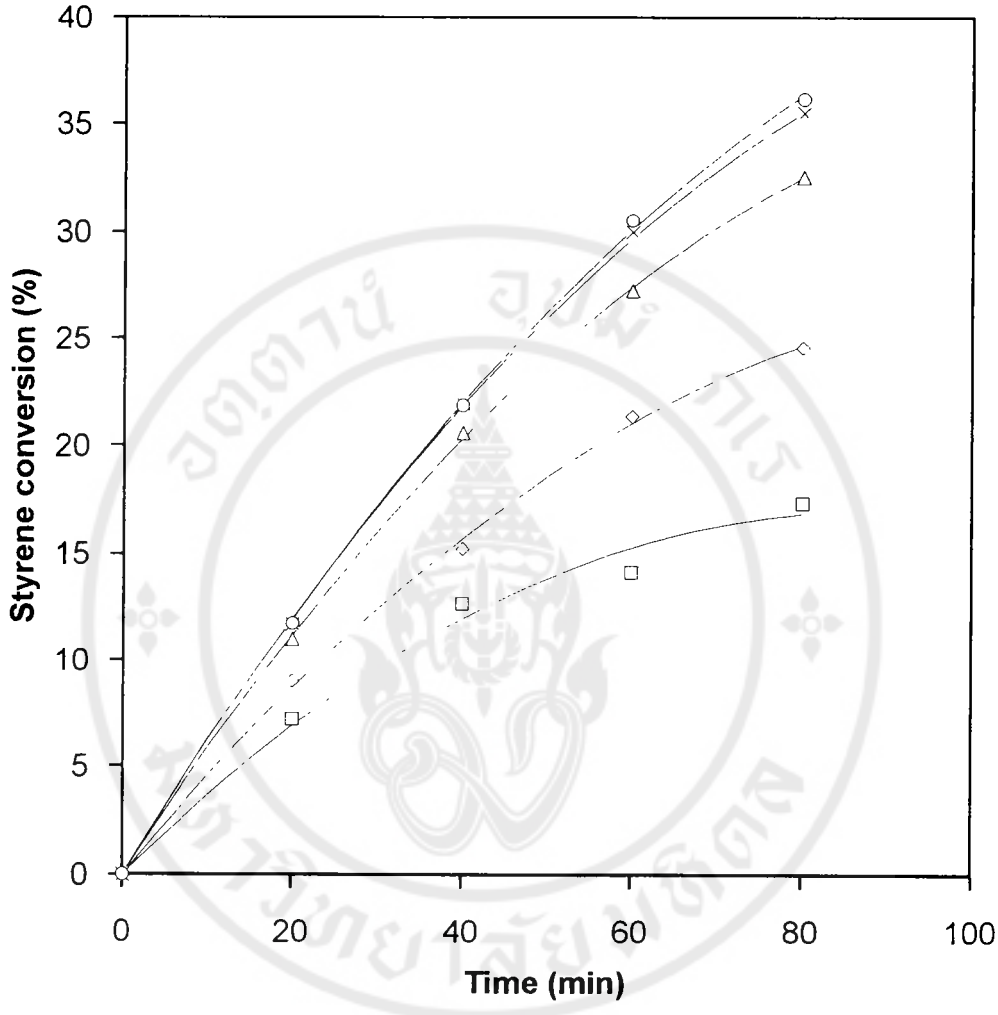


Figure 4-4 The plot of styrene conversion against time (Experiment: (□) Batch I, (◊) Batch J, (Δ) Batch K, (x) Batch L, and (o) Batch M).

Copyright by Mahidol University

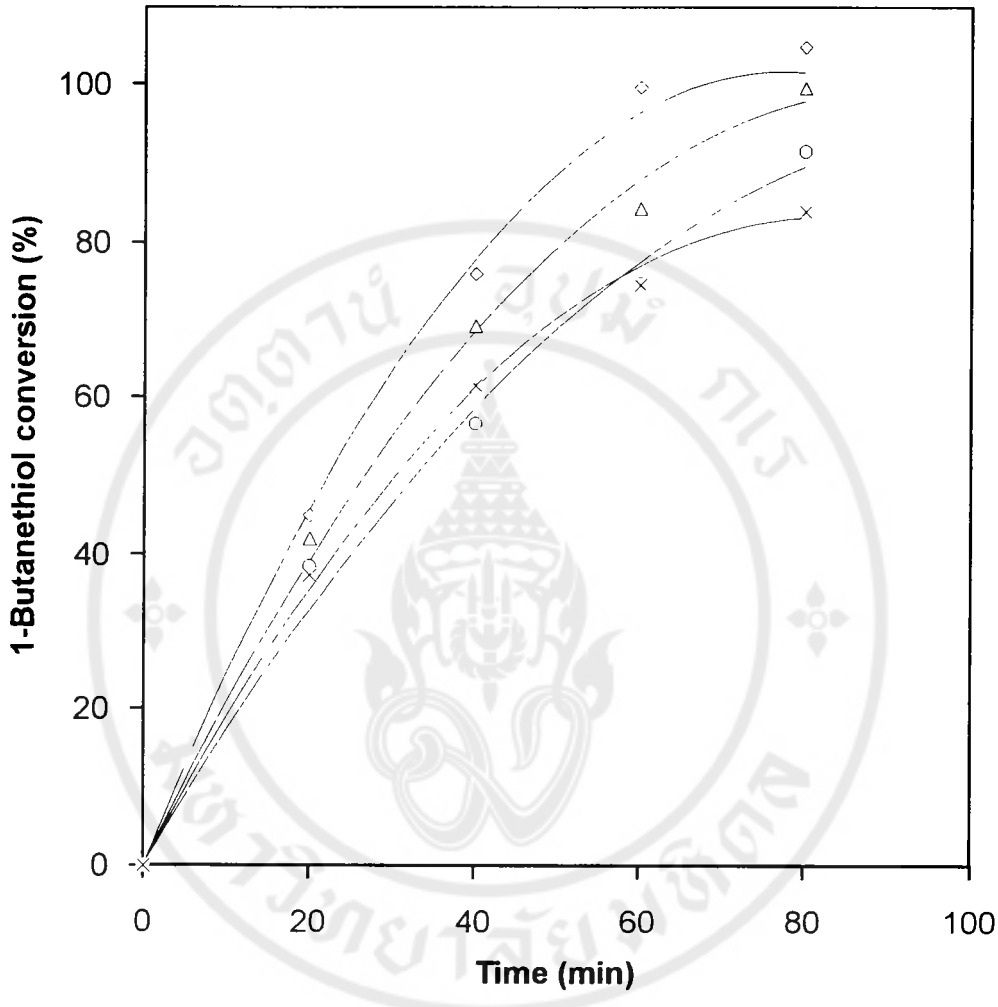


Figure 4-5 The plot of 1-BuSH conversion against time (Experiment: (◊) Batch J, (Δ) Batch K, (x) Batch L, and (o) Batch M).

Copyright by Mahidol University

where a, b and c are the individual exponent for styrene, 1-BuSH and AIBN, respectively, with respective to $R_{ST,1\text{-}BuSH}$. d, e and f are the individual exponent to styrene, 1-BuSH and AIBN, respectively, with respect to $R_{1\text{-}BuSH}$. The total reaction order with respect to $R_{ST,1\text{-}BuSH}$ and to $R_{1\text{-}BuSH}$ is subsequently equal to a+b+c and to d+e+f. $k_{ST,1\text{-}BuSH}$ and $k_{1\text{-}BuSH}$ are the overall rate constants for $R_{ST,1\text{-}BuSH}$ and for $R_{1\text{-}BuSH}$, respectively. Rewritten Equations (4-14) and (4-15) yield Equations (4-16) and (4-18), respectively.

$$R_{ST,1-BuSH} = k'_{ST,1-BuSH}[ST]^a[1-BuSH]^b \quad (4\text{-}16)$$

where

$$k'_{ST,1-BuSH} = k_{ST,1-BuSH}[AIBN]^c \quad (4\text{-}17)$$

$$R_{1-BuSH} = k'_{1-BuSH}[ST]^d[1-BuSH]^e \quad (4\text{-}18)$$

where

$$k'_{1-BuSH} = k_{1-BuSH}[AIBN]^f \quad (4\text{-}19)$$

Employing the same method for the evaluation of p and q, the exponent orders a and b were calculated equal to 0.97 ± 0.21 and 0.24 ± 0.02, respectively. By the same manner, the exponent orders d and e were calculated equal to 1.51 ± 0.29 and 0.38 ± 0.08, respectively. $k'_{ST,1\text{-}BuSH}$ and $k'_{1\text{-}BuSH}$ at 0.0, 2.5 and 5.0 % styrene conversions of the polymerization Batches J-M were calculated; the results are, respectively, shown in Tables 4-2 and 4-3. The average values of $k'_{ST,1\text{-}BuSH}$ and $k'_{1\text{-}BuSH}$ were evaluated equal to $1.01 \times 10^{-4} \pm 6.81 \times 10^{-6}$ l/mol.s and $5.31 \times 10^{-5} \pm 7.88 \times 10^{-6}$ l/mol.s, respectively.

Copyright by Mahidol University

4.3 Mathematical Modelling and Computer Simulation of the Non-Classical Thermal-Initiated Free Radically Solution Polymerization of a Vinyl Monomer in the Presence of a Chain Transfer Agent.

4.3.1 The mathematical modelling

The non-classical kinetics of thermal-labile free radically solution polymerization reaction of a vinyl monomer in the presence of a chain transfer agent proposed in this study are shown in Figure 4-6. Two crucial evidents, being the major differentiations of the kinetics proposed in this study to those proposed by many previous workers, are that :-

(1) the competition reactions between R* with M yielding RM* and R* with CTA yielding CTA* are accounted for the first time in this study, and (2) in the case that CTA is effective (high C_x value) resulting in CTA exhausted in the early stage of the polymerization reaction, the propagation reactions of RM_n^* (where n is an integer ranging from 1 to ∞) with mM (where m is an integer ranging from 1 to ∞) take place continuously.

However, the effect of dead end polymers acting as CTA has not yet been accounted for in this study. Furthermore, other possible reactions such as the reaction between the monomer acting as CTA with the primary chain transfer radicals (Reaction (R-32)) and the coupling

Copyright by Mahidol University

Figure 4-6 The kinetic mechanism of thermal-initiated free radically solution polymerization of a vinyl monomer in the presence of a chain transfer agent.

---

1) Initiator dissociation reaction to give the primary initiator radical

$$I \xrightarrow{\Delta, k_d} 2R^* \qquad \text{(R-1)}$$

2) Preparation of the primary chain transfer agent radical

$$R^* + CTA \xrightarrow{k_{sr}} CTA^* + R \qquad \text{(R-2)}$$

3) Reaction of the primary chain transfer agent radical with monomer

$$CTA^* + M \xrightarrow{k_{i,CTA}} CTAM^* \qquad \text{(R-3)}$$

4) Reaction of the primary initiator radical with monomer

$$R^* + M \xrightarrow{k_i} RM^* \qquad \text{(R-4)}$$

5) Propagation reactions

$$M_n^* + M \xrightarrow{k_p} M_{n+1}^*, \ n=1,2,3,.. \qquad \text{(R-5)}$$

$$RM_n^* + M \xrightarrow{k_p} RM_{n+1}^*, \ n=1,2,3,.. \qquad \text{(R-6)}$$

$$CTAM_n^* + M \xrightarrow{k_p} CTAM_{n+1}^*, \ n=1,2,3,.. \qquad \text{(R-7)}$$

6) Termination reactions

6.1) By combination

a) combination of similar types of growing radicals

$$M_n^* + M_m^* \xrightarrow{k_{tc}} M_{n+m} \qquad \text{(R-8)}$$

Copyright by Mahidol University

$$RM_n^* + RM_m^* \xrightarrow{k_{tc}} RM_{n+m}R \qquad \text{(R-9)}$$

$$CTAM_n^* + CTAM_m^* \xrightarrow{k_{tc}} CTAM_{n+m}CTA \qquad \text{(R-10)}$$

b) combination of different types of growing radicals

$$M_n^* + RM_m^* \xrightarrow{k_{tc}} RM_{n+m} \quad \text{(R-11)}$$

$$M_n^* + CTAM_m^* \xrightarrow{k_{tc}} CTAM_{n+m} \quad \text{(R-12)}$$

$$RM_n^* + CTAM_m^* \xrightarrow{k_{tc}} CTAM_{n+m}R \quad \text{(R-13)}$$

6.2) By disproportionation

$$M_n^* + M_m^* \xrightarrow{k_{td}} M_n + M_m \quad \text{(R-14)}$$

$$M_n^* + RM_m^* \xrightarrow{k_{td}} M_n + RM_m \quad \text{(R-15)}$$

$$M_n^* + CTAM_m^* \xrightarrow{k_{td}} M_n + CTAM_m \quad \text{(R-16)}$$

$$RM_n^* + RM_m^* \xrightarrow{k_{td}} RM_n + RM_m \quad \text{(R-17)}$$

$$RM_n^* + CTAM_m^* \xrightarrow{k_{td}} RM_n + CTAM_m \quad \text{(R-18)}$$

$$CTAM_n^* + CTAM_m^* \xrightarrow{k_{td}} CTAM_n + CTAM_m \quad \text{(R-19)}$$

where m and n are integer greater than 1, and k_{td} is rate coefficient for the termination by disproportionation reaction.

6.3) Reactions leading to inactive primary radicals

$$CTA^* + M_n^* \xrightarrow{k_{tcc}} CTAM_n \quad \text{(R-20)}$$

$$CTA^* + RM_n^* \xrightarrow{k_{tcc}} CTAM_nR \quad \text{(R-21)}$$

$$CTA^* + CTAM_n^* \xrightarrow{k_{tcc}} CTAM_nCTA \quad \text{(R-22)}$$

$$R^* + M_n^* \xrightarrow{k_{tci}} RM_n \quad \text{(R-23)}$$

$$R^* + RM_n^* \xrightarrow{k_{tci}} RM_nR \quad \text{(R-24)}$$

$$R^* + CTAM_n^* \xrightarrow{k_{tci}} RM_nCTA \quad \text{(R-25)}$$

Copyright by Mahidol University

7) Chain transfer reactions

$$M_n^* + CTA \xrightarrow{k_{tr}} M_nX + CTA^* \quad \text{(R-26)}$$

$$RM_n^* + CTA \xrightarrow{k_{tr}} RM_nX + CTA^* \quad \text{(R-27)}$$

$$CTAM_n^* + CTA \xrightarrow{k_{tr}} CTAM_nX + CTA^* \quad \text{(R-28)}$$

$$M_n^* + M \xrightarrow{k_{trm}} M_n + M^* \quad \text{(R-29)}$$

$$RM_n^* + M \xrightarrow{k_{trm}} RM_n + M^* \quad \text{(R-30)}$$

$$CTAM_n^* + M \xrightarrow{k_{trm}} CTAM_n + M^* \quad \text{(R-31)}$$

---

reactions between the primary chain transfer radicals (Reaction (R-33)) were not included in the kinetics proposed in this study.

$$CTA^* + M \xrightarrow{k_{sm}} CTAZ + M^* \quad \text{(R-32)}$$

where Z is atom derive from a monomer molecule.

$$CTA^* + CTA^* \xrightarrow{k_{tss}} CTA\text{-}CTA \quad \text{(R-33)}$$

The mathematical modelling of such the proposed kinetics was established and given in detail in Appendix I. The modelling composed of sets of algebraic equations and differential equations. Solutions for the latter were obtained using the numerical integration techniques as described in the Experimental Chapter III. Computer algorithms for the numerical integrations and for solving the algebraic equations were written in the Microsoft Quick BASIC computer language version 4.5. Computer simulations were carried out using a IBM-compatible personal computer.

Copyright by Mahidol University

Prior to employing the computer programmes for simulation studies, it is essential to test for two resons, including (a) whether or not they are reliable to be utilized and (b) to confine limitations of the computer programmes written. All of the kinetic coefficients presented in the model were examined by monitoring the effect of variation of these values on the monomer and CTA conversions. It was found that k_{sr} (ranged from 10^{-5} to 10^{6} l/mol.s) had very little effect on the monomer and CTA conversions. In contrast, the variation of k_p and k_{tr} values showed a great changes on the monomer and CTA conversions. The obvious limitation of the computer programmes written in this work is that the smaller the incremental time for the numerical algorithms is used, the more the computing time consumes. The greater incremental time results in solutions obtained with high tolerance. The incremental time used for the computer simulations in this study were ranged from 20 to 30 s.

4.3.2 The computer simulations

The computer programmes required to key-in known values of the initial concentrations (mol/l) of monomer, CTA, and initiator as well as those of the kinetic coefficients appeared in the mathematical modelling. The kinetic coefficients used for the computer simulation experiments are shown in Table 3-2. The values of k_p and k_{tr} were varied while the others were constant. Simulation results were presented on the relationship of the monomer or CTA conversions with the reaction time. The results were compared to those obtained experimentally. Disagreement of these results could be dissolved by adjusting firstly the values of k_{tr} until the accordance results of the simulation and experimental CTA conversion-

Copyright by Mahidol University

time were obtained within 5 % tolerance. Secondly, the monomer conversion-time obtained by simulations was compared to that obtained experimentally. Discrepancy of these results rendered adjustment of the k_p value until the agreement was obtained. Subsequently, comparison of the experimental and simulation CTA conversion-time results was again carried out in order to ensure that there was of no great different occurred according to the adjustment of the k_p value.

Table 4-4 shows the k_p and k_{tr} values which gave the best agreement of the monomer and CTA conversion-time results obtained experimentally and computationally. The maximum reaction time used for the computer simulations was 60 minutes. Figures 4-7 - 4-11 show the computer simulation (the continuous line) compared to the experimental results (represented by various symbols). The simulation values (hereafter refer to the values obtained from the computer simulations) of k_p and k_{tr}were different from the values quoted in literatures in the order of 10^2.

Copyright by Mahidol University

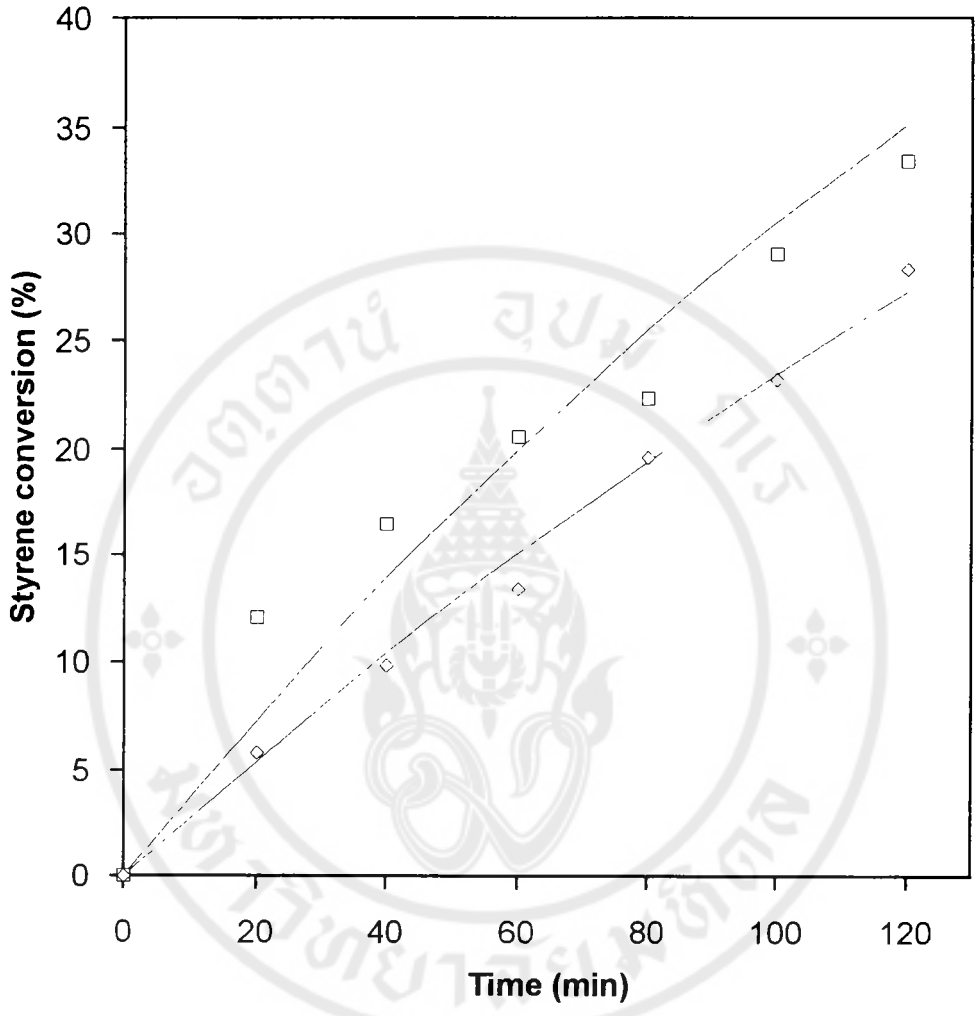


Figure 4-7 The plot of styrene conversion against time (Continuous line (——) for the simulation results, Experiment: (□) Batch A (k_p=320 l/mol.s) and (◊) Batch B (k_p=220 l/mol.s)).

Copyright by Mahidol University

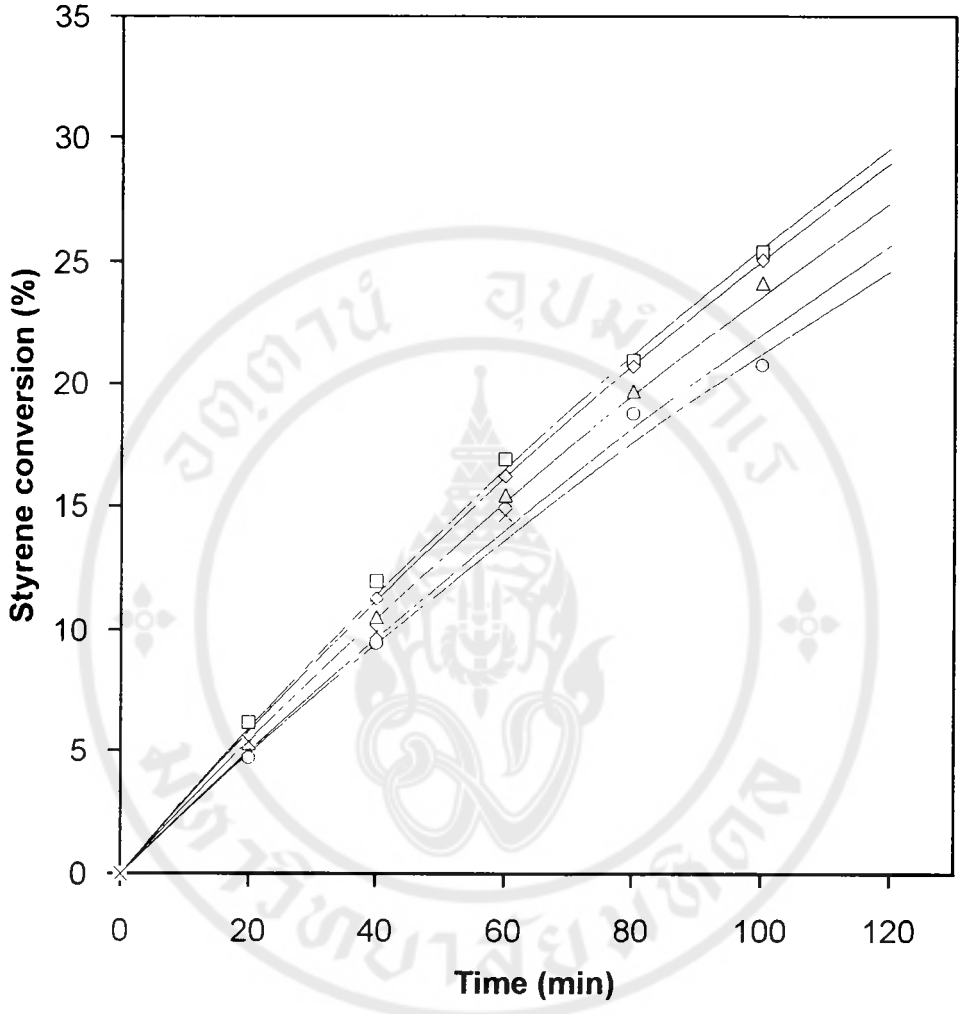


Figure 4-8 The plot of styrene conversion against time (Continuous line (——) for the simulation results, Experiment: (□) Batch C (k_p=245 l/mol.s), (◊) Batch D (k_p=230 l/mol.s), (Δ) Batch E (k_p=220 l/mol.s), (x) Batch G (k_p=200 l/mol.s)and (o) Batch H (k_p=180 l/mol.s)).

Copyright by Mahidol University

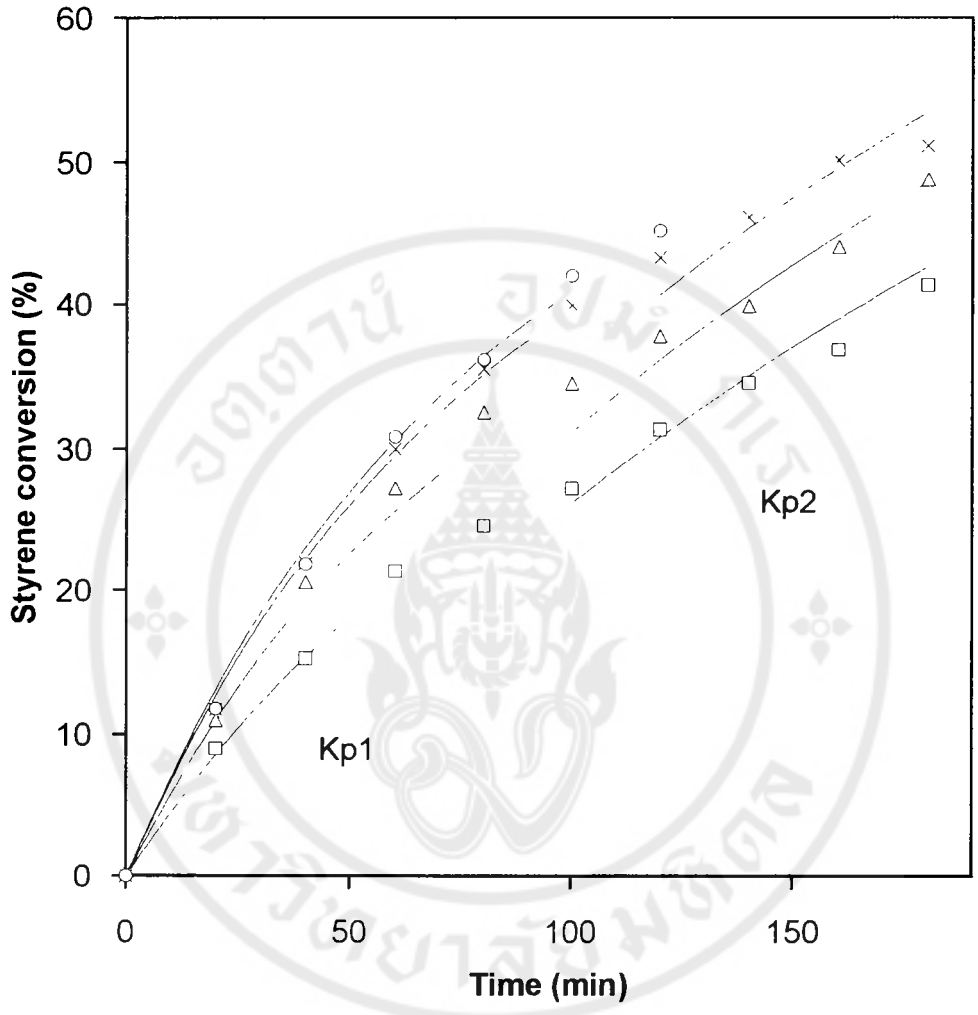


Figure 4-9 The plot of styrene conversion against time (Continuous line (——) for the simulation results, Experiment: (□) Batch J (k_{p1}=250, k_{p2}=170 l/mol.s), (Δ) Batch K(k_{p1}=275, k_{p2}=200 l/mol.s), (x) Batch L (k_{p1}=280, k_{p2}=225 l/mol.s), and (o) Batch M (k_{p1}=285 l/mol.s)).

Copyright by Mahidol University

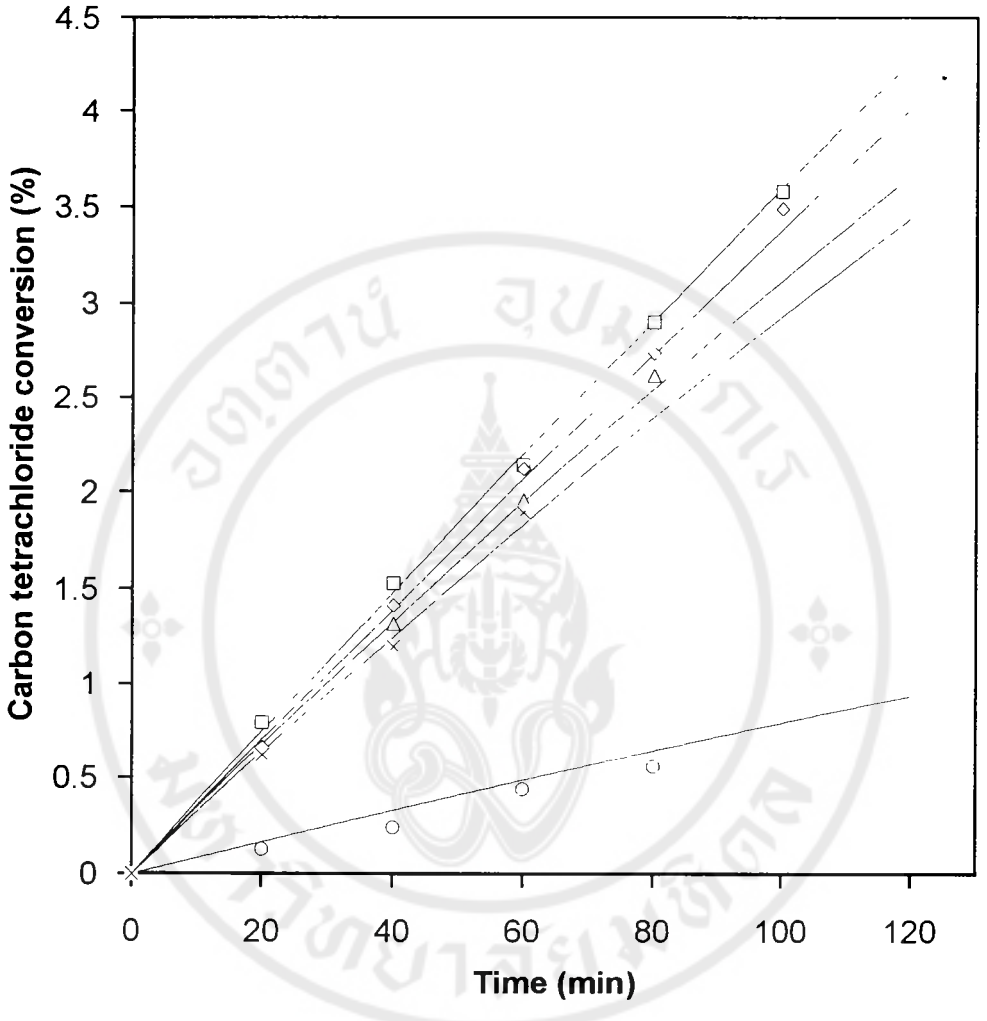


Figure 4-10 The plot of CCl_4 conversion against time (Continuous line (——) for the simulation results, Experiment: (□) Batch C (k_{tr}=4.5x10^{-4} l/mol.s), (◊) Batch D (k_{tr}=1.2 l/mol.s), (Δ) Batch E (k_{tr}=4.5 l/mol.s), (x) Batch G (k_{tr}=4.8 l/mol.s),and (o) Batch H (k_{tr}=5.0 l/mol.s)).

Copyright by Mahidol University

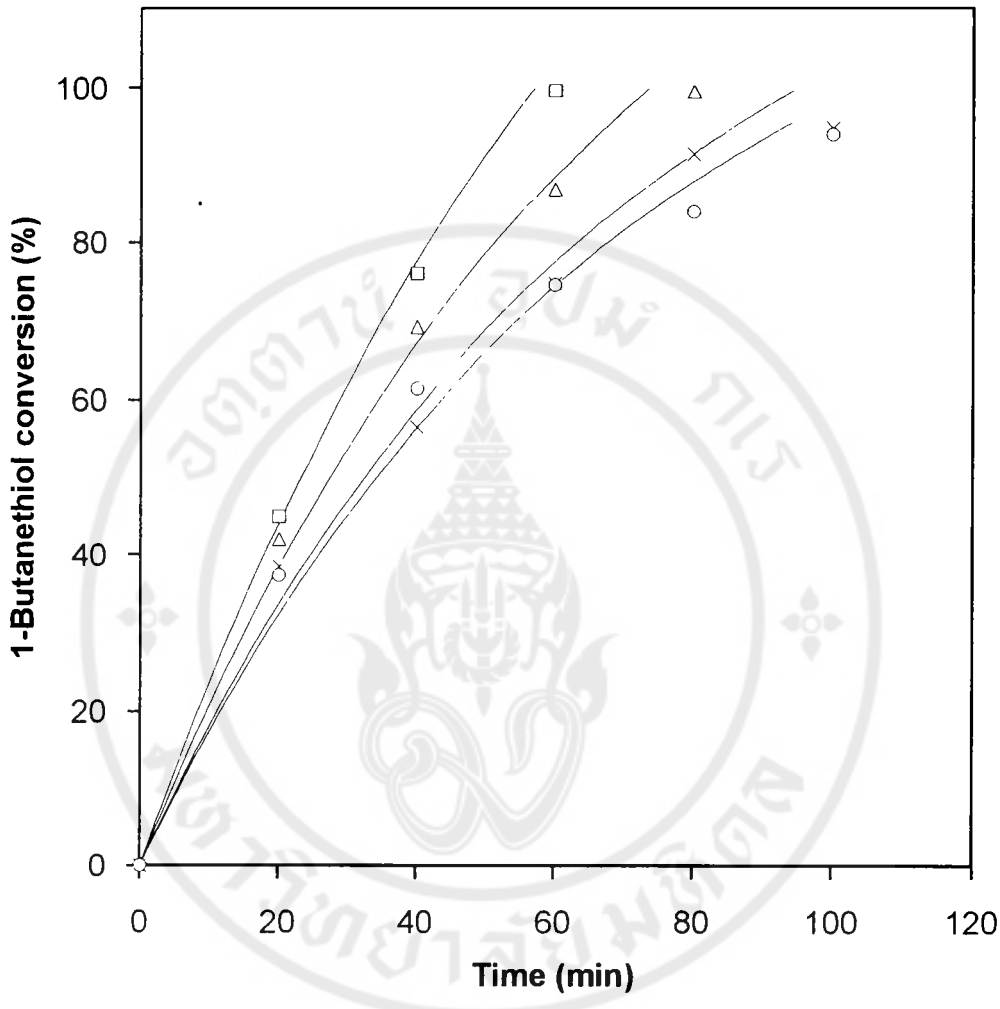


Figure 4-11 The plot of 1-BuSH conversion against time (Continuous line (——) for the simulation results, Experiment: (□) Batch J (k_{tr}=950 l/mol.s), (Δ) Batch K (k_{tr}=1020 l/mol.s), (x) Batch L (k_{tr}=1050 l/mol.s), and (o) Batch M (k_{tr}=1060 l/mol.s)).

Copyright by Mahidol University

Table 4-4 The simulation values of k_p and k_{tr} at 70 °C.

Batch	$k_{p,1}$ (l/mol.s)	$k_{p,2}$ (l/mol.s)	k_{tr} (l/mol.s)
A	320	-	-
B	220	-	-
C	245	-	$4.5X10^{-4}$
D	230	-	1.2
E	220	-	4.5
G	200	-	4.8
H	180	-	5.0
I	-	-	-
J	250	170	950
K	275	220	1020
L	280	225	1050
M	285	-	1060

Copyright by Mahidol University

Considering the styrene conversion-time curves of Batches J-K (Figure 4-9, 1-BuSH was the chain transfer agent), two discrete conversion-time relationships can be observed in those curves. For the polymerisation Batches J and K, the conversion-time relationship deviated after the reaction time of 80 and 100 minutes, respectively. The evidences suggest that the kinetic mechanism after that time should not be the same as that of the earlier stage of the polymerisation reaction. Consideration of the 1-BuSH conversion-time curves suggests that, for the polymerisation Batches J and K, 1-BuSH be all used within 50 and 80 minutes, respectively. The polymerisation kinetics after exhausting of the dedicated chain transfer agent should therefore be the kinetics of free radical polymerisation in the absence of the dedicated chain transfer agent. The similar evidence has been found for the telomerisation of styrene using CBr_4 chain transfer agent [52]. The simulation values, k_{p1} and k_{p2}, were evaluated by fitting the styrene conversion-time curves relevant to those two regions appeared in Figure 4-9. These values are also given in Table 4-4.

In contrast, when CCl_4 was the chain transfer agent, two discrete conversion-time relationship cannot be observed (see Figure 4-8). This result suggested that C_x of 1-BuSH should be much higher than that of CCl_4.

Copyright by Mahidol University

4.4 Comparisons of the Results obtained from the Experiments and from the Computer Simulations.

4.4.1 Molecular weight and molecular weight distribution of polymers

Values used for computer simulations for the particular polymerization batches were the same as those shown in Table 3-2 (see Chapter III) except for those of k_p and k_{tr}. The initial concentrations of styrene, AIBN and CCl_4 or 1-BuSH of the particular polymerization batches were also required for the simulation experiments; these values are shown in Table 3-1 (see Chapter III). The molecular weight (MW) and molecular weight distribution (MWD) obtained by the computer simulations were compared with those obtained experimentally. Figures 4-12 - 4-23 show the overlaid MWD curves, obtained experimentally and computationally, of all polymerization batches. Good agreement between the experimental and simulation MWDs could be observed from the overlaid curves. The values of number-average and weight-average MW (M_n and M_w) obtained experimentally and computationally were in good agreement. These values are also given in Figures 4-12- 4-23. Since the simulation results of both of the MW and MWD were in good accordance with the experimental ones, it can be concluded that the kinetics of thermal-initiated free radical polymerization of a vinyl monomer in the presence of a chain transfer agent proposed in this study are acceptable.

Copyright by Mahidol University

The polymerization Batch H, the large amount of CCl_4 (8.17 mol/l) in comparison to that of styrene (1.74 mol/l) was used. MWD of polymers obtained had two discrete distributions (see Figure 4-18); the smaller had

lower MW than the larger. The low quality of low MW polymer obtained and may have the molecular structure as CCl_3-(CH_2-CH(Ph)-)-Cl (S1), with the number-average degree of polymerization (DP_n) approximately equal to 3. DP_n of the larger MWD was approximately 60; the MWD seemed to be the normal distribution. Tsuchida and Mimashi (1972) proposed the kinetics of free radical telomerisation of styrene using CCl_4 telogen to obtain telomer having the chemical structure S1. They reported that the pure telomer was obtained having normal MWD, which is in contrast to the results found in this study. From the computer simulations, a mixture of dead-end polymers could be observed with a large amount of S1. It is needed further investigation on the chemical structure of the polymers of the larger MWD.

The smallest and the largest amount of 1-BuSH used in this study was 0.36 mol/l (Batch I) and 1.81 mol/l (Batch M). MWD of Batch M had two discrete peaks which left-hand peak had DP_n equal to 1 and the other equal to 2, indicating that the polymers obtained were low MW compounds, i.e., telomers. In contrast to Batch M, MWD of Batch I had only one peak. Except for the reaction time of 180 minutes, right distribution shoulder appeared in the region of high MW. Batch J showed the similar MWD to Batch I, and Batches K and L showed the similar MWD to Batch M. The simulation MWDs of all Batches gave disagreement results to the experimental MWDs, but good agreement in M_n and M_w values obtained. It was observed that the area under the simulation MWD peaks was two times higher than that obtained experimentally. It is because of the inexisting assumption used in the kinetic modelling in particular the kinetic coefficients for the chain transfer

Copyright by Mahidol University

reactions of CTA to various types of active growing radicals were assumed all equal to k_{tr}. There are some least possible reactions in the kinetic model, which appears in reality, such as the combination reactions of $CTAM_n^*$ with $CTAM_m^*$ or with RM_x^*. These reactions caused unacceptable simulation results compared to the experimental. It is noted that this disagreement did not happen when CCl_4 was employed as the chain transfer agent. It can be therefore concluded that the dynamic predictive model proposed in this study is acceptable only when CTA has low C_x value, i.e., high molecular weight polymers obtained.

4.4.2 The chain transfer constants (C_x) of CCl_4 and 1-BuSH

It is well-known that the C_x value of a chain transfer agent can be evaluated using the Mayo Equation by study the relation of DP_n to [CTA]/[M]. In this work, the values of C_x for CCl_4 and 1-BuSH were also obtained at 2.5 and 5.0 % styrene conversions, using the graphical method as described by Mayo (1943), i.e., the plot of reciprocal DP_n against [CTA]/[M] and C_x is the slope of curve. The DP_n values were calculated from M_n which obtained using the GPC technique as described in Chapter III. M_n obtained in this study was M_n of all types of polymers synthesized because fractionation (by separation technique) of the polymers did not perform. Figure 4-24 shows the different in MWD of polymers being separated compared to that being unseparated. It is clearly seen that separation technique causes disappearance of low MW polymers, rendering misleading value of M_n obtained.

Copyright by Mahidol University

4.4.2.1 C_X value for CCl_4

The typical Mayo plot for evaluation of C_X value for CCl_4 is shown in Figure 4-25. The values obtained at 2.5 and 5.0 % conversion of styrene are $1.06x10^{-2}$ and $1.09x10^{-2}$, respectively. The simulation values of C_X were also evaluated equal to $6.7x10^{-3}$ and $6.5x10^{-3}$ at 2.5 and 5.0 % styrene conversions. The simulation and experimental values are not much different.

The attempt has been made to derive the modified Mayo Equation from the kinetic model proposed in Section 4.3 of this Chapter. The detailed derivation of the modified Mayo Equation is given in Appendix V. The differences of the Modified and original Mayo Equations are that DP_n of the former substituted by DP_n-1 and $DP_{n,0}$ (the DP_n of polymers prepared in the absence of CTA). The experimental and simulation C_X values evaluated using the modified Mayo Equation at 2.5 and 5.0 % styrene conversion were evaluated and shown in Table 4-6. The experimental and simulation values obtained from the original and the modified Mayo Equations were not of great differences.

4.4.2.2 C_X value for 1-BuSH

The typical original and modified Mayo plot for evaluation of C_X value for 1-BuSH is shown in Figure 4-26. The experimental and simulation values obtained at 2.5 and 5.0 % conversion of styrene using the original andality ied Mayo Equations are also given in Table 4-6. The experimental and simulation values obtained from the original and the modified Mayo

Copyright by Mahidol University

Equations were of great differences. However, it may be concluded that the modified Mayo Plot gives the better results than the original plot, by comparison with the value quoted in literature.

Table 4-5 Values of [CTA]/[M] and degree of polymerization (DP_n) at 2.5 and 5.0% styrene conversion.

Batch	**2.5 % styrene conversion**		**5.0 % styrene conversion**	
	[CTA]/[M]	DP_n	[CTA]/[M]	DP_n
(CCl_4)				
C	0.06	128.4	0.06	120.7
D	0.12	115.5	0.12	108.8
E	0.18	111.5	0.18	103.5
G	0.30	96.1	0.31	90.7
(1-BuSH)				
J	0 13	2.26	0.11	2.27
K	0.20	2.04	0.18	2.05
L	0.26	2.00	0.25	2.03
M	0 33	1.86	0.32	1.88

Copyright by Mahidol University

Table 4-6 Values of chain transfer constant (C_x) of carbon tetrachloride and 1-butanethiol at 2.5 and 5.0 % styrene conversion.

CTA	Experimental				Computer Simulation				Ref.*
	2.5 % styrene conversion		5.0 % styrene conversion		2.5 % styrene conversion		5.0 % styrene conversion		
	(A)	(B)	(A)	(B)	(A)	(B)	(A)	(B)	
CCl_4	0.011	0.011	0.011	0.011	0.007	0.007	0.007	0.007	0.014
1-BuSH	0.422	1.632	0.406	1.547	0.761	1.243	0.729	1.208	15.0

(A) The Mayo Equation, (B) The modified Mayo Equation

*From Brandrup and Immergut Handbook of Polymer, 3rd Edition, 1975

Copyright by Mahidol University

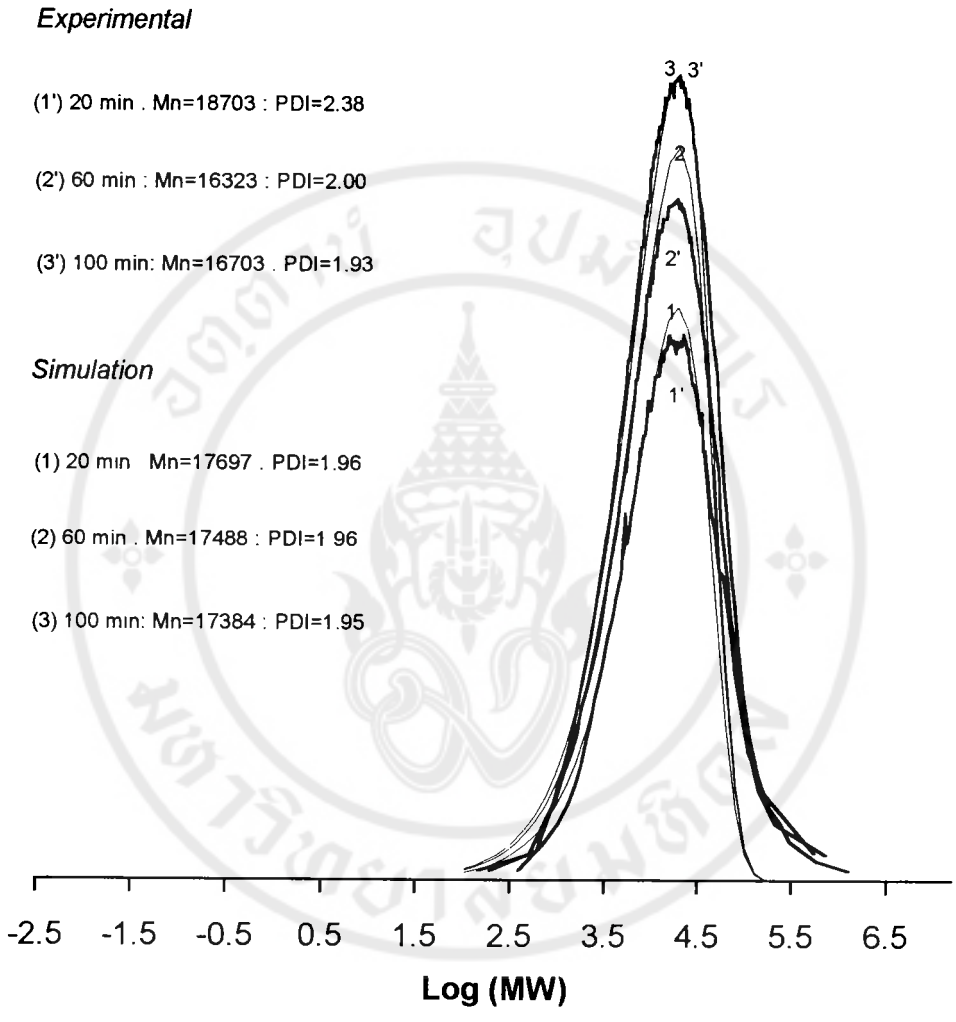


Figure 4-12 Overlaid MWD of the polymerization Batch A (Experiment (——), Computer simulation (——)).

Copyright by Mahidol University

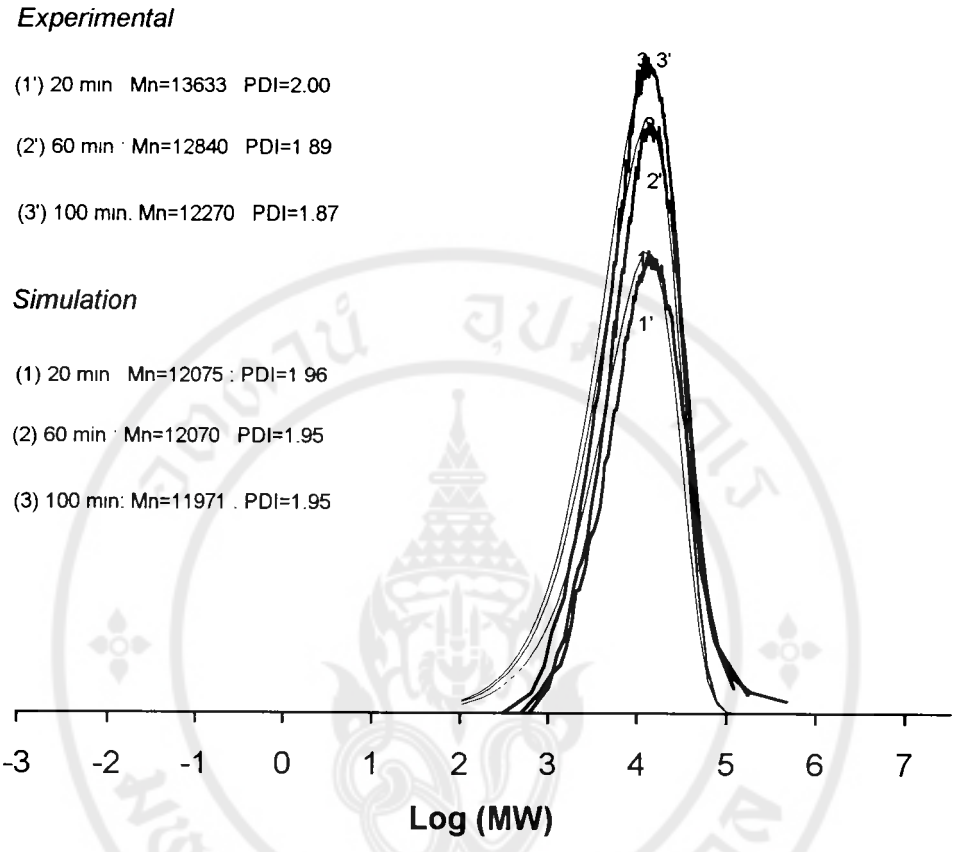


Figure 4-13 Overlaid MWD of the polymerization Batch B (Experiment (——), Computer simulation (——)).

Copyright by Mahidol University

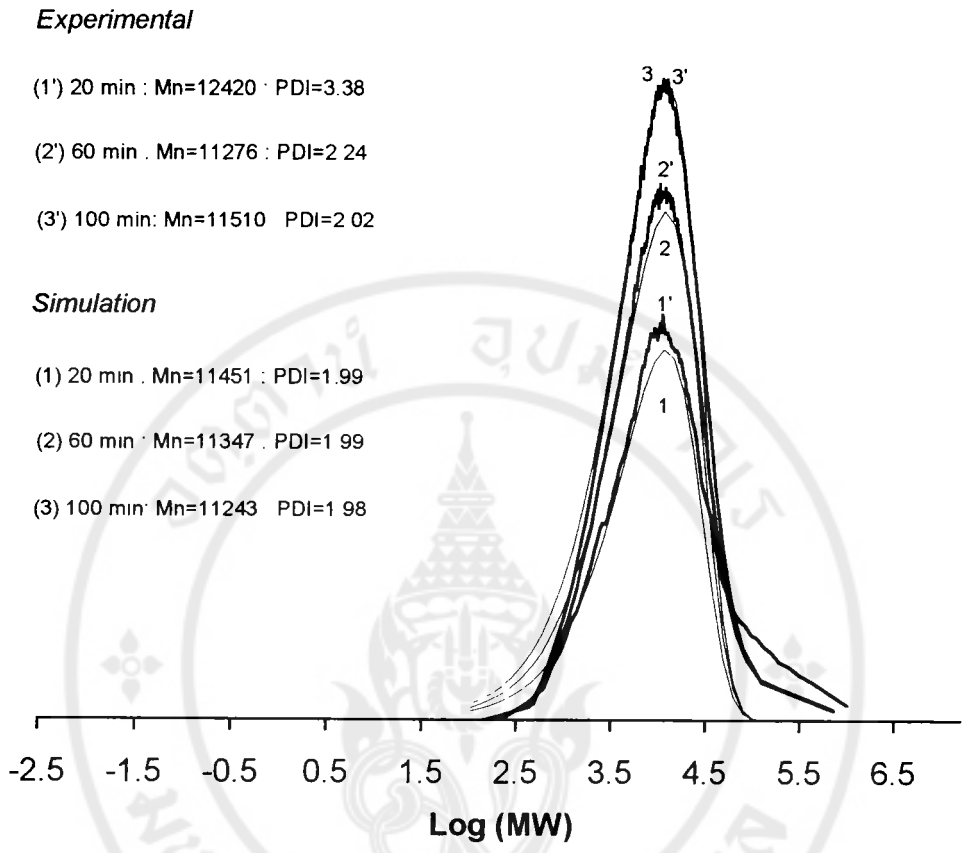


Figure 4-14 Overlaid MWD of the polymerization Batch C (Experiment (——), Computer simulation (——)).

Copyright by Mahidol University

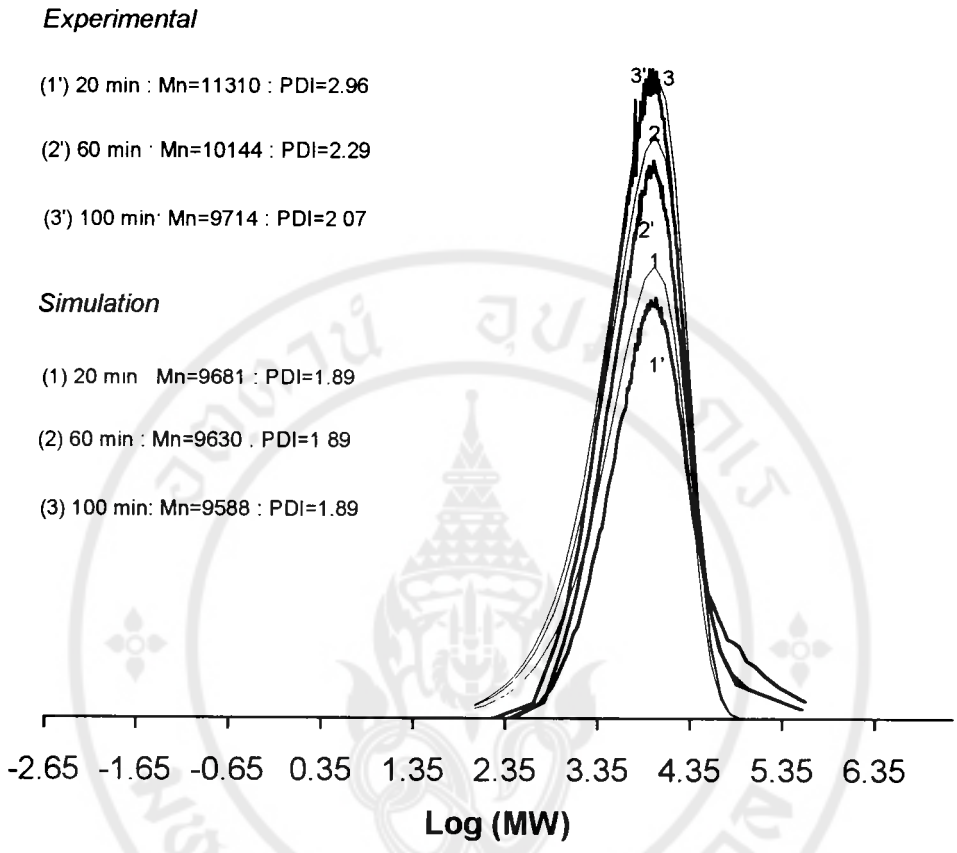


Figure 4-15 Overlaid MWD of the polymerization Batch D (Experiment (——), Computer simulation (——)).

Copyright by Mahidol University

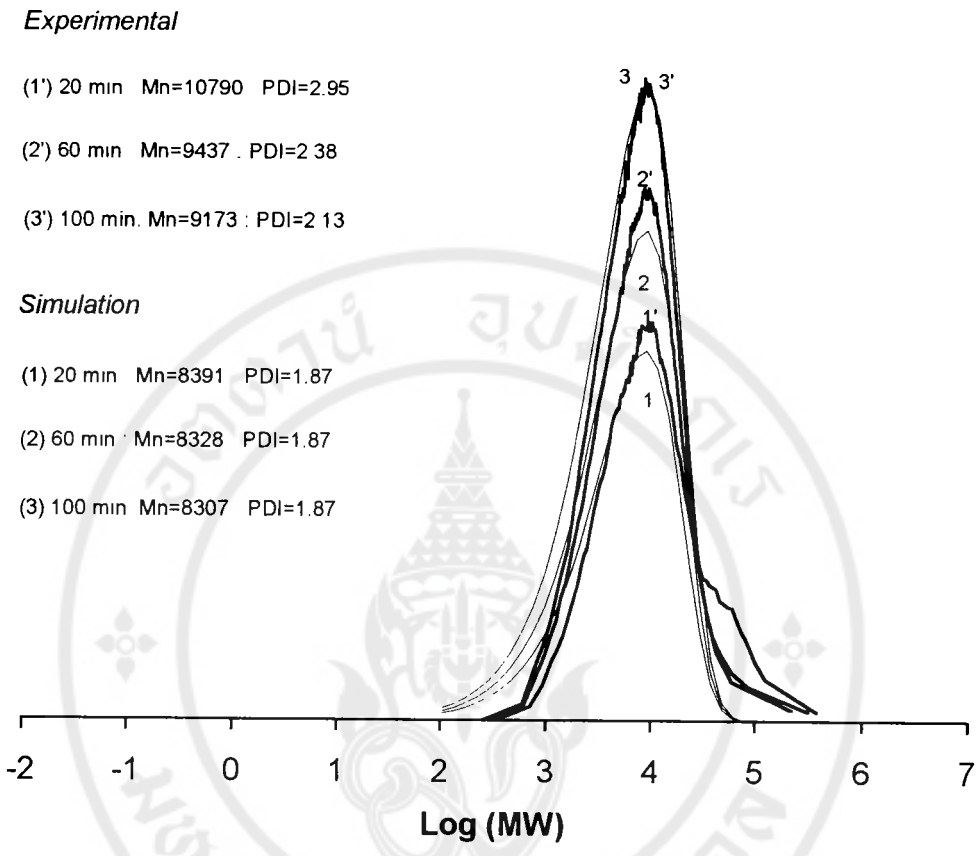


Figure 4-16 Overlaid MWD of the polymerization Batch E (Experiment (——), Computer simulation (——)).

Copyright by Mahidol University

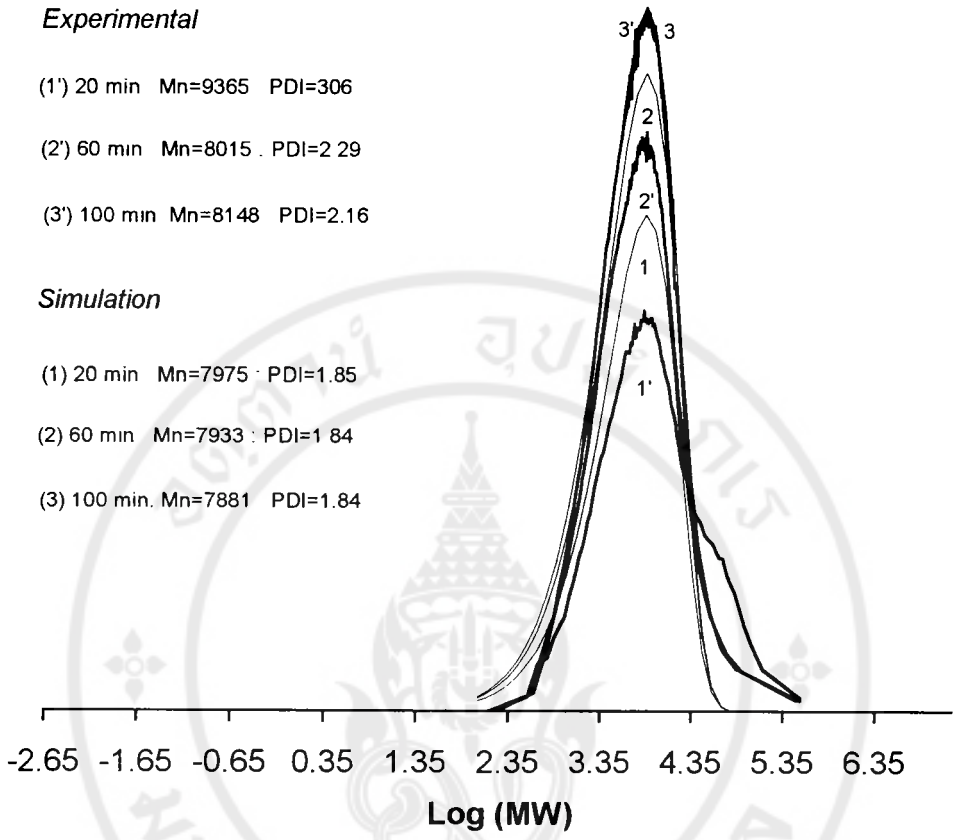


Figure 4-17 Overlaid MWD of the polymerization Batch G (Experiment (——), Computer simulation (——)).

Copyright by Mahidol University

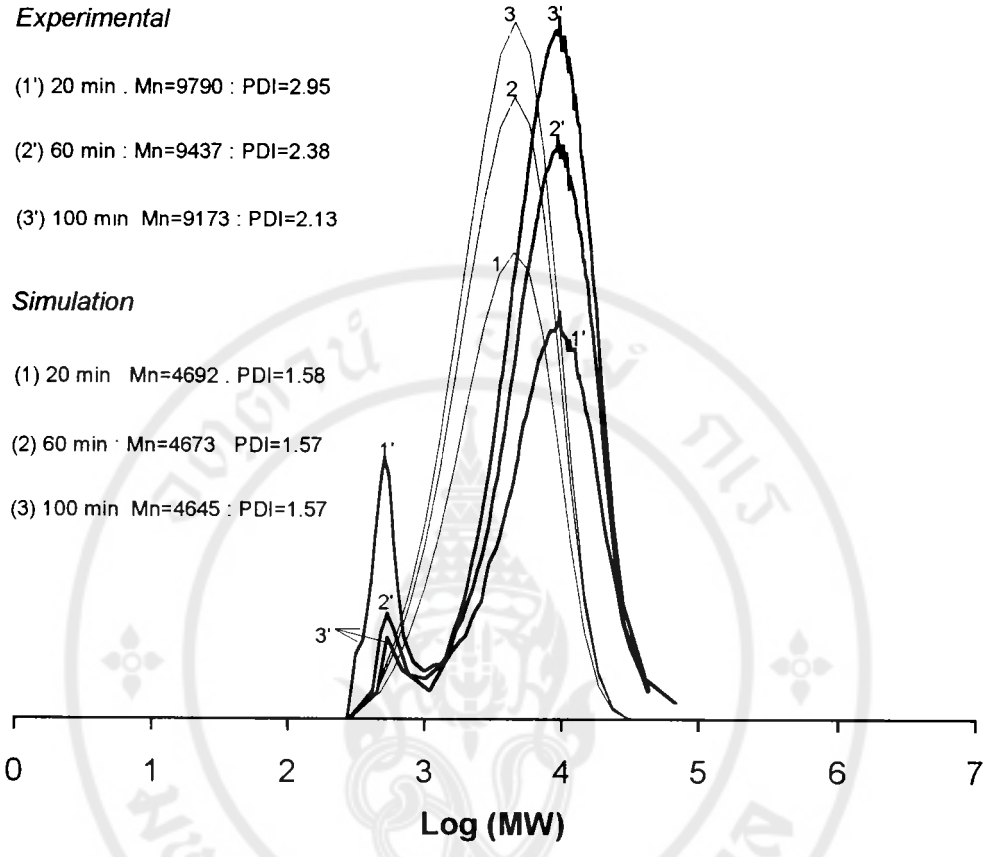


Figure 4-18 Overlaid MWD of the polymerization Batch H (Experiment (——), Computer simulation (——)).

Copyright by Mahidol University

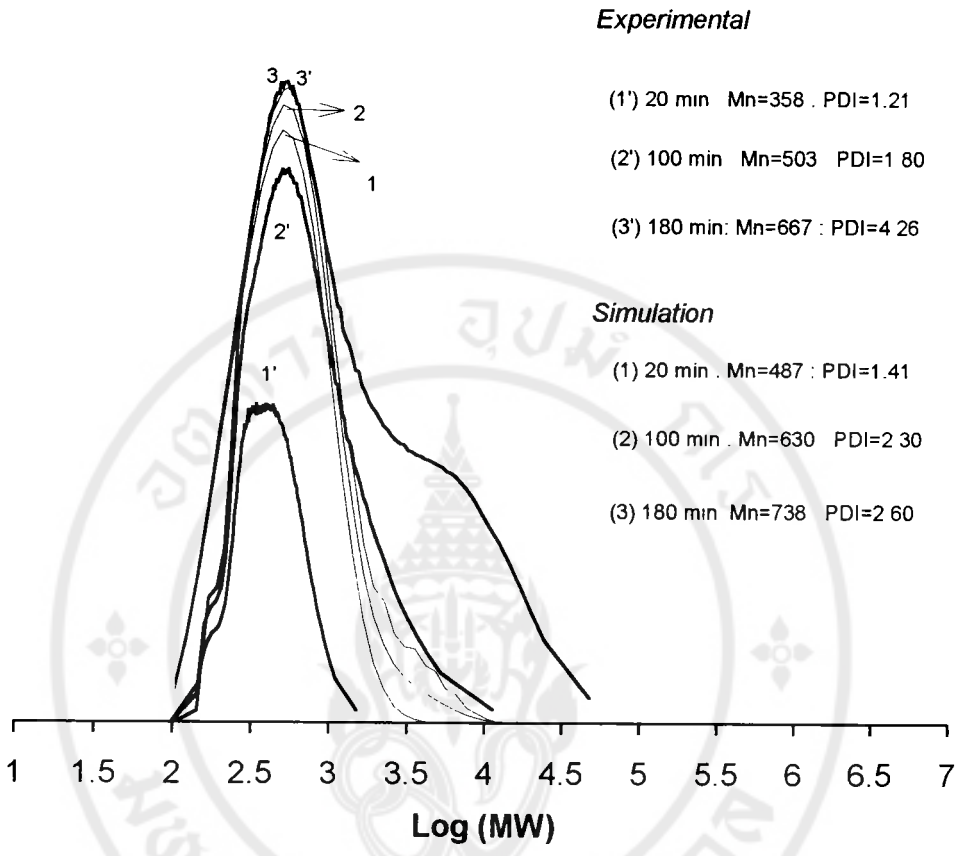


Figure 4-19 Overlaid MWD of the polymerization Batch I (Experiment (——), Computer simulation (——)).

Copyright by Mahidol University

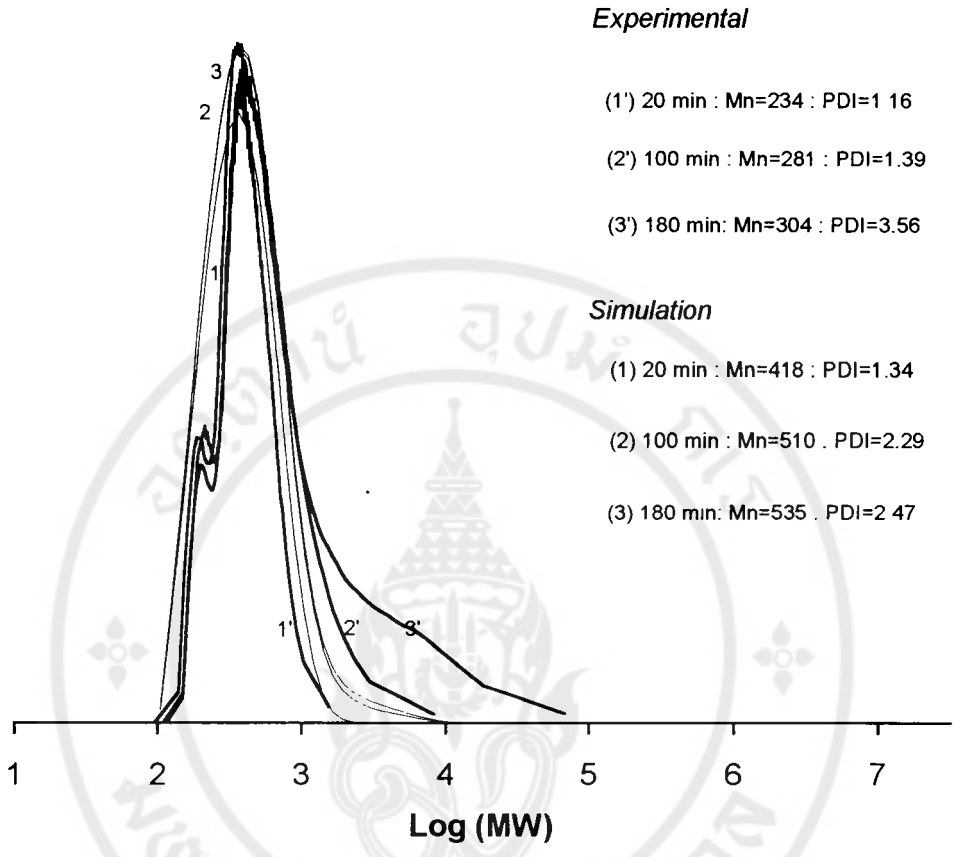


Figure 4-20 Overlaid MWD of the polymerization Batch J (Experiment (——), Computer simulation (——)).

Copyright by Mahidol University

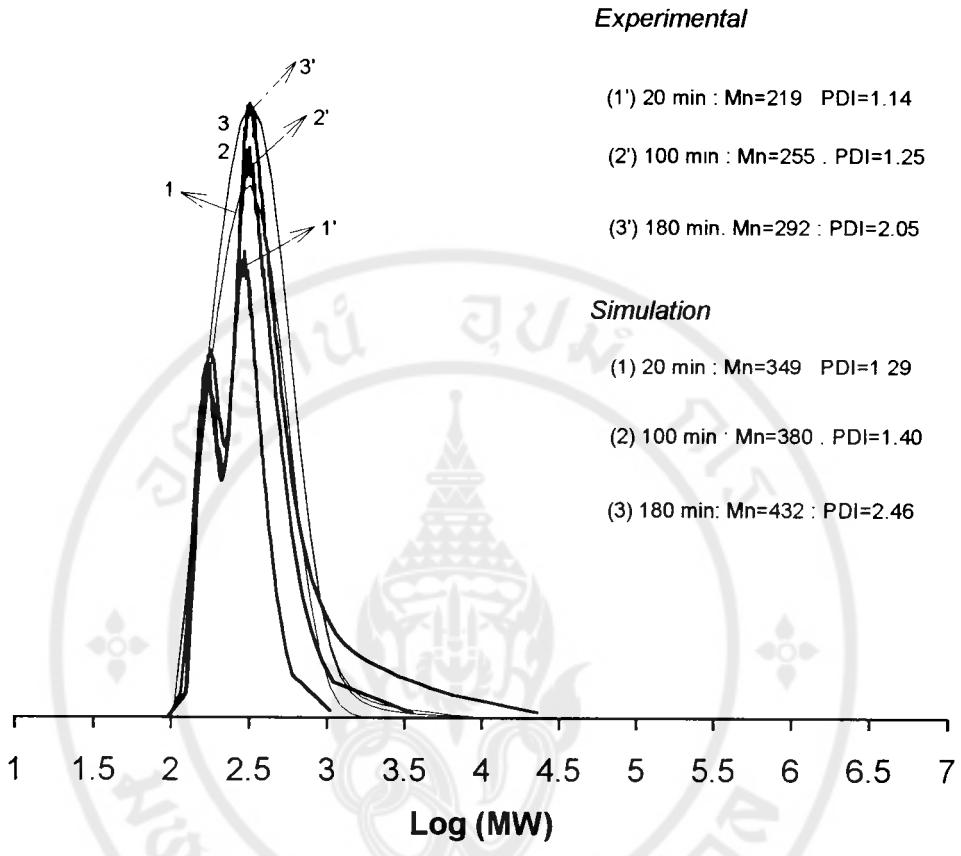


Figure 4-21 Overlaid MWD of the polymerization Batch K (Experiment (——), Computer simulation (——)).

Copyright by Mahidol University

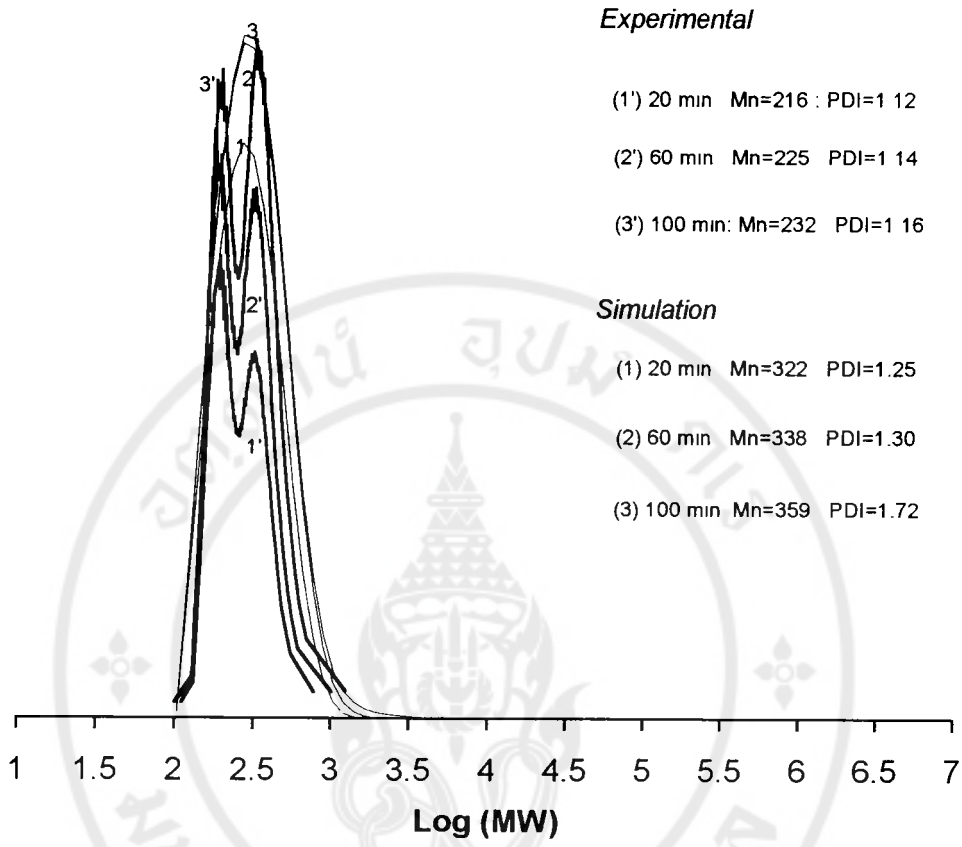


Figure 4-22 Overlaid MWD of the polymerization Batch L (Experiment (——), Computer simulation (——)).

Copyright by Mahidol University

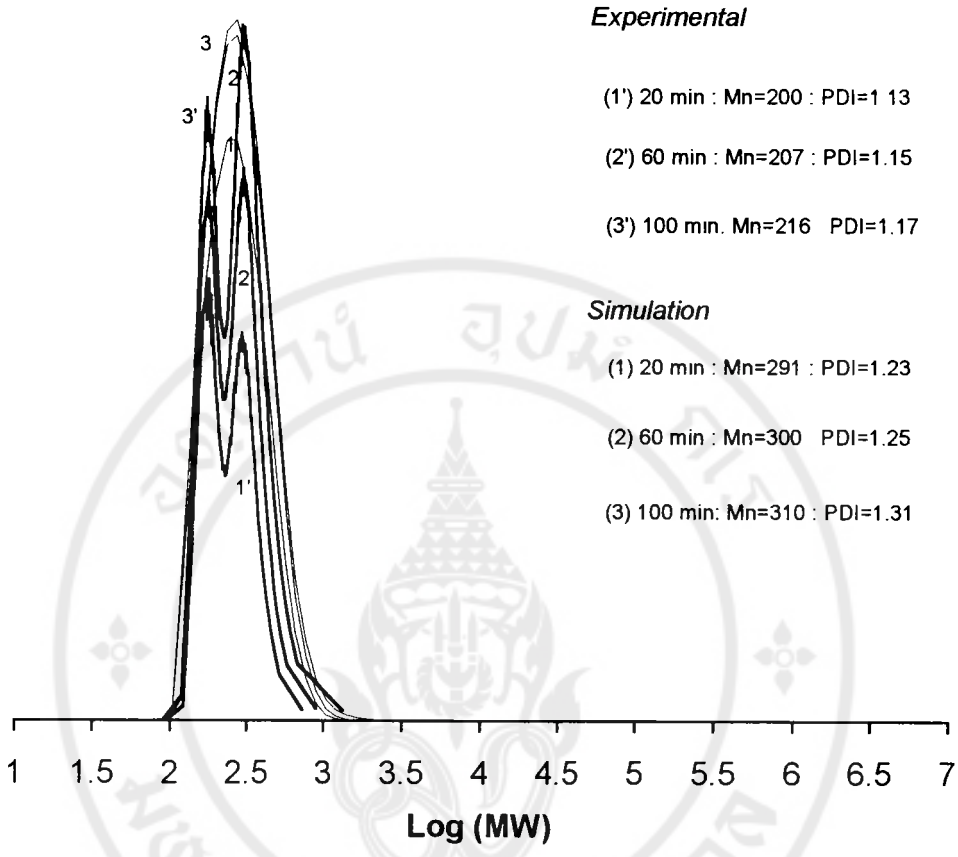


Figure 4-23 Overlaid MWD of the polymerization Batch L (Experiment (——), Computer simulation (——)).

Copyright by Mahidol University

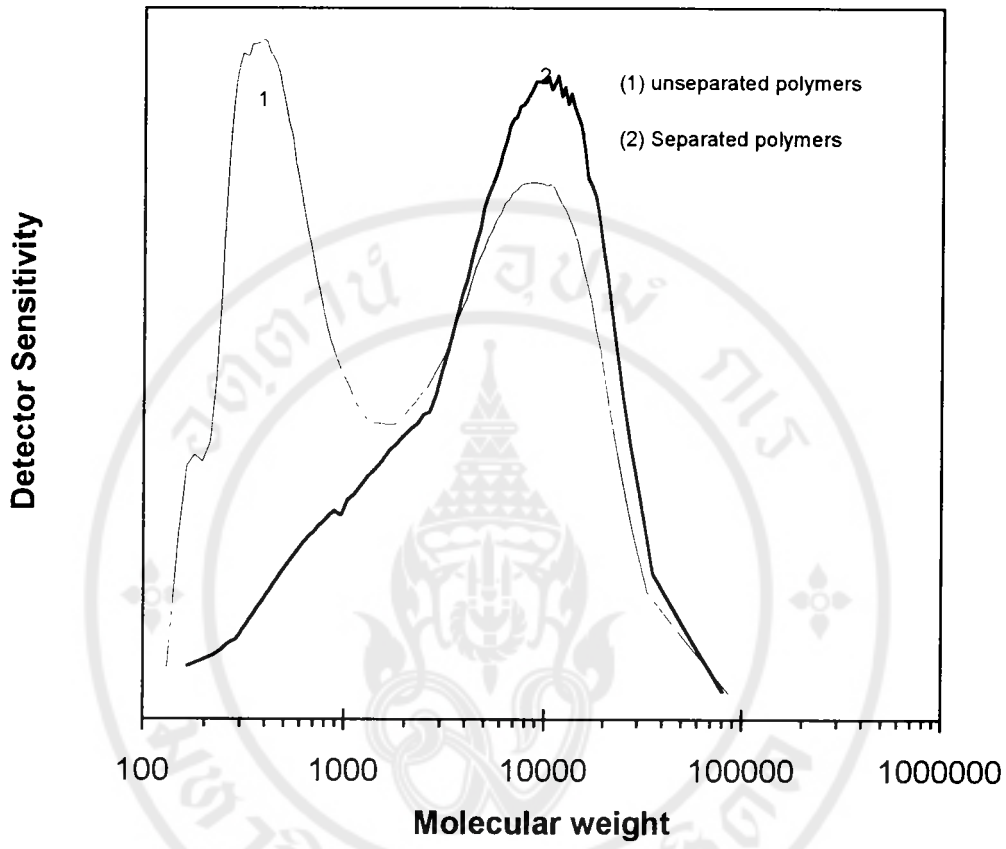


Figure 4-24 Molecular Weight Distribution of Polymers (Batch N): Comparison between separated polymers (——) and unseparated polymers (——).

Copyright by Mahidol University

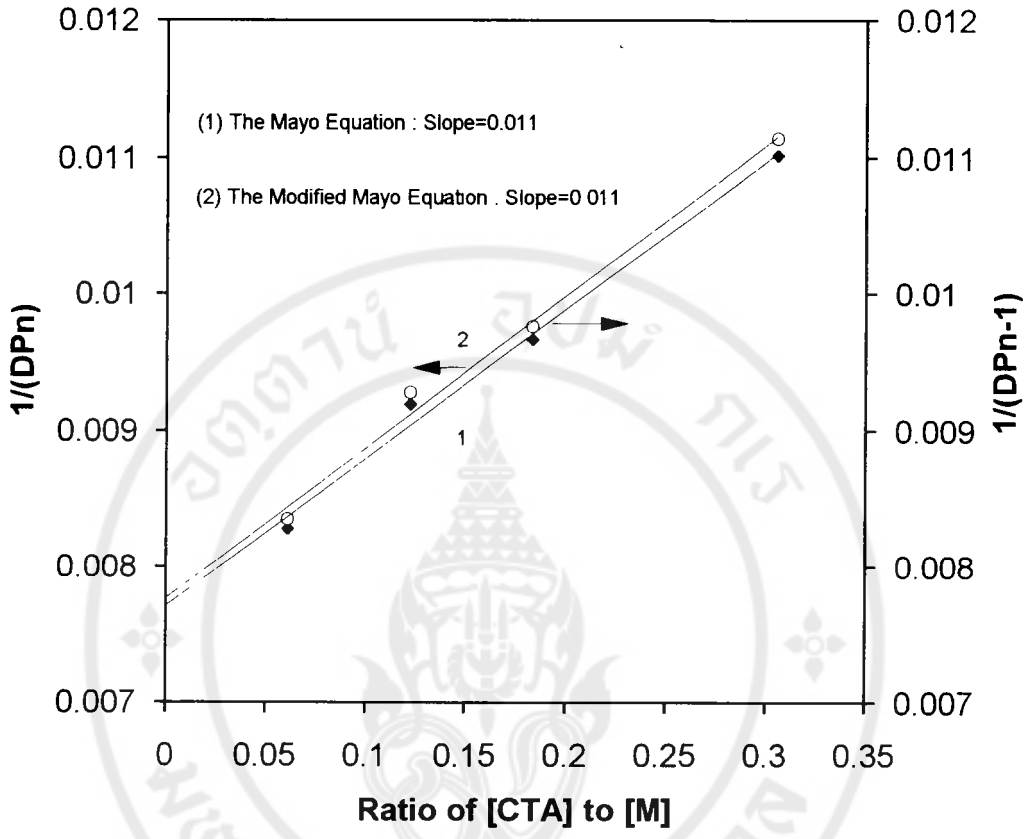


Figure 4-25 The plot of reciprocal degree of polymerization against [CTA]/[M], (CTA is carbon tetrachloride and M is styrene monomer).

Copyright by Mahidol University

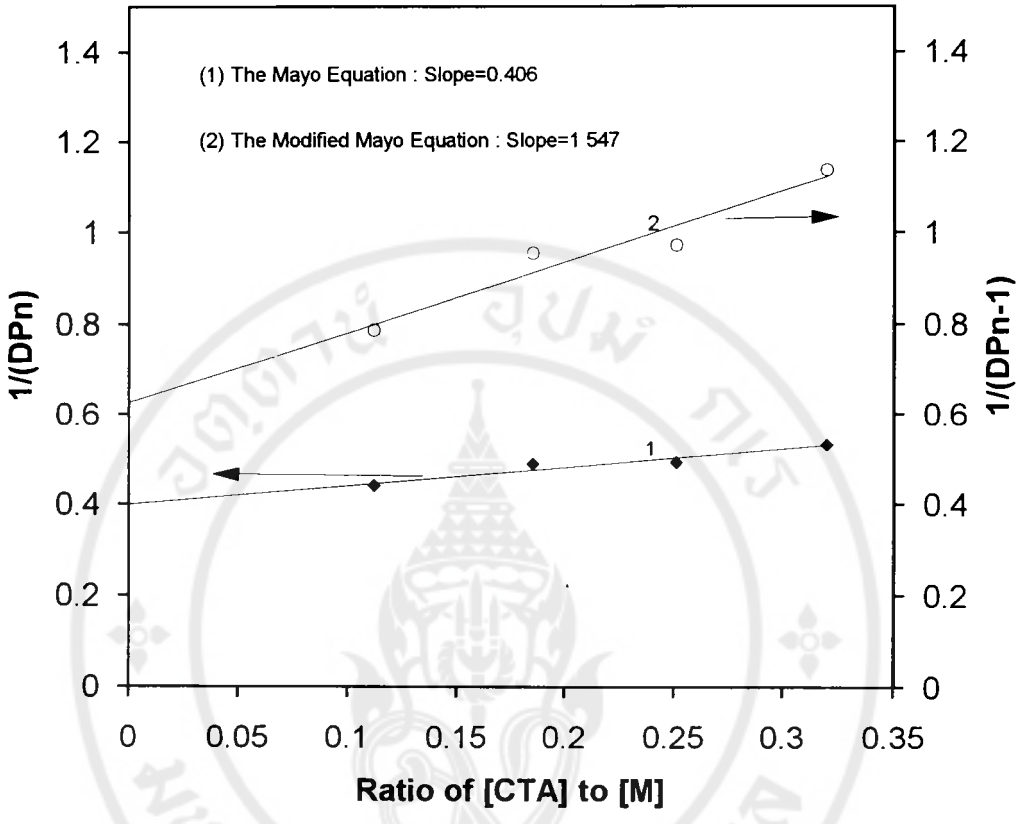


Figure 4-26 The plot of reciprocal degree of polymerization against [CTA]/[M], (CTA is 1-butanethiol and M is styrene monomer).

Copyright by Mahidol University

Chapter V
Conclusions and Suggestions for Further Work

5.1 Conclusions

Raman spectroscopic technique has been employed successfully to study the kinetics of AIBN-initiated free radical polymerization of styrene in the presence of carbon tetrachloride or 1-butanethiol chain transfer agent. Styrene, CCl_4 and 1-BuSH conversion at the particular reaction time were evaluated from the ratio of the Raman peak height or area at the wave number 1631 cm^{-1}, 314 cm^{-1}, 2577 cm^{-1}, respectively, at that reaction time to the initial time. From the conversion-time relationship, the overall rate of polymerization ($R_{St,CTA}$) and that of loss of CTA (R_{CTA}) were determined and the results obtained at 70 °C are :-

$$R_{ST.CCl_4} = k'_{ST,CCl_4}[ST]^{0.95\pm0.01}[CCl_4]^{-0.07\pm0.04}$$

where

$$k'_{ST,CCl_4} = k_{ST,CCl_4}[AIBN]^{r} = 4.88x10\text{-}5 \pm 4.18x10^{-7} \text{ l/mol.s}$$

$$R_{ST,1-BuSH} = k'_{ST,1-BuSH}[ST]^{0.97\pm0.21}[1-BuSH]^{0.24\pm0.02}$$

where

$$k'_{ST,1-BuSH} = k_{ST,1-BuSH}[AIBN]^{c} = 1.01x10\text{-}4 \pm 6.81x10^{-6} \text{ l/mol.s}$$

Copyright by Mahidol University

$$R_{CCl_4} = k'_{CCl_4}[ST]^{0.99\pm0.07}[CCl_4]^{0.91\pm0.03}$$

where

$$k'_{CCl_4} = k_{CCl_4}[AIBN]^{u} = 8.40x10^{-7} \pm 6.28x10^{-9} \text{ l/mol.s}$$

$$R_{1-BuSH} = k'_{1-BuSH}[ST]^{1.51\pm0.29}[1-BuSH]^{0.38\pm0.08}$$

where

$$k'_{1-BuSH} = k_{1-BuSH}[AIBN]^{f} = 5.31x10^{-5}\pm 7.88x10^{-6}\ \text{l/mol.s}$$

The initial overall rates ($R_{p,0}$) of AIBN-initiated free radically bulk and solution polymerization of styrene in the absence of chain transfer agent at 70 °C were also determined equal to $6.17x10^{-4}$ mol/l.s and $2.83x10^{-4}$ mol/l.s, respectively. The initial overall kinetic coefficients ($k_{p,0}$) are equal to 340.36 l/mol.s and 195.07 l/mol.s for the bulk and the solution polymerization, respectively.

The summation of individual exponent order is known as the total reaction order providing information relating to the nature of reactions, such as whether or not the reactions are dependent on the initial concentrations of starting reagents. In classical polymerization reaction, it is well-known that the total reaction order for the polymerization is 1.5. In the case that cage effect happens in the polymerization the total reaction order deviates from 1.5. For non-classical polymerization in the presence of CTA, the total reaction order for the polymerization was found to be ranged from 1.5-2.5, while that for the rate of loss of CTA to be ranged from 2.0-2.5. For the former if the total reaction order is equal to 1.5, the polymerization trends toward the classical polymerization reaction, i.e., CTA has vary little effect on the polymerization or CTA is less effective. For effective CTA, the total reaction order trends toward 2.5, i.e., the rate of polymerization is the function of the initial concentrations of initiator, monomer and CTA. For the latter if the

Copyright by Mahidol University

total reaction order is 2.0, the rate of loss of CTA is independent upon the initiator concentration, i.e., CTA is very effective, and the reaction of CTA with the primary initiator radicals is feasible. The rate of loss of CTA is totally dependent upon the initiator concentration when CTA is less effective to the primary initiator radicals, resulting the total reaction order is also equal to 2.0. From this study, the exponent order with respect to AIBN concentration could not be determined so that the total reaction order could not be obtained.

The kinetics of thermal-initiated free radical polymerization of a vinyl monomer in the presence of a chain transfer agent were proposed, and the deterministic dynamic predictive model represented such the kinetics were established. Computer programmes were written in Microsoft Quick BASIC computer language for computer simulation experiments. The simulation values of k_p and k_{tr} were evaluated by means of curve fitting of the experimental conversion-time curves of styrene and the chain transfer agents. These values were employed for obtaining simulation MWDs of dead-end polymers, compared to the experimental MWDs. GPC was employed to determine MW and MWD of polymers synthesized. It was found that good agreement was obtained when CCl_4 was the chain transfer agent, while discordance of the results obtained when 1-BuSH was the chain transfer agent. It was because the assumption made on that the kinetic coefficient (k_{tr}) for the transfer reaction of different types of growing active radicals to the chain transfer agent are all equal to one another.

Copyright by Mahidol University

Therefore, the amount of non-existing dead-end polymers contributed to the simulation MWD.

The values of chain transfer constant (C_X) of CCl_4 and 1-BuSH were also obtained by means of typical graphical technique as described by Mayo. The modified Mayo Equation was also established from the kinetic mechanism proposed in this study. The equation can be expressed as :-

$$\frac{1}{DP_n - 1} = \left(\frac{1}{DP_n}\right)_0 + C_m + C_x \frac{[CTA]}{[M]}$$

where

$$\left(\frac{1}{DP_n}\right)_0 = \frac{2k_d f[I]}{k_p[M](P_m + P_r + P_c)} - \frac{k_{tci}[R^*]}{k_p[M]}$$

C_X values of both chain transfer agents were obtained using the modified Mayo Equation. The experimental and simulation C_X values at 2.5 and 5.0 % styrene conversions are summarized in a table shown in the following page. Different C_X values at various percentage of styrene conversion are different especially when C_X has high value. It may be concluded that k_{tr} or k_p is not constant during the polymerization.

The modified Mayo Equation was found suitable for evaluation of C_X values of an effective CTA, while the original Mayo Equation is suitable for less effective CTA. C_X values reported in literature were mostly evaluated from the relationship of [CTA]/[M] with reciprocal of viscosity-average MW (M_V), measured using a technique of very

Copyright by Mahidol University

dilute solution. The method is suitable for polymers having high MW, especially over 100,000 g/mol. For lower MW, the Mark-Houwink-Sakurada Equation is not applicable [52]. The method used in this study is therefore claimed to be the most appropriate method to determine C_X values.

CTA	Experimental				Computer Simulation				Ref.*
	2.5 % styrene conversion		5.0 % styrene conversion		2.5 % styrene conversion		5.0 % styrene conversion		
	(A)	(B)	(A)	(B)	(A)	(B)	(A)	(B)	
CCl_4	0.011	0.011	0.011	0.011	0.007	0.007	0.007	0.007	0.014
1-BuSH	0.422	1.632	0.406	1.547	0.761	1.243	0.729	1.208	15.0

(A) The Mayo Equation, (B) The modified Mayo Equation

*From Brandrup and Immergut Handbook of Polymer, 3rdEdition, 1975

5.2 Suggestions for further work

5.2.1 Determination of C_X values of various chain transfer agent using the modified Mayo Equation proposed in this study in order to confirm its availability.

Copyright by Mahidol University

5.2.2 Investigation of chemical structures of low molecular weight polymers using spectroscopic techniques. Understanding of the polymer chemical structures are useful tool for fuller understanding of the kinetic mechanism.

5.2.3 Mathematical modelling and computer simulation

In this model, there are several errors in assumptions made, such as

(a) assumption was made on that the rate coefficients for propagation, termination and chain transfer reaction were not dependent upon viscosity of the reaction medium and upon the kinetic chain length of polymers.

(b) assumption was made on that k_{tr}'s for the chain transfer reactions of the various types of active growing radicals are equal to one another.

Further exercises on accounting the effect of viscosity of the medium into the mathematical model or on discrepancies of the k_{tr} values for the chain transfer reaction of the particular active growing radicals to the chain transfer agent.

Copyright by Mahidol University

Bibliography

1. Flory P.J., The Mechanism of Vinyl Polymerizations, *J.Am.Chem.Soc.*, **59**,(1937) p.241-253
2. Mayo F.R., *J.Am.Chem.Soc.*, **65**, (1943) p.2324-2329
3. Peterson M.D., Weber A.G., *U.S. Pat. 2,395,292,* (1946)
4. Bamford C.H. and Tompa H., The Calculation of Molecular Weight Distributions from Kinetic Schemes, *Trans.Faraday Soc.*, **50**, (1954) p.1097-1118
5. Scalan J., *J.C.S. Trans.Faraday Soc.*, **52**, (1956) 1286
6. Amundson N.R. and Liu S.-L., Calculation of Molecular Weight Distributions in Polymerazition, *Chemical Engineering Science*, **17**, (1962) p.797-802
7. Tsuchida E., Mimashi S., *J.Polym.Sci., Part A*, **3**, (1965) 1401
8. Detar D.F., Detar D.E., *J.Phys.Chem.*, **70**,)1966) 3842
9. Beoit H., Rempp P. and Grubisic Z., *J. Polymer Sci., **B**, Polymer Letters*, **5**, (1967) p.753
10. Detar D.E., *J.Am.Chem.Soc.*, **89**, (1967) 4058
11. Hamielec A.E. and Hodgins J.W., Polymer Reactors and Molecular Weight Distribution, *AIEChEJournal*, **13,** No.6, (1967) p.1087-1090
12. Amundson N.R. and Luss D., Polymer Molecular Weight Distributions, *J.Macromol.Sci.-Revs.*, **C2**, (1967)p.145-152
13. Fröberg C.-E., Chap.2 and Chap.14, *Introduction to Numerical Analysis*, Addison-Wesley Publishing Company, Inc., London, (1968) p.19-25, 244-247
14. Anthony R., Chap.8, *A First Course in Numerical Analysis*, International Student Edition, McGraw-Hill, Inc. New York, (1970)p.318-350
15. Lowry G.G., *Markov chains and Monte Carlo Caculations in Polymer Science*, Dekker, New York (1970)
16. McCaffery E.L., Exp.18, *Laboratory Preparation for Macromolecular Chemistry*, McGraw-Hill, Inc. New York, (1970)p.201-216.
17. Pryor W.A. and Coco J.H., *Macromolecules*, **3**, (1970)500

Copyright by Mahidol University

18. Ray W.H., On the Mathematical Modeling of Polymerization Reactors, *J.Macromol.Sci.-Revs.Macromol.Chem.*, **C8(1)**, (1972)p.1-56

19. Collins E.A., Bares J., Billmeyer F.W., Chap.1, Chap.3 and Chap.7, *Experiments in Polymer Science*, A Wiley Interscienec Publication, New York, (1973)p.8-14, 38-47, 121-172

20. Starks C.M., *Free Radical Telomerization*, Acdemic Press, New York and London (1974)

21. Brandrup J. and Immergut E.H., Chap.2, *Polymer Handbook*, John Wiley & sons, New York, (1975)p.405-418

22. Deb P.C., Non-Ideal Polymerization, *Europ. Polym. J.*, **11,** (1975)p.31-36

23. Bruns W., Motoc I., O'Driscoll, K.F., *Monte Carlo Applications in Polymer Science*, Lecture Notes in Chemistry Series **27**, Edited by Bertheir G., Dewar M.J.S., Fisher H., Fukui, K., Hall G.G., Hartmann H., Jaffé, H.H., Jortner J., Kutzelnigg W., Ruedenberg K., Scrocco E., Springer-Verlag, New York (1981)

24. Demidovich B.P., Maron I.A., Chap.4, *Computational Mathematics*, MIR Publishers, Moscow, (1981)p.127-131

25. Cutter L.A. and Drexler T.D., *Computer Applications in Applied Polymer Science*, Edited by Theodore Provder, American Chemical Society; Published by ACS Symposium Series **197**, (1982)p.13-24

26. Marten F.L. and Hamielec A.E., High-Conversion Diffusion-Controlled Polymerization of Styrene. I, *J. Appl. Poly. Sci.*, **27**, (1982)p.489-505

27. Armitage P.D., *Computer Simulation of Oligomerisation of Styrene*, MSc. Dissertation, School of Control Engineering, University of Bradford (1983)

28. Barbe W. and Rüchardt C., *Makromol.Chem.*, **184.6**, (1983)p.1235

29. Bamford C.H. and Schofield E., Non-classical free-radical polymerization: 3. Diffusion-control in degradative addition, *Polymer*, **24**, (1983)p.433-438

30. Rintelen T., Riederle K. and Kirchner K., Transfer of Kinetics from Batch to Continuous Reators with Special Regard to the Formation of Oligomers and Effect of Retarders during Styrene Polymerization, *Polymer Reaction Engineering*, Edited by Reichert K.H. and Geiseler W.,Hanser Publishers, New York, (1983)p.175-205

Copyright by Mahidol University

31. Tien N.K., Flaschel E. and Renken A., The Thermal Bulk Polymerization of Styrene in a Tubular Reactor: Reactor Behaviour and Product Properties, *Polymer Reaction Engineering*, Edited by Reichert K.H. and Geiseler W.,Hanser Publishers, New York, (1983)p.175-205

32. Villermaux J., Blavie L. and Pons M., A New Method for Modeling Free Radical Polymerization and Predicting Polymer Quality, *Polymer Reaction Engineering*, Edited by Reichert K.H. and Geiseler W., Hanser Publishers, New York, (1983)p.2-20

33. Reichert K.H. and Geiseler W. (editors), *Polymer Reaction Engineering*, Hanser Publishers, New York (1983)

34. Billmeyer F.W., Chap.3, *Textbook of Polymer Science*, A Wiley Interscienec Publication, New York, (1983)p.49-77

35. Carraher C.E. and Seymour R.B., *Appilied Polymer Science*, ASC Symposium Series 285, 2nd Edition, Tess R.W., Poehlein G.W., Editors, *Am. Chem. Soc.*, **13** (1985)

36. Herman F.M., Norbert M.B., Charles G.O., Menges G., Radical Polymerization, *Encyclopedia of Polymer Science and Engineering,* Vol.**13**, A Willey-Interscience publication, New York, (1985)p.708-867

37. Gosden R.G., Johnson A.F., *Royal Society of Chemistry Microcomputer Group Symposium*, Bradford University, January (1986)

38. Rempp P. and Merrill E.W., Chap.3, *Polymer Synthesis*, Huethig & Wepf Verlag Basel, Heidelberg, New York, (1986)p.69-90

39. Binder K., Editor, *Applications of the Monte Carlo Method in Statistical Physics*, 2nd Edition, Berlin, New York, Springer-Verlag (1987)

40. Galbraith M.N., Moad G., Solomon D.H. and Spurling T.H., Influences of the Initiation and Termination Reactions on the Molecular Weight Distribution and Compositional Heterogeneity of Functional Copolymers: An Application of Monte Carlo Simulation, *Macromolecules*, **20**, (1987)p.675-679

41. Press W.H., Flannery B.P., Teukolsky S.A., Vettering W.T., *Numerical Recipes, the art of scientific computing*, Cambridge University Press (1987)

42. Scraton R.E., *Further Numerical Methods in BASIC*, Edward Arnold (1987)

Copyright by Mahidol University

43. Seymour R.B. and Carraher C.E., Chap.4, *Polymer Chemistry*, Marcel Dekker, Inc., New York, (1987)p.91-121

44 Buback M., Garcia-Rubio L.H., Gilbert R.G., Napper D.H., Guillot J., Hamielec A.E., Hill D., Russell G.T., Hill D.J.T., Moad G., O'Driscoll K.F., Shen J., Olaj O.F., Stickler M., Solomon D., and Winnik M.A., Consistent values of rate parameters in free radical polymerization systems, *J. Polym. Sci., Polymer Letters Edn.*, **26,** (1988)p.293-297

45. Schröder E. and Karl-Friedrich Arndt G.M., Gel Permeation Chromatography and Determination of molecular Weight Distribution, *Polymer Characterization*, Hanser Publishers, New York, (1988)p.166

46. Billingham N.C., Molecular Weight Distribution, *Comprehensive Polymer Science*,**3,** (1989)p.43-57

47. Chapra S.C. and Canale R.P., Chap.5 and Chap.15, *Numerical Method for Engineers*, McGrew-Hill Book Co., Singapore, (1989)p.152-158, 606-608

48. Kumar A. and Khandelwal P.K., Kinetic Modeling of Polymerization Reactions, *Handbook of Polymer Science and Technology*, Vol.**1**, Edited by Cheremisinoff N.P., Marcel Dekker, Inc., New York and Basel, (1989)p.375-426

49. Rudin A., Molecular Weight Distributions, *Comprehensive Polymer Science*, Vol. **3,** Edited by Eastmond G.V., Ledwith A., Russo A., Sigwalt P., Pergamon Press, Great Britain, (1989)p.239-245

50. Steinfeld J.I., Francisco J.S. and Hase W.L., Chap.1- Chap.5, *Chemical Kinetics and Dynamics*, Prentice-Hall, Inc., New York, (1989)p.1-199

51. Boutevin B., Telechelic Oligomers by Radical Reactions, *Advances in Polymer Science*, Vol.**94**, Springer-Verlag Berlin, Heidelberg, Germany, (1990)p.71-102

52. Euaperkiati A., *Free Radical Telmerisation Reactions*, Ph.D. Thesis, Chemistry and Chemical Technology Department of Chemical Engineering, University of Bradford (1990)

53. Norman B.C., Lawrence H.D., Stephen E.W., *Introduction to Infrared and Spectroscopy.*, third edition, Academic Press, Inc., New York (1990).

54. Beati E., Pegoraro M., Gianotti G. and Cassani F., Chain transfer in Vinyl Chloride Polymerization in the Presence of Polybutadiene, *Polymer International*, **24**, (1991)p.39-43

Copyright by Mahidol University

55. Illsley D.R., Lehrle R.S., Shortland A., Polymerization of DEC-1-Ene .1. Basic Free-Radical Mechanisms from Kinetic Studies, *European Polymer J.*, **27**, (1991)p.171-176

56. Meehan E., McConville J.A. and Warner F.P., Recent Developments in the GPC Characterisation of Polymeric Materials, *Polymer International*, **26**, (1991)p.23-28

57. Johansson M., Hult A., Influence of Allyl Ethers on Free-Radical Polymerization of Styrene, *J. Polymer Sci. A Polymer Chemistry*, **29**, (1991)p.9-14

58. Meijs G.F., Rizzardo E. and Tam P.T.Le, New Chain Transfer Agents for Free Radical Polymerizations, *Polymer International*, **26**, (1991)p.239-244

59. Patrick H., Catherine J. and Gavin W., Chap.2-7, *Fourier Transform Raman Spectroscopy*, Ellis Horwood, Singapore, (1991)p.17-174

60. Sylvester M.F., Yip S., Argon A.S., Investigation by Atomistic Simulation of Structural and Dynamic Differences in the Glassy and Liquid States of Atatic Poly(propylene), *Computer Simulation of Polymers*, Edited by Roe R.J., Polymer Science and Engineering Series, Prentice Hall Advanced Reference Series, New Jersey, (1991)p.105-121

61. Sumpter B.G., Noid D.W., Wunderlich B., Cheng S.Z.D., Computer Simulations of Structural and Dynamical Properties of Polymers in the Solid States, *Computer Simulation of Polymers*, Edited by Roe R.J., Polymer Science and Engineering Series, Prentice Hall Advanced Reference Series, New Jersey, (1991)p.311-330

62. Yamada B., Kageoka M. and Otsu T., Dependence of Propagation and Termination Rate Constants on Conversion for the Radical Polymerization of Styrene in Bulk As Studied by ESR Spectroscopy, *Macromolecules*, **24**, (1991)p.5234-5236

63. Yuan H.G., Kalfas G. and Ray W.H., Suspension Polymerization, *J. Macromol. Sci-Revs. Macromol. Chem. Phys.*, **C31**(283), (1991)p.215-299

64. Buback M., Gilbert R.G., Russell G.T., Hill D.J.T., Moad G., Odriscoll K.F., Shen J., Winnik M.A., Consistent Values of Rate Parameters in Free-Radical Polymerization Systems .2. Outstanding Dilemmas and Recommendations, *J. Polymer Sci. A Polymer Chemistry*, **30**, (1992)p.851-863

Copyright by Mahidol University

65. Butala D.N., Liang W.R. and Choi K.Y., Multiobjective Dynamic Optimization of Batch Free Radical Polymerization Process Catalyzed by Mixed Initiator Systems, *J. Appl. Poly. Sci.*, **44**, (1992)p.1759-1778

66. Damoun S., Papin R., Ripault G., Rousseau M., Rabadeux J.C., Durand D., Radical Polymerization of Methyl Methacrylate in Solution Monitored and Studied by Raman Spectroscopy, *Journal of Raman Spectroscopy*, **23**, (1992)p.385-389

67. Gilbert R.G., Consistent Values of Rate Parameters in Free-Radical Polymerization Systems, *Pure and Applied Chemistry*, **64**, (1992)p.1563-1567

68. Madruga E.L., Malfeito J.J., Effect of Chain Transfer Agent and the Reaction Medium on the Limiting Conversion of Methyl Methacrylate Free Radical Polymerization, *European Polymer J.*, **28**, (1992)p.863-866

69. Roland P., Chung T. and Solomon D.H., Recent Developments in Free Radical Polymerization, *Progress in Pacific Polymer Science*, Vol.**2**, Edited by Imanishi Y., Springer-Verlag Berlin, Heidelberg, Germany, (1992)p.13-25

70. Venkateshwaran G. and Kumar A., Solution of Free Radical Polymerization, *J. Appl. Poly. Sci.*, **45**, (1992)p.187-215

71. Xia P., Cheng H. and Yan D., Free-Radical Polymerization of Methyl Methacrylate Initiated by Thiol Alone, *J. Appl. Poly. Sci.*, **45**, (1992)p.579-582

72. Yan D., Lin J. and Yuan C., Non-Steady-State Model for Kinetics of Free Radical Polymerization: 4. Direct Thermal Initiation, *Polymer International*, **28**, (1992)p.183-188

73. Yan D.Y., Feng J., Nonsteady State Model for Free-Radical Polymerization, *Makromolekulare Chemie-Macromolecular Symposia*, **63**, (1992)p.189-208

74. Yoon W.J. and Choi K.Y., Free-Radical Polymerization of Styrene with a Binary Mixture of Symmetrical Bifunctional Initiators, *J.Appl.Poly.Sci.*, **46**, (1992)p.1353-1367

75. Braun D., Hempler P., Chain Transfer Constants of Some Aliphatic Agents in the Polymerization of Styrene and Methyl methacrylate, *Poly.Bull.*, **30**, (1993)p.55-60

76. Dais V.A., Priddy D.B., Bell B., Sikkema K.D., Smith P., Investigation of the Free Radical Polymerization of Styrene in the Presence of Several 1,1-Disubstituted Ethylenes, *J. Polymer Sci. A Polymer Chemistry*, **31**, (1993)p.901-908

Copyright by Mahidol University

77. Tobita H., Hamielec A.E., Cross Linking Kinetics in Emulsion Polymerization, *Polymer International*, **30**, (1993)p.177-183

78. Villalobos M.A., Hamielec A.E. and Wood P.E., Bulk and Suspension Polymerization of Styrene in the Presence of *n*-Pentane. An Evaluation of Monofunctional and Bifunctional Initiation, *J. Appl. Poly.Sci.*, **50**, (1993)p.327-343.

Copyright by Mahidol University

Appendix I

Mathematical Modelling of the Kinetics of Thermal-Initiated Free Radical Polymerization Reaction of a Vinyl Monomer in the Presence of a Chain Transfer Agent

Material balances of various species in the kinetic mechanism of thermal-initiated free radical polymerization reaction of a vinyl monomer in the presence of a chain transfer agent presented in Section 4.3, Chapter IV are shown below.

1) Material balances for initiator and chain transfer agent radicals (R*, CTA*)

From reactions (R-1), (R-2), (R-4), (R-23), (R-24), (R-25) and reactions (R-2), (R-3), (R-26), (R-27), (R-28), the materials balances for R* and CTA* species are shown in Equations (AI-1) and (AI-2), respectively.

$$\frac{d}{dt}[R^*]=2k_d f[I] - k_{sr}[R^*][CTA] - k_i[R^*][M]-k_{tci}[R^*]\{P_m+P_r+P_c\} \quad \text{(AI-1)}$$

$$\frac{d}{dt}[CTA^*]=k_{sr}[R^*][CTA] + k_{tr}[CTA]\{P_m+P_r+P_c\}-k_{i,CTA}[CTA^*][M] \quad \text{(AI-2)}$$

2) Material balances for growing polymer radicals, P

P is here abbreviated for the growing polymer radicals or the growing active radical concentrations. An alphabet in the bracket ( ) attached to P indicates types of growing polymer radicals, and the subscription indicate chain length of the growing polymer radicals.

Copyright by Mahidol University

P_X is here abbreviated for the summation of concentrations of growing polymer radicals with the types x. For the growing polymer radicals of the types RM_n^*,

$CTAM_n^*$, M_n^*, the summation of the individual growing polymer radical concentrations are P_r, P_c, and P_m, respectively.

$$\frac{d}{dt}P(r)_1=k_i[R^*][M] - P(r)_1\{k_p[M] + 0.5k_{tc}P_r + k_{tc}P_c + k_{tc}P_m +$$

$$k_{td}P_r + k_{td}P_c + k_{td}P_m + k_{tr}[CTA] + k_{trm}[M] + k_{tci}[R^*]\} \quad \text{(AI-3)}$$

$$\frac{d}{dt}P(r)_n=k_p[M]P(r)_{n-1} - P(r)_n\{k_p[M] + 0.5k_{tc}P_r + k_{tc}P_c + k_{tc}P_m +$$

$$k_{td}P_r + k_{td}P_c + k_{td}P_m + k_{tr}[CTA] + k_{trm}[M] + k_{tci}[R^*]\} \quad \text{(AI-4)}$$

$$\frac{d}{dt}P_r =k_i[R^*][M] - P_r\{0.5k_{tc}P_r + k_{tc}P_c + k_{tc}P_m + k_{td}P_r + k_{td}P_c +$$

$$k_{td}P_m + k_{tr}[CTA] + k_{trm}[M] + k_{tci}[R^*]\} \quad \text{(AI-5)}$$

$$\frac{d}{dt}P(c)_1=k_{i,CTA}[CTA^*][M] - P(c)_1\{k_p[M] + 0.5k_{tc}P_c + k_{tc}P_r +$$

$$k_{tc}P_m + k_{td}P_r + k_{td}P_c + k_{td}P_m + k_{tr}[CTA] + k_{trm}[M] + k_{tci}[R^*]\} \quad \text{(AI-6)}$$

$$\frac{d}{dt}P(c)_n=k_p[M]P(c)_{n-1} - P(c)_n\{k_p[M] + 0.5k_{tc}P_c + k_{tc}P_r + k_{tc}P_m +$$

$$k_{td}P_r + k_{td}P_c + k_{td}P_m + k_{tr}[CTA] + k_{trm}[M] + k_{tci}[R^*]\} \quad \text{(AI-7)}$$

$$\frac{d}{dt}P_c =k_{i,CTA}[CTA^*][M] - P_c\{0.5k_{tc}P_c + k_{tc}P_r + k_{tc}P_m + k_{td}P_r$$

$$+k_{td}P_c + k_{td}P_m + k_{tr}[CTA] + k_{trm}[M] + k_{tci}[R^*]\} \quad \text{(AI-8)}$$

$$\frac{d}{dt}P(m)_1=k_{trm}[M]\{P_m+P_r+P_c\} - P(m)_1\{k_p[M] + 0.5k_{tc}P_m + k_{tc}P_r +$$

$$k_{tc}P_c + k_{td}P_r + k_{td}P_c + k_{td}P_m + k_{tr}[CTA] + k_{trm}[M] + k_{tci}[R^*]\} \quad \text{(AI-9)}$$

$$\frac{d}{dt}P(m)_n=k_p[M]P(m)_{n-1} - P(m)_n\{k_p[M] + 0.5k_{tc}P_m + k_{tc}P_r + k_{tc}P_c +$$

$$k_{td}P_r + k_{td}P_c + k_{td}P_m + k_{tr}[CTA] + k_{trm}[M] + k_{tci}[R^*]\} \quad \text{(AI-10)}$$

$$\frac{d}{dt}P_m =k_{trm}[M]\{P_m+P_r+P_c\} - P_m\{0.5k_{tc}P_m + k_{tc}P_r + k_{tc}P_c + k_{td}P_r$$

$$+ k_{td}P_c + k_{td}P_m + k_{tr}[CTA] + k_{trm}[M] + k_{tci}[R^*]\} \quad \text{(AI-11)}$$

Copyright by Mahidol University

3) Material balances for monomer, M

Material balances for the monomer concentration (Equation (AI-12)) can be obtained from Reactions (R-3), (R-4), (R-5), (R-6), (R-7), (R-29), (R-30), and (R-31).

$$-\frac{d}{dt}[M]=k_i[R^*][M] + k_{i,CTA}[CTA^*][M]+k_p[M]\{P_m+P_r+P_c\}+ k_{trm}[M]\{P_m+P_r+P_c\} \quad \text{(AI-12)}$$

4) Material balances for chain transfer agents, CTA

From reactions (R-2), (R-26), (R-27), (R-28), the material balance for CTA can be written as shown in equation (AI-13).

$$-\frac{d}{dt}[CTA] = k_{sr}[R^*][CTA]+k_{tr}[CTA]\{P_m+P_r+P_c\} \quad \text{(AI-13)}$$

5) Material balances for dead-end polymers (Inactive Polymer)

The termination and chain transfer reactions result in inactive polymer chain or dead-end polymers. The material balances for the dead-end polymers can be obtained from Reactions (R-5)-(R-31), and these are shown in (AI-14)-(AI-22).

$$\frac{d}{dt}[M_n]=P(m)_n\{k_{trm}[M]+k_{td}(P_m+P_c+P_r)\} +0.5k_{tc}\sum_{i=1}^{n-1}P(m)_i P(m)_{n-i} \quad \text{(AI-14)}$$

$$\frac{d}{dt}[M_nX] = k_{tr}[CTA]P(m)_n \quad \text{(AI-15)}$$

$$\frac{d}{dt}[RM_n] = P(r)_n\{k_{trm}[M]+k_{td}(P_m+P_c+P_r)\} +k_{tc}\sum_{i=1}^{n-1}P(r)_i P(m)_{n-i} + k_{tci}[R^*]P(m)_n \quad \text{(AI-16)}$$

Copyright by Mahidol University

$$\frac{d}{dt}[RM_nX] = k_{tr}[CTA]P(r)_n \quad \text{(AI-17)}$$

$$\frac{d}{dt}[RM_nR] = 0.5k_{tc}\sum_{i=1}^{n-1}P(r)_iP(r)_{n-i} + k_{tci}[R^*]P(r)_n \quad \text{(AI-18)}$$

$$\frac{d}{dt}[CTAM_n] = P(c)_n\{k_{trm}[M]+k_{td}(P_m+P_c+P_r)\}+k_{tc}\sum_{i=1}^{n-1}P(c)_iP(m)_{n-i} \quad \text{(AI-19)}$$

$$\frac{d}{dt}[CTAM_nX] = k_{tr}[CTA]P(c)_n \quad \text{(AI-20)}$$

$$\frac{d}{dt}[CTAM_nR] = k_{tc}\sum_{i=1}^{n-1}P(r)_iP(c)_{n-i} + k_{tci}[R^*]P(c)_n \quad \text{(AI-21)}$$

$$\frac{d}{dt}[CTAM_nCTA] = 0.5k_{tc}\sum_{i=1}^{n-1}P(c)_iP(c)_{n-i} \quad \text{(AI-22)}$$

A chemical reaction reaches the stationary state when the rate of change of radical intermediates is much smaller than that of radical formation or that of radical destruction. Let R_i be the overall rate for the radical formation and R_d be the overall rate for the radical destruction. At the stationary state, R_i be equal to R_d and d[Radical]/dt = 0. Let $\varphi = 2k_d f[I]$, thus Equation (AI-1) can be rewritten as :-

$$[R^*] = \frac{\varphi}{k_i[M] + k_{sr}[CTA] + k_{tci}(P_m + P_r + P_c)} \quad \text{(AI-23)}$$

$$\beta_{sr} = \beta_o k_{sr}[CTA] \quad \text{(AI-24)}$$

$$\beta_i = \beta_o k_i[M] \quad \text{(AI-25)}$$

$$\beta_{tci} = \beta_o k_{tci}(P_m + P_r + P_c) \quad \text{(AI-26)}$$

From Equations (AI-4), (AI-7) and (AI-10), propagation probability (α) of three types of growing polymer radicals RM_n^*, $CTAM_n^*$, and M_n^* (abbreviated for α_m, α_r and α_c, respectively) can be determined from Equations (AI-27) ,(AI-28) and (AI-29), respectively.

Copyright by Mahidol University

$$\alpha_m = \frac{k_p[M]}{\chi_m} \quad \text{(AI-27)}$$

$$\alpha_r = \frac{k_p[M]}{\chi_r} \quad \text{(AI-28)}$$

$$\alpha_c = \frac{k_p[M]}{\chi_c} \quad \text{(AI-29)}$$

where

$$\chi_m = k_p[M] + 0.5k_{tc}P_m + k_{tc}P_r + k_{tc}P_c + k_{td}P_r + k_{td}P_c + k_{td}P_m + k_{tr}[CTA] + k_{trm}[M] + k_{tci}[R^*] \quad \text{(AI-30)}$$

$$\chi_r = k_p[M] + 0.5k_{tc}P_r + k_{tc}P_m + k_{tc}P_c + k_{td}P_r + k_{td}P_c + k_{td}P_m + k_{tr}[CTA] + k_{trm}[M] + k_{tci}[R^*] \quad \text{(AI-31)}$$

$$\chi_c = k_p[M] + 0.5k_{tc}P_c + k_{tc}P_r + k_{tc}P_m + k_{td}P_r + k_{td}P_c + k_{td}P_m + k_{tr}[CTA] + k_{trm}[M] + k_{tci}[R^*] \quad \text{(AI-32)}$$

$P(r)_1$, $P(c)_1$, $P(m)_1$, $P(r)_n$, $P(c)_n$ and $P(m)_n$ can be written and shown below.

$$P(m)_1 = \frac{1}{\chi_m} k_{trm}[M]\{P_m + P_r + P_c\} \quad \text{(AI-33)}$$

$$P(r)_1 = \frac{1}{\chi_r} k_i[R^*][M] \quad \text{(AI-34)}$$

$$P(c)_1 = \frac{1}{\chi_c} k_{i,CTA}[CTA^*][M] \quad \text{(AI-35)}$$

$$P(m)_n = P(m)_1 \alpha_m^{n-1} \quad \text{(AI-36)}$$

$$P(r)_n = P(r)_1 \alpha_r^{n-1} \quad \text{(AI-37)}$$

$$P(c)_n = P(c)_1 \alpha_c^{n-1} \quad \text{(AI-38)}$$

Substitutions of $P(r)_1$, $P(c)_1$, $P(m)_1$, $P(r)_n$, $P(c)_n$, $P(m)_n$, P_r, P_c, and P_m into Equations (AI-14)-(AI-22), Equations (AI-39)-(AI-47), where $n \geq 2$, are obtained.

$$\frac{d}{dt}[M_n] = P(m)_1 \alpha_m^{n-1} \{k_{trm}[M] + k_{td}(P_m + P_c + P_r)\} + 0.5 k_{tc} P(m)_1^2 \alpha_m^{n-2}(n-1) \quad \text{(AI-39)}$$

$$\frac{d}{dt}[M_n X] = k_{tr}[CTA]P(m)_1 \alpha_m^{n-1} \quad \text{(AI-40)}$$

$$\frac{d}{dt}[RM_n] = P(r)_1 \alpha_r^{n-1} \{k_{trm}[M] + k_{td}(P_m + P_c + P_r)\} + k_{tci}[R^*]P(m)_1 \alpha_m^{n-1} + k_{tc} \frac{P(r)_1 P(m)_1}{\alpha_r \alpha_m} \psi_{rm} \quad \text{(AI-41)}$$

Copyright by Mahidol University

$$\frac{d}{dt}[RM_nX] = k_{tr}[CTA]P(r)_1\alpha_r^{n-1} \quad \text{(AI-42)}$$

$$\frac{d}{dt}[RM_nR] = 0.5k_{tc}P(r)_1^2\alpha_r^{n-2}(n-1) + k_{tci}[R^*]P(r)_1\alpha_r^{n-1} \quad \text{(AI-43)}$$

$$\frac{d}{dt}[CTAM_n] = P(c)_1\alpha_c^{n-1}\{k_{trm}[M]+k_{td}(P_m+P_c+P_r)\}+k_{tc}\frac{P(c)_1P(m)_1}{\alpha_c\alpha_m}\psi_{cm} \quad \text{(AI-44)}$$

$$\frac{d}{dt}[CTAM_nX] = k_{tr}[CTA]P(c)_1\alpha_c^{n-1} \quad \text{(AI-45)}$$

$$\frac{d}{dt}[CTAM_nR] = k_{tci}[R^*]P(c)_1\alpha_c^{n-1} + k_{tc}\frac{P(c)_1P(r)_1}{\alpha_c\alpha_r}\psi_{rc} \quad \text{(AI-46)}$$

$$\frac{d}{dt}[CTAM_nCTA] = 0.5k_{tc}P(c)_1^2\alpha_c^{n-2}(n-1) \quad \text{(AI-47)}$$

Determination of weight fraction of dead-end polymers

Let D represent one of the following :- $[M_n]$, $[M_nX]$, $[RM_n]$, $[RM_nX]$, $[RM_nR]$, $[CTAM_n]$, $[CTAM_nX]$, $[CTAM_nCTA]$ or $[CTAM_nR]$), and let $\Delta[M] = [M]_o-[M]_t$. The weight fraction $W_n(D)$ of dead-end polymers (D) of given chain length (n) can be calculated from Equation (AI-48).

$$W_n(D) = \frac{1}{\Delta[M]}nD_n \quad \text{(AI-48)}$$

Determination of the number- and weight-average molecular weight (M_n and M_w) of dead-end polymers by means of the generating function

In this work, the r-th moments (Q_r) is defined as :-

$$Q_r = \sum_{n=2}^{n} n^r D_n \text{ ,where } r = 0, 1, 2, 3, \ldots \quad \text{(AI-49)}$$

Copyright by Mahidol University

For the zeroth moment, r = 0 and therefore Equation (AI-49) is rewritten as :-

$$Q_0 = \sum_{n=2}^{n} D_n \quad \text{(AI-50).}$$

For the first moment, r = 1, and Equation (AI-49) is rewritten as :-

$$Q_1 = \sum_{n=2}^{n} nD_n \qquad \text{(AI-51)}$$

For the second moment, r = 2, and Equation (AI-49) is rewritten as :-

$$Q_2 = \sum_{n=2}^{n} n^2 D_n \qquad \text{(AI-52)}$$

M_n and M_W relate to the r-th moment as shown in Equations (AI-53) and (AI-54), respectively (18).

$$M_n = MW\frac{Q_1}{Q_0} \qquad \text{or } M_n = MW.DP_n \qquad \text{(AI-53)}$$

and

$$M_w = MW\frac{Q_2}{Q_1} \qquad \text{or } M_w = MW.DP_w \qquad \text{(AI-54)}$$

where MW is the molecular weight of the monomer. The ratio Q_1/Q_0 in Equation (AI-53) and Q_2/Q_1 in Equation (AI-54) is equal to DP_n (the number-average degree of polymerization) and DP_W (the weight average degree of polymerization), respectively.The r-th moment of dead-end polymers (D) can be evaluated from Equation (AI-55).

$$Q_r = \int_0^t \sum_{n=2}^{n} n^r \frac{d[D_n]}{dt}dt \qquad \text{(AI-55)}$$

1) For M_n species ($D = M_n$)

Let $\Gamma_1 = k_{trm}[M]+k_{td}(P_m+P_c+P_r)$ and $\Gamma_2 = 0.5k_{tc}$ Substitution of Γ_1 and Γ_2 into Equation (AI-39) yields Equation (AI-56).

$$\frac{d}{dt}[M_n] = \Gamma_1 P(m)_1 \alpha_m^{n-1} + \Gamma_2 P(m)_1^2 \alpha_m^{n-2}(n-1) \qquad \text{(AI-56)}$$

Copyright by Mahidol University

Substitution of $\frac{d}{dt}[M_n]$ into Equation (A1-55) to obtain the zeroth, the first, and the second moments (Equations (AI-57), (AI-58), and (AI-59), respectively) by integration of the differentiation term in Equation (AI-55).

$$Q_0 = \int_0^t \left(\Gamma_1 \frac{P(m)_1 \alpha_m}{1-\alpha_m} + \Gamma_2 \frac{P(m)_1^2}{(1-\alpha_m)^2} \right) dt \qquad \text{(AI-57)}$$

$$Q_1 = \int_0^t \left(\Gamma_1 \frac{P(m)_1 (2-\alpha_m)\alpha_m}{(1-\alpha_m)^2} + \Gamma_2 \frac{2P(m)_1^2}{(1-\alpha_m)^3} \right) dt \qquad \text{(AI-58)}$$

$$Q_2 = \int_0^t \left(\Gamma_1 \frac{P(m)_1 (4-3\alpha_m+\alpha_m^2)\alpha_m}{(1-\alpha_m)^3} + \Gamma_2 \frac{2(2+\alpha_m)P(m)_1^2}{(1-\alpha_m)^4} \right) dt \qquad \text{(AI-59)}$$

2) For M_nX species ($D = M_nX$)

Let $\Gamma_3 = k_{tr}[CTA]$, and substitution of Γ_3 into Equation (AI-40) gives Equation (AI-60).

$$\frac{d}{dt}[M_nX] = \Gamma_3 P(m)_1 \alpha_m^{n-1} \qquad \text{(AI-60)}$$

Substitution of $\frac{d}{dt}[M_nX]$ into Equation (AI-55) to obtain the zeroth, the first, and the second moments (Equations (AI-61), (AI-62), and (AI-63), respectively) by integration of the differentiation term in Equation (AI-55).

$$Q_0 = \int_0^t \left(\Gamma_3 \frac{P(m)_1 \alpha_m}{1-\alpha_m} \right) dt \qquad \text{(AI-61)}$$

$$Q_1 = \int_0^t \left(\Gamma_3 \frac{P(m)_1 (2-\alpha_m)\alpha_m}{(1-\alpha_m)^2} \right) dt \qquad \text{(AI-62)}$$

$$Q_2 = \int_0^t \left(\Gamma_3 \frac{P(m)_1 (4-3\alpha_m+\alpha_m^2)\alpha_m}{(1-\alpha_m)^3} \right) dt \qquad \text{(AI-63)}$$

Copyright by Mahidol University

3) For RM_n species ($D = RM_n$)

Let $\Gamma_4 = k_{tci}[R^*]$, $\Gamma_5 = k_{trm}[M]+k_{td}(P_m+P_c+P_r)$ and $\Gamma_6 = k_{tc}\frac{P(r)_1 P(m)_1}{\alpha_r \alpha_m}$

and substitution of Γ_4, Γ_5 and Γ_6 into Equation (AI-41) gives Equation (AI-64).

$$\frac{d}{dt}[RM_n] = \Gamma_4 P(m)_1 \alpha_m^{n-1} + \Gamma_5 P(r)_1 \alpha_r^{n-1} + \Gamma_6 \psi_{rm} \quad \text{(AI-64).}$$

Substitution of $\frac{d}{dt}[RM_n]$ into Equation (AI-55) to obtain the zeroth, the first, and the second moments (Equations (AI-65), (AI-66), and (AI-67), respectively) by integration of the differentiation term in Equation (AI-55).

$$Q_0 = \int_0^t \left(\Gamma_4 \frac{P(m)_1 \alpha_m}{1-\alpha_m} + \Gamma_5 \frac{P(r)_1 \alpha_r}{1-\alpha_r} + \Gamma_6 \sum_{n-2}^{n} \psi_{rm} \right) dt \quad \text{(AI-65)}$$

$$Q_1 = \int_0^t \left(\Gamma_4 \frac{P(m)_1 (2-\alpha_m)\alpha_m}{(1-\alpha_m)^2} + \Gamma_5 \frac{P(r)_1 (2-\alpha_r)\alpha_r}{(1-\alpha_r)^2} + \Gamma_6 \sum_{n-2}^{n} n\psi_{rm} \right) dt \quad \text{(AI-66)}$$

$$Q_2 = \int_0^t \left(\Gamma_4 \frac{P(m)_1 (4-3\alpha_m+\alpha_m{}^2)\alpha_m}{(1-\alpha_m)^3} + \Gamma_5 \frac{P(r)_1 (4-3\alpha_r+\alpha_r{}^2)\alpha_r}{(1-\alpha_r)^2} \right) dt + \int_0^t \Gamma_6 \sum_{n-2}^{n} n^2 \psi_{rm} dt \quad \text{(AI-67)}$$

where

$$\sum_{n=2}^{n} \psi_{rm} = \begin{cases} \frac{\alpha_r^2}{1-\alpha_r}\left[\frac{R_{m/r}}{1-R_{m/r}}\right] & ;\alpha_m < \alpha_r \\ \frac{\alpha_m^2}{1-\alpha_m}\left[\frac{R_{m/r}}{1-R_{m/r}}\right] & ;\alpha_r < \alpha_m \\ \frac{\alpha_m^2}{(1-\alpha_m)^2} & ;\alpha_r = \alpha_m \end{cases}$$

$$\sum_{n=2}^{n} n\psi_{rm} = \begin{cases} \frac{(2-\alpha_r)\alpha_r^2}{(1-\alpha_r)^2}\left[\frac{R_{m/r}}{1-R_{m/r}}\right] & ;\alpha_m < \alpha_r \\ \frac{(2-\alpha_m)\alpha_m^2}{(1-\alpha_m)^2}\left[\frac{R_{m/r}}{1-R_{m/r}}\right] & ;\alpha_r < \alpha_m \\ \frac{2\alpha_m^2}{(1-\alpha_m)^3} & ;\alpha_r = \alpha_m \end{cases}$$

Copyright by Mahidol University

$$\sum_{n=2}^{n} n^2 \psi_{rm} = \begin{cases} \dfrac{(4-3\alpha_r+\alpha_r^2)\alpha_r^2}{(1-\alpha_r)^3}\left[\dfrac{R_{m/r}}{1-R_{m/r}}\right] & ;\alpha_m < \alpha_r \\ \dfrac{(4-3\alpha_m+\alpha_m^2)\alpha_m^2}{(1-\alpha_m)^3}\left[\dfrac{R_{m/r}}{1-R_{m/r}}\right] & ;\alpha_r < \alpha_m \\ \dfrac{2(2+\alpha_m)\alpha_m^2}{(1-\alpha_m)^4} & ;\alpha_r = \alpha_m \end{cases}$$

4) For RM_nX species (D = RM_nX)

Let $\Gamma_7 = k_{tr}[CTA]$ and substitution of Γ_7 into Equation (AI-42) gives Equation (AI-68).

$$\frac{d}{dt}[RM_nX] = \Gamma_7 P(r)_1 \alpha_r^{n-1} \qquad \text{(AI-68)}$$

Substitution of $\frac{d}{dt}[RM_nX]$ into Equation (AI-55) to obtain the zeroth, the first, and the second moments (Equations (AI-69), (AI-70), and (AI-71), respectively) by integration of the differentiation term in Equation (AI-55).

$$Q_0 = \int_0^t \left(\Gamma_7 \frac{P(r)_1 \alpha_r}{1-\alpha_r} \right) dt \qquad \text{(AI-69)}$$

$$Q_1 = \int_0^t \left(\Gamma_7 \frac{P(r)_1 (2-\alpha_r)\alpha_r}{(1-\alpha_r)^2} \right) dt \qquad \text{(AI-70)}$$

$$Q_2 = \int_0^t \left(\Gamma_7 \frac{P(r)_1 (4-3\alpha_r+\alpha_r{}^2)\alpha_r}{(1-\alpha_r)^3} \right) dt \qquad \text{(AI-71)}$$

5) For RM_nR species (D = RM_nR)

Let $\Gamma_8 = k_{tci}[R^*]$ and $\Gamma_9 = 0.5k_{tc}$ and substitution of Γ_8 and Γ_9 into Equation (AI-43) gives Equation (AI-72).

Copyright by Mahidol University

$$\frac{d}{dt}[RM_nR] = \Gamma_8 P(r)_1 \alpha_r^{n-1} + \Gamma_9 P(r)_1{}^2 \alpha_r^{n-2}(n-1) \qquad \text{(AI-72)}$$

Substitution of $\frac{d}{dt}[RM_nR]$ into Equation (AI-55) to obtain the zeroth, the first, and the second moments (Equations (AI-73), (AI-74), and (AI-75), respectively) by integration of the differentiation term in Equation (AI-55).

$$Q_0 = \int_0^t \left(\Gamma_8 \frac{P(r)_1 \alpha_r}{1-\alpha_r} + \Gamma_9 \frac{P(r)_1^2}{(1-\alpha_r)^2} \right) dt \qquad \text{(AI-73)}$$

$$Q_1 = \int_0^t \left(\Gamma_8 \frac{P(r)_1 (2-\alpha_r)\alpha_r}{(1-\alpha_r)^2} + \Gamma_9 \frac{2P(r)_1^2}{(1-\alpha_r)^3} \right) dt \qquad \text{(AI-74)}$$

$$Q_2 = \int_0^t \left(\Gamma_8 \frac{P(r)_1 (4-3\alpha_r+\alpha_r^2)\alpha_r}{(1-\alpha_r)^3} + \Gamma_9 \frac{2(2+\alpha_r)P(r)_1^2}{(1-\alpha_r)^4} \right) dt \qquad \text{(AI-75)}$$

6) For $CTAM_n$ species (D = $CTAM_n$)

Let $\Gamma_{10} = k_{trm}[M]+k_{td}(P_m+P_c+P_r)$ and $\Gamma_{11} = k_{tc}\frac{P(c)_1 P(m)_1}{\alpha_c \alpha_m}$ and substitution of Γ_{10} and Γ_{11} into Equation (AI-44) gives Equation (AI-76).

$$\frac{d}{dt}[CTAM_n] = \Gamma_{10} P(c)_1 \alpha_c^{n-1} + \Gamma_{11} \psi_{cm} \qquad \text{(AI-76)}$$

Substitution of $\frac{d}{dt}[CTAM_n]$ into Equation (AI-55) to obtain the zeroth, the first, and the second moments (Equations (AI-77), (AI-78), and (AI-79), respectively) by integration of the differentiation term in Equation (AI-55).

$$Q_0 = \int_0^t \left(\Gamma_{10} \frac{P(c)_1 \alpha_c}{1-\alpha_c} + \Gamma_{11} \sum_{n-2}^{n} \psi_{cm} \right) dt \qquad \text{(AI-77)}$$

$$Q_1 = \int_0^t \left(\Gamma_{10} \frac{P(c)_1 (2-\alpha_c)\alpha_c}{(1-\alpha_c)^2} + \Gamma_{11} \sum_{n-2}^{n} n\psi_{cm} \right) dt \qquad \text{(AI-78)}$$

$$Q_2 = \int_0^t \left(\Gamma_{10} \frac{P(c)_1 (4-3\alpha_c+\alpha_c^2)\alpha_c}{(1-\alpha_c)^3} + \Gamma_{11} \sum_{n=2}^{n} n^2 \psi_{cm} \right) dt \qquad \text{(AI-79)}$$

Copyright by Mahidol University

where

$$\sum_{n=2}^{n}\psi_{cm}=\begin{cases}\dfrac{\alpha_c^2}{1-\alpha_c}\left[\dfrac{R_{m/c}}{1-R_{m/c}}\right] & ;\alpha_m<\alpha_c\\ \dfrac{\alpha_m^2}{1-\alpha_m}\left[\dfrac{R_{m/c}}{1-R_{m/c}}\right] & ;\alpha_c<\alpha_m\\ \dfrac{\alpha_m^2}{(1-\alpha_m)^2} & ;\alpha_c=\alpha_m\end{cases}$$

$$\sum_{n=2}^{n}n\psi_{cm}=\begin{cases}\dfrac{(2-\alpha_c)\alpha_c^2}{(1-\alpha_c)^2}\left[\dfrac{R_{m/c}}{1-R_{m/c}}\right] & ;\alpha_m<\alpha_c\\ \dfrac{(2-\alpha_m)\alpha_m^2}{(1-\alpha_m)^2}\left[\dfrac{R_{m/c}}{1-R_{m/c}}\right] & ;\alpha_c<\alpha_m\\ \dfrac{2\alpha_m^2}{(1-\alpha_m)^3} & ;\alpha_c=\alpha_m\end{cases}$$

$$\sum_{n=2}^{n}n^2\psi_{cm}=\begin{cases}\dfrac{(4-3\alpha_c+\alpha_c^2)\alpha_c^2}{(1-\alpha_c)^3}\left[\dfrac{R_{m/c}}{1-R_{m/c}}\right] & ;\alpha_m<\alpha_c\\ \dfrac{(4-3\alpha_m+\alpha_m^2)\alpha_m^2}{(1-\alpha_m)^3}\left[\dfrac{R_{m/c}}{1-R_{m/c}}\right] & ;\alpha_c<\alpha_m\\ \dfrac{2(2+\alpha_m)\alpha_m^2}{(1-\alpha_m)^4} & ;\alpha_c=\alpha_m\end{cases}$$

7) For $CTAM_nX$ species (D = $CTAM_nX$)

Let $\Gamma_{12} = k_{tr}[CTA]$ and substitution of Γ_{12} into Equation (AI-45) gives Equation (AI-80).

$$\frac{d}{dt}[CTAM_nX] = \Gamma_{12}P(c)1\alpha_c^{n-1} \qquad \text{(AI-80)}.$$

Copyright by Mahidol University

Substitution of $\frac{d}{dt}[CTAM_nX]$ into Equation (AI-55) to obtain the zeroth, the first, and the second moments (Equations (AI-81), (AI-82), and (AI-83), respectively) by integration of the differentiation term in Equation (AI-55)

$$Q_0 = \int_0^t \left(\Gamma_{12} \frac{P(c)_1 \alpha_c}{1-\alpha_c} \right) dt \qquad \text{(AI-81)}$$

$$Q_1 = \int_0^t \left(\Gamma_{12} \frac{P(c)_1 (2-\alpha_c)\alpha_c}{(1-\alpha_c)^2} \right) dt \qquad \text{(AI-82)}$$

$$Q_2 = \int_0^t \left(\Gamma_{12} \frac{P(c)_1 (4-3\alpha_c+\alpha_c{}^2)\alpha_c}{(1-\alpha_c)^3} \right) dt \qquad \text{(AI-83)}$$

8) For $CTAM_nCTA$ species (D = $CTAM_nCTA$)

Let $\Gamma_{13} = 0.5k_{tc}$ and substitution of Γ_{13} into Equation (AI-46) gives Equation (AI-84).

$$\frac{d}{dt}[CTAM_nCTA] = \Gamma_{13} P(c)_1{}^2 \alpha_c{}^{n-2}(n-1) \qquad \text{(AI-84)}$$

Substitution of $\frac{d}{dt}[CTAM_nCTA]$ into Equation (AI-55) to obtain the zeroth, the first, and the second moments (Equations (AI-85), (AI-86), and (AI-87), respectively) by integration of the differentiation term in Equation (AI-55).

$$Q_0 = \int_0^t \left(\Gamma_{13} \frac{P(c)_1{}^2}{(1-\alpha_c)^2} \right) dt \qquad \text{(AI-85)}$$

$$Q_1 = \int_0^t \left(\Gamma_{13} \frac{2P(c)_1{}^2}{(1-\alpha_c)^3} \right) dt \qquad \text{(AI-86)}$$

$$Q_2 = \int_0^t \left(\Gamma_{13} \frac{2(2+\alpha_c)P(c)_1{}^2}{(1-\alpha_c)^4} \right) dt \qquad \text{(AI-87)}$$

Copyright by Mahidol University

9) For $CTAM_nR$ species (D = $CTAM_nR$)

Let $\Gamma_{14} = k_{tci}[R^*]$, $\Gamma_{15} = k_{tc}\frac{P(c)_1 P(r)_1}{\alpha_c \alpha_r}$ and substitution of Γ_{14} and Γ_{15} into Equation (AI-47) gives Equation (AI-88).

$$\frac{d}{dt}[CTAM_nR] = \Gamma_{14}[R^*]P(c)_1\alpha_c^{n-1} + \Gamma_{15}\psi_{rc} \qquad \text{(AI-88)}.$$

Substitution of $\frac{d}{dt}[CTAM_nR]$ into Equation (AI-55) to obtain the zeroth, the first, and the second moments (Equations (AI-89), (AI-90), and (AI-91), respectively) by integration of the differentiation term in Equation (AI-55).

$$Q_0 = \int_0^t \left(\Gamma_{14}\frac{P(c)_1\alpha_c}{1-\alpha_c} + \Gamma_{15}\sum_{n-2}^{n}\psi_{rc} \right) dt \qquad \text{(AI-89)}$$

$$Q_1 = \int_0^t \left(\Gamma_{14}\frac{P(c)_1(2-\alpha_c)\alpha_c}{(1-\alpha_c)^2} + \Gamma_{15}\sum_{n-2}^{n} n\psi_{rc} \right) dt \qquad \text{(AI-90)}$$

$$Q_2 = \int_0^t \left(\Gamma_{14}\frac{P(c)_1(4-3\alpha_c+\alpha_c{}^2)\alpha_c}{(1-\alpha_c)^3} + \Gamma_{15}\sum_{n=2}^{n} n^2\psi_{rc} \right) dt \qquad \text{(AI-91)}$$

where

$$\sum_{n=2}^{n}\psi_{rc} = \begin{cases} \frac{\alpha_r^2}{1-\alpha_r}\left[\frac{R_{c/r}}{1-R_{c/r}}\right] & ;\alpha_c < \alpha_r \\ \frac{\alpha_c^2}{1-\alpha_c}\left[\frac{R_{c/r}}{1-R_{c/r}}\right] & ;\alpha_r < \alpha_c \\ \frac{\alpha_c^2}{(1-\alpha_c)^2} & ;\alpha_r = \alpha_c \end{cases}$$

Copyright by Mahidol University

$$\sum_{n=2}^{n} n\psi_{rc} = \begin{cases} \dfrac{(2-\alpha_r)\alpha_r^2}{(1-\alpha_r)^2}\left[\dfrac{R_{c/r}}{1-R_{c/r}}\right] & ;\alpha_c < \alpha_r \\ \dfrac{(2-\alpha_c)\alpha_c^2}{(1-\alpha_c)^2}\left[\dfrac{R_{c/r}}{1-R_{c/r}}\right] & ;\alpha_r < \alpha_c \\ \dfrac{2\alpha_c^2}{(1-\alpha_c)^3} & ;\alpha_r = \alpha_c \end{cases}$$

$$\sum_{n=2}^{n} n^2\psi_{rc} = \begin{cases} \dfrac{(4-3\alpha_r+\alpha_r^2)\alpha_r^2}{(1-\alpha_r)^3}\left[\dfrac{R_{c/r}}{1-R_{c/r}}\right] & ;\alpha_c < \alpha_r \\ \dfrac{(4-3\alpha_c+\alpha_c^2)\alpha_c^2}{(1-\alpha_c)^3}\left[\dfrac{R_{c/r}}{1-R_{c/r}}\right] & ;\alpha_r < \alpha_c \\ \dfrac{2(2+\alpha_c)\alpha_c^2}{(1-\alpha_c)^4} & ;\alpha_r = \alpha_c \end{cases}$$

Copyright by Mahidol University

Appendix II

Overall Velocity Coefficient of the AIBN-initiated Polymerization Reactions (k) and Evaluation of the Exponent Order p, q and r.

The relationship of $R_{ST,CCl4}$ to [ST], [AIBN], and/or [CCl_4] is assumed to be:-

$$R_{ST,CCl4} = k'_{ST,CCl4}[ST]^p[CCl_4]^q \quad \text{(AII-1)}$$

where

$$k'_{ST,CCl4} = k_{ST,CCl4}[AIBN]^r \quad \text{(AII-2)}$$

where p, q, and r are the individual exponent order for ST, CCl_4, and AIBN, respectively. The total reaction order with respect to $R_{ST,CCl4}$ is equal to p+q+r. Taking two sets (i and j) of data at the same percentage conversion of styrene, and dividing the first set by the second, gives:-

$$\frac{(R_{ST,CCl_4})_i}{(R_{ST,CCl_4})_j} = \left(\frac{[ST]_i}{[ST]_j}\right)^p \cdot \left(\frac{[CCl_4]_i}{[CCl_4]_j}\right)^q \quad \text{(AII-3)}$$

Taking the natural logarithm of (AII-3) gives:-

$$\ln\left(\frac{(R_{ST,CCl_4})_i}{(R_{ST,CCl_4})_j}\right) = p.\ln\left(\frac{[ST]_i}{[ST]_j}\right) + q.\ln\left(\frac{[CCl_4]_i}{[CCl_4]_j}\right) \quad \text{(AII-4)}$$

Taking the two sets of data (1 and 2) as (AII-5) and (AII-6):-

$$\ln\left(\frac{(R_{ST,CCl_4})_i}{(R_{ST,CCl_4})_j}\right)_1 = p.\ln\left(\frac{[ST]_i}{[ST]_j}\right)_1 + q.\ln\left(\frac{[CCl_4]_i}{[CCl_4]_j}\right)_1 \quad \text{(AII-5)}$$

Copyright by Mahidol University

$$\ln\left(\frac{(R_{ST,CCl_4})_i}{(R_{ST,CCl_4})_j}\right)_2 = p.\ln\left(\frac{[ST]_i}{[ST]_j}\right)_2 + q.\ln\left(\frac{[CCl_4]_i}{[CCl_4]_j}\right)_2 \quad \text{(AII-6)}$$

Therefore, for the two sets of data at two level of $[CCl_4]_0/[ST]_0$, p and q can be obtained by using the expressions (AII-7) and (AII-8) and $k'_{ST,CCl4}$ can be obtained by using equation (AII-9):-

$$q = \frac{(A_1 - A_2) - (B_1 - B_2)}{(C_1 - C_2)} \qquad \text{(AII-7)}$$

$$p = \frac{A_1 - q.C_1}{B_1} \qquad \text{(AII-8)}$$

$$k'_{ST,CCl4} = R_{ST,CCl4}/([ST]^p[CCl_4]^q) \qquad \text{(AII-9)}$$

where

$$A_1 = \ln\left(\frac{(R_{ST,CCl_4})_i}{(R_{ST,CCl_4})_j}\right)_1 \text{ and } A_2 = \ln\left(\frac{(R_{ST,CCl_4})_i}{(R_{ST,CCl_4})_j}\right)_2$$

$$B_1 = \ln\left(\frac{[ST]_i}{[ST]_j}\right)_1 \text{ and } B_2 = \ln\left(\frac{[ST]_i}{[ST]_j}\right)_2$$

$$C_1 = \ln\left(\frac{[CCl_4]_i}{[CCl_4]_j}\right)_1 \text{ and } C_2 = \ln\left(\frac{[CCl_4]_i}{[CCl_4]_j}\right)_2$$

If the exponent order r is known, $k_{ST,CCl4}$ can be obtained by using Equation (AII-2).

Copyright by Mahidol University

Appendix III
Computer Programmes For the Dynamic Predictive Thermal-Initiated Free Radical Polymerization of a Vinyl monomer in the Presence of a Chain Transfer Agent

1) The main control programme (MT.BAS)

```
DECLARE SUB InstallP ()
DECLARE SUB FunckeyInsI ()
DECLARE SUB FunckeyIns ()
DECLARE SUB Insxxx (xcount!, owncolor!)
DECLARE SUB FuckeyMenu ()
DECLARE SUB Mxxx (xcount!, owncolor!)
DECLARE SUB Position (row!, col!, Text$, Tcolor!,
Bcolor!)
DECLARE SUB Center (row!, Text$, Tcolor!, Bcolor!)
DECLARE SUB CenterI (row!, Text$, Tcolor!, Bcolor!)
DECLARE SUB FunckeyTitle ()
DECLARE SUB MMenu ()
DECLARE SUB FunckeyUtil ()
DECLARE SUB FunckeyUtiII ()
DECLARE SUB CommandMenu ()
DECLARE SUB Cxxx (xcount!, owncolor!)
DECLARE SUB EraseScreen ()
DIM SHARED Comm$(1 TO 4)
DIM SHARED Menu$(1 TO 3)
DIM SHARED InstallPrint$(1 TO 2)
Menu$(1) = "Command"
Menu$(2) = "Quit!"
Comm$(1) = "Weight fraction simulation"
Comm$(2) = "Modify Mayo relationship"
Comm$(3) = "Install program capture & print"
Comm$(4) = "Quit!"
InstallPrint$(1) = "NEC P2200XE"
InstallPrint$(2) = "HP LaserJet III"
SHELL "TT.EXE"
    DO
     FunckeyTitle
     FuckeyMenu
     CALL Mxxx(count, 15)
     DO
      LOCATE 1, 2· COLOR 15, 6: PRINT TIME$
      KBD$ = INKEY$
     LOOP UNTIL KBD$ <> ""
     SELECT CASE KBD$
         CASE CHR$(0) + "K"     'Left Arrow
              count = count - 1
              CALL Mxxx(count + 1, 0)
              IF count < 1 THEN count = 2
         CASE CHR$(0) + "M"     'Right Arrow
              count = count + 1
              CALL Mxxx(count - 1, 0)
              IF count > 2 THEN count = 1
         CASE CHR$(13)
              inpcount = count
              SELECT CASE inpcount
                   CASE 1
                      CommandMenu
                   CASE 2
                      COLOR 10, 0: CLS : END
                END SELECT
        END SELECT
    LOOP UNTIL KBD$ = CHR$(27)
    SHELL "MUCOM.EXE"

    SUB Center (row, Text$, Tcolor, Bcolor)
      COLOR Tcolor, Bcolor
      LOCATE row, 41 - LEN(Text$) / 2
      PRINT Text$;
    END SUB

    SUB CenterI (row, Text$, Tcolor, Bcolor)
      COLOR Tcolor, Bcolor
      LOCATE row, (41 - LEN(Text$) / 2) + 2
      PRINT Text$;
    END SUB

    SUB CommandMenu
        FunckeyUtiII
        FunckeyUtil
      DO
        CALL Cxxx(count, 15)
        DO
         LOCATE 1, 2. COLOR 15, 6: PRINT TIME$
         KBD$ = INKEY$
        LOOP UNTIL KBD$ <> ""
        SELECT CASE KBD$
            CASE CHR$(0) + "H"
                 count = count - 1
                 CALL Cxxx(count + 1, 0)
                 IF count < 1 THEN count = 4
            CASE CHR$(0) + "P"     'Right Arrow
                 count = count + 1
                 CALL Cxxx(count - 1, 0)
                 IF count > 4 THEN count = 1
            CASE CHR$(13)
                 inpcount = count
                 SELECT CASE inpcount
                     CASE 1
                        SHELL "MTW.EXE"
                        EXIT SUB
                     CASE 2
                        SHELL "MTM EXE"
                        EXIT SUB
                     CASE 3
                        InstallP
                        'SHELL "c."
                        'SHELL "cd\pzp-nec"
                        'SHELL "pzp.com"
                        'SHELL "c:"
                        'SHELL "cd\simulate"
                        EXIT SUB
                     CASE 4
                        COLOR 10, 0: CLS · END
                 END SELECT
        END SELECT
      LOOP UNTIL KBD$ = CHR$(27)
      EraseScreen
    END SUB

    SUB Cxxx (xcount, owncolor)
        SELECT CASE xcount
            CASE 1
              CALL Center(13, Comm$(1), owncolor, 3)
            CASE 2
              CALL Center(14, Comm$(2), owncolor, 3)
            CASE 3
              CALL Center(15, Comm$(3), owncolor, 3)
            CASE 4
              CALL Center(16, Comm$(4), owncolor, 3)
        END SELECT
    END SUB
```

Copyright by Mahidol University

```
SUB EraseScreen
   LOCATE 5, 1
   COLOR 0, 1
   FOR i = 1 TO 15
   PRINT STRING$(80, " ")
   NEXT i
END SUB

SUB FuckeyMenu
   LOCATE 2, 1  COLOR 15, 15: PRINT STRING$(80,
CHR$(219))
   LOCATE 3, 1: COLOR , 3. PRINT STRING$(80, " ")
   CALL Position(3, 2, Menu$(1), 0, 3)
   CALL Position(3, 12, Menu$(2), 0, 3)
END SUB

SUB FunckeyIns
      X = 15  Y = 4  i = 8
      CALL Center(i + 1, " ----------", X, Y)
      CALL Center(i + 2, "³                ³", X, Y)
      CALL Center(i + 3, " ----------", X, Y)
      CALL Center(i + 4, "³                ³", X, Y)
      CALL Center(i + 5, "³                ³", X, Y)
      CALL Center(i + 6, "³                ³", X, Y)
      CALL Center(i + 7, "³                ³", X, Y)
      CALL Center(i + 8, "³                ³", X, Y)
      CALL Center(i + 9, "³                ³", X, Y)
      CALL Center(i + 10, " ----------", X, Y)
      CALL Center(10, "INSTALL PROGRAM CAPTURE
& PRINTING OUT", 14, 4)
      CALL Center(13, InstallPrint$(1), 0, 4)
      CALL Center(14, InstallPrint$(2), 0, 4)
END SUB

SUB FunckeyInsI
      X = 0: Y = 0: i = 9
      CALL Center(i + 1, " ----------", X, Y)
      CALL Center(i + 2, "³                ³", X, Y)
      CALL Center(i + 3, " ----------", X, Y)
      CALL Center(i + 4, "³                ³", X, Y)
      CALL Center(i + 5, "³                ³", X, Y)
      CALL Center(i + 6, "³                ³", X, Y)
      CALL Center(i + 7, "³                ³", X, Y)
      CALL Center(i + 8, "³                ³", X, Y)
      CALL Center(i + 9, "³                ³", X, Y)
      CALL Center(i + 10, " ----------", X, Y)
END SUB

SUB FunckeyTitle
      COLOR 0, 1. CLS
      LOCATE 1, 1: COLOR , 6: PRINT STRING$(80, " ")
      CALL Center(1, "Computer Simulation and Modelling
", 14, 6)
END SUB

SUB FunckeyUtil
      X = 0  Y = 3: i = 8
      CALL Center(i + 1, " ----------", X, Y)
      CALL Center(i + 2, "³                ³", X, Y)
      CALL Center(i + 3, " ----------", X, Y)
      CALL Center(i + 4, "³                ³", X, Y)
      CALL Center(i + 5, "³                ³", X, Y)
      CALL Center(i + 6, "³                ³", X, Y)
      CALL Center(i + 7, "³                ³", X, Y)
      CALL Center(i + 8, "³                ³", X, Y)
      CALL Center(i + 9, "³                ³", X, Y)
      CALL Center(i + 10, " ----------", X, Y)
      CALL Center(10, "COMMAND MENU", 14, 3)
      CALL Center(13, Comm$(1), 0, 3)
      CALL Center(14, Comm$(2), 0, 3)
      CALL Center(15, Comm$(3), 0, 3)
      CALL Center(16, Comm$(4), 0, 3)
END SUB

SUB FunckeyUtilI
      X = 0: Y = 0: i = 9
      CALL Center(i + 1, " ----------", X, Y)
      CALL Center(i + 2, "³                ³", X, Y)
      CALL Center(i + 3, " ----------", X, Y)
      CALL Center(i + 4, "³                ³", X, Y)
      CALL Center(i + 5, "³                ³", X, Y)
      CALL Center(i + 6, "³                ³", X, Y)
      CALL Center(i + 7, "³                ³", X, Y)
      CALL Center(i + 8, "³                ³", X, Y)
      CALL Center(i + 9, "³                ³", X, Y)
      CALL Center(i + 10, " ----------", X, Y)
END SUB

SUB InstallP
      DO
      FunckeyInsI
      FunckeyIns
      CALL Insxxx(count, 15)
      DO
       LOCATE 1, 2: COLOR 15, 6: PRINT TIME$
       KBD$ = INKEY$
      LOOP UNTIL KBD$ <> ""
      SELECT CASE KBD$
          CASE CHR$(0) + "H"
                count = count - 1
                CALL Insxxx(count + 1, 0)
                IF count < 1 THEN count = 2
          CASE CHR$(0) + "P"     'Right Arrow
                count = count + 1
                CALL Insxxx(count - 1, 0)
                IF count > 2 THEN count = 1
          CASE CHR$(13)
                inpcount = count
                SELECT CASE inpcount
                   CASE 1
                      SHELL "c:"
                      SHELL "cd\SIMULATE\necP"
                      SHELL "pzp com"
                      SHELL "c:"
                      SHELL "cd\simulate"
                      EXIT SUB
                   CASE 2
                      SHELL "c."
                      SHELL "cd\SIMULATE\\hpP"
                      SHELL "pzp.com"
                      SHELL "c."
                      SHELL "cd\simulate"
                      EXIT SUB
                END SELECT
      END SELECT
    LOOP UNTIL KBD$ = CHR$(27)
    EraseScreen
END SUB

SUB Insxxx (xcount, owncolor)
      SELECT CASE xcount
          CASE 1
                CALL Center(13, InstallPrint$(1), owncolor,
4)
          CASE 2
                CALL Center(14, InstallPrint$(2), owncolor,
4)
       END SELECT
END SUB

SUB Left (row, Text$, Tcolor, Bcolor)
    COLOR Tcolor, Bcolor
    LOCATE row, 1
```

Copyright by Mahidol University

```
    PRINT Text$;
END SUB

SUB MMenu
    DO
     FuckeyMenu
     DO
      CALL Mxxx(count, 15)
     DO
      LOCATE 1, 2  COLOR 15, 15: PRINT TIME$
      KBD$ = INKEY$
     LOOP UNTIL KBD$ <> ""
     SELECT CASE KBD$
         CASE CHR$(0) + "K"     'Left Arrow
              count = count - 1
              CALL Mxxx(count + 1, 0)
              IF count < 1 THEN count = 1
         CASE CHR$(0) + "M"     'Right Arrow
              count = count + 1
              CALL Mxxx(count - 1, 0)
              IF count > 2 THEN count = 2
         CASE CHR$(13)
              inpcount = count
              SELECT CASE inpcount
                  CASE 1
                     CommandMenu
                  CASE 2
                     COLOR 10, 0: CLS . END
              END SELECT
    END SELECT
  LOOP UNTIL KBD$ = CHR$(27)
  LOOP UNTIL KBD$ = CHR$(27)
END SUB

SUB Mxxx (xcount, owncolor)
    SELECT CASE xcount
         CASE 1
           CALL Position(3, 2, Menu$(1), owncolor, 3)
         CASE 2
           CALL Position(3, 12, Menu$(2), owncolor, 3)
      END SELECT
END SUB

SUB Position (row, col, Text$, Tcolor, Bcolor)
  COLOR Tcolor, Bcolor
  LOCATE row, col
  PRINT Text$,
END SUB
```

2) Menu programmes for calculation of dead-end polymers (MTW.BAS)

```
DECLARE SUB ManaulInput ()
DECLARE SUB ConfirmBoxM ()
DECLARE SUB Manaulxxx (num!, fc!, bc!)
DECLARE SUB ManaulMenuSelect (num!)
DECLARE SUB CurveLoadMenuSelect (num!)
DECLARE SUB CondSaveMenuSelect (num!)
DECLARE SUB CondSaveMenu1 ()
DECLARE SUB CondLoadMenu1 ()
DECLARE SUB CurveLoadMenu ()
DECLARE SUB CurveLoadMenu1 ()
DECLARE SUB CondLoadMenuSelect (num!)
DECLARE SUB CondLoadMenu ()
DECLARE SUB CondSaveMenu ()
DECLARE SUB UtilMenu1 ()
DECLARE SUB CurveSaveMenu1 ()
DECLARE SUB Utilxxx (num!, fc!, bc!)
DECLARE SUB CurveSaveMenu ()
DECLARE SUB CurveSaveMenuSelect (num!)
DECLARE SUB CurveSaveMenu2 ()
DECLARE SUB CurveSaveMenu3 ()
DECLARE SUB CurveSave ()
DECLARE SUB Center (row!, Text$, Tcolor!, Bcolor!)
DECLARE SUB ResultMenu ()
DECLARE SUB Sxxx (num!, fc!, bc!)
DECLARE SUB ResultMenuSelect (num!)
DECLARE SUB ResultSubMenu1 ()
DECLARE SUB ResultSubMenu2 ()
DECLARE SUB GraphicsMenu ()
DECLARE SUB GraphicsSubMenu2 ()
DECLARE SUB GraphicsSubMenu3 ()
DECLARE SUB Gxxx (num!, fc!, bc!)
DECLARE SUB GraphicsMenuSelect (num!)
DECLARE SUB GraphicsSubMenu1 ()
DECLARE SUB SentTemp ()
DECLARE SUB HelpForInputValue ()
DECLARE SUB HelpForInputText ()
DECLARE SUB HelpForDrive ()
DECLARE SUB HelpForFile ()
DECLARE SUB HelpForEditRate ()
DECLARE SUB RunSubMenu5 ()
DECLARE SUB ParameterSubMenu3 ()
DECLARE SUB SentConstant ()
DECLARE SUB RunSubMenu3 ()
DECLARE SUB RunMenuSelect (num!)
DECLARE SUB FileSubMenu4 (YLL!, Text$)
DECLARE SUB LoadCondFile ()
DECLARE SUB ParaMenuSelect (num!)
DECLARE SUB RunSubMenu4 ()
DECLARE SUB SaveCondFile ()
DECLARE SUB LoadFile ()
DECLARE SUB FileSubMenu3 ()
DECLARE SUB ConfirmBox ()
DECLARE SUB FileMenuSelect (num!)
DECLARE SUB RunSubMenu2 ()
DECLARE SUB FileSubMenu2 ()
DECLARE SUB ParameterSubMenu2 ()
DECLARE SUB ClearScreen (xc!)
DECLARE SUB DBox ()
DECLARE SUB DBackBox ()
DECLARE SUB RunSubMenu1 ()
DECLARE SUB ParameterSubMenu1 ()
DECLARE SUB ParameterMenu ()
DECLARE SUB RunMenu ()
DECLARE SUB FileSubMenu1 ()
DECLARE SUB FileMenu ()
DECLARE SUB PrintBar ()
DECLARE SUB SetScreen ()
DECLARE SUB PrintText (XL!, YL!, fc!, bc!, Text$)
DECLARE SUB PrintSameText (x1!, y1!, lx!, ly!, fc!, bc!,
Text$)
DECLARE SUB Fxxx (num!, fc!, bc!)
DECLARE SUB Pxxx (num!, fc!, bc!)
DECLARE SUB Rxxx (num!, fc!, bc!)
DECLARE SUB Mxxx (num!, bc!)
COMMON SHARED MoName$, IoName$, SoName$
COMMON SHARED Mo, Io, CTA, Temp, Time, TimeS
COMMON SHARED Mt, CTAt, It, Pr, Pc
COMMON SHARED Kp, Ktd, Ktc, Ktrm, Ki, Kd, f, Ktci,
Ksr, Ktr
COMMON SHARED NumberPoint, StepTime, Xscale,
Yscale
COMMON SHARED SaveCondDrive$, SaveCondName$,
LoadCondDrive$
```

Copyright by Chiang Mai University

```
COMMON SHARED SaveCurveDrive$,
SaveCurveName$, LoadCurveDrive$, LoadCurveName$
COMMON SHARED OpenCondName$
COMMON SHARED GelEffect$
COMMON SHARED XoCOOR, YoCOOR, XaxisLength,
YaxisLength, ZaxisLength
COMMON SHARED LeftAngle, RightAngle, NCLstep,
THRstep
XoCOOR = 550
YoCOOR = 380
XaxisLength = 250
YaxisLength = 250
ZaxisLength = 190
LeftAngle = 30
RightAngle = 15
NCLstep = 1
THRstep = 1
SetScreen
PrintBar
StepTime = 20
NumberPoint = 150
Xscale = 10
Yscale = 10
GelEffect$ = "OFF"
ColCount = 0
DO
  DO
  FOR i = 0 TO 18
     LOCATE 3 + i, 1: COLOR 15, 0  PRINT
STRING$(80, CHR$(219))
  NEXT i
  CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
  CALL Center(22, "Press <Enter> To Select, Right & Left
Arrow Key To Move Or Esc To Abort", 15, 1)
    CALL Mxxx(ColCount, 8)
     DO
       kbd$ = INKEY$
       LOCATE 23, 72  COLOR 15, 11: PRINT TIME$
     LOOP UNTIL kbd$ <> ""
     SELECT CASE kbd$
         CASE CHR$(0) + "K"     'Left Arrow
              ColCount = ColCount - 1
              CALL Mxxx(ColCount + 1, 3)
              IF ColCount < 1 THEN ColCount = 5
         CASE CHR$(0) + "M"     'Right Arrow
              ColCount = ColCount + 1
              CALL Mxxx(ColCount - 1, 3)
              IF ColCount > 5 THEN ColCount = 1
         CASE CHR$(27)
              CLS : END
     END SELECT
     LOOP UNTIL kbd$ = CHR$(13)
     SELECT CASE ColCount
         CASE 1
            FileMenu
         CASE 2
            ParameterMenu
         CASE 3
            RunMenu
         CASE 4
            GraphicsMenu
         CASE 5
            ResultMenu
     END SELECT
LOOP

SUB Center (row, Text$, Tcolor, Bcolor)
   COLOR Tcolor, Bcolor
   LOCATE row, 41 - LEN(Text$) / 2
   PRINT Text$,
END SUB

SUB ClearScreen (xc)
     CALL PrintSameText(38, 5, 42, xc, 0, 0, " ")
END SUB

SUB CondLoadMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
     DBackBox
     DBox
     CondLoadMenu1
     RowCount = 0
     DO
      CALL Utilxxx(RowCount, 15, 8)
     DO
        kbd$ = INKEY$
        LOCATE 23, 72: COLOR 15, 11: PRINT TIME$
     LOOP UNTIL kbd$ <> ""
     SELECT CASE kbd$
          CASE CHR$(0) + "H"     'Up Arrow
               RowCount = RowCount - 1
               CALL Utilxxx(RowCount + 1, 0, 3)
               IF RowCount < 1 THEN RowCount = 3
          CASE CHR$(0) + "P"     'Down Arrow
               RowCount = RowCount + 1
               CALL Utilxxx(RowCount - 1, 0, 3)
               IF RowCount > 3 THEN RowCount = 1
     END SELECT
     LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
    IF kbd$ = CHR$(13) THEN CALL
CondLoadMenuSelect(RowCount)
  LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB CondLoadMenu1
     CALL PrintText(5, 5, 15, 11, "LOAD PARAMETER
& CONDITION")
     UtilMenu1
END SUB

SUB CondLoadMenuSelect (num)
   SELECT CASE num
      CASE 2
           HelpForDrive
           CALL PrintText(57, 8, 15, 9, "            ")
           LOCATE 8, 60
           INPUT "", LoadCondDrive$
           IF LoadCondDrive$ = "" THEN LoadCondDrive$
= "C:\SIMULATE\PACKAGE"
           CALL PrintText(57, 8, 0, 0, "                  ")
           CALL PrintText(57, 8, 15, 0, LoadCondDrive$)
      CASE 3
           ImageDrive$ = LEFT$(LoadCondDrive$, 2)
           SHELL ImageDrive$
           SHELL "dir *.CNW /B/S/O" + " > " +
"ROOT.SCN"
           OPEN "Mdrive.DV" FOR OUTPUT AS #12
              PRINT #12, ImageDrive$
           CLOSE #12
           SHELL "MTF.EXE"
           OPEN "Ldrive.DV" FOR INPUT AS #13
              INPUT #13, OCN$
           CLOSE #13
           IF OCN$ <> "" THEN
              OpenCondName$ = OCN$
              LoadCondFile
           END IF
           SetScreen
```

Copyright by Mahidol University

```
        PrintBar
        FOR i = 0 TO 18
            LOCATE 3 + i, 1: COLOR 15, 0: PRINT
STRING$(80, CHR$(219))
        NEXT i
        CALL ClearScreen(13)
        CALL PrintText(38, 4, 14, 1, "
INFORMATION  & SUMARY          ")
        FileSubMenu2
        FileSubMenu3
   END SELECT
END SUB

SUB CondSaveMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
     DBackBox
     DBox
     CondSaveMenu1
     RowCount = 0
     DO
      CALL Utilxxx(RowCount, 15, 8)
     DO
        kbd$ = INKEY$
        LOCATE 23, 72: COLOR 15, 11  PRINT TIME$
     LOOP UNTIL kbd$ <> ""
     SELECT CASE kbd$
        CASE CHR$(0) + "H"     'Up Arrow
            RowCount = RowCount - 1
            CALL Utilxxx(RowCount + 1, 0, 3)
            IF RowCount < 1 THEN RowCount = 3
        CASE CHR$(0) + "P"     'Down Arrow
            RowCount = RowCount + 1
            CALL Utilxxx(RowCount - 1, 0, 3)
            IF RowCount > 3 THEN RowCount = 1
     END SELECT
     LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
     IF kbd$ = CHR$(13) THEN CALL
CondSaveMenuSelect(RowCount)
   LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB CondSaveMenu1
     CALL PrintText(5, 5, 15, 11, "SAVE PARAMETER
& CONDITION")
     UtilMenu1
END SUB

SUB CondSaveMenuSelect (num)
   SELECT CASE num
     CASE 1
        REM Input Curve File name
        HelpForFile
        CALL PrintText(57, 10, 15, 9, "          ")
        LOCATE 10, 60
        INPUT "", SaveCondName$
        IF SaveCondName$ = "" THEN SaveCondName$
= "MTEC"
        CALL PrintText(57, 10, 0, 0, "          ")
        CALL PrintText(57, 10, 15, 0, SaveCondName$)
     CASE 2
        HelpForDrive
        CALL PrintText(57, 11, 15, 9, "          ")
        LOCATE 11, 60
        INPUT "", SaveCondDrive$
        IF SaveCondDrive$ = "" THEN SaveCondDrive$
= "C:\SIMULATE\PACKAGE"
        CALL PrintText(57, 11, 0, 0, "          ")
        CALL PrintText(57, 11, 15, 0, SaveCondDrive$)
     CASE 3
        IF SaveCondName$ <> "" AND SaveCondDrive$
<> "" THEN
            SaveCondFile
        END IF
   END SELECT

END SUB

SUB ConfirmBox

     CALL PrintText(8, 12, 15, 11, " ----------")
     CALL PrintText(8, 13, 15, 11, "³          ³")
     CALL PrintText(8, 14, 15, 11, "³          ³")
     CALL PrintText(8, 15, 15, 11, "³          ³")
     CALL PrintText(8, 16, 15, 11, " ----------")
     CALL PrintText(8, 17, 15, 11, "³  Yes or No  ³")
     CALL PrintText(8, 18, 15, 11, " ----------")
     CALL PrintText(14, 17, 16, 11, "Y")
     CALL PrintText(21, 17, 16, 11, "N")
END SUB

SUB ConfirmBoxM

     CALL PrintText(8, 12, 15, 11, " ----------")
     CALL PrintText(8, 13, 15, 11, "³          ³")
     CALL PrintText(8, 14, 15, 11, "³          ³")
     CALL PrintText(8, 15, 15, 11, "³          ³")
     CALL PrintText(8, 16, 15, 11, " ----------")
     CALL PrintText(8, 17, 15, 11, "³  Manaul or Edit  ³")
     CALL PrintText(8, 18, 15, 11, " ----------")
     CALL PrintText(12, 17, 16, 11, "M")
     CALL PrintText(22, 17, 16, 11, "E")

END SUB

SUB CurveLoadMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
     DBackBox
     DBox
     CurveLoadMenu1
     RowCount = 0
     DO
      CALL Utilxxx(RowCount, 15, 8)
     DO
        kbd$ = INKEY$
        LOCATE 23, 72: COLOR 15, 11. PRINT TIME$
     LOOP UNTIL kbd$ <> ""
     SELECT CASE kbd$
        CASE CHR$(0) + "H"     'Up Arrow
            RowCount = RowCount - 1
            CALL Utilxxx(RowCount + 1, 0, 3)
            IF RowCount < 1 THEN RowCount = 3
        CASE CHR$(0) + "P"     'Down Arrow
            RowCount = RowCount + 1
            CALL Utilxxx(RowCount - 1, 0, 3)
            IF RowCount > 3 THEN RowCount = 1
     END SELECT
     LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
     IF kbd$ = CHR$(13) THEN CALL
CurveLoadMenuSelect(RowCount)
   LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB
```

Copyright by Mahidol University

```
SUB CurveLoadMenu1
    CALL PrintText(9, 5, 15, 11, "LOAD SIMULATED
CURVE")
    UtilMenu1
END SUB

SUB CurveLoadMenuSelect (num)
  SELECT CASE num
    CASE 2
        HelpForDrive
        CALL PrintText(57, 17, 15, 9, "          ")
        LOCATE 17, 60
        INPUT "", LoadCurveDrive$
        IF LoadCurveDrive$ = "" THEN
LoadCurveDrive$ = "C:\SIMULATE\PACKAGE"
        CALL PrintText(57, 17, 0, 0, "          ")
        CALL PrintText(57, 17, 15, 0, LoadCurveDrive$)
    CASE 3
        ImageDrive$ = LEFT$(LoadCurveDrive$, 2)
        SHELL ImageDrive$
        SHELL "dir * GPH /B/S/O" + " > " +
"ROOT SCN"
        OPEN "Mdrive DV" FOR OUTPUT AS #12
            PRINT #12, ImageDrive$
        CLOSE #12
        SHELL "MTF.EXE"
        OPEN "Ldrive DV" FOR INPUT AS #13
            INPUT #13, OCN$
        CLOSE #13
        IF OCN$ <> "" THEN
            LoadCurveName$ = OCN$
            OPEN "CURVE.LV" FOR OUTPUT AS #22
                PRINT #22, LoadCurveName$
                PRINT #22, LoadCurveDrive$
            CLOSE #22
        SHELL "MTL EXE"
        END IF
        SetScreen
        PrintBar
        FOR i = 0 TO 18
            LOCATE 3 + i, 1, COLOR 15, 0: PRINT
STRING$(80, CHR$(219))
        NEXT i
        CALL ClearScreen(13)
        CALL PrintText(38, 4, 14, 1, "
INFORMATION  & SUMARY          ")
        FileSubMenu2
        FileSubMenu3
  END SELECT
END SUB

SUB CurveSaveMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
    DBackBox
    DBox
    CurveSaveMenu1
    RowCount = 0
    DO
     CALL Utilxxx(RowCount, 15, 8)
    DO
        kbd$ = INKEY$
        LOCATE 23, 72: COLOR 15, 11, PRINT TIME$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"     'Up Arrow
            RowCount = RowCount - 1
            CALL Utilxxx(RowCount + 1, 0, 3)
            IF RowCount < 1 THEN RowCount = 3
        CASE CHR$(0) + "P"     'Down Arrow
            RowCount = RowCount + 1
            CALL Utilxxx(RowCount - 1, 0, 3)
            IF RowCount > 3 THEN RowCount = 1
    END SELECT
    LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
    IF kbd$ = CHR$(13) THEN CALL
CurveSaveMenuSelect(RowCount)
  LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB CurveSaveMenu1
    CALL PrintText(9, 5, 15, 11, "SAVE SIMULATED
CURVE")
    UtilMenu1
END SUB

SUB CurveSaveMenuSelect (num)
  SELECT CASE num
    CASE 1
        REM Input Curve File name
        HelpForFile
        CALL PrintText(57, 13, 15, 9, "          ")
        LOCATE 13, 60
        INPUT "", SaveCurveName$
        IF SaveCurveName$ = "" THEN
SaveCurveName$ = "MTEC"
        CALL PrintText(57, 13, 0, 0, "          ")
        CALL PrintText(57, 13, 15, 0, SaveCurveName$)
    CASE 2
        HelpForDrive
        CALL PrintText(57, 14, 15, 9, "          ")
        LOCATE 14, 60
        INPUT "", SaveCurveDrive$
        IF SaveCurveDrive$ = "" THEN
SaveCurveDrive$ = "C:\SIMULATE\PACKAGE"
        CALL PrintText(57, 14, 0, 0, "          ")
        CALL PrintText(57, 14, 15, 0, SaveCurveDrive$)
    CASE 3
        IF SaveCurveName$ <> "" AND
SaveCurveDrive$ <> "" THEN
        REM extension ".GPH" , consist with file
extension " GP1" to ".GP5"
        OPEN "CURVE.SV" FOR OUTPUT AS #21
            PRINT #21, SaveCurveName$
            PRINT #21, SaveCurveDrive$
        CLOSE #21
        SHELL "MTS.EXE"
        END IF
  END SELECT
END SUB

SUB DBackBox
    CALL PrintSameText(5, 5, 31, 15, 11, 8, " ")
END SUB

SUB DBox
    CALL PrintSameText(3, 4, 30, 15, 0, 11, " ")
    CALL PrintSameText(3, 4, 30, 0, 0, 11, "Ä")
    CALL PrintSameText(3, 4 + 2, 30, 0, 0, 11, "Ä")
    CALL PrintSameText(3, 4 + 15, 30, 0, 0, 11, "Ä")
    FOR k = 0 TO 15
        CALL PrintSameText(3, 4 + k, 1, 0, 0, 11, "³")
        CALL PrintSameText(3 + 30, 4 + k, 1, 0, 0, 11,
"³")
    NEXT k
    CALL PrintSameText(3, 4, 1, 0, 0, 11, "Ú")
    CALL PrintSameText(3 + 30, 4, 1, 0, 0, 11, "¿")
    CALL PrintSameText(3, 4 + 2, 1, 0, 0, 11, "Ã")
    CALL PrintSameText(3 + 30, 4 + 2, 1, 0, 0, 11, "´")
```

Copyright by Mahidol University

```
      CALL PrintSameText(3, 4 + 15, 1, 0, 0, 11, "À")
      CALL PrintSameText(3 + 30, 4 + 15, 1, 0, 0, 11, "Ù")
END SUB

SUB FileMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
      DBackBox
      DBox
      CALL ClearScreen(13)
      CALL PrintText(38, 4, 14, 1, "        INFORMATION
& SUMARY          ")
      CALL PrintText(5, 2, 15, 8, " File ")
      CALL PrintText(5, 2, 10, 8, " F")
      FileSubMenu1
      FileSubMenu2
      FileSubMenu3
      RowCount = 0
      DO
       CALL Fxxx(RowCount, 15, 8)
      DO
           kbd$ = INKEY$
           LOCATE 23, 72: COLOR 15, 11: PRINT TIME$
      LOOP UNTIL kbd$ <> ""
      SELECT CASE kbd$
           CASE CHR$(0) + "H"     'Up Arrow
                RowCount = RowCount - 1
                CALL Fxxx(RowCount + 1, 0, 3)
                IF RowCount < 1 THEN RowCount = 6
           CASE CHR$(0) + "P"     'Down Arrow
                RowCount = RowCount + 1
                CALL Fxxx(RowCount - 1, 0, 3)
                IF RowCount > 6 THEN RowCount = 1
      END SELECT
      LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
      IF kbd$ = CHR$(13) THEN CALL
FileMenuSelect(RowCount)
   LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB FileMenuSelect (num)
   SELECT CASE num
      CASE 1
           Temp = 60
           Time = 60
           MoName$ = "PS"
           Mo = 8
           IoName$ = "BP"
           Io = .01
           SoName$ = "EB"
           CTA =  1
      CASE 2
           ConfirmBox
           CALL PrintText(14, 13, 6, 11, "CONFIRM !")
           CALL PrintText(11, 14, 0, 11, "YOUR
COMMAND IS")
           CALL PrintText(11, 15, 9, 11, "LOAD
PARAMETERS")
           DO
           CondLoading$ = UCASE$(INPUT$(1))
           LOOP WHILE INSTR("YN", CondLoading$) = 0
           CALL PrintText(8, 17, 15, 11, "³      Yes      ³")
           SELECT CASE CondLoading$
                CASE "Y"
                     CondLoadMenu
           END SELECT
      CASE 3
           ConfirmBox
           CALL PrintText(14, 13, 6, 11, "CONFIRM !")
           CALL PrintText(11, 14, 0, 11, "YOUR
 COMMAND IS")
           CALL PrintText(11, 15, 9, 11, "SAVE
 PARAMETERS")
           DO
           CondSaving$ = UCASE$(INPUT$(1))
           LOOP WHILE INSTR("YN", CondSaving$) = 0
           CALL PrintText(8, 17, 15, 11, "³      Yes      ³")
           SELECT CASE CondSaving$
                CASE "Y"
                     CondSaveMenu
           END SELECT
      CASE 4
           ConfirmBox
           CALL PrintText(14, 13, 6, 11, "CONFIRM !")
           CALL PrintText(11, 14, 0, 11, "YOUR
COMMAND IS")
           CALL PrintText(13, 15, 9, 11, "SAVE CURVE")
           DO
           CurveSaving$ = UCASE$(INPUT$(1))
           LOOP WHILE INSTR("YN", CurveSaving$) = 0
           CALL PrintText(8, 17, 15, 11, "³      Yes      ³")
           SELECT CASE CurveSaving$
                CASE "Y"
                     CurveSaveMenu
           END SELECT
      CASE 5
           ConfirmBox
           CALL PrintText(14, 13, 6, 11, "CONFIRM !")
           CALL PrintText(11, 14, 0, 11, "YOUR
COMMAND IS")
           CALL PrintText(13, 15, 9, 11, "LOAD CURVE")
           DO
           CurveLoading$ = UCASE$(INPUT$(1))
           LOOP WHILE INSTR("YN", CurveLoading$) =
0
           CALL PrintText(8, 17, 15, 11, "³      Yes      ³")
           SELECT CASE CurveLoading$
                CASE "Y"
                     CurveLoadMenu
           END SELECT
      CASE 6
           ConfirmBox
           CALL PrintText(14, 13, 6, 11, "WARNING !")
           CALL PrintText(11, 14, 0, 11, "YOUR
COMMAND IS")
           CALL PrintText(13, 15, 0, 11, "EXIT TO DOS")
           DO
           ChangeYourMine$ = UCASE$(INPUT$(1))
           LOOP WHILE INSTR("YN",
ChangeYourMine$) = 0
           SELECT CASE ChangeYourMine$
                CASE "Y"
                     COLOR 15, 1  CLS : END
           END SELECT
   END SELECT
END SUB

SUB FileSubMenu1
      CALL PrintText(10, 5, 15, 11, "FILE MANAGER
MENU")
      CALL PrintText(4, 7, 0, 11, "   New  Parameters
")
      CALL PrintText(4, 8, 0, 11, "   Load Parameters
")
      CALL PrintText(4, 9, 0, 11, "   Save Parameters      ")
      CALL PrintText(4, 10, 0, 11, "   Save Simulation
")
      CALL PrintText(4, 11, 0, 11, "   Load Simulation
")
```

Copyright by Mahidol University

```
    CALL PrintText(4, 12, 0, 11, "    Exit to DOS          ")
END SUB

SUB FileSubMenu2
    CALL PrintText(40, 6, 15, 11, " PARAMETER
STATUS(NOW)")
    CALL PrintText(45, 7, 14, 0, "File Name .")
    CALL PrintText(45, 8, 14, 0, "Path Name  ")
    CALL PrintText(40, 9, 15, 11, " SAVE
PARAMETERS FILE ")
    CALL PrintText(45, 10, 14, 0, "File Name  ")
    CALL PrintText(45, 11, 14, 0, "Path Name :")
    CALL PrintText(40, 12, 15, 11, " SAVE
SIMULATED CURVE ")
    CALL PrintText(45, 13, 14, 0, "File Name .")
    CALL PrintText(45, 14, 14, 0, "Path Name  ")
    CALL PrintText(40, 15, 15, 11, " LOAD
SIMULATED CURVE ")
    CALL PrintText(45, 16, 14, 0, "File Name  ")
    CALL PrintText(45, 17, 14, 0, "Path Name :")
END SUB

SUB FileSubMenu3
    CALL PrintText(45, 7, 12, 0, OpenCondName$)
    CALL FileSubMenu4(10, SaveCondName$)
    CALL FileSubMenu4(11, SaveCondDrive$)
    CALL FileSubMenu4(13, SaveCurveName$)
    CALL FileSubMenu4(14, SaveCurveDrive$)
    CALL PrintText(45, 16, 12, 0, LoadCurveName$)
    CALL FileSubMenu4(17, LoadCurveDrive$)
END SUB

SUB FileSubMenu4 (YLL, Text$)
    IF Text$ <> "" THEN
        CALL PrintText(57, YLL, 15, 0, Text$)
    ELSE
        CALL PrintText(57, YLL, 15, 0, "Untitle")
    END IF
END SUB

SUB Fxxx (num, fc, bc)
     IF num < 1 THEN num = 1
     IF num > 6 THEN num = 6
     SELECT CASE num
          CASE 1
               CALL PrintText(4, 7, fc, bc, "   New
Parameters       ")
          CASE 2
               CALL PrintText(4, 8, fc, bc, "   Load
Parameters       ")
          CASE 3
               CALL PrintText(4, 9, fc, bc, "   Save
Parameters       ")
          CASE 4
               CALL PrintText(4, 10, fc, bc, "   Save
Simulation       ")
          CASE 5
               CALL PrintText(4, 11, fc, bc, "   Load
Simulation       ")
          CASE 6
               CALL PrintText(4, 12, fc, bc, "   Exit to
DOS          ")
     END SELECT
END SUB

SUB GraphicsMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
    DBackBox
    DBox
    CALL ClearScreen(11)
    CALL PrintText(38, 4, 14, 1, "       INFORMATION
& SUMARY         ")
    CALL PrintText(31, 2, 15, 8, " Graphics ")
    CALL PrintText(31, 2, 10, 8, " G")
    GraphicsSubMenu1
    GraphicsSubMenu2
    GraphicsSubMenu3
    RowCount = 0
    DO
     CALL Gxxx(RowCount, 15, 8)
    DO
        kbd$ = INKEY$
        LOCATE 23, 72: COLOR 15, 11  PRINT TIME$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"     'Up Arrow
            RowCount = RowCount - 1
            CALL Gxxx(RowCount + 1, 0, 3)
            IF RowCount < 1 THEN RowCount = 9
        CASE CHR$(0) + "P"     'Down Arrow
            RowCount = RowCount + 1
            CALL Gxxx(RowCount - 1, 0, 3)
            IF RowCount > 9 THEN RowCount = 1
    END SELECT
    LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
    IF kbd$ = CHR$(13) THEN CALL
GraphicsMenuSelect(RowCount)
  LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB GraphicsMenuSelect (num)
  SELECT CASE num
    CASE 1
        HelpForInputValue
        CALL PrintText(60, 7, 15, 9, "     ")
        LOCATE 7, 62
        INPUT "", XoCOOR
        IF XoCOOR = 0 THEN XoCOOR = 550
    CASE 2
        HelpForInputValue
        CALL PrintText(60, 8, 15, 9, "     ")
        LOCATE 8, 62
        INPUT "", YoCOOR
        IF YoCOOR = 0 THEN YoCOOR = 380
    CASE 3
        HelpForInputValue
        CALL PrintText(60, 9, 15, 9, "     ")
        LOCATE 9, 62
        INPUT "", XaxisLength
        IF XaxisLength = 0 THEN XaxisLength = 250
    CASE 4
        HelpForInputValue
        CALL PrintText(60, 10, 15, 9, "     ")
        LOCATE 10, 62
        INPUT "", YaxisLength
        IF YaxisLength = 0 THEN YaxisLength = 250
    CASE 5
        HelpForInputValue
        CALL PrintText(60, 11, 15, 9, "     ")
        LOCATE 11, 62
        INPUT "", ZaxisLength
        IF ZaxisLength = 0 THEN ZaxisLength = 200
    CASE 6
        HelpForInputValue
        CALL PrintText(60, 12, 15, 9, "     ")
        LOCATE 12, 62
        INPUT "", LeftAngle
        IF LeftAngle = 0 THEN LeftAngle = 30
```

Copyright by Mahidol University

```
      CASE 7
          HelpForInputValue
          CALL PrintText(60, 13, 15, 9, "     ")
          LOCATE 13, 62
          INPUT "", RightAngle
          IF RightAngle = 0 THEN RightAngle = 15
      CASE 8
          HelpForInputValue
          CALL PrintText(60, 14, 15, 9, "     ")
          LOCATE 14, 62
          INPUT "", NCLstep
          IF NCLstep = 0 THEN NCLstep = 1
      CASE 9
          HelpForInputValue
          CALL PrintText(60, 15, 15, 9, "     ")
          LOCATE 15, 62
          INPUT "", THRstep
          IF THRstep = 0 THEN THRstep = 1
    END SELECT
END SUB

SUB GraphicsSubMenu1
      fc = 0: bc = 11
      CALL PrintText(6, 5, 15, bc, "GRAPHICS 3-
DIMENSION PLOT")
      CALL PrintText(4, 7, fc, bc, "   Xo coordination
")
      CALL PrintText(4, 8, fc, bc, "   Yo coordination
")
      CALL PrintText(4, 9, fc, bc, "   X-axis length       ")
      CALL PrintText(4, 10, fc, bc, "   Y-axis length
")
      CALL PrintText(4, 11, fc, bc, "   Z-axis length
")
      CALL PrintText(4, 12, fc, bc, "   Left  angle        ")
      CALL PrintText(4, 13, fc, bc, "   Right angle        ")
      CALL PrintText(4, 14, fc, bc, "   NCL  step          ")
      CALL PrintText(4, 15, fc, bc, "   Time step          ")
END SUB

SUB GraphicsSubMenu2
      CALL PrintText(40, 6, 15, 11, " GRAPHICS STATUS
")
      CALL PrintText(41, 7, 14, 0, "Xo coordination
pixel")
      CALL PrintText(41, 8, 14, 0, "Yo coordination
pixel")
      CALL PrintText(41, 9, 14, 0, "X-axis length
pixel")
      CALL PrintText(41, 10, 14, 0, "X-axis length
pixel")
      CALL PrintText(41, 11, 14, 0, "X-axis length
pixel")
      CALL PrintText(41, 12, 14, 0, "Left  angle
degree")
      CALL PrintText(41, 13, 14, 0, "Right angle
degree")
      CALL PrintText(41, 14, 14, 0, "NCL  step
point/step")
      CALL PrintText(41, 15, 14, 0, "Time step
point/step")
END SUB

SUB GraphicsSubMenu3
   COLOR 15, 0
   LOCATE 7, 61: PRINT USING "###"; XoCOOR
   LOCATE 8, 61: PRINT USING "###"; YoCOOR
   LOCATE 9, 61: PRINT USING "###"; XaxisLength
   LOCATE 10, 61  PRINT USING "###"; YaxisLength
   LOCATE 11, 61: PRINT USING "###"; ZaxisLength
   LOCATE 12, 61. PRINT USING "###"; LeftAngle
   LOCATE 13, 61. PRINT USING "###"; RightAngle
   LOCATE 14, 61. PRINT USING "###", NCLstep
   LOCATE 15, 61  PRINT USING "###", THRstep
END SUB

SUB Gxxx (num, fc, bc)
      IF num < 1 THEN num = 1
      IF num > 9 THEN num = 9
SELECT CASE num
          CASE 1
          CALL PrintText(4, 7, fc, bc, " Xo coordination
")
CASE 2
       CALL PrintText(4, 8, fc, bc, " Yo coordination
")
          CASE 3
          CALL PrintText(4, 9, fc, bc, "   X-axis length  ")
       CASE 4
       CALL PrintText(4, 10, fc, bc, "   Y-axis length
")
          CASE 5
          CALL PrintText(4, 11, fc, bc, "   Z-axis length
")
          CASE 6
          CALL PrintText(4, 12, fc, bc, "   Left  angle
")
          CASE 7
          CALL PrintText(4, 13, fc, bc, "   Right angle
")
          CASE 8
          CALL PrintText(4, 14, fc, bc, "   NCL  step
")
          CASE 9
          CALL PrintText(4, 15, fc, bc, "   Time step
")
      END SELECT
END SUB

SUB HelpForDrive
      CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
      CALL PrintText(21, 22, 15, 9, "Input!  Directory Or
Enter To Use Defualt.")
END SUB

SUB HelpForEditRate
      CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
      CALL PrintText(30, 22, 15, 9, "Enter to Edit Values")
END SUB

SUB HelpForFile
      CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
      CALL PrintText(22, 22, 15, 9, "Input!  File Name
[Maximum 8 Character].")
END SUB

SUB HelpForInputText
      CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
      CALL PrintText(20, 22, 15, 9, "Input!  Text Or Enter
If You don't known.")
END SUB

SUB HelpForInputValue
      CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
      CALL PrintText(30, 22, 15, 9, "Input!  Numeric
Value")
END SUB

SUB LoadCondFile
   OPEN OpenCondName$ FOR INPUT AS #1
       INPUT #1, MoName$
       INPUT #1, IoName$
       INPUT #1, SoName$
       INPUT #1, Mo
```

Copyright by Mahidol University

```
        INPUT #1, Io
        INPUT #1, CTA
        INPUT #1, Temp
        INPUT #1, Time
        INPUT #1, Kp
        INPUT #1, Ktc
        INPUT #1, Ktd
        INPUT #1, Ktrm
        INPUT #1, Ki
        INPUT #1, Kd
        INPUT #1, f
        INPUT #1, Ktci
        INPUT #1, Ksr
        INPUT #1, Ktr
        INPUT #1, NumberPoint
        INPUT #1, StepTime
        INPUT #1, Xscale
        INPUT #1, Yscale
        INPUT #1, GelEffect$
        INPUT #1, XoCOOR
        INPUT #1, YoCOOR
        INPUT #1, XaxisLength
        INPUT #1, YaxisLength
        INPUT #1, ZaxisLength
        INPUT #1, LeftAngle
        INPUT #1, RightAngle
        INPUT #1, NCLstep
        INPUT #1, THRstep
    CLOSE #1
END SUB

SUB ManaulInput
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
        RunSubMenu4
        RunSubMenu5
        RowCount = 0
        DO
          CALL Manaulxxx(RowCount, 15, 9)
        DO
            kbd$ = INKEY$
            LOCATE 23, 72  COLOR 15, 11: PRINT TIME$
        LOOP UNTIL kbd$ <> ""
        SELECT CASE kbd$
            CASE CHR$(0) + "H"     'Up Arrow
                RowCount = RowCount - 1
                CALL Manaulxxx(RowCount + 1, 15, 0)
                IF RowCount < 1 THEN RowCount = 10
            CASE CHR$(0) + "P"     'Down Arrow
                RowCount = RowCount + 1
                CALL Manaulxxx(RowCount - 1, 15, 0)
                IF RowCount > 10 THEN RowCount = 1
        END SELECT
        LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
        IF kbd$ = CHR$(13) THEN CALL
ManaulMenuSelect(RowCount)
    LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB ManaulMenuSelect (num)
    SELECT CASE num
        CASE 1
            HelpForInputValue
            CALL PrintText(59, 11, 15, 9, "          ")
            LOCATE 11, 60
            INPUT "", Kp
        CASE 2
            HelpForInputValue
            CALL PrintText(59, 12, 15, 9, "          ")
            LOCATE 12, 60
            INPUT "", Ktd
        CASE 3
            HelpForInputValue
            CALL PrintText(59, 13, 15, 9, "          ")
            LOCATE 13, 60
            INPUT "", Ktc
        CASE 4
            HelpForInputValue
            CALL PrintText(59, 14, 15, 9, "          ")
            LOCATE 14, 60
            INPUT "", Ktrm
        CASE 5
            HelpForInputValue
            CALL PrintText(59, 15, 15, 9, "          ")
            LOCATE 15, 60
            INPUT "", Ki
        CASE 6
            HelpForInputValue
            CALL PrintText(59, 16, 15, 9, "          ")
            LOCATE 16, 60
            INPUT "", Kd
        CASE 7
            HelpForInputValue
            CALL PrintText(59, 17, 15, 9, "          ")
            LOCATE 17, 60
            INPUT "", f
        CASE 8
            HelpForInputValue
            CALL PrintText(59, 18, 15, 9, "          ")
            LOCATE 18, 60
            INPUT "", Ktci
        CASE 9
            HelpForInputValue
            CALL PrintText(59, 19, 15, 9, "          ")
            LOCATE 19, 60
            INPUT "", Ksr
        CASE 10
            HelpForInputValue
            CALL PrintText(59, 20, 15, 9, "          ")
            LOCATE 20, 60
            INPUT "", Ktr
    END SELECT
END SUB

SUB Manaulxxx (num, fc, bc)
        IF num < 1 THEN num = 1
        IF num > 10 THEN num = 10
        SELECT CASE num
            CASE 1
                COLOR fc, bc
                LOCATE 11, 60. PRINT USING
"##.###^^^^ "; Kp
            CASE 2
                COLOR fc, bc
                LOCATE 12, 60. PRINT USING
"## ###^^^^ "; Ktd
            CASE 3
                COLOR fc, bc
                LOCATE 13, 60: PRINT USING
"##.###^^^^ "; Ktc
            CASE 4
                COLOR fc, bc
                LOCATE 14, 60: PRINT USING
"##.###^^^^ "; Ktrm
            CASE 5
                COLOR fc, bc
                LOCATE 15, 60: PRINT USING
"##.###^^^^ "; Ki
            CASE 6
```

Copyright by Mahidol University

```
            COLOR fc, bc
            LOCATE 16, 60  PRINT USING
"##.###^^^^ "; Kd
        CASE 7
            COLOR fc, bc
            LOCATE 17, 60  PRINT USING "##.####
"; f
        CASE 8
            COLOR fc, bc
            LOCATE 18, 60: PRINT USING
"##.###^^^^ "; Ktc1
        CASE 9
            COLOR fc, bc
            LOCATE 19, 60  PRINT USING
"##.###^^^^ ", Ksr
        CASE 10
            COLOR fc, bc
            LOCATE 20, 60: PRINT USING
"##.###^^^^ ", Ktr
    END SELECT

END SUB

SUB Mxxx (num, bc)
    IF num < 1 THEN num = 1
    IF num > 5 THEN num = 5
    SELECT CASE num
        CASE 1
            CALL PrintText(5, 2, 15, bc, " File ")
            CALL PrintText(5, 2, 10, bc, " F")
        CASE 2
            CALL PrintText(13, 2, 15, bc, " Parameter ")
            CALL PrintText(13, 2, 10, bc, " P")
        CASE 3
            CALL PrintText(25, 2, 15, bc, " Run ")
            CALL PrintText(25, 2, 10, bc, " R")
        CASE 4
            CALL PrintText(31, 2, 15, bc, " Graphics ")
            CALL PrintText(31, 2, 10, bc, " G")
        CASE 5
            CALL PrintText(41, 2, 15, bc, " Simulated
result ")
            CALL PrintText(41, 2, 10, bc, " S")
    END SELECT
END SUB

SUB ParaMenuSelect (num)
  SELECT CASE num
    CASE 1
        HelpForInputValue
        CALL PrintText(62, 7, 15, 9, "        ")
        LOCATE 7, 62
        INPUT "", Temp
    CASE 2
        HelpForInputValue
        CALL PrintText(62, 8, 15, 9, "        ")
        LOCATE 8, 62
        INPUT "", Time
    CASE 3
        HelpForInputText
        CALL PrintText(62, 9, 15, 9, "        ")
        LOCATE 9, 62
        INPUT "", MoName$
        IF MoName$ = "" THEN MoName$ =
"UNKNOWN"
    CASE 4
        HelpForInputValue
        CALL PrintText(62, 10, 15, 9, "        ")
        LOCATE 10, 62
        INPUT "", Mo
    CASE 5
        HelpForInputText
        CALL PrintText(62, 11, 15, 9, "        ")
        LOCATE 11, 62
        INPUT "", IoName$
        IF IoName$ = "" THEN IoName$ =
"UNKNOWN"
    CASE 6
        HelpForInputValue
        CALL PrintText(62, 12, 15, 9, "        ")
        LOCATE 12, 62
        INPUT "", Io
    CASE 7
        HelpForInputText
        CALL PrintText(62, 13, 15, 9, "        ")
        LOCATE 13, 62
        INPUT "", SoName$
        IF SoName$ = "" THEN SoName$ =
"UNKNOWN"
    CASE 8
        HelpForInputValue
        CALL PrintText(62, 14, 15, 9, "        ")
        LOCATE 14, 62
        INPUT "", CTA
  END SELECT
END SUB

SUB ParameterMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
    DBackBox
    DBox
    CALL ClearScreen(10)
    CALL PrintText(38, 4, 14, 1, "        INFORMATION
& SUMARY        ")
    CALL PrintText(13, 2, 15, 8, " Parameter ")
    CALL PrintText(13, 2, 10, 8, " P")
    ParameterSubMenu1
    ParameterSubMenu2
    ParameterSubMenu3
    RowCount = 0
    DO
     CALL Pxxx(RowCount, 15, 8)
    DO
        kbd$ = INKEY$
        LOCATE 23, 72: COLOR 15, 11: PRINT TIME$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"     'Up Arrow
            RowCount = RowCount - 1
            CALL Pxxx(RowCount + 1, 0, 3)
            IF RowCount < 1 THEN RowCount = 8
        CASE CHR$(0) + "P"     'Down Arrow
            RowCount = RowCount + 1
            CALL Pxxx(RowCount - 1, 0, 3)
            IF RowCount > 8 THEN RowCount = 1
    END SELECT
    LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
    IF kbd$ = CHR$(13) THEN CALL
ParaMenuSelect(RowCount)
  LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB ParameterSubMenu1
    fc = 0: bc = 11
    CALL PrintText(7, 5, 15, bc, "PARAMETERS &
CONDITIONS")
    CALL PrintText(4, 7, fc, bc, "    Temperature        ")
```

Copyright by Mahidol University

```
    CALL PrintText(4, 8, fc, bc, "   Reaction Time
")
    CALL PrintText(4, 9, fc, bc, "   Monomer  Type
")
    CALL PrintText(4, 10, fc, bc, "   Monomer  Conc.
")
    CALL PrintText(4, 11, fc, bc, "   Initiator Type
")
    CALL PrintText(4, 12, fc, bc, "   Initiator Conc
")
    CALL PrintText(4, 13, fc, bc, "   Chain Transfer Type
")
    CALL PrintText(4, 14, fc, bc, "   Chain Transfer Conc.
")
END SUB

SUB ParameterSubMenu2
    CALL PrintText(40, 6, 15, 11, " PARAMETER
STATUS ")
    CALL PrintText(41, 7, 14, 0, "Temperature
øC")
    CALL PrintText(41, 8, 14, 0, "Reaction Time
Min")
    CALL PrintText(41, 9, 14, 0, "Monomer Type
Name")
    CALL PrintText(41, 10, 14, 0, "Monomer conc
mol/l")
    CALL PrintText(41, 11, 14, 0, "Initiator Type
name")
    CALL PrintText(41, 12, 14, 0, "Initiator Conc.
mol/l")
    CALL PrintText(41, 13, 14, 0, "Chain Transfer Type
name")
    CALL PrintText(41, 14, 14, 0, "Chain Transfer Conc.
mol/l")
END SUB

SUB ParameterSubMenu3
  COLOR 15, 0
  LOCATE 7, 63: PRINT USING "###.##"; Temp
  LOCATE 8, 63  PRINT USING "####.#"; Time
  LOCATE 9, 65  PRINT UCASE$(MoName$)
  LOCATE 10, 63  PRINT USING "###.####", Mo
  LOCATE 11, 65  PRINT UCASE$(IoName$)
  LOCATE 12, 63  PRINT USING "###.####", Io
  LOCATE 13, 65  PRINT UCASE$(SoName$)
  LOCATE 14, 63: PRINT USING "###.####"; CTA
END SUB

SUB PrintBar
    CALL PrintText(5, 2, 15, 3, " File ")
    CALL PrintText(5, 2, 10, 3, " F")
    CALL PrintText(13, 2, 15, 3, " Parameter ")
    CALL PrintText(13, 2, 10, 3, " P")
    CALL PrintText(25, 2, 15, 3, " Run ")
    CALL PrintText(25, 2, 10, 3, " R")
    CALL PrintText(31, 2, 15, 3, " Graphics ")
    CALL PrintText(31, 2, 10, 3, " G")
    CALL PrintText(41, 2, 15, 3, " Simulated result ")
    CALL PrintText(41, 2, 10, 3, " S")

END SUB

SUB PrintSameText (x1, y1, lx, ly, fc, bc, Text$)
COLOR fc, bc
  FOR i = 0 TO ly
    LOCATE y1 + i, x1
    PRINT STRING$(lx, Text$)
  NEXT i
END SUB

SUB PrintText (XL, YL, fc, bc, Text$)
  LOCATE YL, XL: COLOR fc, bc: PRINT Text$
END SUB

SUB Pxxx (num, fc, bc)
    IF num < 1 THEN num = 1
    IF num > 8 THEN num = 8
    SELECT CASE num
        CASE 1
            CALL PrintText(4, 7, fc, bc, "   Temperature
")
        CASE 2
            CALL PrintText(4, 8, fc, bc, "   Reaction
Time       ")
        CASE 3
            CALL PrintText(4, 9, fc, bc, "   Monomer
Type       ")
        CASE 4
            CALL PrintText(4, 10, fc, bc, "   Monomer
Conc.      ")
        CASE 5
            CALL PrintText(4, 11, fc, bc, "   Initiator
Type       ")
        CASE 6
            CALL PrintText(4, 12, fc, bc, "   Initiator
Conc       ")
        CASE 7
            CALL PrintText(4, 13, fc, bc, "   Chain
Transfer Type    ")
        CASE 8
            CALL PrintText(4, 14, fc, bc, "   Chain
Transfer Conc    ")
    END SELECT
END SUB

SUB ResultMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
    DBackBox
    DBox
    CALL ClearScreen(4)
    CALL PrintText(38, 4, 14, 1, "       INFORMATION
& SUMARY       ")
    CALL PrintText(41, 2, 15, 8, " Simulated result ")
    CALL PrintText(41, 2, 10, 8, " S")
    ResultSubMenu1
    ResultSubMenu2
    RowCount = 0
    DO
     CALL Sxxx(RowCount, 15, 8)
    DO
        kbd$ = INKEY$
        LOCATE 23, 72: COLOR 15, 11: PRINT TIME$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"     'Up Arrow
            RowCount = RowCount - 1
            CALL Sxxx(RowCount + 1, 0, 3)
            IF RowCount < 1 THEN RowCount = 2
        CASE CHR$(0) + "P"     'Down Arrow
            RowCount = RowCount + 1
            CALL Sxxx(RowCount - 1, 0, 3)
            IF RowCount > 2 THEN RowCount = 1
    END SELECT
    LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
    IF kbd$ = CHR$(13) THEN CALL
ResultMenuSelect(RowCount)
  LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
```

Copyright by Mahidol University

```
END SUB

SUB ResultMenuSelect (num)
  SELECT CASE num
    CASE 1
        SHELL "PLOTW-2D.EXE"
        SCREEN 12: SCREEN 0, 0
        SetScreen
        PrintBar
        CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
        CALL PrintText(10, 22, 10, 9, "Press <Enter> To
Select, U&D Arrow Key To Move Or Esc To Abort")
    CASE 2
        SHELL "PLOTW-3D.EXE"
        SCREEN 12: SCREEN 0, 0
        SetScreen
        PrintBar
        CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
        CALL PrintText(10, 22, 10, 9, "Press <Enter> To
Select, U&D Arrow Key To Move Or Esc To Abort")
  END SELECT
END SUB

SUB ResultSubMenu1
    fc = 0: bc = 11
    CALL PrintText(8, 5, 15, bc, "SHOW SIMULATED
RESULTS")
    CALL PrintText(4, 7, fc, bc, "   Plot 2-D Weight
Distrib.  ")
    CALL PrintText(4, 8, fc, bc, "   Plot 3-D Weight
Distrib.  ")
END SUB

SUB ResultSubMenu2
    CALL PrintText(40, 6, 15, 11, " DESCRIPTION     ")
    CALL PrintText(43, 7, 14, 0, "Plot two   dimension
")
    CALL PrintText(43, 8, 14, 0, "Plot three dimension
")
END SUB

SUB RunMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
    DBackBox
    DBox
    CALL ClearScreen(15)
    CALL PrintText(38, 4, 14, 1, "         INFORMATION
& SUMARY         ")
    CALL PrintText(25, 2, 15, 8, " Run ")
    CALL PrintText(25, 2, 10, 8, " R")
    RunSubMenu1
    RunSubMenu2
    RunSubMenu3
    RunSubMenu4
    RunSubMenu5
    IF GelEffect$ = "ON" THEN
        CALL PrintText(30, 13, 9, 11, "ON")
    ELSE
        CALL PrintText(30, 13, 9, 11, "OFF")
    END IF

    RowCount = 0
    DO
     CALL Rxxx(RowCount, 15, 8)
    DO
        kbd$ = INKEY$
        LOCATE 23, 72  COLOR 15, 11  PRINT TIME$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"     'Up Arrow
            RowCount = RowCount - 1
            CALL Rxxx(RowCount + 1, 0, 3)
            IF RowCount < 1 THEN RowCount = 7
        CASE CHR$(0) + "P"     'Down Arrow
            RowCount = RowCount + 1
            CALL Rxxx(RowCount - 1, 0, 3)
            IF RowCount > 7 THEN RowCount = 1
    END SELECT
    LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
    IF kbd$ = CHR$(13) THEN CALL
RunMenuSelect(RowCount)
  LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB RunMenuSelect (num)
  SELECT CASE num
    CASE 1
        SentConstant
        DBackBox
        DBox
        CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
        CALL PrintText(20, 22, 14, 9, "Wait !  Now
calculate  (Interupt by F10 key)")
        CALL PrintText(13, 5, 15, 11, "RUN STATUS")
        CALL PrintText(9, 8, 0, 11, "Estimate Working
(%)")
        CALL PrintText(8, 8, 1, 11, "Ú")
        CALL PrintText(8, 9, 1, 11, "ÀÄÄ" + CHR$(16))
        CALL PrintText(9, 10, 0, 11, "Monomer
Conversion(%)")
        CALL PrintText(8, 10, 1, 11, "Ú")
        CALL PrintText(8, 11, 1, 11, "ÀÄÄ" +
CHR$(16))
        CALL PrintText(9, 12, 0, 11, "Chain Transfer
Conv(%)")
        CALL PrintText(8, 12, 1, 11, "U")
        CALL PrintText(8, 13, 1, 11, "ÀAA" +
CHR$(16))
        SHELL "CALQWW.EXE"
        SHELL "SUCCESS EXE"
        SCREEN 12: SCREEN 0, 0
        SetScreen
        PrintBar
        CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
        CALL PrintText(10, 22, 10, 9, "Press <Enter> To
Select, U&D Arrow Key To Move Or Esc To Abort")
    CASE 2
        HelpForInputValue
        CALL PrintText(60, 7, 15, 9, "       ")
        LOCATE 7, 60
        INPUT "", Yscale
        IF Yscale <= 0 THEN Yscale = 10
    CASE 3
        HelpForInputValue
        CALL PrintText(60, 8, 15, 9, "       ")
        LOCATE 8, 60
        INPUT "", Xscale
        IF Xscale <= 0 THEN Xscale = 10
    CASE 4
        HelpForInputValue
        CALL PrintText(60, 9, 15, 9, "       ")
        LOCATE 9, 60
        INPUT "", StepTime
        IF StepTime <= 0 THEN StepTime = 20
    CASE 5
        HelpForInputValue
        CALL PrintText(60, 10, 15, 9, "       ")
```

Copyright by Mahidol University

```
            LOCATE 10, 60
            INPUT "", NumberPoint
            IF NumberPoint <= 0 THEN NumberPoint = 150
        CASE 6
            ConfirmBoxM
            CALL PrintText(14, 13, 6, 11, "CONFIRM !")
            CALL PrintText(11, 14, 0, 11, "Manaul
input<M>")
            CALL PrintText(11, 15, 9, 11, "Edit in ED  <E>")
            DO
            EDrate$ = UCASE$(INPUT$(1))
            LOOP WHILE INSTR("ME", EDrate$) = 0
            SELECT CASE EDrate$
                CASE "M"
                    CALL PrintText(8, 17, 15, 11, "3
Manaul input   3")
                    ManaulInput
                CASE "E"
                    CALL PrintText(8, 17, 15, 11, "3     Edit
in ED    3")
            HelpForEditRate
            SentTemp
            SHELL "ED.EXE"
          OPEN "ACEDIT.AV" FOR INPUT AS #22
            INPUT #22, MacceptText$
            INPUT #22, IacceptText$
            INPUT #22, CacceptText$
          CLOSE #22
          OPEN "SENTMIAN.CTN" FOR INPUT AS #21
          SELECT CASE MacceptText$
            CASE "Y"
            INPUT #21, Kp
            INPUT #21, Ktd
            INPUT #21, Ktc
            INPUT #21, Ktrm
          END SELECT
          SELECT CASE IacceptText$
            CASE "Y"
            INPUT #21, Ki
            INPUT #21, Kd
            INPUT #21, f
            INPUT #21, Ktci
          END SELECT
          SELECT CASE CacceptText$
            CASE "Y"
            INPUT #21, Ksr
            INPUT #21, Ktr
          END SELECT
          CLOSE #21
            FOR i = 0 TO 23
                COLOR 0, 0
                LOCATE 1 + i, 1: PRINT STRING$(80, " ")
            NEXT i
            SetScreen
            DBackBox
            DBox
            PrintBar

    END SELECT
    CASE 7
        ConfirmBox
        CALL PrintText(14, 13, 6, 11, "WARNING !")
        CALL PrintText(11, 14, 0, 11, "YOU WANT TO
ADD")
        CALL PrintText(13, 15, 0, 11, "GEL EFFECTS")
        DO
        XGelEffect$ = UCASE$(INPUT$(1))
        LOOP WHILE INSTR("YN", XGelEffect$) = 0
        SELECT CASE XGelEffect$
            CASE "Y"
                GelEffect$ = "ON"
            CASE "N"
                GelEffect$ = "OFF"
        END SELECT
    END SELECT
END SUB

SUB RunSubMenu1
    fc = 0: bc = 11
    CALL PrintText(5, 5, 15, bc, "ITERATIVE VALUE
FOR ROUTINE")
    CALL PrintText(4, 7, fc, bc, "   Start to Simulation
")
    CALL PrintText(4, 8, fc, bc, "   Y-axis Scale          ")
    CALL PrintText(4, 9, fc, bc, "   X-axis Scale          ")
    CALL PrintText(4, 10, fc, bc, "   Increment Time
")
    CALL PrintText(4, 11, fc, bc, "   Number of Data
")
    CALL PrintText(4, 12, fc, bc, "   Edit Rate Coefficient
")
    CALL PrintText(4, 13, fc, bc, "   Edit Gel Effect
")
END SUB

SUB RunSubMenu2
    CALL PrintText(40, 6, 15, 11, " RUN STATUS     ")
    CALL PrintText(43, 7, 14, 0, "Y-axis Scale
Time of Wn")
    CALL PrintText(43, 8, 14, 0, "X-axis Scale
Log(NCL) ")
    CALL PrintText(43, 9, 14, 0, "Iterative Time
Sec")
    CALL PrintText(43, 10, 14, 0, "Number of Data
Point")
END SUB

SUB RunSubMenu3
        COLOR 15, 0
        LOCATE 7, 61: PRINT USING "###.##"; Yscale
        LOCATE 8, 61: PRINT USING "### ##"; Xscale
        LOCATE 9, 60: PRINT USING "#### ##";
StepTime
        LOCATE 10, 62: PRINT USING "#####";
NumberPoint
END SUB

SUB RunSubMenu4
    CALL PrintText(43, 11, 14, 0, "Propagate ,Kp
l/mol.s")
    CALL PrintText(43, 12, 14, 0, "Terminate ,Ktd
l/mol.s")
    CALL PrintText(43, 13, 14, 0, "Terminate ,Ktc
l/mol s")
    CALL PrintText(43, 14, 14, 0, "Transfer ,Ktrm
l/mol.s")
    CALL PrintText(43, 15, 14, 0, "Initiate  ,Ki
l/mol.s")
    CALL PrintText(43, 16, 14, 0, "Dissociate,Kd
1/sec")
    CALL PrintText(43, 17, 14, 0, "efficiency,f  :
-  ")
    CALL PrintText(43, 18, 14, 0, "1øTermina ,Ktci
l/mol s")
    CALL PrintText(43, 19, 14, 0, "CTAøPrepare,Ksr:
l/mol.s")
    CALL PrintText(43, 20, 14, 0, "Transfer  ,Ktr :
l/mol.s")

END SUB

SUB RunSubMenu5
        COLOR 15, 0
```

Copyright by Mahidol University

```
        LOCATE 11, 60: PRINT USING "##.###^^^^ ";
Kp
        LOCATE 12, 60: PRINT USING "## ###^^^^ ",
Ktd
        LOCATE 13, 60. PRINT USING "##.###^^^^ ",
Ktc
        LOCATE 14, 60  PRINT USING "## ###^^^^ ",
Ktrm
        LOCATE 15, 60  PRINT USING "##.###^^^^ ";
Ki
        LOCATE 16, 60  PRINT USING "## ###^^^^ ",
Kd
        LOCATE 17, 60  PRINT USING "##.####    "; f
        LOCATE 18, 60  PRINT USING "##.###^^^^ ",
Ktci
        LOCATE 19, 60. PRINT USING "##.###^^^^ ";
Ksr
        LOCATE 20, 60  PRINT USING "##.###^^^^ ",
Ktr

END SUB

SUB Rxxx (num, fc, bc)
      IF num < 1 THEN num = 1
      IF num > 7 THEN num = 7
      SELECT CASE num
          CASE 1
              CALL PrintText(4, 7, fc, bc, "    Start to
Simulation     ")
          CASE 2
              CALL PrintText(4, 8, fc, bc, "    Y-axis Scale
")
          CASE 3
              CALL PrintText(4, 9, fc, bc, "    X-axis Scale
")
          CASE 4
              CALL PrintText(4, 10, fc, bc, "    Increment
Time        ")
          CASE 5
              CALL PrintText(4, 11, fc, bc, "    Number of
Data        ")
          CASE 6
              CALL PrintText(4, 12, fc, bc, "    Edit Rate
Coefficient   ")
          CASE 7
              CALL PrintText(4, 13, fc, bc, "    Edit Gel
Effect       ")
              IF GelEffect$ = "ON" THEN
              CALL PrintText(30, 13, 9, bc, "ON")
              ELSE
              CALL PrintText(30, 13, 9, bc, "OFF")
              END IF
      END SELECT
END SUB

SUB SaveCondFile
IF SaveCondDrive$ <> "" AND SaveCondName$ <> "" 
THEN
   OPEN SaveCondDrive$ + "\" + SaveCondName$ +
".CNW" FOR OUTPUT AS #1
        PRINT #1, MoName$
        PRINT #1, IoName$
        PRINT #1, SoName$
        PRINT #1, Mo
        PRINT #1, Io
        PRINT #1, CTA
        PRINT #1, Temp
        PRINT #1, Time
        PRINT #1, Kp
        PRINT #1, Ktc
        PRINT #1, Ktd
        PRINT #1, Ktrm
        PRINT #1, Ki
        PRINT #1, Kd
        PRINT #1, f
        PRINT #1, Ktci
        PRINT #1, Ksr
        PRINT #1, Ktr
        PRINT #1, NumberPoint
        PRINT #1, StepTime
        PRINT #1, Xscale
        PRINT #1, Yscale
        PRINT #1, GelEffect$
        PRINT #1, XoCOOR
        PRINT #1, YoCOOR
        PRINT #1, XaxisLength
        PRINT #1, YaxisLength
        PRINT #1, ZaxisLength
        PRINT #1, LeftAngle
        PRINT #1, RightAngle
        PRINT #1, NCLstep
        PRINT #1, THRstep
   CLOSE #1
   END IF
   END SUB

   SUB SentConstant
     OPEN "CONST.MTX" FOR OUTPUT AS #3
        PRINT #3, MoName$
        PRINT #3, IoName$
        PRINT #3, SoName$
        PRINT #3, Mo
        PRINT #3, Io
        PRINT #3, CTA
        PRINT #3, Temp
        PRINT #3, Time
        PRINT #3, Kp
        PRINT #3, Ktc
        PRINT #3, Ktd
        PRINT #3, Ktrm
        PRINT #3, Ki
        PRINT #3, Kd
        PRINT #3, f
        PRINT #3, Ktci
        PRINT #3, Ksr
        PRINT #3, Ktr
        PRINT #3, NumberPoint
        PRINT #3, StepTime
        PRINT #3, Xscale
        PRINT #3, Yscale
        PRINT #3, GelEffect$
        PRINT #3, XoCOOR
        PRINT #3, YoCOOR
        PRINT #3, XaxisLength
        PRINT #3, YaxisLength
        PRINT #3, ZaxisLength
        PRINT #3, LeftAngle
        PRINT #3, RightAngle
        PRINT #3, NCLstep
        PRINT #3, THRstep
     CLOSE #3
   END SUB

   SUB SentTemp
     OPEN "SENTED.CTN" FOR OUTPUT AS #20
        PRINT #20, Temp
     CLOSE #20
   END SUB

   SUB SetScreen
   COLOR 0, 13: CLS
   FOR i = 0 TO 18
        LOCATE 3 + i, 1: COLOR 15, 0: PRINT
   STRING$(80, CHR$(219))
```

Copyright by Mahidol University

```
NEXT i
CALL PrintSameText(1, 1, 80, 0, 15, 4, " ")
CALL PrintSameText(1, 2, 80, 0, 15, 3, " ")
CALL PrintSameText(1, 23, 80, 0, 15, 3, " ")
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(1, "WEIGHT DISTRIBUTION
SIMULATION", 15, 4)
CALL PrintText(38, 4, 14, 1, "        INFORMATION &
SUMARY        ")
CALL PrintText(2, 23, 5, 3, "Action:- <enter>=select item
<arrow key>=move   <escape>=abort")
END SUB

SUB Sxxx (num, fc, bc)
      IF num < 1 THEN num = 1
      IF num > 2 THEN num = 2
      SELECT CASE num
          CASE 1
              CALL PrintText(4, 7, fc, bc, "   Plot 2-D
Weight Distrib.  ")
          CASE 2
              CALL PrintText(4, 8, fc, bc, "   Plot 3-D
Weight Distrib.  ")
      END SELECT

END SUB

SUB UtilMenu1
     CALL PrintText(4, 7, 0, 11, "   File Name            ")
     CALL PrintText(4, 8, 0, 11, "   Path Name            ")
     CALL PrintText(4, 9, 0, 11, "   Go                 ")
END SUB

SUB Utilxxx (num, fc, bc)
      IF num < 1 THEN num = 1
      IF num > 3 THEN num = 3
      SELECT CASE num
          CASE 1
          CALL PrintText(4, 7, fc, bc, "   File Name      ")
          CASE 2
          CALL PrintText(4, 8, fc, bc, "   Path Name      ")
          CASE 3
          CALL PrintText(4, 9, fc, bc, "   Go             ")
      END SELECT
END SUB
```

Copyright by Mahidol University

3) The menu programmes for the Modified Mayo Equation (MTM.BAS)

```
DECLARE SUB ConfirmBoxM ()
DECLARE SUB ManaulInput ()
DECLARE SUB Manaulxxx (num!, fc!, bc!)
DECLARE SUB ManaulMenuSelect (num!)
DECLARE SUB CondLoadMenu ()
DECLARE SUB CondSaveMenu ()
DECLARE SUB CurveLoadMenu ()
DECLARE SUB CurveSaveMenu ()
DECLARE SUB CurveLoadMenuSelect (num!)
DECLARE SUB CondSaveMenuSelect (num!)
DECLARE SUB CondSaveMenu1 ()
DECLARE SUB CondLoadMenu1 ()
DECLARE SUB CurveLoadMenu1 ()
DECLARE SUB CondLoadMenuSelect (num!)
DECLARE SUB UtilMenu1 ()
DECLARE SUB CurveSaveMenu1 ()
DECLARE SUB Utilxxx (num!, fc!, bc!)
DECLARE SUB CurveSaveMenuSelect (num!)
DECLARE SUB Center (row!, Text$, Tcolor!, Bcolor!)
DECLARE SUB ResultMenu ()
DECLARE SUB Sxxx (num!, fc!, bc!)
DECLARE SUB ResultMenuSelect (num!)
DECLARE SUB ResultSubMenu1 ()
DECLARE SUB ResultSubMenu2 ()
DECLARE SUB SentTemp ()
DECLARE SUB HelpForInputValue ()
DECLARE SUB HelpForInputText ()
DECLARE SUB HelpForDrive ()
DECLARE SUB HelpForFile ()
DECLARE SUB HelpForEditRate ()
DECLARE SUB RunSubMenu5 ()
DECLARE SUB ParameterSubMenu3 ()
DECLARE SUB SentConstant ()
DECLARE SUB RunSubMenu3 ()
DECLARE SUB RunMenuSelect (num!)
DECLARE SUB FileSubMenu4 (YLL!, Text$)
DECLARE SUB LoadCondFile ()
DECLARE SUB ParaMenuSelect (num!)
DECLARE SUB RunSubMenu4 ()
DECLARE SUB SaveCondFile ()
DECLARE SUB LoadFile ()
DECLARE SUB FileSubMenu3 ()
DECLARE SUB ConfirmBox ()
DECLARE SUB FileMenuSelect (num!)
DECLARE SUB RunSubMenu2 ()
DECLARE SUB FileSubMenu2 ()
DECLARE SUB ParameterSubMenu2 ()
DECLARE SUB ClearScreen (xc!)
DECLARE SUB DBox ()
DECLARE SUB DBackBox ()
DECLARE SUB RunSubMenu1 ()
DECLARE SUB ParameterSubMenu1 ()
DECLARE SUB ParameterMenu ()
DECLARE SUB RunMenu ()
DECLARE SUB FileSubMenu1 ()
DECLARE SUB FileMenu ()
DECLARE SUB PrintBar ()
DECLARE SUB SetScreen ()
DECLARE SUB PrintText (XL!, YL!, fc!, bc!, Text$)
DECLARE SUB PrintSameText (x1!, y1!, lx!, ly!, fc!, bc!,
Text$)
DECLARE SUB Fxxx (num!, fc!, bc!)
DECLARE SUB Pxxx (num!, fc!, bc!)
DECLARE SUB Rxxx (num!, fc!, bc!)
DECLARE SUB Mxxx (num!, bc!)
COMMON SHARED MoName$, IoName$, SoName$
COMMON SHARED Mo, Io, Temp, Time
COMMON SHARED CTAo1, CTAo2, PercentSampling
COMMON SHARED Kp, Ktd, Ktc, Ktrm, Ki, Kd, f, Ktci,
Ksr, Ktr
COMMON SHARED numberDATA, StepTime,
numberNCL, Xscale
COMMON SHARED SaveCondDrive$, SaveCondName$,
LoadCondDrive$
COMMON SHARED SaveCurveDrive$,
SaveCurveName$, LoadCurveDrive$, LoadCurveName$
COMMON SHARED OpenCondName$
COMMON SHARED GelEffect$
CLS
SetScreen
PrintBar
Xscale = 8
StepTime = 20
numberNCL = 150
numberDATA = 50
PercentSampling = 5
GelEffect$ = "OFF"
ColCount = 0
DO
  DO
  FOR i = 0 TO 18
      LOCATE 3 + i, 1: COLOR 15, 0: PRINT
STRING$(80, CHR$(219))
  NEXT i
  CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
  CALL Center(22, "Press <Enter> To Select, Right & Left
Arrow Key To Move Or Esc To Abort", 15, 1)
    CALL Mxxx(ColCount, 8)
    DO
       kbd$ = INKEY$
       LOCATE 23, 72: COLOR 15, 3. PRINT TIME$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "K"     'Left Arrow
             ColCount = ColCount - 1
             CALL Mxxx(ColCount + 1, 3)
             IF ColCount < 1 THEN ColCount = 5
        CASE CHR$(0) + "M"     'Right Arrow
             ColCount = ColCount + 1
             CALL Mxxx(ColCount - 1, 3)
             IF ColCount > 5 THEN ColCount = 1
        CASE CHR$(27)
             CLS : END
    END SELECT
    LOOP UNTIL kbd$ = CHR$(13)
    SELECT CASE ColCount
        CASE 1
           FileMenu
        CASE 2
           ParameterMenu
        CASE 3
           RunMenu
        CASE 5
           ResultMenu
    END SELECT
LOOP

SUB Center (row, Text$, Tcolor, Bcolor)
  COLOR Tcolor, Bcolor
  LOCATE row, 41 - LEN(Text$) / 2
  PRINT Text$;
END SUB

SUB ClearScreen (xc)
    CALL PrintSameText(38, 5, 42, xc, 0, 0, " ")
END SUB

SUB CondLoadMenu
```

Copyright by Payap University

```
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
      DBackBox
      DBox
      CondLoadMenu1
      RowCount = 0
      DO
       CALL Utilxxx(RowCount, 15, 8)
      DO
           kbd$ = INKEY$
           LOCATE 23, 72. COLOR 15, 11: PRINT TIME$
      LOOP UNTIL kbd$ <> ""
      SELECT CASE kbd$
           CASE CHR$(0) + "H"     'Up Arrow
                RowCount = RowCount - 1
                CALL Utilxxx(RowCount + 1, 0, 3)
                IF RowCount < 1 THEN RowCount = 3
           CASE CHR$(0) + "P"     'Down Arrow
                RowCount = RowCount + 1
                CALL Utilxxx(RowCount - 1, 0, 3)
                IF RowCount > 3 THEN RowCount = 1
      END SELECT
      LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
      IF kbd$ = CHR$(13) THEN CALL
CondLoadMenuSelect(RowCount)
   LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB CondLoadMenu1
      CALL PrintText(5, 5, 15, 11, "LOAD PARAMETER
& CONDITION")
      UtilMenu1
END SUB

SUB CondLoadMenuSelect (num)
   SELECT CASE num
      CASE 2
           HelpForDrive
           CALL PrintText(57, 8, 15, 9, "            ")
           LOCATE 8, 60
           INPUT "", LoadCondDrive$
           IF LoadCondDrive$ = "" THEN LoadCondDrive$
= "C:\SIMULATE\PACKAGE"
           CALL PrintText(57, 8, 0, 0, "                ")
           CALL PrintText(57, 8, 15, 0, LoadCondDrive$)
      CASE 3
           ImageDrive$ = LEFT$(LoadCondDrive$, 2)
           SHELL ImageDrive$
           SHELL "dir * CNM /B/S/O" + " > " +
"ROOT.SCN"
           OPEN "Mdrive DV" FOR OUTPUT AS #12
              PRINT #12, ImageDrive$
           CLOSE #12
           SHELL "MTF EXE"
           OPEN "Ldrive.DV" FOR INPUT AS #13
                INPUT #13, OCN$
           CLOSE #13
           IF OCN$ <> "" THEN
                OpenCondName$ = OCN$
                LoadCondFile
           END IF
           SetScreen
           PrintBar
           FOR i = 0 TO 18
                LOCATE 3 + i, 1: COLOR 15, 0: PRINT
STRING$(80, CHR$(219))
           NEXT i
           CALL ClearScreen(13)
           CALL PrintText(38, 4, 14, 1, "
INFORMATION  & SUMARY          ")
           FileSubMenu2
           FileSubMenu3
   END SELECT
END SUB

SUB CondSaveMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
      DBackBox
      DBox
      CondSaveMenu1
      RowCount = 0
      DO
       CALL Utilxxx(RowCount, 15, 8)
      DO
           kbd$ = INKEY$
           LOCATE 23, 72: COLOR 15, 11. PRINT TIME$
      LOOP UNTIL kbd$ <> ""
      SELECT CASE kbd$
           CASE CHR$(0) + "H"     'Up Arrow
                RowCount = RowCount - 1
                CALL Utilxxx(RowCount + 1, 0, 3)
                IF RowCount < 1 THEN RowCount = 3
           CASE CHR$(0) + "P"     'Down Arrow
                RowCount = RowCount + 1
                CALL Utilxxx(RowCount - 1, 0, 3)
                IF RowCount > 3 THEN RowCount = 1
      END SELECT
      LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
      IF kbd$ = CHR$(13) THEN CALL
CondSaveMenuSelect(RowCount)
   LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB CondSaveMenu1
      CALL PrintText(5, 5, 15, 11, "SAVE PARAMETER
& CONDITION")
      UtilMenu1
END SUB

SUB CondSaveMenuSelect (num)
   SELECT CASE num
      CASE 1
           REM Input Curve File name
           HelpForFile
           CALL PrintText(57, 10, 15, 9, "            ")
           LOCATE 10, 60
           INPUT "", SaveCondName$
           IF SaveCondName$ = "" THEN SaveCondName$
= "MTEC"
           CALL PrintText(57, 10, 0, 0, "            ")
           CALL PrintText(57, 10, 15, 0, SaveCondName$)
      CASE 2
           HelpForDrive
           CALL PrintText(57, 11, 15, 9, "            ")
           LOCATE 11, 60
           INPUT "", SaveCondDrive$
           IF SaveCondDrive$ = "" THEN SaveCondDrive$
= "C:\SIMULATE\PACKAGE"
           CALL PrintText(57, 11, 0, 0, "            ")
           CALL PrintText(57, 11, 15, 0, SaveCondDrive$)
      CASE 3
           IF SaveCondName$ <> "" AND SaveCondDrive$
<> "" THEN
```

Copyright by Mahidol University

```
            SaveCondFile
        END IF
    END SELECT

END SUB

SUB ConfirmBox

    CALL PrintText(8, 12, 15, 11, " ----------")
    CALL PrintText(8, 13, 15, 11, "³          ³")
    CALL PrintText(8, 14, 15, 11, "³          ³")
    CALL PrintText(8, 15, 15, 11, "³          ³")
    CALL PrintText(8, 16, 15, 11, " ----------")
    CALL PrintText(8, 17, 15, 11, "³ Yes or No ³")
    CALL PrintText(8, 18, 15, 11, " ----------")
    CALL PrintText(14, 17, 16, 11, "Y")
    CALL PrintText(21, 17, 16, 11, "N")
END SUB

SUB ConfirmBoxM
    CALL PrintText(8, 12, 15, 11, " ----------")
    CALL PrintText(8, 13, 15, 11, "³          ³")
    CALL PrintText(8, 14, 15, 11, "³          ³")
    CALL PrintText(8, 15, 15, 11, "³          ³")
    CALL PrintText(8, 16, 15, 11, " ----------")
    CALL PrintText(8, 17, 15, 11, "³ Manaul or Edit ³")
    CALL PrintText(8, 18, 15, 11, " ----------")
    CALL PrintText(12, 17, 16, 11, "M")
    CALL PrintText(22, 17, 16, 11, "E")

END SUB

SUB CurveLoadMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
    DBackBox
    DBox
    CurveLoadMenu1
    RowCount = 0
    DO
     CALL Utilxxx(RowCount, 15, 8)
    DO
        kbd$ = INKEY$
        LOCATE 23, 72: COLOR 15, 11  PRINT TIME$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"     'Up Arrow
            RowCount = RowCount - 1
            CALL Utilxxx(RowCount + 1, 0, 3)
            IF RowCount < 1 THEN RowCount = 3
        CASE CHR$(0) + "P"     'Down Arrow
            RowCount = RowCount + 1
            CALL Utilxxx(RowCount - 1, 0, 3)
            IF RowCount > 3 THEN RowCount = 1
    END SELECT
    LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
    IF kbd$ = CHR$(13) THEN CALL
CurveLoadMenuSelect(RowCount)
  LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB CurveLoadMenu1
    CALL PrintText(9, 5, 15, 11, "LOAD SIMULATED
CURVE")
    UtilMenu1
END SUB

SUB CurveLoadMenuSelect (num)
  SELECT CASE num
    CASE 2
        HelpForDrive
        CALL PrintText(57, 17, 15, 9, "          ")
        LOCATE 17, 60
        INPUT "", LoadCurveDrive$
        IF LoadCurveDrive$ = "" THEN
LoadCurveDrive$ = "C:\SIMULATE\PACKAGE"
        CALL PrintText(57, 17, 0, 0, "          ")
        CALL PrintText(57, 17, 15, 0, LoadCurveDrive$)
    CASE 3
        ImageDrive$ = LEFT$(LoadCurveDrive$, 2)
        SHELL ImageDrive$
        SHELL "dir *.GPP /B/S/O" + " > " +
"ROOT SCN"
        OPEN "Mdrive.DV" FOR OUTPUT AS #12
            PRINT #12, ImageDrive$
        CLOSE #12
        SHELL "MTF EXE"
        OPEN "Ldrive.DV" FOR INPUT AS #13
            INPUT #13, OCN$
        CLOSE #13
        IF OCN$ <> "" THEN
            LoadCurveName$ = OCN$
            OPEN "CURVE.LV" FOR OUTPUT AS #22
                PRINT #22, LoadCurveName$
                PRINT #22, LoadCurveDrive$
            CLOSE #22
        SHELL "MTLM.EXE"
        END IF
        SetScreen
        PrintBar
        FOR i = 0 TO 18
            LOCATE 3 + i, 1  COLOR 15, 0: PRINT
STRING$(80, CHR$(219))
        NEXT i
        CALL ClearScreen(13)
        CALL PrintText(38, 4, 14, 1, "
INFORMATION  & SUMARY          ")
        FileSubMenu2
        FileSubMenu3
  END SELECT
END SUB

SUB CurveSaveMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
    DBackBox
    DBox
    CurveSaveMenu1
    RowCount = 0
    DO
     CALL Utilxxx(RowCount, 15, 8)
    DO
        kbd$ = INKEY$
        LOCATE 23, 72: COLOR 15, 11: PRINT TIME$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"     'Up Arrow
            RowCount = RowCount - 1
            CALL Utilxxx(RowCount + 1, 0, 3)
            IF RowCount < 1 THEN RowCount = 3
        CASE CHR$(0) + "P"     'Down Arrow
            RowCount = RowCount + 1
            CALL Utilxxx(RowCount - 1, 0, 3)
            IF RowCount > 3 THEN RowCount = 1
    END SELECT
```

Copyright by Mahidol University

```
    LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
    IF kbd$ = CHR$(13) THEN CALL
CurveSaveMenuSelect(RowCount)
  LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB CurveSaveMenu1
    CALL PrintText(9, 5, 15, 11, "SAVE SIMULATED
CURVE")
    UtilMenu1
END SUB

SUB CurveSaveMenuSelect (num)
  SELECT CASE num
    CASE 1
        REM Input Curve File name
        HelpForFile
        CALL PrintText(57, 13, 15, 9, "          ")
        LOCATE 13, 60
        INPUT "", SaveCurveName$
        IF SaveCurveName$ = "" THEN
SaveCurveName$ = "MTEC"
        CALL PrintText(57, 13, 0, 0, "          ")
        CALL PrintText(57, 13, 15, 0, SaveCurveName$)
    CASE 2
        HelpForDrive
        CALL PrintText(57, 14, 15, 9, "          ")
        LOCATE 14, 60
        INPUT "", SaveCurveDrive$
        IF SaveCurveDrive$ = "" THEN
SaveCurveDrive$ = "C:\SIMULATE\PACKAGE"
        CALL PrintText(57, 14, 0, 0, "          ")
        CALL PrintText(57, 14, 15, 0, SaveCurveDrive$)
    CASE 3
        IF SaveCurveName$ <> "" AND
SaveCurveDrive$ <> "" THEN
        REM extension " GPP" , consist with file extension
".GM1" to " GM2"
        OPEN "CURVE.SV" FOR OUTPUT AS #21
            PRINT #21, SaveCurveName$
            PRINT #21, SaveCurveDrive$
        CLOSE #21
        SHELL "MTSM.EXE"
        END IF
  END SELECT
END SUB

SUB DBackBox
    CALL PrintSameText(5, 5, 31, 15, 11, 8, " ")
END SUB

SUB DBox
    CALL PrintSameText(3, 4, 30, 15, 0, 11, " ")
    CALL PrintSameText(3, 4, 30, 0, 0, 11, "Ä")
    CALL PrintSameText(3, 4 + 2, 30, 0, 0, 11, "Ä")
    CALL PrintSameText(3, 4 + 15, 30, 0, 0, 11, "Ä")
    FOR k = 0 TO 15
        CALL PrintSameText(3, 4 + k, 1, 0, 0, 11, "³")
        CALL PrintSameText(3 + 30, 4 + k, 1, 0, 0, 11,
"³")
    NEXT k
    CALL PrintSameText(3, 4, 1, 0, 0, 11, "Ú")
    CALL PrintSameText(3 + 30, 4, 1, 0, 0, 11, "¿")
    CALL PrintSameText(3, 4 + 2, 1, 0, 0, 11, "Ã")
    CALL PrintSameText(3 + 30, 4 + 2, 1, 0, 0, 11, "´")
    CALL PrintSameText(3, 4 + 15, 1, 0, 0, 11, "À")
    CALL PrintSameText(3 + 30, 4 + 15, 1, 0, 0, 11, "Ù")
END SUB

SUB FileMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 1)
    DBackBox
    DBox
    ClearScreen (13)
    CALL PrintText(38, 4, 14, 1, "        INFORMATION
& SUMARY        ")
    CALL PrintText(5, 2, 15, 8, " File ")
    CALL PrintText(5, 2, 10, 8, " F")
    FileSubMenu1
    FileSubMenu2
    FileSubMenu3
    RowCount = 0
    DO
     CALL Fxxx(RowCount, 15, 8)
    DO
        kbd$ = INKEY$
        LOCATE 23, 72: COLOR 15, 3: PRINT TIME$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"     'Up Arrow
            RowCount = RowCount - 1
            CALL Fxxx(RowCount + 1, 0, 3)
            IF RowCount < 1 THEN RowCount = 6
        CASE CHR$(0) + "P"     'Down Arrow
            RowCount = RowCount + 1
            CALL Fxxx(RowCount - 1, 0, 3)
            IF RowCount > 6 THEN RowCount = 1
    END SELECT
    LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
    IF kbd$ = CHR$(13) THEN CALL
FileMenuSelect(RowCount)
  LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB FileMenuSelect (num)
  SELECT CASE num
  CASE 1
        Temp = 60
        Time = 60000
        MoName$ = "PS"
        Mo = 8
        IoName$ = "BP"
        Io = .01
        SoName$ = "UNKNOWN"
        CTAo1 = .05
        CTAo2 =  5
    CASE 2
        CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
        CALL Center(22, "Press Y if you want to
continue, N if you want to reject", 14, 1)
        ConfirmBox
        CALL PrintText(14, 13, 6, 11, "CONFIRM !")
        CALL PrintText(11, 14, 0, 11, "YOUR
COMMAND IS")
        CALL PrintText(11, 15, 9, 11, "LOAD
PARAMETERS")
        DO
        CondLoading$ = UCASE$(INPUT$(1))
        LOOP WHILE INSTR("YN", CondLoading$) = 0
        CALL PrintText(8, 17, 15, 11, "³      Yes      ³")
        SELECT CASE CondLoading$
            CASE "Y"
                CondLoadMenu
        END SELECT
    CASE 3
```

Copyright by Mahidol University

```
        CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
        CALL Center(22, "Press Y if you want to
continue, N if you want to reject", 14, 1)
        ConfirmBox
        CALL PrintText(14, 13, 6, 11, "CONFIRM !")
        CALL PrintText(11, 14, 0, 11, "YOUR
COMMAND IS")
        CALL PrintText(11, 15, 9, 11, "SAVE
PARAMETERS")
        DO
        CondSaving$ = UCASE$(INPUT$(1))
        LOOP WHILE INSTR("YN", CondSaving$) = 0
        CALL PrintText(8, 17, 15, 11, "³      Yes      ³")
        SELECT CASE CondSaving$
            CASE "Y"
                CondSaveMenu
        END SELECT
    CASE 4
        CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
        CALL Center(22, "Press Y if you want to
continue, N if you want to reject", 14, 1)
        ConfirmBox
        CALL PrintText(14, 13, 6, 11, "CONFIRM !")
        CALL PrintText(11, 14, 0, 11, "YOUR
COMMAND IS")
        CALL PrintText(13, 15, 9, 11, "SAVE CURVE")
        DO
        CurveSaving$ = UCASE$(INPUT$(1))
        LOOP WHILE INSTR("YN", CurveSaving$) = 0
        CALL PrintText(8, 17, 15, 11, "³      Yes      ³")
        SELECT CASE CurveSaving$
            CASE "Y"
                CurveSaveMenu
        END SELECT
    CASE 5
        CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
        CALL Center(22, "Press Y if you want to
continue, N if you want to reject", 14, 1)
        ConfirmBox
        CALL PrintText(14, 13, 6, 11, "CONFIRM !")
        CALL PrintText(11, 14, 0, 11, "YOUR
COMMAND IS")
        CALL PrintText(13, 15, 9, 11, "LOAD CURVE")
        DO
        CurveLoading$ = UCASE$(INPUT$(1))
        LOOP WHILE INSTR("YN", CurveLoading$) =
0
        CALL PrintText(8, 17, 15, 11, "³      Yes      ³")
        SELECT CASE CurveLoading$
            CASE "Y"
                CurveLoadMenu
        END SELECT
    CASE 6
        CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
        CALL Center(22, "Press Y if you want to
continue, N if you want to reject", 14, 1)
        ConfirmBox
        CALL PrintText(14, 13, 6, 11, "WARNING !")
        CALL PrintText(11, 14, 0, 11, "YOUR
COMMAND IS")
        CALL PrintText(13, 15, 0, 11, "EXIT TO DOS")
        DO
        ChangeYourMine$ = UCASE$(INPUT$(1))
        LOOP WHILE INSTR("YN",
ChangeYourMine$) = 0
        SELECT CASE ChangeYourMine$
            CASE "Y"
                COLOR 15, 1: CLS : END
        END SELECT
  END SELECT
END SUB

SUB FileSubMenu1
    CALL PrintText(10, 5, 15, 11, "FILE MANAGER
MENU")
    CALL PrintText(4, 7, 0, 11, "   New Parameters
")
    CALL PrintText(4, 8, 0, 11, "   Load Parameters
")
    CALL PrintText(4, 9, 0, 11, "   Save Parameters      ")
    CALL PrintText(4, 10, 0, 11, "   Save Simulation
")
    CALL PrintText(4, 11, 0, 11, "   Load Simulation
")
    CALL PrintText(4, 12, 0, 11, "   Exit to DOS          ")
END SUB

SUB FileSubMenu2
    CALL PrintText(40, 6, 15, 11, " PARAMETER
STATUS(NOW)")
    CALL PrintText(45, 7, 14, 0, "File Name :")
    CALL PrintText(45, 8, 14, 0, "Path Name :")
    CALL PrintText(40, 9, 15, 11, " SAVE
PARAMETERS FILE ")
    CALL PrintText(45, 10, 14, 0, "File Name :")
    CALL PrintText(45, 11, 14, 0, "Path Name :")
    CALL PrintText(40, 12, 15, 11, " SAVE
SIMULATED CURVE ")
    CALL PrintText(45, 13, 14, 0, "File Name :")
    CALL PrintText(45, 14, 14, 0, "Path Name :")
    CALL PrintText(40, 15, 15, 11, " LOAD
SIMULATED CURVE ")
    CALL PrintText(45, 16, 14, 0, "File Name :")
    CALL PrintText(45, 17, 14, 0, "Path Name :")
END SUB

SUB FileSubMenu3
    CALL PrintText(45, 7, 12, 0, OpenCondName$)
    CALL FileSubMenu4(10, SaveCondName$)
    CALL FileSubMenu4(11, SaveCondDrive$)
    CALL FileSubMenu4(13, SaveCurveName$)
    CALL FileSubMenu4(14, SaveCurveDrive$)
    CALL PrintText(45, 16, 12, 0, LoadCurveName$)
    CALL FileSubMenu4(17, LoadCurveDrive$)
END SUB

SUB FileSubMenu4 (YLL, Text$)
    IF Text$ <> "" THEN
        CALL PrintText(57, YLL, 15, 0, Text$)
    ELSE
        CALL PrintText(57, YLL, 15, 0, "Untitle")
    END IF
END SUB

SUB Fxxx (num, fc, bc)
     IF num < 1 THEN num = 1
     IF num > 6 THEN num = 6
     SELECT CASE num
         CASE 1
             CALL PrintText(4, 7, fc, bc, "   New
Parameters      ")
         CASE 2
             CALL PrintText(4, 8, fc, bc, "   Load
Parameters      ")
         CASE 3
             CALL PrintText(4, 9, fc, bc, "   Save
Parameters      ")
         CASE 4
             CALL PrintText(4, 10, fc, bc, "   Save
Simulation      ")
         CASE 5
             CALL PrintText(4, 11, fc, bc, "   Load
Simulation      ")
         CASE 6
```

Copyright by Mahidol University

```
            CALL PrintText(4, 12, fc, bc, "    Exit to
DOS          ")
      END SELECT
END SUB

SUB HelpForDrive
      CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
      CALL PrintText(21, 22, 15, 9, "Input!  Directory Or
Enter To Use Defualt.")
END SUB

SUB HelpForEditRate
      CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
      CALL PrintText(30, 22, 15, 9, "Enter to Edit Values")
END SUB

SUB HelpForFile
      CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
      CALL PrintText(22, 22, 15, 9, "Input!  File Name
[Maximum 8 Character].")
END SUB

SUB HelpForInputText
      CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
      CALL PrintText(20, 22, 15, 9, "Input! Text Or Enter
If You don't known ")
END SUB

SUB HelpForInputValue
      CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
      CALL PrintText(30, 22, 15, 9, "Input! Numeric
Value")
END SUB

SUB LoadCondFile
  OPEN OpenCondName$ FOR INPUT AS #1
      INPUT #1, MoName$
      INPUT #1, IoName$
      INPUT #1, SoName$
      INPUT #1, Mo
      INPUT #1, Io
      INPUT #1, CTAo1
      INPUT #1, CTAo2
      INPUT #1, PercentSampling
      INPUT #1, Temp
      INPUT #1, Time
      INPUT #1, Kp
      INPUT #1, Ktc
      INPUT #1, Ktd
      INPUT #1, Ktrm
      INPUT #1, Ki
      INPUT #1, Kd
      INPUT #1, f
      INPUT #1, Ktci
      INPUT #1, Ksr
      INPUT #1, Ktr
      INPUT #1, numberNCL
      INPUT #1, Xscale
      INPUT #1, StepTime
      INPUT #1, numberDATA
      INPUT #1, GelEffect$
  CLOSE #1
END SUB

SUB LoadFile
        CALL PrintText(57, 8, 15, 9, "          ")
        LOCATE 8, 60
        INPUT "", DataDrive$
        ImageDrive$ = LEFT$(DataDrive$, 2)
        SHELL ImageDrive$
        SHELL "dir *.CNM /B/S/O" + " > " +
"ROOT.SCN"
        OPEN "Mdrive.DV" FOR OUTPUT AS #12
            PRINT #12, ImageDrive$
        CLOSE #12
        SHELL "MTF.EXE"
END SUB

SUB ManaulInput
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 9)
    RunSubMenu4
    RunSubMenu5
    RowCount = 0
    DO
     CALL ManaulXXX(RowCount, 15, 9)
    DO
        kbd$ = INKEY$
        LOCATE 23, 72: COLOR 15, 11: PRINT TIME$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"     'Up Arrow
            RowCount = RowCount - 1
            CALL ManaulXXX(RowCount + 1, 15, 0)
            IF RowCount < 1 THEN RowCount = 10
        CASE CHR$(0) + "P"     'Down Arrow
            RowCount = RowCount + 1
            CALL ManaulXXX(RowCount - 1, 15, 0)
            IF RowCount > 10 THEN RowCount = 1
    END SELECT
    LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
    IF kbd$ = CHR$(13) THEN CALL
ManaulMenuSelect(RowCount)
  LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"

END SUB

SUB ManaulMenuSelect (num)
  SELECT CASE num
    CASE 1
        HelpForInputValue
        CALL PrintText(59, 11, 15, 9, "          ")
   bd$ = INKEY$TE 11, 60
        INPUT "", Kp
    CASE 2
        HelpForInputValue
        CALL PrintText(59, 12, 15, 9, "          ")
        LOCATE 12, 60
        INPUT "", Ktd
    CASE 3
        HelpForInputValue
        CALL PrintText(59, 13, 15, 9, "          ")
        LOCATE 13, 60
        INPUT "", Ktc
    CASE 4
        HelpForInputValue
        CALL PrintText(59, 14, 15, 9, "          ")
        LOCATE 14, 60
        INPUT "", Ktrm
    CASE 5
        HelpForInputValue
        CALL PrintText(59, 15, 15, 9, "          ")
        LOCATE 15, 60
        INPUT "", Ki
    CASE 6
        HelpForInputValue
        CALL PrintText(59, 16, 15, 9, "          ")
        LOCATE 16, 60
        INPUT "", Kd
```

Copyright by Mahidol University

```
    CASE 7
        HelpForInputValue
        CALL PrintText(59, 17, 15, 9, "          ")
        LOCATE 17, 60
        INPUT "", f
    CASE 8
        HelpForInputValue
        CALL PrintText(59, 18, 15, 9, "          ")
        LOCATE 18, 60
        INPUT "", Ktci
    CASE 9
        HelpForInputValue
        CALL PrintText(59, 19, 15, 9, "          ")
        LOCATE 19, 60
        INPUT "", Ksr
    CASE 10
        HelpForInputValue
        CALL PrintText(59, 20, 15, 9, "          ")
        LOCATE 20, 60
        INPUT "", Ktr
  END SELECT

END SUB

SUB Manaulxxx (num, fc, bc)
    IF num < 1 THEN num = 1
    IF num > 10 THEN num = 10
    SELECT CASE num
        CASE 1
            COLOR fc, bc
            LOCATE 11, 60: PRINT USING
"##.###^^^^ ", Kp
        CASE 2
            COLOR fc, bc
            LOCATE 12, 60: PRINT USING
"##.###^^^^ "; Ktd
        CASE 3
            COLOR fc, bc
            LOCATE 13, 60: PRINT USING
"##.###^^^^ "; Ktc
        CASE 4
            COLOR fc, bc
            LOCATE 14, 60: PRINT USING
"## ###^^^^ ", Ktrm
        CASE 5
            COLOR fc, bc
            LOCATE 15, 60: PRINT USING
"##.###^^^^ "; Ki
        CASE 6
            COLOR fc, bc
            LOCATE 16, 60. PRINT USING
"##.###^^^^ ", Kd
        CASE 7
            COLOR fc, bc
            LOCATE 17, 60: PRINT USING "##.####
"; f
        CASE 8
            COLOR fc, bc
            LOCATE 18, 60: PRINT USING
"##.###^^^^ "; Ktci
        CASE 9
            COLOR fc, bc
            LOCATE 19, 60: PRINT USING
"##.###^^^^ "; Ksr
        CASE 10
            COLOR fc, bc
            LOCATE 20, 60: PRINT USING
"##.###^^^^ "; Ktr
    END SELECT
END SUB

SUB Mxxx (num, bc)
    IF num < 1 THEN num = 1
    IF num > 5 THEN num = 5
    SELECT CASE num
        CASE 1
            CALL PrintText(5, 2, 15, bc, " File ")
            CALL PrintText(5, 2, 10, bc, " F")
        CASE 2
            CALL PrintText(13, 2, 15, bc, " Parameter ")
            CALL PrintText(13, 2, 10, bc, " P")
        CASE 3
            CALL PrintText(25, 2, 15, bc, " Run ")
            CALL PrintText(25, 2, 10, bc, " R")
        CASE 4
            CALL PrintText(31, 2, 15, bc, " Graphics ")
            CALL PrintText(31, 2, 10, bc, " G")
        CASE 5
            CALL PrintText(41, 2, 15, bc, " Simulated
result ")
            CALL PrintText(41, 2, 10, bc, " S")
    END SELECT
END SUB

SUB ParaMenuSelect (num)
  SELECT CASE num
    CASE 1
        HelpForInputValue
        CALL PrintText(62, 7, 15, 9, "          ")
        LOCATE 7, 62
        INPUT "", Temp
    CASE 2
        HelpForInputValue
        CALL PrintText(62, 8, 15, 9, "          ")
        LOCATE 8, 62
        INPUT "", Time
    CASE 3
        HelpForInputText
        CALL PrintText(62, 9, 15, 9, "          ")
        LOCATE 9, 62
        INPUT "", MoName$
        IF MoName$ = "" THEN MoName$ =
"UNKNOWN"
    CASE 4
        HelpForInputValue
        CALL PrintText(62, 10, 15, 9, "          ")
        LOCATE 10, 62
        INPUT "", Mo
        IF Mo = 0 THEN Mo = 8.728
    CASE 5
        HelpForInputText
        CALL PrintText(62, 11, 15, 9, "          ")
        LOCATE 11, 62
        INPUT "", IoName$
        IF IoName$ = "" THEN IoName$ =
"UNKNOWN"
    CASE 6
        HelpForInputValue
        CALL PrintText(62, 12, 15, 9, "          ")
        LOCATE 12, 62
        INPUT "", Io
        IF Io = 0 THEN Io = .01
    CASE 7
        HelpForInputText
        CALL PrintText(62, 13, 15, 9, "          ")
        LOCATE 13, 62
        INPUT "", SoName$
        IF SoName$ = "" THEN SoName$ =
"UNKNOWN"
    CASE 8
        HelpForInputValue
        CALL PrintText(62, 14, 15, 9, "          ")
        LOCATE 14, 62
        INPUT "", CTAo1
```

Copyright by Mahidol University

```
    CASE 9
        HelpForInputValue
        CALL PrintText(62, 15, 15, 9, "        ")
        LOCATE 15, 62
        INPUT "", CTAo2
    CASE 10
        HelpForInputValue
        CALL PrintText(62, 16, 15, 9, "        ")
        LOCATE 16, 62
        INPUT "", PercentSampling
        IF PercentSampling = 0 THEN PercentSampling =
5
  END SELECT
END SUB

SUB ParameterMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 1)
    DBackBox
    DBox
    CALL ClearScreen(12)
    CALL PrintText(38, 4, 14, 1, "        INFORMATION
& SUMARY        ")
    CALL PrintText(13, 2, 15, 8, " Parameter ")
    CALL PrintText(13, 2, 10, 8, " P")
    ParameterSubMenu1
    ParameterSubMenu2
    ParameterSubMenu3
    RowCount = 0
    DO
     CALL Pxxx(RowCount, 15, 8)
    DO
        kbd$ = INKEY$
        LOCATE 23, 72: COLOR 15, 3: PRINT TIME$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"     'Up Arrow
            RowCount = RowCount - 1
            CALL Pxxx(RowCount + 1, 0, 3)
            IF RowCount < 1 THEN RowCount = 10
        CASE CHR$(0) + "P"     'Down Arrow
            RowCount = RowCount + 1
            CALL Pxxx(RowCount - 1, 0, 3)
            IF RowCount > 10 THEN RowCount = 1
    END SELECT
    LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
    IF kbd$ = CHR$(13) THEN CALL
ParaMenuSelect(RowCount)
  LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB ParameterSubMenu1
    fc = 0: bc = 11
    CALL PrintText(7, 5, 15, bc, "PARAMETERS &
CONDITIONS")
    CALL PrintText(4, 7, fc, bc, "   Temperature          ")
    CALL PrintText(4, 8, fc, bc, "   Reaction Time
")
    CALL PrintText(4, 9, fc, bc, "   Monomer  Type
")
    CALL PrintText(4, 10, fc, bc, "   Monomer  Conc.
")
    CALL PrintText(4, 11, fc, bc, "   Initiator Type
")
    CALL PrintText(4, 12, fc, bc, "   Initiator Conc
")
    CALL PrintText(4, 13, fc, bc, "   Chain Transfer Type
")
    CALL PrintText(4, 14, fc, bc, "    -Conc. range from
")
    CALL PrintText(4, 15, fc, bc, "    -          to    ")
    CALL PrintText(4, 16, fc, bc, "   Sampling
at(monomer)    ")
END SUB

SUB ParameterSubMenu2
    CALL PrintText(40, 6, 15, 11, " PARAMETER
STATUS ")
    CALL PrintText(41, 7, 14, 0, "Temperature
øC")
    CALL PrintText(41, 8, 14, 0, "Reaction Time
Min")
    CALL PrintText(41, 9, 14, 0, "Monomer Type
Name")
    CALL PrintText(41, 10, 14, 0, "Monomer conc.
mol/l")
    CALL PrintText(41, 11, 14, 0, "Initiator Type
name")
    CALL PrintText(41, 12, 14, 0, "Initiator Conc.
mol/l")
    CALL PrintText(41, 13, 14, 0, "Chain Transfer Type
name")
    CALL PrintText(41, 14, 14, 0, "  -Conc. range from
mol/l")
    CALL PrintText(41, 15, 14, 0, "  -          to
mol/l")
    CALL PrintText(41, 16, 14, 0, "Sampling at(monomer)
%CONV")
END SUB

SUB ParameterSubMenu3
  COLOR 15, 0
  LOCATE 7, 63: PRINT USING "###.##"; Temp
  LOCATE 8, 63. PRINT USING "######", Time
  LOCATE 9, 65: PRINT UCASE$(MoName$)
  LOCATE 10, 63: PRINT USING "###.##"; Mo
  LOCATE 11, 65: PRINT UCASE$(IoName$)
  LOCATE 12, 63: PRINT USING "###.####", Io
  LOCATE 13, 65: PRINT UCASE$(SoName$)
  LOCATE 14, 63: PRINT USING "### ####", CTAo1
  LOCATE 15, 63: PRINT USING "### ####", CTAo2
  LOCATE 16, 63· PRINT USING "###.#",
PercentSampling
END SUB

SUB PrintBar
    CALL PrintText(5, 2, 15, 3, " File ")
    CALL PrintText(5, 2, 10, 3, " F")
    CALL PrintText(13, 2, 15, 3, " Parameter ")
    CALL PrintText(13, 2, 10, 3, " P")
    CALL PrintText(25, 2, 15, 3, " Run ")
    CALL PrintText(25, 2, 10, 3, " R")
    CALL PrintText(31, 2, 15, 3, " Graphics ")
    CALL PrintText(31, 2, 10, 3, " G")
    CALL PrintText(41, 2, 15, 3, " Simulated result ")
    CALL PrintText(41, 2, 10, 3, " S")

END SUB

SUB PrintSameText (x1, y1, lx, ly, fc, bc, Text$)
COLOR fc, bc
  FOR i = 0 TO ly
    LOCATE y1 + i, x1
    PRINT STRING$(lx, Text$)
  NEXT i
END SUB

SUB PrintText (XL, YL, fc, bc, Text$)
```

Copyright by Mahidol University

```
   LOCATE YL, XL: COLOR fc, bc: PRINT Text$
END SUB

SUB Pxxx (num, fc, bc)
      IF num < 1 THEN num = 1
      IF num > 10 THEN num = 10
      SELECT CASE num
          CASE 1
              CALL PrintText(4, 7, fc, bc, "   Temperature
")
          CASE 2
              CALL PrintText(4, 8, fc, bc, "   Reaction
Time       ")
          CASE 3
              CALL PrintText(4, 9, fc, bc, "   Monomer
Type       ")
          CASE 4
              CALL PrintText(4, 10, fc, bc, "   Monomer
Conc       ")
          CASE 5
              CALL PrintText(4, 11, fc, bc, "   Initiator
Type       ")
          CASE 6
              CALL PrintText(4, 12, fc, bc, "   Initiator
Conc       ")
          CASE 7
              CALL PrintText(4, 13, fc, bc, "   Chain
Transfer Type    ")
          CASE 8
              CALL PrintText(4, 14, fc, bc, "    -Conc.
range from    ")
          CASE 9
              CALL PrintText(4, 15, fc, bc, "    -
to    ")
          CASE 10
              CALL PrintText(4, 16, fc, bc, "   Sampling
at(monomer)    ")
      END SELECT
END SUB

SUB ResultMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 1)
    DBackBox
    DBox
    CALL ClearScreen(4)
    CALL PrintText(38, 4, 14, 1, "        INFORMATION
& SUMARY        ")
    CALL PrintText(41, 2, 15, 8, " Simulated result ")
    CALL PrintText(41, 2, 10, 8, " S")
    ResultSubMenu1
    ResultSubMenu2
    RowCount = 1
    DO
     CALL Sxxx(RowCount, 15, 8)
    DO
        kbd$ = INKEY$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"     'Up Arrow
            RowCount = RowCount - 1
            CALL Sxxx(RowCount + 1, 0, 3)
            IF RowCount < 1 THEN RowCount = 1
        CASE CHR$(0) + "P"     'Down Arrow
            RowCount = RowCount + 1
            CALL Sxxx(RowCount - 1, 0, 3)
            IF RowCount > 1 THEN RowCount = 1
    END SELECT
    LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
    IF kbd$ = CHR$(13) THEN CALL
ResultMenuSelect(RowCount)
  LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"

END SUB

SUB ResultMenuSelect (num)
  SELECT CASE num
    CASE 1
        SHELL "PLOTM-2D.EXE"
        SCREEN 12: SCREEN 0, 0
        SetScreen
        PrintBar
        CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
        CALL PrintText(10, 22, 10, 9, "Press <Enter> To
Select, U&D Arrow Key To Move Or Esc To Abort")
  END SELECT
END SUB

SUB ResultSubMenu1
    fc = 0: bc = 11
    CALL PrintText(8, 5, 15, bc, "SHOW SIMULATED
RESULTS")
    CALL PrintText(4, 7, fc, bc, " Recall old Modify Mayo
Plot ")

END SUB

SUB ResultSubMenu2
    CALL PrintText(40, 6, 15, 11, " DESCRIPTION    ")
    CALL PrintText(42, 8, 14, 0, " Recall old Modify
Mayo Plot ")
END SUB

SUB RunMenu
DO
CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
CALL Center(22, "Press <Enter> To Select, Up & Down
Arrow Key To Move Or Esc To Abort", 14, 1)
    DBackBox
    DBox
    CALL ClearScreen(15)
    CALL PrintText(38, 4, 14, 1, "        INFORMATION
& SUMARY        ")
    CALL PrintText(25, 2, 15, 8, " Run ")
    CALL PrintText(25, 2, 10, 8, " R")
    RunSubMenu1
    RunSubMenu2
    RunSubMenu3
    RunSubMenu4
    RunSubMenu5
    IF GelEffect$ = "ON" THEN
        CALL PrintText(30, 13, 9, 11, "ON")
    ELSE
        CALL PrintText(30, 13, 9, 11, "OFF")
    END IF
    RowCount = 0
    DO
     CALL Rxxx(RowCount, 15, 8)
    DO
        kbd$ = INKEY$
        LOCATE 23, 72: COLOR 15, 3: PRINT TIME$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"     'Up Arrow
            RowCount = RowCount - 1
            CALL Rxxx(RowCount + 1, 0, 3)
            IF RowCount < 1 THEN RowCount = 7
```

Copyright by Mahidol University

```
            CASE CHR$(0) + "P"      'Down Arrow
                RowCount = RowCount + 1
                CALL Rxxx(RowCount - 1, 0, 3)
                IF RowCount > 7 THEN RowCount = 1
        END SELECT
        LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
CHR$(27) OR kbd$ = CHR$(0) + "K" OR kbd$ =
CHR$(0) + "M"
        IF kbd$ = CHR$(13) THEN CALL
RunMenuSelect(RowCount)
    LOOP UNTIL kbd$ = CHR$(27) OR kbd$ = CHR$(0) +
"K" OR kbd$ = CHR$(0) + "M"
END SUB

SUB RunMenuSelect (num)
    SELECT CASE num
        CASE 1
            SentConstant
            DBackBox
            DBox
            CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
            CALL PrintText(20, 22, 14, 9, "Wait !  Now
calculate. (Interupt by F10 key)")
            CALL PrintText(13, 5, 15, 11, "RUN STATUS")
            CALL PrintText(9, 8, 0, 11, "Estimate Working
(%)")
            CALL PrintText(8, 8, 1, 11, "Ú")
            CALL PrintText(8, 9, 1, 11, "ÀÃÃ" + CHR$(16))
            CALL PrintText(9, 10, 0, 11, "Monomer
Conversion(%)")
            CALL PrintText(8, 10, 1, 11, "Ú")
            CALL PrintText(8, 11, 1, 11, "ÀÃÃ" +
CHR$(16))
            CALL PrintText(9, 12, 0, 11, "Chain Transfer
Conv(%)")
            CALL PrintText(8, 12, 1, 11, "Ú")
            CALL PrintText(8, 13, 1, 11, "ÀÃÃ" +
CHR$(16))
            CALL PrintText(9, 14, 0, 11, "Sampling time at "
+ STR$(PercentSampling) + " %")
            CALL PrintText(8, 14, 1, 11, "Ú")
            CALL PrintText(8, 15, 1, 11, "ÀÃÃ" +
CHR$(16))
            SHELL "CALQMM.EXE"
            SHELL "SUCCESS.EXE"
            SCREEN 12: SCREEN 0, 0
            SetScreen
            PrintBar
            CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
            CALL PrintText(10, 22, 10, 9, "Press <Enter> To
Select, U&D Arrow Key To Move Or Esc To Abort")
        CASE 2
            HelpForInputValue
            CALL PrintText(60, 7, 15, 9, "       ")
            LOCATE 7, 60
            INPUT "", numberNCL
            IF numberNCL <= 0 THEN numberNCL = 150
        CASE 3
            HelpForInputValue
            CALL PrintText(60, 8, 15, 9, "       ")
            LOCATE 8, 60
            INPUT "", Xscale
            IF Xscale <= 0 THEN Xscale = 8
        CASE 4
            HelpForInputValue
            CALL PrintText(60, 9, 15, 9, "       ")
            LOCATE 9, 60
            INPUT "", StepTime
            IF StepTime <= 0 THEN StepTime = 20
        CASE 5
            HelpForInputValue
            CALL PrintText(60, 10, 15, 9, "       ")
            LOCATE 10, 60
            INPUT "", numberDATA
            IF numberDATA <= 0 THEN numberDATA = 50
        CASE 6
            ConfirmBoxM
            CALL PrintText(14, 13, 6, 11, "CONFIRM !")
            CALL PrintText(11, 14, 0, 11, "Manaul
input<M>")
            CALL PrintText(11, 15, 9, 11, "Edit in ED  <E>")
            DO
            EDrate$ = UCASE$(INPUT$(1))
            LOOP WHILE INSTR("ME", EDrate$) = 0
            SELECT CASE EDrate$
                CASE "M"
                    CALL PrintText(8, 17, 15, 11, "³
Manaul input   ³")
                    ManaulInput
                CASE "E"
                    CALL PrintText(8, 17, 15, 11, "³    Edit
in ED   ³")
            HelpForEditRate
            SentTemp
            SHELL "ED.EXE"
        OPEN "ACEDIT.AV" FOR INPUT AS #22
            INPUT #22, MacceptText$
            INPUT #22, IacceptText$
            INPUT #22, CacceptText$
        CLOSE #22
        OPEN "SENTMIAN.CTN" FOR INPUT AS #21
        SELECT CASE MacceptText$
            CASE "Y"
            INPUT #21, Kp
            INPUT #21, Ktd
            INPUT #21, Ktc
            INPUT #21, Ktrm
        END SELECT
        SELECT CASE IacceptText$
            CASE "Y"
            INPUT #21, Ki
            INPUT #21, Kd
            INPUT #21, f
            INPUT #21, Ktci
        END SELECT
        SELECT CASE CacceptText$
            CASE "Y"
            INPUT #21, Ksr
            INPUT #21, Ktr
        END SELECT
        CLOSE #21
            FOR i = 0 TO 23
                COLOR 0, 0
                LOCATE 1 + i, 1: PRINT STRING$(80, " ")
            NEXT i
            SetScreen
            DBackBox
            DBox
            PrintBar

    END SELECT
    CASE 7
        ConfirmBox
        CALL PrintText(14, 13, 6, 11, "WARNING !")
        CALL PrintText(11, 14, 0, 11, "YOU WANT TO
ADD")
        CALL PrintText(13, 15, 0, 11, "GEL EFFECTS")
        DO
        XGelEffect$ = UCASE$(INPUT$(1))
        LOOP WHILE INSTR("YN", XGelEffect$) = 0
        SELECT CASE XGelEffect$
            CASE "Y"
                GelEffect$ = "ON"
            CASE "N"
```

Copyright by Mahidol University

```
            GelEffect$ = "OFF"
        END SELECT
    END SELECT
END SUB

SUB RunSubMenu1
    fc = 0  bc = 11
    CALL PrintText(5, 5, 15, bc, "ITERATIVE VALUE
FOR ROUTINE")
    CALL PrintText(4, 7, fc, bc, "   Start to Simulation
")
    CALL PrintText(4, 8, fc, bc, "   Number of NCL point
")
    CALL PrintText(4, 9, fc, bc, "   NCL-axis scale
")
    CALL PrintText(4, 10, fc, bc, "   Increment Time
")
    CALL PrintText(4, 11, fc, bc, "   Number of CTA
point    ")
    CALL PrintText(4, 12, fc, bc, "   Edit Rate Coefficient
")
    CALL PrintText(4, 13, fc, bc, "   Edit Gel Effect
")
END SUB

SUB RunSubMenu2
    CALL PrintText(40, 6, 15, 11, " RUN STATUS     ")
    CALL PrintText(43, 7, 14, 0, "No.of NCL point
Point")
    CALL PrintText(43, 8, 14, 0, "NCL-axis scale .
log(NCL)")
    CALL PrintText(43, 9, 14, 0, "Iterative Time :
Sec")
    CALL PrintText(43, 10, 14, 0, "No.of CTA point:
Point")
END SUB

SUB RunSubMenu3
        COLOR 15, 0
        LOCATE 7, 63  PRINT USING "####";
numberNCL
        LOCATE 8, 63  PRINT USING "## ##", Xscale
        LOCATE 9, 62. PRINT USING "####.#";
StepTime
        LOCATE 10, 61. PRINT USING "#####";
numberDATA
END SUB

SUB RunSubMenu4
    CALL PrintText(43, 11, 14, 0, "Propagate ,Kp  :
l/mol s")
    CALL PrintText(43, 12, 14, 0, "Terminate ,Ktd :
l/mol.s")
    CALL PrintText(43, 13, 14, 0, "Terminate ,Ktc :
l/mol.s")
    CALL PrintText(43, 14, 14, 0, "Transfer ,Ktrm .
l/mol.s")
    CALL PrintText(43, 15, 14, 0, "Initiate  ,Ki
l/mol.s")
    CALL PrintText(43, 16, 14, 0, "Dissociate,Kd  :
1/sec")
    CALL PrintText(43, 17, 14, 0, "efficiency,f   :
")
    CALL PrintText(43, 18, 14, 0, "1øTermina.,Ktci.
l/mol.s")
    CALL PrintText(43, 19, 14, 0, "CTAøPrepare,Ksr:
l/mol.s")
    CALL PrintText(43, 20, 14, 0, "Transfer  ,Ktr :
l/mol.s")

END SUB

SUB RunSubMenu5
        COLOR 15, 0
        LOCATE 11, 60: PRINT USING "## ###^^^^ ";
Kp
        LOCATE 12, 60: PRINT USING "##.###^^^^ ",
Ktd
        LOCATE 13, 60: PRINT USING "##.###^^^^ ";
Ktc
        LOCATE 14, 60: PRINT USING "## ###^^^^ ",
Ktrm
        LOCATE 15, 60: PRINT USING "## ###^^^^ ",
Ki
        LOCATE 16, 60: PRINT USING "## ###^^^^ ";
Kd
        LOCATE 17, 60: PRINT USING "##.####    "; f
        LOCATE 18, 60: PRINT USING "##.###^^^^ ",
Ktci
        LOCATE 19, 60: PRINT USING "## ###^^^^ ";
Ksr
        LOCATE 20, 60: PRINT USING "##.###^^^^ ";
Ktr
END SUB

SUB Rxxx (num, fc, bc)
      IF num < 1 THEN num = 1
      IF num > 7 THEN num = 7
      SELECT CASE num
          CASE 1
              CALL PrintText(4, 7, fc, bc, "   Start to
Simulation    ")
          CASE 2
              CALL PrintText(4, 8, fc, bc, "   Number of
NCL point    ")
          CASE 3
              CALL PrintText(4, 9, fc, bc, "   NCL-axis
scale        ")
          CASE 4
              CALL PrintText(4, 10, fc, bc, "   Increment
Time        ")
          CASE 5
              CALL PrintText(4, 11, fc, bc, "   Number of
CTA point    ")
          CASE 6
              CALL PrintText(4, 12, fc, bc, "   Edit Rate
Coefficient   ")
          CASE 7
              CALL PrintText(4, 13, fc, bc, "   Edit Gel
Effect       ")
              IF GelEffect$ = "ON" THEN
              CALL PrintText(30, 13, 9, bc, "ON")
              ELSE
              CALL PrintText(30, 13, 9, bc, "OFF")
              END IF
      END SELECT
END SUB

SUB SaveCondFile
   OPEN SaveCondDrive$ + "\" + SaveCondName$ +
".CNM" FOR OUTPUT AS #1
      PRINT #1, MoName$
      PRINT #1, IoName$
      PRINT #1, SoName$
      PRINT #1, Mo
      PRINT #1, Io
      PRINT #1, CTAo1
      PRINT #1, CTAo2
      PRINT #1, PercentSampling
      PRINT #1, Temp
      PRINT #1, Time
      PRINT #1, Kp
      PRINT #1, Ktc
      PRINT #1, Ktd
```

Copyright by Mahidol University

```
        PRINT #1, Ktrm
        PRINT #1, Ki
        PRINT #1, Kd
        PRINT #1, f
        PRINT #1, Ktci
        PRINT #1, Ksr
        PRINT #1, Ktr
        PRINT #1, numberNCL
        PRINT #1, Xscale
        PRINT #1, StepTime
        PRINT #1, numberDATA
        PRINT #1, GelEffect$
    CLOSE #1
END SUB

SUB SentConstant
    OPEN "CONST.MTY" FOR OUTPUT AS #3
        PRINT #3, MoName$
        PRINT #3, IoName$
        PRINT #3, SoName$
        PRINT #3, Mo
        PRINT #3, Io
        PRINT #3, CTAo1
        PRINT #3, CTAo2
        PRINT #3, PercentSampling
        PRINT #3, Temp
        PRINT #3, Time
        PRINT #3, Kp
        PRINT #3, Ktc
        PRINT #3, Ktd
        PRINT #3, Ktrm
        PRINT #3, Ki
        PRINT #3, Kd
        PRINT #3, f
        PRINT #3, Ktci
        PRINT #3, Ksr
        PRINT #3, Ktr
        PRINT #3, numberNCL
        PRINT #3, Xscale
        PRINT #3, StepTime
        PRINT #3, numberDATA
        PRINT #3, GelEffect$
    CLOSE #3
END SUB

SUB SentTemp
    OPEN "SENTED CTN" FOR OUTPUT AS #20
        PRINT #20, Temp
    CLOSE #20
END SUB

SUB SetScreen
    COLOR 0, 13  CLS
    FOR i = 0 TO 18
        LOCATE 3 + i, 1  COLOR 15, 0. PRINT
STRING$(80, CHR$(219))
    NEXT i
    CALL PrintSameText(1, 1, 80, 0, 15, 4, " ")
    CALL PrintSameText(1, 2, 80, 0, 15, 3, " ")
    CALL PrintSameText(1, 23, 80, 0, 15, 3, " ")
    CALL PrintSameText(1, 22, 80, 0, 15, 9, " ")
    CALL Center(1, "MODIFY MAYO'S
RELATIONSHIP", 15, 4)
    CALL PrintText(38, 4, 14, 1, "          INFORMATION
& SUMARY          ")
    CALL PrintText(2, 23, 5, 3, "Action:- <enter>=select
item   <arrow key>=move   <escape>=abort")
END SUB

SUB Sxxx (num, fc, bc)
    SELECT CASE num
        CASE 1
            CALL PrintText(4, 7, fc, bc, " Recall old
Modify Mayo Plot ")
    END SELECT

END SUB

SUB Utilmenu1
    CALL PrintText(4, 7, 0, 11, "  File Name          ")
    CALL PrintText(4, 8, 0, 11, "  Path Name          ")
    CALL PrintText(4, 9, 0, 11, "  Go                 ")
END SUB

SUB Utilxxx (num, fc, bc)
    IF num < 1 THEN num = 1
    IF num > 3 THEN num = 3
    SELECT CASE num
        CASE 1
            CALL PrintText(4, 7, fc, bc, "  File Name
")
        CASE 2
            CALL PrintText(4, 8, fc, bc, "  Path Name
")
        CASE 3
            CALL PrintText(4, 9, fc, bc, "  Go
")
    END SELECT
END SUB
```

Copyright by Mahidol University

4) Programmes for calculation for the weight distribution of dead-end polymers (CALQW.BAS)

```
DECLARE SUB CreateWeightFractionOfAllSpecies ()
DECLARE FUNCTION KtdGelEffect! (Xi!)
DECLARE FUNCTION KtcGelEffect! (Xi!)
DECLARE FUNCTION KtrmGelEffect! (Xi!)
DECLARE SUB NewtonRaphson (t!, x!, y!)
DECLARE FUNCTION PrimaryRadical! (t!, x!, y!)
DECLARE FUNCTION ODEDn1! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn2! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn3! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn4! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn5! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn6! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn7! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn8! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn9! (num!, t!, x!, y!)
DECLARE FUNCTION ALPHAr! (t!, x!, y!)
DECLARE FUNCTION ALPHAc! (t!, x!, y!)
DECLARE FUNCTION ALPHAm! (t!, x!, y!)
DECLARE FUNCTION ODEM! (t!, x!, y!)
DECLARE FUNCTION ODEC! (t!, x!, y!)
DECLARE SUB LoadAllParameter ()
DECLARE FUNCTION PriPr! (t!, x!, y!)
DECLARE FUNCTION PriPc! (t!, x!, y!)
DECLARE FUNCTION PriPm! (t!, x!, y!)
COMMON SHARED Mo, Io, CTA, Temp, Time, FT
COMMON SHARED Kp, Ktd, Ktc, Ktrm, Ki, Kd, f, Kfci,
Ksr, Ktr, Ktrm0, Ktd0, Ktc0
COMMON SHARED Pr, Pc, Pm
COMMON SHARED number, h, Xscale, GelEffect$
COMMON SHARED Tox, Nox
LoadAllParameter
DIM SHARED WeightFraction(1 TO 10, 1 TO number +
1)
DIM SHARED NCL(1 TO number + 1)
FT = Time * 60
Ktrm0 = Ktrm
Ktc0 = Ktc
Ktd0 = Ktd
OPEN "PLOTW.3D" FOR OUTPUT AS #2
Ncount = 1
Mcount = 0
FOR v = 0 TO Xscale STEP Xscale / number
  n = 10 ^ v
  NCL(Ncount) = n
      Stepnumber = FT / h
      ITime = 0
      Mi = Mo
      Ci = CTA
    OPEN "PLOTCONV 2D" FOR OUTPUT AS #1
    count = 0
    REM define initial weight fraction at start reaction time
    Dx1x = 0: Dx2x = 0  Dx3x = 0: Dx4x = 0  Dx5x = 0
Dx6x = 0: Dx7x = 0  Dx8x = 0: Dx9x = 0: Dx10x = 0
    FOR i = 0 TO Stepnumber
      IF INKEY$ = CHR$(0) + "D" THEN END
      IF GelEffect$ = "ON" THEN
      Ktrm = KtrmGelEffect((Mo - Mi) / Mo)
      Ktc = KtcGelEffect((Mo - Mi) / Mo)
      Ktd = KtdGelEffect((Mo - Mi) / Mo)
      ELSE
      Ktrm = Ktrm0
      Ktc = Ktc0
      Ktd = Ktd0
      END IF
      CALL NewtonRaphson(ITime, Mi, Ci)
      REM Monomer at time T
      IF Mi <= 0 THEN EXIT FOR
        k1M = h * ODEM(ITime, Mi, Ci)
        k2M = h * ODEM(ITime +  5 * h, Mi +  5 *
k1M, Ci)
        k3M = h * ODEM(ITime + .5 * h, Mi +  5 *
k2M, Ci)
        k4M = h * ODEM(ITime + h, Mi + k3M, Ci)
        Mi = Mi + (k1M + 2 * k2M + 2 * k3M +
k4M) / 6
      REM Chain transfer agent at time T
      IF Ci > 0 THEN
        k1C = h * ODEC(ITime, Ci, Mi)
        k2C = h * ODEC(ITime + .5 * h, Mi +  5
* k1C)
        k3C = h * ODEC(ITime +  5 * h, Mi, Ci + .5
* k2C)
        k4C = h * ODEC(ITime + h, Mi, Ci + k3C)
        Ci = Ci + (k1C + 2 * k2C + 2 * k3C + k4C) /
6
        IF Ci <= 0 THEN Ci = 0
      END IF
    REM Species Mn
      k1Dn1 = h * ODEDn1(n, ITime, Mi, Ci)
      k2Dn1 = h * ODEDn1(n, ITime + .5 * h, Mi, Ci +
.5 * k1Dn1)
      k3Dn1 = h * ODEDn1(n, ITime + .5 * h, Mi, Ci +
5 * k2Dn1)
      k4Dn1 = h * ODEDn1(n, ITime + h, Mi, Ci +
k3Dn1)
      Dn1 = (k1Dn1 + 2 * k2Dn1 + 2 * k3Dn1 +
k4Dn1) / 6
      Wn1 = n * Dn1 / (Mo - Mi)
      WeightFraction(1, Ncount) = Wn1 + Dx1x
      Dx1x = WeightFraction(1, Ncount)
    REM Species MnX
      k1Dn2 = h * ODEDn2(n, ITime, Mi, Ci)
      k2Dn2 = h * ODEDn2(n, ITime +  5 * h, Mi, Ci +
5 * k1Dn2)
      k3Dn2 = h * ODEDn2(n, ITime + .5 * h, Mi, Ci +
.5 * k2Dn2)
      k4Dn2 = h * ODEDn2(n, ITime + h, Mi, Ci +
k3Dn1)
      Dn2 = (k1Dn2 + 2 * k2Dn2 + 2 * k3Dn2 +
k4Dn2) / 6
      Wn2 = n * Dn2 / (Mo - Mi)
      WeightFraction(2, Ncount) = Wn2 + Dx2x
      Dx2x = WeightFraction(2, Ncount)
    REM Species RMn
      k1Dn3 = h * ODEDn3(n, ITime, Mi, Ci)
      k2Dn3 = h * ODEDn3(n, ITime + .5 * h, Mi, Ci +
.5 * k1Dn3)
      k3Dn3 = h * ODEDn3(n, ITime + .5 * h, Mi, Ci +
.5 * k2Dn3)
      k4Dn3 = h * ODEDn3(n, ITime + h, Mi, Ci +
k3Dn3)
      Dn3 = (k1Dn3 + 2 * k2Dn3 + 2 * k3Dn3 +
k4Dn3) / 6
      Wn3 = n * Dn3 / (Mo - Mi)
      WeightFraction(3, Ncount) = Wn3 + Dx3x
      Dx3x = WeightFraction(3, Ncount)
    REM Species RMnX
      k1Dn4 = h * ODEDn4(n, ITime, Mi, Ci)
      k2Dn4 = h * ODEDn4(n, ITime + .5 * h, Mi, Ci +
.5 * k1Dn4)
      k3Dn4 = h * ODEDn4(n, ITime + .5 * h, Mi, Ci +
.5 * k2Dn4)
      k4Dn4 = h * ODEDn4(n, ITime + h, Mi, Ci +
k3Dn4)
      Dn4 = (k1Dn4 + 2 * k2Dn4 + 2 * k3Dn4 +
k4Dn4) / 6
```

Copyright by Mahidol University

```
        Wn4 = n * Dn4 / (Mo - Mi)
        WeightFraction(4, Ncount) = Wn4 + Dx4x
        Dx4x = WeightFraction(4, Ncount)
    REM Species RMnR
        k1Dn5 = h * ODEDn5(n, ITime, M1, C1)
        k2Dn5 = h * ODEDn5(n, ITime + .5 * h, M1, C1 +
.5 * k1Dn5)
        k3Dn5 = h * ODEDn5(n, ITime + .5 * h, M1, Ci +
.5 * k2Dn5)
        k4Dn5 = h * ODEDn5(n, ITime + h, M1, C1 +
k3Dn5)
        Dn5 = (k1Dn5 + 2 * k2Dn5 + 2 * k3Dn5 +
k4Dn5) / 6
        Wn5 = n * Dn5 / (Mo - Mi)
        WeightFraction(5, Ncount) = Wn5 + Dx5x
        Dx5x = WeightFraction(5, Ncount)
    REM Species CTAMn
        k1Dn6 = h * ODEDn6(n, ITime, M1, C1)
        k2Dn6 = h * ODEDn6(n, ITime + .5 * h, Mi, C1 +
 5 * k1Dn6)
        k3Dn6 = h * ODEDn6(n, ITime +  5 * h, Mi, C1 +
.5 * k2Dn6)
        k4Dn6 = h * ODEDn6(n, ITime + h, M1, C1 +
k3Dn6)
        Dn6 = (k1Dn6 + 2 * k2Dn6 + 2 * k3Dn6 +
k4Dn6) / 6
        Wn6 = n * Dn6 / (Mo - Mi)
        WeightFraction(6, Ncount) = Wn6 + Dx6x
        Dx6x = WeightFraction(6, Ncount)
    REM Species CTAMnX
        k1Dn7 = h * ODEDn7(n, ITime, Mi, C1)
        k2Dn7 = h * ODEDn7(n, ITime +  5 * h, M1, Ci +
.5 * k1Dn7)
        k3Dn7 = h * ODEDn7(n, ITime + .5 * h, Mi, C1 +
 5 * k2Dn7)
        k4Dn7 = h * ODEDn7(n, ITime + h, M1, C1 +
k3Dn7)
        Dn7 = (k1Dn7 + 2 * k2Dn7 + 2 * k3Dn7 +
k4Dn7) / 6
        Wn7 = n * Dn7 / (Mo - Mi)
        WeightFraction(7, Ncount) = Wn7 + Dx7x
        Dx7x = WeightFraction(7, Ncount)
    REM Species CTAMnCTA
        k1Dn8 = h * ODEDn8(n, ITime, Mi, C1)
        k2Dn8 = h * ODEDn8(n, ITime +  5 * h, M1, Ci +
 5 * k1Dn8)
        k3Dn8 = h * ODEDn8(n, ITime + .5 * h, Mi, C1 +
.5 * k2Dn8)
        k4Dn8 = h * ODEDn8(n, ITime + h, M1, C1 +
k3Dn8)
        Dn8 = (k1Dn8 + 2 * k2Dn8 + 2 * k3Dn8 +
k4Dn8) / 6
        Wn8 = n * Dn8 / (Mo - Mi)
        WeightFraction(8, Ncount) = Wn8 + Dx8x
        Dx8x = WeightFraction(8, Ncount)
    REM Species CTAMnR
        k1Dn9 = h * ODEDn9(n, ITime, Mi, C1)
        k2Dn9 = h * ODEDn9(n, ITime + .5 * h, M1, C1 +
.5 * k1Dn9)
        k3Dn9 = h * ODEDn9(n, ITime +  5 * h, Mi, C1 +
 5 * k2Dn9)
        k4Dn9 = h * ODEDn9(n, ITime + h, Mi, Ci +
k3Dn9)
        Dn9 = (k1Dn9 + 2 * k2Dn9 + 2 * k3Dn9 +
k4Dn9) / 6
        Wn9 = n * Dn9 / (Mo - M1)
        WeightFraction(9, Ncount) = Wn9 + Dx9x
        Dx9x = WeightFraction(9, Ncount)
    REM Species ALL
        Dn10 = Dn1 + Dn2 + Dn3 + Dn4 + Dn5 + Dn6 +
Dn7 + Dn8 + Dn9
        Wn10 = n * Dn10 / (Mo - Mi)
        WeightFraction(10, Ncount) = Wn10 + Dx10x
        Dx10x = WeightFraction(10, Ncount)

    REM Save datafile for read in plotting 2,3D graph after
simulation
        REM Conversion curve data
          PRINT #1, Mi, C1, ITime
        REM 3D Simulation curve
        IF Mcount MOD Nox = 0 THEN
          IF count MOD Tox = 0 THEN
            PRINT #2, Wn10, n, ITime
          END IF
        END IF

    REM Printing percent work
        LOCATE 9, 13: COLOR 0, 11
        percent = (Ncount / number) * 100
        IF percent > 100 THEN percent = 100
            PRINT USING "###.##", percent
            LOCATE 11, 13  COLOR 0, 11
        MOpercent = ((Mo - M1) / Mo) * 100
        IF MOpercent > 100 THEN MOpercent = 100
            PRINT USING "###.##"; MOpercent
        IF CTA > 0 THEN
            LOCATE 13, 13  COLOR 0, 11
            CTApercent = ((CTA - Ci) / CTA) * 100
            IF CTApercent > 100 THEN CTApercent =
100
            PRINT USING "###.##", CTApercent
        END IF

    ITime = ITime + h
    count = count + 1
    NEXT i
    CLOSE #1
  Ncount = Ncount + 1
  Mcount = Mcount + 1
NEXT v
CLOSE #2
CreateWeightFractionOfAllSpecies
END

FUNCTION ALPHAc (t, x, y) STATIC
    Radical = PrimaryRadical(t, x, y)
    a1 = Kp * x
    a2 = Ktrm * x
    a3 = Ktr * y
    a4 = Ktc * ((.5 * Pc) + Pr + Pm)
    a5 = Ktd * (Pr + Pc + Pm)
    a6 = Ktci * Radical
    a = a1 + a2 + a3 + a4 + a5 + a6
    ALPHAc = a1 / a
END FUNCTION

FUNCTION ALPHAm (t, x, y) STATIC
    Radical = PrimaryRadical(t, x, y)
    a1 = Kp * x
    a2 = Ktrm * x
    a3 = Ktr * y
    a4 = Ktc * ((.5 * Pm) + Pr + Pc)
    a5 = Ktd * (Pr + Pc + Pm)
    a6 = Ktci * Radical
    a = a1 + a2 + a3 + a4 + a5 + a6
    ALPHAm = a1 / a
END FUNCTION

FUNCTION ALPHAr (t, x, y) STATIC
    Radical = PrimaryRadical(t, x, y)
    a1 = Kp * x
    a2 = Ktrm * x
    a3 = Ktr * y
    a4 = Ktc * ((.5 * Pr) + Pc + Pm)
```

Copyright by Mahidol University

```
    a5 = Ktd * (Pr + Pc + Pm)
    a6 = Ktci * Radical
    a = a1 + a2 + a3 + a4 + a5 + a6
    ALPHAr = a1 / a
END FUNCTION

SUB CreateWeightFractionOfAllSpecies
    OPEN "PLOTW.2DN" FOR OUTPUT AS #5
      FOR i = 1 TO number
        PRINT #5, NCL(i)
      NEXT i
    CLOSE #5
    OPEN "PLOTW 2D1" FOR OUTPUT AS #6
      FOR i = 1 TO number
        PRINT #6, WeightFraction(1, i)
      NEXT i
    CLOSE #6
    OPEN "PLOTW.2D2" FOR OUTPUT AS #7
      FOR i = 1 TO number
        PRINT #7, WeightFraction(2, i)
      NEXT i
    CLOSE #7
    OPEN "PLOTW 2D3" FOR OUTPUT AS #8
      FOR i = 1 TO number
        PRINT #8, WeightFraction(3, i)
      NEXT i
    CLOSE #8
    OPEN "PLOTW.2D4" FOR OUTPUT AS #9
      FOR i = 1 TO number
        PRINT #9, WeightFraction(4, i)
      NEXT i
    CLOSE #9
    OPEN "PLOTW.2D5" FOR OUTPUT AS #10
      FOR i = 1 TO number
        PRINT #10, WeightFraction(5, i)
      NEXT i
    CLOSE #10
    OPEN "PLOTW 2D6" FOR OUTPUT AS #11
      FOR i = 1 TO number
        PRINT #11, WeightFraction(6, i)
      NEXT i
    CLOSE #11
    OPEN "PLOTW.2D7" FOR OUTPUT AS #12
      FOR i = 1 TO number
        PRINT #12, WeightFraction(7, i)
      NEXT i
    CLOSE #12
    OPEN "PLOTW 2D8" FOR OUTPUT AS #13
      FOR i = 1 TO number
        PRINT #13, WeightFraction(8, i)
      NEXT i
    CLOSE #13
    OPEN "PLOTW.2D9" FOR OUTPUT AS #14
      FOR i = 1 TO number
        PRINT #14, WeightFraction(9, i)
      NEXT i
    CLOSE #14
    OPEN "PLOTW 2DT" FOR OUTPUT AS #15
      FOR i = 1 TO number
        PRINT #15, WeightFraction(10, i)
      NEXT i
    CLOSE #15
END SUB

FUNCTION KtcGelEffect (Xi) STATIC
    B = 2.5882 - (3.4852 *  001 * (Temp + 273.3))
    C = 4.3108 - (1 2635 * .01 * (Temp + 273.3))
    D = -4 6488 + (1.9026 * .01 * (Temp + 273.3))
    E = (B * Xi) + (C * Xi * Xi) + (D * Xi * Xi * Xi)
    KtcGelEffect = Ktc0 * EXP(-2 * E)
END FUNCTION

FUNCTION KtdGelEffect (Xi) STATIC
    B = 2.5882 - (3.4852 *  001 * (Temp + 273.3))
    C = 4.3108 - (1.2635 *  01 * (Temp + 273 3))
    D = -4.6488 + (1.9026 *  01 * (Temp + 273.3))
    E = (B * Xi) + (C * Xi * Xi) + (D * Xi * Xi * Xi)
    KtdGelEffect = Ktd0 * EXP(-2 * E)
END FUNCTION

FUNCTION KtrmGelEffect (Xi) STATIC
    B = 2.5882 - (3 4852 * .001 * (Temp + 273 3))
    C = 4.3108 - (1.2635 * .01 * (Temp + 273.3))
    D = -4.6488 + (1.9026 * .01 * (Temp + 273.3))
    E = (B * Xi) + (C * Xi * Xi) + (D * Xi * Xi * Xi)
    KtrmGelEffect = Ktrm0 * EXP(-2 * E)
END FUNCTION

SUB LoadAllParameter
    OPEN "CONST MTX" FOR INPUT AS #1
      INPUT #1, MoName$
      INPUT #1, IoName$
      INPUT #1, SoName$
      INPUT #1, Mo
      INPUT #1, Io
      INPUT #1, CTA
      INPUT #1, Temp
      INPUT #1, Time
      INPUT #1, Kp
      INPUT #1, Ktc
      INPUT #1, Ktd
      INPUT #1, Ktrm
      INPUT #1, Ki
      INPUT #1, Kd
      INPUT #1, f
      INPUT #1, Ktci
      INPUT #1, Ksr
      INPUT #1, Ktr
      INPUT #1, number
      INPUT #1, h
      INPUT #1, Xscale
      INPUT #1, Yscale
      INPUT #1, GelEffect$
      INPUT #1, XoCOOR
      INPUT #1, YoCOOR
      INPUT #1, XaxisLength
      INPUT #1, YaxisLength
      INPUT #1, ZaxisLength
      INPUT #1, LeftAngle
      INPUT #1, RightAngle
      INPUT #1, Nox
      INPUT #1, Tox
    CLOSE #1
END SUB

SUB NewtonRaphson (t, x, y) STATIC
    It = Io * EXP(-Kd * t)
    Prapx = SQR(2 * Kd * f * It / ((.5 * Ktc) + Ktd))
    IF Ktrm > 0 THEN
        Pmapx = Prapx
    ELSE
        Pmapx = 0
    END IF
    IF y > 0 AND Ktr > 0 THEN
        Pcapx = Prapx
    ELSE
        Pcapx = 0
    END IF
FOR n = 1 TO 100

r1 = 2 * Ki * x * Kd * f * It * Ktci
r2 = (Ki * x) + (Ksr * y) + (Ktci * (Prapx + Pcapx +
Pmapx))
r3 = Ktc * (Prapx + Pcapx + Pmapx)
```

Copyright by Mahidol University

```
r4 = Ktd * ((2 * Prapx) + Pcapx + Pmapx)
r5 = (Ktr * y) + (Ktrm * x)
r6 = 2 * Kd * f * It * Ktci
r7 = (Ki * x) + (Ksr * y) + (Ktci * (Pcapx + Pmapx))
r8 = 2 * Kd * f * It
r10 = r8 / r2          '1st radical
r11 = Ki * x * r10
r12 = Prapx * Ktc * (( 5 * Prapx) + Pcapx + Pmapx)
r13 = Prapx * Ktd * (Prapx + Pcapx + Pmapx)
r14 = (Ktr * y * Prapx) + (Ktrm * x * Prapx)
r15 = Ktci * r10 * Prapx
dFr = (r1 / (r2 ^ 2)) + (r3 + r4 + r5) + ((r6 * r7) / (r2 ^ 2))
Fr = -r11 + (r12 + r13 + r14 + r15)
Prt = Prapx - (Fr / dFr)

IF y > 0 AND Ktr > 0 THEN
c1 = 2 * Ksr * y * Kd * f * It * Ktci
c2 = (Ki * x) + (Ksr * y) + (Ktci * (Prapx + Pcapx +
Pmapx))
c3 = Ktc * (Prapx + Pcapx + Pmapx)
c4 = Ktd * ((2 * Pcapx) + Prapx + Pmapx)
c5 = Ktrm * x
c6 = 2 * Kd * f * It * Ktci
c7 = (Ki * x) + (Ksr * y) + (Ktci * (Prapx + Pmapx))
c8 = 2 * Kd * f * It
c10 = c8 / c2          '1st radical
c11 = (Ksr * y * c10) + (Ktr * y * (Pcapx + Prapx +
Pmapx))
c12 = Pcapx * Ktc * (( 5 * Pcapx) + Prapx + Pmapx)
c13 = Pcapx * Ktd * (Pcapx + Prapx + Pmapx)
c14 = (Ktr * y * Pcapx) + (Ktrm * x * Pcapx)
c15 = Ktci * c10 * Pcapx
dFc = (c1 / (c2 ^ 2)) + (c3 + c4 + c5) + ((c6 * c7) / (c2 ^ 2))
Fc = -c11 + (c12 + c13 + c14 + c15)
Pct = Pcapx - (Fc / dFc)
END IF

IF Ktrm > 0 THEN
m2 = (Ki * x) + (Ksr * y) + (Ktci * (Prapx + Pcapx +
Pmapx))
m3 = Ktc * (Prapx + Pcapx + Pmapx)
m4 = Ktd * ((2 * Pmapx) + Prapx + Pcapx)
m5 = Ktr * y
m6 = 2 * Kd * f * It * Ktci
m7 = (Ki * x) + (Ksr * y) + (Ktci * (Prapx + Pcapx))
m8 = 2 * Kd * f * It
m10 = m8 / m2          '1st radical
m11 = Ktrm * x * (Pcapx + Prapx + Pmapx)
m12 = Pmapx * Ktc * ((.5 * Pmapx) + Prapx + Pcapx)
m13 = Pmapx * Ktd * (Pcapx + Prapx + Pmapx)
m14 = (Ktr * y * Pmapx) + (Ktrm * x * Pmapx)
m15 = Ktci * m10 * Pmapx
dFm = (m3 + m4 + m5) + ((m6 * m7) / (m2 ^ 2))
Fm = -m11 + (m12 + m13 + m14 + m15)
Pmt = Pmapx - (Fm / dFm)
END IF

IF Prt > 0 AND Pct > 0 AND Pmt > 0 THEN
    IF ABS((Pmt - Pmapx) / Pmt) <= .0001 AND
ABS((Pct - Pcapx) / Pct) <= .0001 AND ABS((Prt - Prapx) /
Prt) <= .0001 THEN EXIT FOR
ELSEIF Prt > 0 AND Pct > 0 AND Pmt = 0 THEN
    IF ABS((Pct - Pcapx) / Pct) <= .0001 AND ABS((Prt -
Prapx) / Prt) <= .0001 THEN EXIT FOR
ELSEIF Prt > 0 AND Pct = 0 AND Pmt > 0 THEN
    IF ABS((Pmt - Pmapx) / Pmt) <= .0001 AND
ABS((Prt - Prapx) / Prt) <= .0001 THEN EXIT FOR
ELSEIF Prt > 0 AND Pct = 0 AND Pmt = 0 THEN
    IF ABS((Prt - Prapx) / Prt) <= .0001 THEN EXIT
FOR
END IF
Prapx = Prt
Pcapx = Pct
Pmapx = Pmt
NEXT n
Pr = Prt
Pc = Pct
Pm = Pmt
END SUB

FUNCTION ODEC (t, x, y) STATIC
'Routine which define ordinary differential equation of
[CTA] (-d[CTA]/dt)
'-d[CTA]/dt = Ksr[R][CTA] + Ktr[CTA](Pr+Pc+Pm)
    B = PrimaryRadical(t, x, y)
    a = (Ksr * B * y) + (Ktr * y * (Pr + Pc + Pm))
    ODEC = -a
END FUNCTION

FUNCTION ODEDn1 (num, t, x, y) STATIC
    a = (Ktrm * x) + (Ktd * (Pm + Pr + Pc))
    B = PriPm(t, x, y)
    C = ALPHAm(t, x, y)
    D = B * (C ^ (num - 1))
    d1 = (num - 1) * .5 * Ktc * (B ^ 2) * (C ^ (num - 2))
    ODEDn1 = (D * a) + d1
END FUNCTION

FUNCTION ODEDn2 (num, t, x, y) STATIC
    IF y > 0 THEN
    a = (Ktr * y)
    B = PriPm(t, x, y)
    C = ALPHAm(t, x, y)
    D = B * (C ^ (num - 1))
    END IF
    ODEDn2 = D * a
END FUNCTION

FUNCTION ODEDn3 (num, t, x, y) STATIC
    a = PrimaryRadical(t, x, y)
    B = PriPm(t, x, y)
    C = PriPr(t, x, y)
    d1 = ALPHAm(t, x, y)
    d2 = ALPHAr(t, x, y)
    a1 = (Ktrm * x) + (Ktd * (Pm + Pr + Pc))
    c31 = d1 ^ -1
    c32 = d2 ^ -1
    IF d1 > d2 THEN
        c5 = d2 / d1
        c4 = d1 ^ num
    END IF
    IF d2 > d1 THEN
        c5 = d1 / d2
        c4 = d2 ^ num
    END IF
    IF c5 < 1 AND c5 > 0 THEN
        c6 = 1 / (1 - c5)
    ELSEIF c5 = 1 THEN
        c6 = num - 1
    END IF
    c1 = Ktci * a * B * (d1 ^ (num - 1))
    c2 = a1 * C * (d2 ^ (num - 1))
    c7 = Ktc * B * C * c31 * c32 * c4 * c6
    ODEDn3 = c1 + c2 + c7
END FUNCTION

FUNCTION ODEDn4 (num, t, x, y) STATIC
    IF y > 0 THEN
    a = (Ktr * y)
    B = PriPr(t, x, y)
    C = ALPHAr(t, x, y)
    D = B * (C ^ (num - 1))
    END IF
    ODEDn4 = D * a
```

Copyright by Mahidol University

```
END FUNCTION

FUNCTION ODEDn5 (num, t, x, y) STATIC
    a = PrimaryRadical(t, x, y)
    B = PriPr(t, x, y)
    C = ALPHAr(t, x, y)
    D = Ktci * a * B * (C ^ (num - 1))
    d1 = (num - 1) * .5 * Ktc * (B ^ 2) * (C ^ (num - 2))
    ODEDn5 = D + d1
END FUNCTION

FUNCTION ODEDn6 (num, t, x, y) STATIC
    B = PriPm(t, x, y)
    d1 = ALPHAm(t, x, y)
    a1 = (Ktrm * x) + (Ktd * (Pm + Pr + Pc))
    IF y > 0 AND Ktr > 0 THEN
        C = PriPc(t, x, y)
        d2 = ALPHAc(t, x, y)
        c31 = d2 ^ -1
        c32 = d1 ^ -1
        IF d1 > d2 THEN
            c5 = d2 / d1
            c4 = d1 ^ num
        END IF
        IF d2 > d1 THEN
            c5 = d1 / d2
            c4 = d2 ^ num
        END IF
        IF c5 < 1 AND c5 > 0 THEN
            c6 = 1 / (1 - c5)
        ELSEIF c5 = 1 THEN
            c6 = num - 1
        END IF
    END IF
    c2 = a1 * C * (d2 ^ (num - 1))
    c7 = Ktc * B * C * c31 * c32 * c4 * c6
    ODEDn6 = c2 + c7
END FUNCTION

FUNCTION ODEDn7 (num, t, x, y) STATIC
    IF y > 0 THEN
    a = (Ktr * y)
    B = PriPc(t, x, y)
    C = ALPHAc(t, x, y)
    D = B * (C ^ (num - 1))
    END IF
    ODEDn7 = D * a
    REM ODEDn7 = 0
END FUNCTION

FUNCTION ODEDn8 (num, t, x, y) STATIC
    IF y > 0 THEN
    a = PriPc(t, x, y)
    B = ALPHAc(t, x, y)
    C = (num - 1) * .5 * Ktc * (a ^ 2) * (B ^ (num - 2))
    END IF
    ODEDn8 = C
END FUNCTION

FUNCTION ODEDn9 (num, t, x, y) STATIC
    d2 = ALPHAr(t, x, y)
    c3 = d2 ^ -1
    IF y > 0 AND Ktr > 0 THEN
        C = PriPc(t, x, y)
        d1 = ALPHAc(t, x, y)
        c31 = d2 ^ -1
        c32 = d1 ^ -1
        IF d1 > d2 THEN
            c5 = d2 / d1
            c4 = d1 ^ num
        END IF
        IF d2 > d1 THEN
            c5 = d1 / d2
            c4 = d2 ^ num
        END IF
        IF c5 < 1 AND c5 > 0 THEN
            c6 = 1 / (1 - c5)
        ELSEIF c5 = 1 THEN
            c6 = num - 1
        END IF
    END IF
    c1 = Ktci * PrimaryRadical(t, x, y) * C * (d1 ^ (num -
1))
    c7 = Ktc * PriPr(t, x, y) * C * c31 * c32 * c4 * c6
    ODEDn9 = c1 + c7
END FUNCTION

FUNCTION ODEM (t, x, y) STATIC
'Routine which define ordinary differential equation of [M] (-
d[M]/dt)
'-d[M]/dt = 2KdfI + (Pr+Pc+Pm){Kp.M + Ktrm.M +
Ktr CTA -Ktci.[R]}
    B = PrimaryRadical(t, x, y)
    a = (Pr + Pc + Pm) * ((Kp * x) + (Ktrm * x) + (Ktr * y)
- (Ktci * B))
    ODEM = -(a + a1)
END FUNCTION

FUNCTION PrimaryRadical (t, x, y) STATIC
    It = Io * EXP(-Kd * t)
    a1 = 2 * Kd * f * It
    a2 = (Ki * x) + (Ksr * y) + (Ktci * (Pr + Pc + Pm))
    PrimaryRadical = a1 / a2
END FUNCTION

FUNCTION PriPc (t, x, y) STATIC
  a = ALPHAc(t, x, y)
  B = PrimaryRadical(t, x, y)
  a1 = Ksr * y * B * a
  a2 = Ktr * y * (Pr + Pc + Pm) * a
  a3 = Kp * x
  PriPc = (a1 + a2) / a3
END FUNCTION

FUNCTION PriPm (t, x, y) STATIC
  a = ALPHAm(t, x, y)
  a1 = Ktrm * x * (Pr + Pc + Pm) * a
  a2 = Kp * x
  PriPm = a1 / a2
END FUNCTION

FUNCTION PriPr (t, x, y) STATIC
  a = ALPHAr(t, x, y)
  B = PrimaryRadical(t, x, y)
  a1 = Ki * x * B * a
  a2 = Kp * x
  PriPr = a1 / a2
END FUNCTION
```

Copyright by Mahidol University

5) Programmes for calculation of the Modified Mayo Equation (CALQM.BAS)

```
DECLARE SUB CalculateDPn ()
DECLARE FUNCTION ODEDn1! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn2! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn3! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn4! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn5! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn6! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn7! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn8! (num!, t!, x!, y!)
DECLARE FUNCTION ODEDn9! (num!, t!, x!, y!)
DECLARE FUNCTION KtcGelEffect! (Xi!)
DECLARE FUNCTION KtdGelEffect! (Xi!)
DECLARE FUNCTION KtrmGelEffect! (Xi!)
DECLARE FUNCTION PrimaryRadical! (t!, x!, y!)
DECLARE FUNCTION ALPHAr! (t!, x!, y!)
DECLARE FUNCTION ALPHAc! (t!, x!, y!)
DECLARE FUNCTION ALPHAm! (t!, x!, y!)
DECLARE FUNCTION ODEM! (t!, x!, y!)
DECLARE FUNCTION ODEC! (t!, x!, y!)
DECLARE SUB LoadAllParameter ()
DECLARE FUNCTION PriPr! (t!, x!, y!)
DECLARE FUNCTION PriPc! (t!, x!, y!)
DECLARE FUNCTION PriPm! (t!, x!, y!)
DECLARE FUNCTION CALQ! (r!, D!, xMi!, xCi!,
xTime!)
DECLARE FUNCTION CalLOWERDPn! ()
DECLARE FUNCTION CalUPPERDPn! ()
DECLARE SUB CALLOOP1 (jCTA!)
DECLARE SUB CALLOOP2 (jCTA!, jM!, jn!, Ncount!)
DECLARE SUB NewtonRaphson (t!, x!, y!)
COMMON SHARED Mo, Io, CTAo1, CTAo2, Temp,
Time, FT, PercentSampling
COMMON SHARED Kp, Ktd, Ktc, Ktrm, Ki, Kd, f, Ktci,
Ksr, Ktr, Ktrm0, Ktd0, Ktc0
COMMON SHARED Pr, Pc, Pm, RatioCtoM,
SamplingTime
COMMON SHARED numberNCL, Xscale, h,
numberDATA, GelEffect$
LoadAllParameter
DIM SHARED NCL(1 TO numberNCL + 1)
DIM SHARED Wn(1 TO 9, 1 TO numberNCL + 1)
DIM SHARED DPn(1 TO 9)
DIM SHARED PeakArea(1 TO 10)
FT = Time * 60
Ktrm0 = Ktrm
Ktc0 = Ktc
Ktd0 = Ktd
OPEN "T-R-DP.MMR" FOR OUTPUT AS #1
Ccount = 0
FOR j = CTAo1 TO CTAo2 STEP (CTAo2 - CTAo1) /
numberDATA
    Mi = Mo
    CALL CALLOOP1(j)
    jNcount = 1
    FOR v = 0 TO Xscale STEP Xscale / numberNCL
        n = 10 ^ v
        NCL(jNcount) = n
        CALL CALLOOP2(j, Mi, v, jNcount)
        jNcount = jNcount + 1
    NEXT v
    CalculateDPn
    REM TDPn is the total degree of polymerization
(number average)
    TDPn = DPn(10)
    REM Save datafile for read in plotting 2-D graph after
simulation
    PRINT #1, SamplingTime, RatioCtoM, TDPn
        LOCATE 9, 13. COLOR 0, 11
        percent = (Ccount / numberDATA) * 100
        IF percent > 100 THEN percent = 100
        PRINT USING "###.##"; percent
  Ccount = Ccount + 1
NEXT j
CLOSE #1
END

FUNCTION ALPHAc (t, x, y) STATIC
    Radical = PrimaryRadical(t, x, y)
    a1 = Kp * x
    a2 = Ktrm * x
    a3 = Ktr * y
    a4 = Ktc * ((.5 * Pc) + Pr + Pm)
    a5 = Ktd * (Pr + Pc + Pm)
    a6 = Ktci * Radical
    a = a1 + a2 + a3 + a4 + a5 + a6
    ALPHAc = a1 / a
END FUNCTION

FUNCTION ALPHAm (t, x, y) STATIC
    Radical = PrimaryRadical(t, x, y)
    a1 = Kp * x
    a2 = Ktrm * x
    a3 = Ktr * y
    a4 = Ktc * (( 5 * Pm) + Pr + Pc)
    a5 = Ktd * (Pr + Pc + Pm)
    a6 = Ktci * Radical
    a = a1 + a2 + a3 + a4 + a5 + a6
    ALPHAm = a1 / a
END FUNCTION

FUNCTION ALPHAr (t, x, y) STATIC
    Radical = PrimaryRadical(t, x, y)
    a1 = Kp * x
    a2 = Ktrm * x
    a3 = Ktr * y
    a4 = Ktc * ((.5 * Pr) + Pc + Pm)
    a5 = Ktd * (Pr + Pc + Pm)
    a6 = Ktci * Radical
    a = a1 + a2 + a3 + a4 + a5 + a6
    ALPHAr = a1 / a
END FUNCTION

SUB CalculateDPn STATIC
    REM Defination.-          SUM{r=1->n}
[fraction(r).NCL(r)]
    REM          DPn(i),ave = ----------------------------
    REM                       SUM{r=1->n}  [fraction(r)]
->dDegreePol
    REM
    REM Calculate Degree of polymerization, DPn(i)
      FOR i = 1 TO 9
        uD = 0: dD = 0
        FOR r = 1 TO numberNCL
          uDegreePol = (Wn(i, r) * NCL(r)) + uD
          uD = uDegreePol
          dDegreePol = Wn(i, r) + dD
          dD = dDegreePol
        NEXT r
        IF dDegreePol > 0 THEN DPn(i) = uDegreePol /
dDegreePol ELSE DPn(i) = 0
      NEXT i
    REM  Peak area(i) = SUM{r=1->n} fraction(r).NCL(r)
    REM
    REM  Calculate of peak area(i) of distribution curve
    REM  to use in calculation of DPn,ave(total)
      FOR i = 1 TO 9
        uD = 0: dD = 0
        FOR r = 1 TO numberNCL
```

Copyright by Mahidol University

```
            uDegreePol = (Wn(i, r) * (LOG(NCL(r)) /
LOG(10))) + uD
            uD = uDegreePol
          NEXT r
          PeakArea(i) = uDegreePol
        NEXT i
    REM Calculation of Average degree of
polymerisation,DPn
    REM Defination - DPn,ave = SUM{i=1->9}
[DPn(i)*Peak area(i)]/SUM{i=1->9} [Peak area(i)]
    REM
        uD = 0: dD = 0: uW = 0: dW = 0
        FOR i = 1 TO 9
            uDPn10 = (PeakArea(i) * DPn(i)) + uD
            dDPn10 = PeakArea(i) + dD
            uD = uDPn10
            dD = dDPn10
        NEXT i
        IF dDPn10 > 0 THEN DPn(10) = uDPn10 /
dDPn10
END SUB

SUB CALLOOP1 (jCTA)
  Mi = Mo
  Ci = jCTA
  ITime = 0
  Stepnumber = FT / h
  count = 0
    FOR i = 0 TO Stepnumber
        IF GelEffect$ = "ON" THEN
        Ktrm = KtrmGelEffect((Mo - Mi) / Mo)
        Ktc = KtcGelEffect((Mo - Mi) / Mo)
        Ktd = KtdGelEffect((Mo - Mi) / Mo)
        ELSE
        Ktrm = Ktrm0
        Ktc = Ktc0
        Ktd = Ktd0
        END IF
        CALL NewtonRaphson(ITime, Mi, Ci)
        'Monomer at time T
        IF Mi <= 0 THEN EXIT FOR
            k1M = h * ODEM(ITime, Mi, Ci)
            k2M = h * ODEM(ITime + .5 * h, Mi + .5 *
k1M, Ci)
            k3M = h * ODEM(ITime + .5 * h, Mi + .5 *
k2M, Ci)
            k4M = h * ODEM(ITime + h, Mi + k3M, Ci)
            Mi = Mi + (k1M + 2 * k2M + 2 * k3M +
k4M) / 6
        'Chain transfer agent at time T
        IF Ci > 0 THEN
            k1C = h * ODEC(ITime, Ci, Mi)
            k2C = h * ODEC(ITime + .5 * h, Mi, Ci + .5
* k1C)
            k3C = h * ODEC(ITime + .5 * h, Mi, Ci + .5
* k2C)
            k4C = h * ODEC(ITime + h, Mi, Ci + k3C)
            Ci = Ci + (k1C + 2 * k2C + 2 * k3C + k4C) /
6
            IF Ci <= 0 THEN Ci = 0
        END IF
        LOCATE 11, 13: COLOR 0, 11
        MOpercent = ((Mo - Mi) / Mo) * 100
        IF MOpercent > 100 THEN MOpercent = 100
        PRINT USING "###.##"; MOpercent
        IF jCTA > 0 THEN
            LOCATE 13, 13: COLOR 0, 11
            CTApercent = ((jCTA - Ci) / jCTA) * 100
            IF CTApercent > 100 THEN CTApercent =
100
            PRINT USING "###.##"; CTApercent
        END IF
        LOCATE 15, 13: COLOR 0, 11
        PRINT USING "######     min"; ITime / 60
    REM SAMPLING
    MOpercent = ((Mo - Mi) / Mo) * 100
    IF MOpercent >= PercentSampling AND Mi > 0
THEN RatioCtoM = Ci / Mi: SamplingTime = ITime / 60:
EXIT FOR
    ITime = ITime + h
   count = count + 1
  NEXT i
END SUB

SUB CALLOOP2 (jCTA, jM, jn, Ncount)
  Mi = jM
  Ci = jCTA
  n = 10 ^ jn
    ITime = 0
    Stepnumber = FT / h
    count = 0
    REM define initial weight fraction at start reaction
time
    Dx1x = 0: Dx2x = 0: Dx3x = 0  Dx4x = 0: Dx5x =
0: Dx6x = 0: Dx7x = 0: Dx8x = 0  Dx9x = 0. DxTx = 0
    FOR i = 0 TO Stepnumber
      IF INKEY$ = CHR$(0) + "D" THEN END
      IF GelEffect$ = "ON" THEN
      Ktrm = KtrmGelEffect((Mo - Mi) / Mo)
      Ktc = KtcGelEffect((Mo - Mi) / Mo)
      Ktd = KtdGelEffect((Mo - Mi) / Mo)
      ELSE
      Ktrm = Ktrm0
      Ktc = Ktc0
      Ktd = Ktd0
      END IF
      CALL NewtonRaphson(ITime, Mi, Ci)
      'Monomer at time T
      IF Mi <= 0 THEN EXIT FOR
          k1M = h * ODEM(ITime, Mi, Ci)
          k2M = h * ODEM(ITime + .5 * h, Mi + .5 *
k1M, Ci)
          k3M = h * ODEM(ITime + .5 * h, Mi + .5 *
k2M, Ci)
          k4M = h * ODEM(ITime + h, Mi + k3M, Ci)
          Mi = Mi + (k1M + 2 * k2M + 2 * k3M +
k4M) / 6
      'Chain transfer agent at time T
      IF Ci > 0 THEN
          k1C = h * ODEC(ITime, Ci, Mi)
          k2C = h * ODEC(ITime + .5 * h, Mi, Ci + .5
* k1C)
          k3C = h * ODEC(ITime + .5 * h, Mi, Ci + .5
* k2C)
          k4C = h * ODEC(ITime + h, Mi, Ci + k3C)
          Ci = Ci + (k1C + 2 * k2C + 2 * k3C + k4C) /
6
          IF Ci <= 0 THEN Ci = 0
      END IF
      LOCATE 11, 13  COLOR 0, 11
      MOpercent = ((Mo - Mi) / Mo) * 100
      IF MOpercent > 100 THEN MOpercent = 100
      PRINT USING "### ##"; MOpercent
      IF jCTA > 0 THEN
          LOCATE 13, 13. COLOR 0, 11
          CTApercent = ((jCTA - Ci) / jCTA) * 100
          IF CTApercent > 100 THEN CTApercent =
100
          PRINT USING "###.##"; CTApercent
      END IF
      LOCATE 15, 13: COLOR 0, 11
      PRINT USING "######     min"; ITime / 60

  REM Calculate weight of each species
```

Copyright by Mahidol University

```
    REM Species Mn
        k1Dn1 = h * ODEDn1(n, ITime, Mi, Ci)
        k2Dn1 = h * ODEDn1(n, ITime + .5 * h, Mi, Ci +
.5 * k1Dn1)
        k3Dn1 = h * ODEDn1(n, ITime + .5 * h, Mi, Ci +
.5 * k2Dn1)
        k4Dn1 = h * ODEDn1(n, ITime + h, Mi, Ci +
k3Dn1)
        Dn1 = (k1Dn1 + 2 * k2Dn1 + 2 * k3Dn1 +
k4Dn1) / 6
        IF (Mo - Mi) > 0 THEN
            Wn1 = n * Dn1 / (Mo - Mi)
            Wn(1, Ncount) = Wn1 + Dx1x
            Dx1x = Wn(1, Ncount)
        END IF
    REM Species MnX
        k1Dn2 = h * ODEDn2(n, ITime, Mi, Ci)
        k2Dn2 = h * ODEDn2(n, ITime + .5 * h, Mi, Ci +
.5 * k1Dn2)
        k3Dn2 = h * ODEDn2(n, ITime + .5 * h, Mi, Ci +
.5 * k2Dn2)
        k4Dn2 = h * ODEDn2(n, ITime + h, Mi, Ci +
k3Dn1)
        Dn2 = (k1Dn2 + 2 * k2Dn2 + 2 * k3Dn2 +
k4Dn2) / 6
        IF (Mo - Mi) > 0 THEN
            Wn2 = n * Dn2 / (Mo - Mi)
            Wn(2, Ncount) = Wn2 + Dx2x
            Dx2x = Wn(2, Ncount)
        END IF
    REM Species RMn
        k1Dn3 = h * ODEDn3(n, ITime, Mi, Ci)
        k2Dn3 = h * ODEDn3(n, ITime + .5 * h, Mi, Ci +
.5 * k1Dn3)
        k3Dn3 = h * ODEDn3(n, ITime + .5 * h, Mi, Ci +
.5 * k2Dn3)
        k4Dn3 = h * ODEDn3(n, ITime + h, Mi, Ci +
k3Dn3)
        Dn3 = (k1Dn3 + 2 * k2Dn3 + 2 * k3Dn3 +
k4Dn3) / 6
        IF (Mo - Mi) > 0 THEN
            Wn3 = n * Dn3 / (Mo - Mi)
            Wn(3, Ncount) = Wn3 + Dx3x
            Dx3x = Wn(3, Ncount)
        END IF
    REM Species RMnX
        k1Dn4 = h * ODEDn4(n, ITime, Mi, Ci)
        k2Dn4 = h * ODEDn4(n, ITime + .5 * h, Mi, Ci +
.5 * k1Dn4)
        k3Dn4 = h * ODEDn4(n, ITime + .5 * h, Mi, Ci +
.5 * k2Dn4)
        k4Dn4 = h * ODEDn4(n, ITime + h, Mi, Ci +
k3Dn4)
        Dn4 = (k1Dn4 + 2 * k2Dn4 + 2 * k3Dn4 +
k4Dn4) / 6
        IF (Mo - Mi) > 0 THEN
            Wn4 = n * Dn4 / (Mo - Mi)
            Wn(4, Ncount) = Wn4 + Dx4x
            Dx4x = Wn(4, Ncount)
        END IF
    REM Species RMnR
        k1Dn5 = h * ODEDn5(n, ITime, Mi, Ci)
        k2Dn5 = h * ODEDn5(n, ITime + .5 * h, Mi, Ci +
 5 * k1Dn5)
        k3Dn5 = h * ODEDn5(n, ITime + .5 * h, Mi, Ci +
.5 * k2Dn5)
        k4Dn5 = h * ODEDn5(n, ITime + h, Mi, Ci +
k3Dn5)
        Dn5 = (k1Dn5 + 2 * k2Dn5 + 2 * k3Dn5 +
k4Dn5) / 6
        IF (Mo - Mi) > 0 THEN
            Wn5 = n * Dn5 / (Mo - Mi)
            Wn(5, Ncount) = Wn5 + Dx5x
            Dx5x = Wn(5, Ncount)
        END IF
    REM Species CTAMn
        k1Dn6 = h * ODEDn6(n, ITime, Mi, Ci)
        k2Dn6 = h * ODEDn6(n, ITime + .5 * h, Mi, Ci +
 5 * k1Dn6)
        k3Dn6 = h * ODEDn6(n, ITime + .5 * h, Mi, Ci +
 5 * k2Dn6)
        k4Dn6 = h * ODEDn6(n, ITime + h, Mi, Ci +
k3Dn6)
        Dn6 = (k1Dn6 + 2 * k2Dn6 + 2 * k3Dn6 +
k4Dn6) / 6
        IF (Mo - Mi) > 0 THEN
            Wn6 = n * Dn6 / (Mo - Mi)
            Wn(6, Ncount) = Wn6 + Dx6x
            Dx6x = Wn(6, Ncount)
        END IF
    REM Species CTAMnX
        k1Dn7 = h * ODEDn7(n, ITime, Mi, Ci)
        k2Dn7 = h * ODEDn7(n, ITime + .5 * h, Mi, Ci +
 5 * k1Dn7)
        k3Dn7 = h * ODEDn7(n, ITime + .5 * h, Mi, Ci +
 5 * k2Dn7)
        k4Dn7 = h * ODEDn7(n, ITime + h, Mi, Ci +
k3Dn7)
        Dn7 = (k1Dn7 + 2 * k2Dn7 + 2 * k3Dn7 +
k4Dn7) / 6
        IF (Mo - Mi) > 0 THEN
            Wn7 = n * Dn7 / (Mo - Mi)
            Wn(7, Ncount) = Wn7 + Dx7x
            Dx7x = Wn(7, Ncount)
        END IF
    REM Species CTAMnCTA
        k1Dn8 = h * ODEDn8(n, ITime, Mi, Ci)
        k2Dn8 = h * ODEDn8(n, ITime + .5 * h, Mi, Ci +
 5 * k1Dn8)
        k3Dn8 = h * ODEDn8(n, ITime + .5 * h, Mi, Ci +
.5 * k2Dn8)
        k4Dn8 = h * ODEDn8(n, ITime + h, Mi, Ci +
k3Dn8)
        Dn8 = (k1Dn8 + 2 * k2Dn8 + 2 * k3Dn8 +
k4Dn8) / 6
        IF (Mo - Mi) > 0 THEN
            Wn8 = n * Dn8 / (Mo - Mi)
            Wn(8, Ncount) = Wn8 + Dx8x
            Dx8x = Wn(8, Ncount)
        END IF
    REM Species CTAMnR
        k1Dn9 = h * ODEDn9(n, ITime, Mi, Ci)
        k2Dn9 = h * ODEDn9(n, ITime + .5 * h, Mi, Ci +
 5 * k1Dn9)
        k3Dn9 = h * ODEDn9(n, ITime + .5 * h, Mi, Ci +
 5 * k2Dn9)
        k4Dn9 = h * ODEDn9(n, ITime + h, Mi, Ci +
k3Dn9)
        Dn9 = (k1Dn9 + 2 * k2Dn9 + 2 * k3Dn9 +
k4Dn9) / 6
        IF (Mo - Mi) > 0 THEN
            Wn9 = n * Dn9 / (Mo - Mi)
            Wn(9, Ncount) = Wn9 + Dx9x
            Dx9x = Wn(9, Ncount)
        END IF
    REM ALL Species

        Wn(10, Ncount) = Dx1x + Dx2x + Dx3x + Dx4x
+ Dx5x + Dx6x + Dx7x + Dx8x + Dx9x

    REM SAMPLING
    IF MOpercent >= PercentSampling AND Mi > 0
THEN EXIT FOR
    ITime = ITime + h
```

Copyright by Mahidol University

```
        count = count - 1
    NEXT i
END SUB

FUNCTION CALQ (r, D, xMi, xCi, xTime) STATIC
    SELECT CASE D
        REM Mn
        CASE 1
            SELECT CASE r
                CASE 0
                IF Ktrm > 0 THEN
                    a1 = ALPHAm(xTime, xMi, xCi)
                    b1 = 1 - a1
                    a2 = PriPm(xTime, xMi, xCi)
                    T1 = (Ktrm * xMi) + (Ktd * (Pm + Pr +
Pc))
                    T2 = 5 * Ktc
                    a3 = T1 * a1 * a2 / b1
                    a4 = T2 * a2 * a2 / (b1 * b1)
                    CALQ = a3 + a4
                END IF
                CASE 1
                IF Ktrm > 0 THEN
                    a1 = ALPHAm(xTime, xMi, xCi)
                    b1 = 1 - a1
                    b2 = 2 - a1
                    a2 = PriPm(xTime, xMi, xCi)
                    T1 = (Ktrm * xMi) + (Ktd * (Pm + Pr +
Pc))
                    T2 = 5 * Ktc
                    a3 = T1 * a1 * a2 * b2 / (b1 * b1)
                    a4 = T2 * a2 * a2 * 2 / (b1 * b1 * b1)
                    CALQ = a3 + a4
                END IF
                CASE 2
                IF Ktrm > 0 THEN
                    a1 = ALPHAm(xTime, xMi, xCi)
                    b1 = 1 - a1
                    b2 = 4 - (3 * a1) + (a1 * a1)
                    b3 = 2 + a1
                    a2 = PriPm(xTime, xMi, xCi)
                    T1 = (Ktrm * xMi) + (Ktd * (Pm + Pr +
Pc))
                    T2 = 5 * Ktc
                    a3 = T1 * a1 * a2 * b2 / (b1 * b1 * b1)
                    a4 = T2 * a2 * a2 * 2 * b3 / (b1 * b1 *
b1 * b1)
                    CALQ = a3 + a4
                END IF
            END SELECT
        REM MnX
        CASE 2
            SELECT CASE r
                CASE 0
                IF Ktr > 0 AND xCi > 0 AND Ktrm > 0
THEN
                    a1 = ALPHAm(xTime, xMi, xCi)
                    b1 = 1 - a1
                    a2 = PriPm(xTime, xMi, xCi)
                    T1 = Ktr * xCi
                    a3 = T1 * a1 * a2 / b1
                    CALQ = a3
                END IF
                CASE 1
                IF Ktr > 0 AND xCi > 0 AND Ktrm > 0
THEN
                    a1 = ALPHAm(xTime, xMi, xCi)
                    b1 = 1 - a1
                    b2 = 2 - a1
                    a2 = PriPm(xTime, xMi, xCi)
                    T1 = Ktr * xCi
                    a3 = T1 * a1 * a2 * b2 / (b1 * b1)
                    CALQ = a3
                END IF
                CASE 2
                IF Ktr > 0 AND xCi > 0 AND Ktrm > 0
THEN
                    a1 = ALPHAm(xTime, xMi, xCi)
                    b1 = 1 - a1
                    b2 = 4 - (3 * a1) + (a1 * a1)
                    a2 = PriPm(xTime, xMi, xCi)
                    T1 = Ktr * xCi
                    a3 = T1 * a1 * a2 * b2 / (b1 * b1 * b1)
                    CALQ = a3
                END IF
            END SELECT
        REM RMn
        CASE 3
            SELECT CASE r
                CASE 0
                    a0 = PrimaryRadical(xTime, xMi, xCi)
                    a1 = ALPHAm(xTime, xMi, xCi)
                    b1 = 1 - a1
                    a2 = ALPHAr(xTime, xMi, xCi)
                    b2 = 1 - a2
                    a3 = PriPm(xTime, xMi, xCi)
                    a4 = PriPr(xTime, xMi, xCi)
                    T1 = Ktci * a0
                    T2 = (Ktrm * xMi) + (Ktd * (Pm + Pr +
Pc))
                    T3 = Ktc * a3 * a4 / (a1 * a2)
                    a5 = T1 * a3 * a1 / b1
                    a6 = T2 * a4 * a2 / b2
                    IF (a1 / a2) < 1 THEN
                        a7 = T3 * (a2 * a2 - b2) * ((a1 / a2)
/ (1 - (a1 / a2)))
                    ELSEIF (a2 / a1) < 1 THEN
                        a7 = T3 * (a1 * a1 / b1) * ((a2 / a1)
/ (1 - (a2 / a1)))
                    ELSEIF (a2 / a1) = 1 THEN
                        a7 = T3 * a2 * a2 / (b2 * b2)
                    END IF
                    CALQ = a5 + a6 + a7
                CASE 1
                    a0 = PrimaryRadical(xTime, xMi, xCi)
                    a1 = ALPHAm(xTime, xMi, xCi)
                    b1 = 1 - a1
                    a2 = ALPHAr(xTime, xMi, xCi)
                    b2 = 1 - a2
                    a3 = PriPm(xTime, xMi, xCi)
                    a4 = PriPr(xTime, xMi, xCi)
                    T1 = Ktci * a0
                    T2 = (Ktrm * xMi) + (Ktd * (Pm + Pr +
Pc))
                    T3 = Ktc * a3 * a4 / (a1 * a2)
                    a5 = T1 * a3 * a1 * (2 - a1) / (b1 * b1)
                    a6 = T2 * a4 * a2 * (2 - a2) / (b2 * b2)
                    IF (a1 / a2) < 1 THEN
                        a7 = T3 * ((2 - a2) * a2 * a2 / (b2 *
b2)) * ((a1 / a2) / (1 - (a1 / a2)))
                    ELSEIF (a2 / a1) < 1 THEN
                        a7 = T3 * ((2 - a1) * a1 * a1 / (b1 *
b1)) * ((a2 / a1) / (1 - (a2 / a1)))
                    ELSEIF (a2 / a1) = 1 THEN
                        a7 = T3 * 2 * a2 * a2 / (b2 * b2 *
b2)
                    END IF
                    CALQ = a5 + a6 + a7
                CASE 2
                    a0 = PrimaryRadical(xTime, xMi, xCi)
                    a1 = ALPHAm(xTime, xMi, xCi)
                    b1 = 1 - a1
                    a2 = ALPHAr(xTime, xMi, xCi)
                    b2 = 1 - a2
```

Copyright by Mahidol University

```
                a3 = PriPm(xTime, xMi, xCi)
                a4 = PriPr(xTime, xMi, xCi)
                T1 = Ktci * a0
                T2 = (Ktrm * xMi) + (Ktd * (Pm + Pr +
Pc))
                T3 = Ktc * a3 * a4 / (a1 * a2)
                a5 = T1 * a3 * a1 * (4 - (3 * a1) + (a1 *
a1)) / (b1 * b1 * b1)
                a6 = T2 * a4 * a2 * (4 - (3 * a2) + (a2 *
a2)) / (b2 * b2 * b2)
                IF (a1 / a2) < 1 THEN
                    a7 = T3 * ((4 - (3 * a2) + (a2 * a2))
* a2 * a2 / (b2 * b2 * b2)) * ((a1 / a2) / (1 - (a1 / a2)))
                ELSEIF (a2 / a1) < 1 THEN
                    a7 = T3 * ((4 - (3 * a1) + (a1 * a1))
* a1 * a1 / (b1 * b1 * b1)) * ((a2 / a1) / (1 - (a2 / a1)))
                ELSEIF (a2 / a1) = 1 THEN
                    a7 = T3 * 2 * (2 + a2) * a2 * a2 /
(b2 * b2 * b2 * b2)
                END IF
                CALQ = a5 + a6 + a7
            END SELECT
        REM RMnX
        CASE 4
            SELECT CASE r
              CASE 0
                IF Ktr > 0 AND xCi > 0 THEN
                    a1 = ALPHAr(xTime, xMi, xCi)
                    b1 = 1 - a1
                    a2 = PriPr(xTime, xMi, xCi)
                    T1 = Ktr * xCi
                    a3 = T1 * a1 * a2 / b1
                    CALQ = a3
                END IF
              CASE 1
                IF Ktr > 0 AND xCi > 0 THEN
                    a1 = ALPHAr(xTime, xMi, xCi)
                    b1 = 1 - a1
                    b2 = 2 - a1
                    a2 = PriPr(xTime, xMi, xCi)
                    T1 = Ktr * xCi
                    a3 = T1 * a1 * a2 * b2 / (b1 * b1)
                    CALQ = a3
                END IF
              CASE 2
                IF Ktr > 0 AND xCi > 0 THEN
                    a1 = ALPHAr(xTime, xMi, xCi)
                    b1 = 1 - a1
                    b2 = 4 - (3 * a1) + (a1 * a1)
                    a2 = PriPr(xTime, xMi, xCi)
                    T1 = Ktr * xCi
                    a3 = T1 * a1 * a2 * b2 / (b1 * b1 * b1)
                    CALQ = a3
                END IF
            END SELECT
        REM RMnR
        CASE 5
            SELECT CASE r
              CASE 0
                    a0 = PrimaryRadical(xTime, xMi, xCi)
                    a1 = ALPHAr(xTime, xMi, xCi)
                    b1 = 1 - a1
                    a2 = PriPr(xTime, xMi, xCi)
                    T1 = Ktci * a0
                    T2 = .5 * Ktc
                    a3 = T1 * a1 * a2 / b1
                    a4 = T2 * a2 * a2 / (b1 * b1)
                    CALQ = a3 + a4
              CASE 1
                    a0 = PrimaryRadical(xTime, xMi, xCi)
                    a1 = ALPHAr(xTime, xMi, xCi)
                    b1 = 1 - a1
                    b2 = 2 - a1
                    a2 = PriPr(xTime, xMi, xCi)
                    T1 = Ktci * a0
                    T2 = 5 * Ktc
                    a3 = T1 * a1 * a2 * b2 / (b1 * b1)
                    a4 = T2 * a2 * a2 * 2 / (b1 * b1 * b1)
                    CALQ = a3 + a4
              CASE 2
                    a0 = PrimaryRadical(xTime, xMi, xCi)
                    a1 = ALPHAr(xTime, xMi, xCi)
                    b1 = 1 - a1
                    b2 = 4 - (3 * a1) + (a1 * a1)
ÿÿ#l$$!
3 = 2 + a1          a2 = PriPr(xTime, xMi, xCi)
                    T1 = Ktci * a0
                    T2 = .5 * Ktc
                    a3 = T1 * a1 * a2 * b2 / (b1 * b1 * b1)
                    a4 = T2 * a2 * a2 * 2 * b3 / (b1 * b1 *
b1 * b1)
                    CALQ = a3 + a4
            END SELECT
        REM CTAMn
        CASE 6
            SELECT CASE r
              CASE 0
                    a1 = ALPHAm(xTime, xMi, xCi)
                    b1 = 1 - a1
                    a2 = ALPHAc(xTime, xMi, xCi)
                    b2 = 1 - a2
                    a3 = PriPm(xTime, xMi, xCi)
                    a4 = PriPc(xTime, xMi, xCi)
                    T1 = (Ktrm * xMi) + (Ktd * (Pm + Pr +
Pc))
                    T2 = Ktc * a3 * a4 / (a1 * a2)
                    a5 = T1 * a4 * a2 / b2
                    IF (a1 / a2) < 1 THEN
                        a6 = T2 * (a2 * a2 / b2) * ((a1 / a2)
/ (1 - (a1 / a2)))
                    ELSEIF (a2 / a1) < 1 THEN
                        a6 = T2 * (a1 * a1 / b1) * ((a2 / a1)
/ (1 - (a2 / a1)))
                    ELSEIF (a2 / a1) = 1 THEN
                        a6 = T2 * a2 * a2 / (b2 * b2)
                    END IF
                    CALQ = a5 + a6
              CASE 1
                    a1 = ALPHAm(xTime, xMi, xCi)
                    b1 = 1 - a1
                    a2 = ALPHAc(xTime, xMi, xCi)
                    b2 = 1 - a2
                    a3 = PriPm(xTime, xMi, xCi)
                    a4 = PriPc(xTime, xMi, xCi)
                    T1 = (Ktrm * xMi) + (Ktd * (Pm + Pr +
Pc))
                    T2 = Ktc * a3 * a4 / (a1 * a2)
                    a5 = T1 * a4 * a2 * (2 - a2) / (b2 * b2)
                    IF (a1 / a2) < 1 THEN
                        a6 = T2 * ((2 - a2) * a2 * a2 / (b2 *
b2)) * ((a1 / a2) / (1 - (a1 / a2)))
                    ELSEIF (a2 / a1) < 1 THEN
                        a6 = T2 * ((2 - a1) * a1 * a1 / (b1 *
b1)) * ((a2 / a1) / (1 - (a2 / a1)))
                    ELSEIF (a2 / a1) = 1 THEN
                        a6 = T2 * 2 * a2 * a2 / (b2 * b2 *
b2)
                    END IF
                    CALQ = a5 + a6
              CASE 2
                    a1 = ALPHAm(xTime, xMi, xCi)
                    b1 = 1 - a1
                    a2 = ALPHAc(xTime, xMi, xCi)
                    b2 = 1 - a2
```

ลิขสิทธิ์มหาวิทยาลัยมหิดล
Copyright by Mahidol University

```
                a3 = PriPm(xTime, xMi, xCi)
                a4 = PriPc(xTime, xMi, xCi)
                T1 = (Ktrm * xMi) + (Ktd * (Pm + Pr +
Pc))
                T2 = Ktc * a3 * a4 / (a1 * a2)
                a5 = T1 * a4 * a2 * (4 - (3 * a2) + (a2 *
a2)) / (b2 * b2 * b2)
                IF (a1 / a2) < 1 THEN
                    a6 = T2 * ((4 - (3 * a2) + (a2 * a2))
* a2 * a2 / (b2 * b2 * b2)) * ((a1 / a2) / (1 - (a1 / a2)))
                ELSEIF (a2 / a1) < 1 THEN
                    a6 = T2 * ((4 - (3 * a1) + (a1 * a1))
* a1 * a1 / (b1 * b1 * b1)) * ((a2 / a1) / (1 - (a2 / a1)))
                ELSEIF (a2 / a1) = 1 THEN
                    a6 = T2 * 2 * (2 + a2) * a2 * a2 /
(b2 * b2 * b2 * b2)
                END IF
                CALQ = a5 + a6
            END SELECT
        REM CTAMnX
        CASE 7
            SELECT CASE r
              CASE 0
                IF Ktr > 0 AND xC1 > 0 THEN
                    a1 = ALPHAc(xTime, xMi, xCi)
                    b1 = 1 - a1
                    a2 = PriPc(xTime, xMi, xC1)
                    T1 = Ktr * xC1
                    a3 = T1 * a1 * a2 / b1
                    CALQ = a3
                END IF
              CASE 1
                IF Ktr > 0 AND xC1 > 0 THEN
                    a1 = ALPHAc(xTime, xMi, xC1)
                    b1 = 1 - a1
                    b2 = 2 - a1
                    a2 = PriPc(xTime, xMi, xCi)
                    T1 = Ktr * xCi
                    a3 = T1 * a1 * a2 * b2 / (b1 * b1)
                    CALQ = a3
                END IF
              CASE 2
                IF Ktr > 0 AND xC1 > 0 THEN
                    a1 = ALPHAc(xTime, xMi, xC1)
                    b1 = 1 - a1
                    b2 = 4 - (3 * a1) + (a1 * a1)
                    a2 = PriPc(xTime, xMi, xCi)
                    T1 = Ktr * xC1
                    a3 = T1 * a1 * a2 * b2 / (b1 * b1 * b1)
                    CALQ = a3
                END IF
            END SELECT
        REM CTAMnCTA
        CASE 8
            SELECT CASE r
              CASE 0
                IF Ktr > 0 AND xCi > 0 THEN
                    a1 = ALPHAc(xTime, xM1, xCi)
                    b1 = 1 - a1
                    a2 = PriPc(xTime, xM1, xCi)
                    T1 = .5 * Ktc
                    a3 = T1 * a2 * a2 / (b1 * b1)
                    CALQ = a3
                END IF
              CASE 1
                IF Ktr > 0 AND xCi > 0 THEN
                    a1 = ALPHAc(xTime, xMi, xCi)
                    b1 = 1 - a1
                    a2 = PriPc(xTime, xMi, xCi)
                    T1 = .5 * Ktc
                    a3 = T1 * a2 * a2 * 2 / (b1 * b1 * b1)
                    CALQ = a3
                END IF
              CASE 2
                IF Ktr > 0 AND xCi > 0 THEN
                    a1 = ALPHAc(xTime, xMi, xC1)
                    b1 = 1 - a1
                    a2 = PriPc(xTime, xMi, xCi)
                    T1 = .5 * Ktc
                    a3 = T1 * a2 * a2 * 2 * (2 + a1) / (b1 *
b1 * b1 * b1)
                    CALQ = a3
                END IF
            END SELECT
        REM CTAMnR
        CASE 9
            SELECT CASE r
              CASE 0
                    a0 = PrimaryRadical(xTime, xMi, xCi)
                    a1 = ALPHAc(xTime, xMi, xCi)
                    b1 = 1 - a1
                    a2 = ALPHAr(xTime, xM1, xCi)
                    b2 = 1 - a2
                    a3 = PriPc(xTime, xM1, xCi)
                    a4 = PriPr(xTime, xMi, xCi)
                    T1 = Ktci * a0
                    T2 = Ktc * a3 * a4 / (a1 * a2)
                    a5 = T1 * a3 * a1 / b1
                    IF (a1 / a2) < 1 THEN
                        a6 = T2 * (a2 * a2 / b2) * ((a1 / a2)
/ (1 - (a1 / a2)))
                    ELSEIF (a2 / a1) < 1 THEN
                        a6 = T2 * (a1 * a1 / b1) * ((a2 / a1)
/ (1 - (a2 / a1)))
                    ELSEIF (a2 / a1) = 1 THEN
                        a6 = T2 * a2 * a2 / (b2 * b2)
                    END IF
                    CALQ = a5 + a6
              CASE 1
                    a0 = PrimaryRadical(xTime, xM1, xCi)
                    a1 = ALPHAc(xTime, xM1, xCi)
                    b1 = 1 - a1
                    a2 = ALPHAr(xTime, xMi, xCi)
                    b2 = 1 - a2
                    a3 = PriPc(xTime, xMi, xCi)
                    a4 = PriPr(xTime, xM1, xCi)
                    T1 = Ktci * a0
                    T2 = Ktc * a3 * a4 / (a1 * a2)
                    a5 = T1 * a3 * a1 * (2 - a1) / (b1 * b1)
                    IF (a1 / a2) < 1 THEN
                        a6 = T2 * ((2 - a2) * a2 * a2 / (b2 *
b2)) * ((a1 / a2) / (1 - (a1 / a2)))
                    ELSEIF (a2 / a1) < 1 THEN
                        a6 = T2 * ((2 - a1) * a1 * a1 / (b1 *
b1)) * ((a2 / a1) / (1 - (a2 / a1)))
                    ELSEIF (a2 / a1) = 1 THEN
                        a6 = T2 * 2 * a2 * a2 / (b2 * b2 *
b2)
                    END IF
                    CALQ = a5 + a6
              CASE 2
                    a0 = PrimaryRadical(xTime, xMi, xC1)
                    a1 = ALPHAc(xTime, xM1, xCi)
                    b1 = 1 - a1
                    a2 = ALPHAr(xTime, xMi, xC1)
                    b2 = 1 - a2
                    a3 = PriPc(xTime, xM1, xCi)
                    a4 = PriPr(xTime, xMi, xCi)
                    T1 = Ktci * a0
                    T2 = Ktc * a3 * a4 / (a1 * a2)
                    a5 = T1 * a3 * a1 * (4 - (3 * a1) + (a1 *
a1)) / (b1 * b1 * b1)
                    IF (a1 / a2) < 1 THEN
```

Copyright by Mahidol University

```
                    a6 = T2 * ((4 - (3 * a2) + (a2 * a2))
* a2 * a2 / (b2 * b2 * b2)) * ((a1 / a2) / (1 - (a1 / a2)))
                ELSEIF (a2 / a1) = 1 THEN
                    a6 = T2 * ((4 - (3 * a1) + (a1 * a1))
* a1 * a1 / (b1 * b1 * b1)) * ((a2 / a1) / (1 - (a2 / a1)))
                ELSEIF (a2 / a1) = 1 THEN
                    a6 = T2 * 2 * (2 + a2) * a2 * a2 /
(b2 * b2 * b2 * b2)
                END IF
                CALQ = a5 + a6
            END SELECT
    END SELECT
END FUNCTION

FUNCTION KtcGelEffect (X1) STATIC
    B = 2.5882 - (3.4852 * .001 * (Temp + 273.3))
    C = 4.3108 - (1.2635 * .01 * (Temp + 273.3))
    D = -4.6488 + (1.9026 * .01 * (Temp + 273.3))
    E = (B * Xi) + (C * X1 * Xi) + (D * Xi * X1 * X1)
    KtcGelEffect = Ktc0 * EXP(-2 * E)
END FUNCTION

FUNCTION KtdGelEffect (X1) STATIC
    B = 2.5882 - (3.4852 * .001 * (Temp + 273.3))
    C = 4.3108 - (1.2635 * .01 * (Temp + 273.3))
    D = -4.6488 + (1.9026 * .01 * (Temp + 273.3))
    E = (B * Xi) + (C * Xi * X1) + (D * X1 * Xi * Xi)
    KtdGelEffect = Ktd0 * EXP(-2 * E)
END FUNCTION

FUNCTION KtrmGelEffect (Xi) STATIC
    B = 2.5882 - (3.4852 * .001 * (Temp + 273.3))
    C = 4.3108 - (1.2635 * .01 * (Temp + 273.3))
    D = -4.6488 + (1.9026 * .01 * (Temp + 273.3))
    E = (B * X1) + (C * Xi * Xi) + (D * X1 * Xi * Xi)
    KtrmGelEffect = Ktrm0 * EXP(-2 * E)
END FUNCTION

SUB LoadAllParameter
    OPEN "CONST.MTY" FOR INPUT AS #1
      INPUT #1, MoName$
      INPUT #1, IoName$
      INPUT #1, SoName$
      INPUT #1, Mo
      INPUT #1, Io
      INPUT #1, CTAo1
      INPUT #1, CTAo2
      INPUT #1, PercentSampling
      INPUT #1, Temp
      INPUT #1, Time
      INPUT #1, Kp
      INPUT #1, Ktc
      INPUT #1, Ktd
      INPUT #1, Ktrm
      INPUT #1, K1
      INPUT #1, Kd
      INPUT #1, f
      INPUT #1, Kte1
      INPUT #1, Ksr
      INPUT #1, Ktr
      INPUT #1, numberNCL
      INPUT #1, Xscale
      INPUT #1, h
      INPUT #1, numberDATA
      INPUT #1, GelEffect$
    CLOSE #1
END SUB

SUB NewtonRaphson (t, x, y) STATIC
    It = Io * EXP(-Kd * t)
    Prapx = SQR(2 * Kd * f * It / ((.5 * Ktc) + Ktd))
    IF Ktrm > 0 THEN
        Pmapx = Prapx
    ELSE
        Pmapx = 0
    END IF
    IF y > 0 AND Ktr > 0 THEN
        Pcapx = Prapx
    ELSE
        Pcapx = 0
    END IF
FOR n = 1 TO 100

r1 = 2 * Ki * x * Kd * f * It * Kte1
r2 = (Ki * x) + (Ksr * y) + (Kte1 * (Prapx + Pcapx +
Pmapx))
r3 = Ktc * (Prapx + Pcapx + Pmapx)
r4 = Ktd * ((2 * Prapx) + Pcapx + Pmapx)
r5 = (Ktr * y) + (Ktrm * x)
r6 = 2 * Kd * f * It * Kte1
r7 = (Ki * x) + (Ksr * y) + (Kte1 * (Pcapx + Pmapx))
r8 = 2 * Kd * f * It
r10 = r8 / r2          '1st radical
r11 = K1 * x * r10
r12 = Prapx * Ktc * (( 5 * Prapx) + Pcapx + Pmapx)
r13 = Prapx * Ktd * (Prapx + Pcapx + Pmapx)
r14 = (Ktr * y * Prapx) + (Ktrm * x * Prapx)
r15 = Kte1 * r10 * Prapx
dFr = (r1 / (r2 ^ 2)) + (r3 + r4 + r5) + ((r6 * r7) / (r2 ^ 2))
Fr = -r11 + (r12 + r13 + r14 + r15)
Prt = Prapx - (Fr / dFr)

IF y > 0 AND Ktr > 0 THEN
c1 = 2 * Ksr * y * Kd * f * It * Kte1
c2 = (Ki * x) + (Ksr * y) + (Kte1 * (Prapx + Pcapx +
Pmapx))
c3 = Ktc * (Prapx + Pcapx + Pmapx)
c4 = Ktd * ((2 * Pcapx) + Prapx + Pmapx)
c5 = Ktrm * x
c6 = 2 * Kd * f * It * Kte1
c7 = (K1 * x) + (Ksr * y) + (Kte1 * (Prapx + Pmapx))
c8 = 2 * Kd * f * It
c10 = c8 / c2          '1st radical
c11 = (Ksr * y * c10) + (Ktr * y * (Pcapx + Prapx +
Pmapx))
c12 = Pcapx * Ktc * ((.5 * Pcapx) + Prapx + Pmapx)
c13 = Pcapx * Ktd * (Pcapx + Prapx + Pmapx)
c14 = (Ktr * y * Pcapx) + (Ktrm * x * Pcapx)
c15 = Kte1 * c10 * Pcapx
dFc = (c1 / (c2 ^ 2)) + (c3 + c4 + c5) + ((c6 * c7) / (c2 ^ 2))
Fc = -c11 + (c12 + c13 + c14 + c15)
Pct = Pcapx - (Fc / dFc)
END IF

IF Ktrm > 0 THEN
m2 = (Ki * x) + (Ksr * y) + (Kte1 * (Prapx + Pcapx +
Pmapx))
m3 = Ktc * (Prapx + Pcapx + Pmapx)
m4 = Ktd * ((2 * Pmapx) + Prapx + Pcapx)
m5 = Ktr * y
m6 = 2 * Kd * f * It * Kte1
m7 = (Ki * x) + (Ksr * y) + (Kte1 * (Prapx + Pcapx))
m8 = 2 * Kd * f * It
m10 = m8 / m2          '1st radical
m11 = Ktrm * x * (Pcapx + Prapx + Pmapx)
m12 = Pmapx * Ktc * ((.5 * Pmapx) + Prapx + Pcapx)
m13 = Pmapx * Ktd * (Pcapx + Prapx + Pmapx)
m14 = (Ktr * y * Pmapx) + (Ktrm * x * Pmapx)
m15 = Kte1 * m10 * Pmapx
dFm = (m3 + m4 + m5) + ((m6 * m7) / (m2 ^ 2))
Fm = -m11 + (m12 + m13 + m14 + m15)
Pmt = Pmapx - (Fm / dFm)
END IF
```

Copyright by Chulalongkorn University

```
IF Prt > 0 AND Pct > 0 AND Pmt > 0 THEN
    IF ABS((Pmt - Pmapx) / Pmt) <= .0001 AND
ABS((Pct - Pcapx) / Pct) <= .0001 AND ABS((Prt - Prapx) /
Prt) <= .0001 THEN EXIT FOR
ELSEIF Prt > 0 AND Pct > 0 AND Pmt = 0 THEN
    IF ABS((Pct - Pcapx) / Pct) <= .0001 AND ABS((Prt -
Prapx) / Prt) <= .0001 THEN EXIT FOR
ELSEIF Prt > 0 AND Pct = 0 AND Pmt > 0 THEN
    IF ABS((Pmt - Pmapx) / Pmt) <= .0001 AND
ABS((Prt - Prapx) / Prt) <= .0001 THEN EXIT FOR
ELSEIF Prt > 0 AND Pct = 0 AND Pmt = 0 THEN
    IF ABS((Prt - Prapx) / Prt) <= .0001 THEN EXIT
FOR
END IF
Prapx = Prt
Pcapx = Pct
Pmapx = Pmt
NEXT n
Pr = Prt
Pc = Pct
Pm = Pmt
END SUB

FUNCTION ODEC (t, x, y) STATIC
'Routine which define ordinary differential equation of
[CTA] (-d[CTA]/dt)
'-d[CTA]/dt = Ksr[R][CTA] + Ktr[CTA](Pr+Pc+Pm)
    B = PrimaryRadical(t, x, y)
    a = (Ksr * B * y) + (Ktr * y * (Pr + Pc + Pm))
    ODEC = -a
END FUNCTION

FUNCTION ODEDn1 (num, t, x, y) STATIC
    a = (Ktrm * x) + (Ktd * (Pm + Pr + Pc))
    B = PriPm(t, x, y)
    C = ALPHAm(t, x, y)
    D = B * (C ^ (num - 1))
    d1 = (num - 1) * .5 * Ktc * (B ^ 2) * (C ^ (num - 2))
    ODEDn1 = (D * a) + d1
END FUNCTION

FUNCTION ODEDn2 (num, t, x, y) STATIC
    IF y > 0 THEN
    a = (Ktr * y)
    B = PriPm(t, x, y)
    C = ALPHAm(t, x, y)
    D = B * (C ^ (num - 1))
    END IF
    ODEDn2 = D * a
END FUNCTION

FUNCTION ODEDn3 (num, t, x, y) STATIC
    a = PrimaryRadical(t, x, y)
    B = PriPm(t, x, y)
    C = PriPr(t, x, y)
    d1 = ALPHAm(t, x, y)
    d2 = ALPHAr(t, x, y)
    a1 = (Ktrm * x) + (Ktd * (Pm + Pr + Pc))
    c31 = d1 ^ -1
    c32 = d2 ^ -1
    IF d1 > d2 THEN
        c5 = d2 / d1
        c4 = d1 ^ num
    END IF
    IF d2 > d1 THEN
        c5 = d1 / d2
        c4 = d2 ^ num
    END IF
    IF c5 < 1 AND c5 > 0 THEN
        c6 = 1 / (1 - c5)
    ELSEIF c5 = 1 THEN
        c6 = num - 1
    END IF
    c1 = Ktc1 * a * B * (d1 ^ (num - 1))
    c2 = a1 * C * (d2 ^ (num - 1))
    c7 = Ktc * B * C * c31 * c32 * c4 * c6
    ODEDn3 = c1 + c2 + c7
END FUNCTION

FUNCTION ODEDn4 (num, t, x, y) STATIC
    IF y > 0 THEN
    a = (Ktr * y)
    B = PriPr(t, x, y)
    C = ALPHAr(t, x, y)
    D = B * (C ^ (num - 1))
    END IF
    ODEDn4 = D * a
END FUNCTION

FUNCTION ODEDn5 (num, t, x, y) STATIC
    a = PrimaryRadical(t, x, y)
    B = PriPr(t, x, y)
    C = ALPHAr(t, x, y)
    D = Ktc1 * a * B * (C ^ (num - 1))
    d1 = (num - 1) *  5 * Ktc * (B ^ 2) * (C ^ (num - 2))
    ODEDn5 = D + d1
END FUNCTION

FUNCTION ODEDn6 (num, t, x, y) STATIC
    B = PriPm(t, x, y)
    d1 = ALPHAm(t, x, y)
    a1 = (Ktrm * x) + (Ktd * (Pm + Pr + Pc))
    IF y > 0 AND Ktr > 0 THEN
        C = PriPc(t, x, y)
        d2 = ALPHAc(t, x, y)
        c31 = d2 ^ -1
        c32 = d1 ^ -1
        IF d1 > d2 THEN
            c5 = d2 / d1
            c4 = d1 ^ num
        END IF
        IF d2 > d1 THEN
            c5 = d1 / d2
            c4 = d2 ^ num
        END IF
        IF c5 < 1 AND c5 > 0 THEN
            c6 = 1 / (1 - c5)
        ELSEIF c5 = 1 THEN
            c6 = num - 1
        END IF
    END IF
    c2 = a1 * C * (d2 ^ (num - 1))
    c7 = Ktc * B * C * c31 * c32 * c4 * c6
    ODEDn6 = c2 + c7
END FUNCTION

FUNCTION ODEDn7 (num, t, x, y) STATIC

    IF y > 0 THEN
    a = (Ktr * y)
    B = PriPc(t, x, y)
    C = ALPHAc(t, x, y)
    D = B * (C ^ (num - 1))
    END IF
    ODEDn7 = D * a
    REM ODEDn7 = 0
END FUNCTION

FUNCTION ODEDn8 (num, t, x, y) STATIC
    IF y > 0 THEN
    a = PriPc(t, x, y)
    B = ALPHAc(t, x, y)
    C = (num - 1) * .5 * Ktc * (a ^ 2) * (B ^ (num - 2))
    END IF
```

Copyright by Mahidol University

```
        ODEDn8 = C
END FUNCTION

FUNCTION ODEDn9 (num, t, x, y) STATIC
        d2 = ALPHAr(t, x, y)
        c3 = d2 ^ -1
        IF y > 0 AND Ktr > 0 THEN
            C = PriPc(t, x, y)
            d1 = ALPHAc(t, x, y)
            c31 = d2 ^ -1
            c32 = d1 ^ -1
            IF d1 > d2 THEN
                c5 = d2 / d1
                c4 = d1 ^ num
            END IF
            IF d2 > d1 THEN
                c5 = d1 / d2
                c4 = d2 ^ num
            END IF
            IF c5 < 1 AND c5 > 0 THEN
                c6 = 1 / (1 - c5)
            ELSEIF c5 = 1 THEN
                c6 = num - 1
            END IF
        END IF
        c1 = Ktci * PrimaryRadical(t, x, y) * C * (d1 ^ (num -
1))
        c7 = Ktc * PriPr(t, x, y) * C * c31 * c32 * c4 * c6
        ODEDn9 = c1 + c7
END FUNCTION

FUNCTION ODEM (t, x, y) STATIC
'Routine which define ordinary differential equation of [M] (-
d[M]/dt)
'-d[M]/dt = 2KdfI + (Pr+Pc+Pm){Kp.M + Ktrm.M +
Ktr.CTA -Ktci.[R]}
        B = PrimaryRadical(t, x, y)
        a = (Pr + Pc + Pm) * ((Kp * x) + (Ktrm * x) + (Ktr * y)
- (Ktci * B))
        ODEM = -(a + a1)
END FUNCTION

FUNCTION PrimaryRadical (t, x, y) STATIC
        It = Io * EXP(-Kd * t)
        a1 = 2 * Kd * f * It
        a2 = (Ki * x) + (Ksr * y) + (Ktci * (Pr + Pc + Pm))
        PrimaryRadical = a1 / a2
END FUNCTION

FUNCTION PriPc (t, x, y) STATIC
    a = ALPHAc(t, x, y)
    B = PrimaryRadical(t, x, y)
    a1 = Ksr * y * B * a
    a2 = Ktr * y * (Pr + Pc + Pm) * a
    a3 = Kp * x
    PriPc = (a1 + a2) / a3
END FUNCTION

FUNCTION PriPm (t, x, y) STATIC
    a = ALPHAm(t, x, y)
    a1 = Ktrm * x * (Pr + Pc + Pm) * a
    a2 = Kp * x
    PriPm = a1 / a2
END FUNCTION

FUNCTION PriPr (t, x, y) STATIC
    a = ALPHAr(t, x, y)
    B = PrimaryRadical(t, x, y)
    a1 = Ki * x * B * a
    a2 = Kp * x
    PriPr = a1 / a2
END FUNCTION
```

Copyright by Mahidol University

6) Programmes for displaying the weight distribution of dead-end polymers (PLOTW-2D.BAS)

```
DECLARE SUB CalculateDPnAndPeakA ()
DECLARE SUB DrawSymol (YLP!, XLP!)
DECLARE SUB Legends (YLP!, XLP!)
DECLARE SUB LabelPeak ()
DECLARE SUB LabelPeakTT ()
DECLARE SUB LabelPeakCumul ()
DECLARE SUB AllSpecies ()
DECLARE SUB ParametersBOXp ()
DECLARE SUB PlotCreateReport ()
DECLARE SUB PlotTotal ()
DECLARE SUB PlotTotalForReport ()
DECLARE SUB AxesForReport ()
DECLARE SUB EraseScreen ()
DECLARE SUB FunckeyView ()
DECLARE SUB ViewPlot ()
DECLARE SUB ViewMenu (xcount!, owncolor!)
DECLARE SUB PlotConver ()
DECLARE SUB ParametersBOX1 ()
DECLARE SUB ParametersBOX2 ()
DECLARE SUB Position (row!, col!, Text$, Tcolor!)
DECLARE SUB PlotGraphI (Xs!, Zs!, n!, Line1!)
DECLARE SUB ReadTransferFile ()
DECLARE SUB Characteristic ()
DECLARE SUB PlotOverLay ()
DECLARE SUB CurveMenu (xcount!, owncolor!)
DECLARE SUB center (row!, Text$, Tcolor!)
DECLARE SUB Left (row!, Text$, Tcolor!)
DECLARE SUB Right (row!, Text$, Tcolor!)
DECLARE SUB PlotCumulative ()
DECLARE SUB LoadAllParameter ()
DECLARE SUB FunckeyMain ()
DECLARE SUB Axes ()
DECLARE SUB Para ()
DECLARE SUB Momt ()
DECLARE SUB PeakA ()
DECLARE SUB MaxW ()
DECLARE SUB Replot ()
DECLARE SUB DLine (Xbegin!, Ybegin!, Xend!, Yend!,
LineColor!)
DECLARE SUB CalMaxWn ()
DECLARE SUB TotalCurve ()
DECLARE SUB PlotWeightDistribute ()
DECLARE SUB PlotGraph (Xs!, Zs!, n!, Line1!)
COMMON SHARED MoName$, IoName$, SoName$, Mo,
Io, CTA, Temp, Time
COMMON SHARED number, h, Xscale, Yscale, FT
COMMON SHARED WMAX, MMAX
COMMON SHARED TotalCount
COMMON SHARED LGXM, CWnx, MCONV,
CTACONV, Xo, Yo, lx, ly, ReportResult$
LG2 = LOG(2) / LOG(10)
SCREEN 12
COLOR 15
LOCATE 15, 25
PRINT "Wait ! .. Now System working."
LoadAllParameter
LGXM = Xscale
FT = Time * 60
DIM SHARED NCL(1 TO number + 1)
DIM SHARED Wn(1 TO 10, 1 TO number + 1)
DIM SHARED PTime(0 TO INT(FT / h) + 1)
DIM SHARED Mt(0 TO INT(FT / h) + 1)
DIM SHARED CTAt(0 TO INT(FT / h) + 1)
DIM SHARED PeakArea(1 TO 9)
DIM SHARED Nmax(1 TO 10)
DIM SHARED Wnx(1 TO 10)
DIM SHARED Mnx(1 TO 10)
DIM SHARED Char(1 TO 10)
DIM SHARED MChar(1 TO 10)
DIM SHARED DPn(1 TO 10)
DIM SHARED DPw(1 TO 10)
DIM SHARED PDI(1 TO 10)
DIM SHARED Menu$(1 TO 6)
DIM SHARED View$(1 TO 7)
ReadTransferFile
Menu$(1) = "View Graph"
Menu$(2) = "Parameters"
Menu$(3) = "Distribute moments"
Menu$(4) = "Peak area"
Menu$(5) = "Maximum W(n)"
Menu$(6) = "Clear"
View$(1) = "Conversion"
View$(2) = "Total curve"
View$(3) = "All species"
View$(4) = "Overlay"
View$(5) = "Cumulative"
View$(6) = "Report"
View$(7) = "Clear"
CalMaxWn
Characteristic
CLS
SCREEN 12
  DO: Xo = 100. Yo = 120: lx = 440. ly = 240
    FunckeyMain
    count = 1
    DO
     CALL CurveMenu(count, 12)
    DO
     KBD$ = INKEY$
    LOOP UNTIL KBD$ <> ""
    SELECT CASE KBD$
        CASE CHR$(0) + "K"  'left
           count = count - 1
           IF count < 1 THEN count = 1
           CALL CurveMenu(count + 1, 15)
        CASE CHR$(0) + "M"  'right
           count = count + 1
           IF count > 10 THEN count = 10
           CALL CurveMenu(count - 1, 15)
        CASE CHR$(0) + "H"  'up
           count = count - 5
           IF count < 1 THEN count = 1
           CALL CurveMenu(count + 5, 15)
        CASE CHR$(0) + "P"  'down
           count = count + 5
           IF count > 6 THEN count = 6
           CALL CurveMenu(count - 5, 15)
    END SELECT
    inpcount = count
    LOOP UNTIL KBD$ = CHR$(13) OR KBD$ =
CHR$(27)
    IF KBD$ = CHR$(27) THEN END
    SELECT CASE inpcount
        CASE 1
           ViewPlot
        CASE 2
           Para
        CASE 3
           Momt
        CASE 4
           PeakA
        CASE 5
           MaxW
        CASE 6
           EraseScreen
    END SELECT
```

©Copyright by Mahidol University

```
  LOOP UNTIL KBD$ = CHR$(27)
END

SUB AllSpecies
    Axes
    CALL Left(7, "    Weight fraction", 13)
    LINE (Xo, Yo)-(Xo + lx, Yo + ly), 13, B
    CALL Legends(10, 49)
    Replot
    Axes
END SUB

SUB Axes
SCREEN 12
    CALL DLine(Xo - 4, Yo + ly, Xo + lx, Yo + ly, 14)
    CALL center(26, "LOG(Number of chain
length,NCL)", 14)
    COLOR 14
    Xint = (lx / 10)
    PRESET (Xo - Xint, Yo + ly + 4)
    FOR i = 0 TO 10 STEP 1
      PSET STEP(Xint, -4)
      LINE -STEP(0, 4)
    NEXT i
    COLOR 14
    Xint = (lx / 5)
    PRESET (Xo - Xint, Yo + ly + 6)
    FOR i = 0 TO 5 STEP 1
      PSET STEP(Xint, -6)
      LINE -STEP(0, 6)
    NEXT i
    FOR i = 0 TO 5
      COLOR 14
      LOCATE 24, 11 + 11 * i
      PRINT USING "## #"; (i / 5) * LGXM
    NEXT i
END SUB

SUB AxesForReport
    LINE (Xo, Yo)-(Xo, Yo + ly), 14
    LINE (Xo, Yo - ly)-(Xo + lx, Yo + ly), 14
    CALL center(26, "  log(Number of chain length)", 15)
    COLOR 14
    Xint = (lx / 5)
    PRESET (Xo - Xint, Yo + ly + 6)
    FOR i = 0 TO 5 STEP 1
      PSET STEP(Xint, -6)
      LINE -STEP(0, 6)
    NEXT i
    FOR i = 0 TO 5
      COLOR 14
      LOCATE 25, 11 + 12 * i
      PRINT USING "## #", (i / 5) * LGXM
    NEXT i
END SUB

SUB CalculateDPnAndPeakA STATIC
    REM Defination -          SUM{r=1->n}
[fraction(r) NCL(r)] --->uDegreePol
    REM          DPn(i),ave = ----------------------------
    REM                       SUM{r=1->n}  [fraction(r)]   -
-->dDegreePol
    REM
    REM Calculate Degree of polymerization, DPn(i)
      FOR i = 1 TO 9
        uD = 0  dD = 0
        FOR r = 1 TO number
          uDegreePol = (Wn(i, r) * NCL(r)) + uD
          uD = uDegreePol
          dDegreePol = Wn(i, r) + dD
          dD = dDegreePol
        NEXT r
        IF dDegreePol > 0 THEN DPn(i) = uDegreePol /
dDegreePol ELSE DPn(i) = 0
      NEXT i

    REM Defination -          SUM{r=1->n}
[fraction(r).NCL^2(r)]  --->uDegreePol
    REM          DPw(i),ave = ----------------------------
    REM                       SUM{r=1->n}
[fraction(r).NCL(r)]   --->dDegreePol
    REM
    REM Calculate Degree of polymerization, DPw(i)
      FOR i = 1 TO 9
        uD = 0: dD = 0
        FOR r = 1 TO number
          uDegreePol = (Wn(i, r) * (NCL(r) ^ 2)) + uD
          uD = uDegreePol
          dDegreePol = (Wn(i, r) * NCL(r)) + dD
          dD = dDegreePol
        NEXT r
        IF dDegreePol > 0 THEN DPw(i) = uDegreePol /
dDegreePol ELSE DPw(i) = 0
      NEXT i

    REM  Peak area(i) = SUM{r=1->n} fraction(r).NCL(r)
    REM
    REM  Calculate of peak area(i) of distribution curve
    REM  to use in calculation of DPn,ave(total)
      FOR i = 1 TO 9
        uD = 0  dD = 0
        FOR r = 1 TO number
          uDegreePol = (Wn(i, r) * (LOG(NCL(r)) /
LOG(10))) + uD
          uD = uDegreePol
        NEXT r
        PeakArea(i) = uDegreePol
      NEXT i

    REM Calculation of Average degree of
polymerisation,DPn,DPw
    REM Defination:- DPn,ave = SUM{i=1->9}
[DPn(i)*Peak area(i)]/SUM{i=1->9} [Peak area(i)]
    REM
      uD = 0: dD = 0: uW = 0: dW = 0
      FOR i = 1 TO 9
        uDPn10 = (PeakArea(i) * DPn(i)) + uD
        dDPn10 = PeakArea(i) + dD
        uD = uDPn10
        dD = dDPn10
        uDPw10 = (PeakArea(i) * DPw(i)) + uW
        dDPw10 = PeakArea(i) + dW
        uW = uDPw10
        dW = dDPw10
      NEXT i
      IF dDPn10 > 0 THEN DPn(10) = uDPn10 /
dDPn10
      IF dDPw10 > 0 THEN DPw(10) = uDPw10 /
dDPw10
    REM Calculation of Polydispersity Index (PDI)
    REM Defination - PDI(i) = DPw(i)/DPn(i)
    REM
      FOR i = 1 TO 10
        IF DPn(i) > 0 THEN PDI(i) = DPw(i) / DPn(i)
      NEXT i
END SUB

SUB CalMaxWn
'Routine for calculation of maximum value of weight
fraction
  FOR i = 1 TO 10
    Wnx(i) = Wn(i, 1)
    FOR n = 1 TO number
```

Copyright by Mahidol University

```
        IF Wnx(i) > Wn(i, n) THEN Wnx(ı) = Wnx(ı)
 ELSE Wnx(ı) = Wn(ı, n)
      NEXT n
    NEXT i
      WMAX = Wnx(10)
    FOR ı = 1 TO 10
      FOR n = 1 TO number
          IF Wn(i, n) = Wnx(ı) THEN Nmax(i) = NCL(n)
      NEXT n
    NEXT i
 END SUB

 SUB center (row, Text$, Tcolor)
    COLOR Tcolor
    LOCATE row, 41 - LEN(Text$) / 2
    PRINT Text$,
 END SUB

 SUB Characteristic
    'Calculation of Y-scale(characteristıs)
     FOR ı = 1 TO 10
      IF Wnx(ı) > 0 THEN
        Cha1 = LOG(Wnx(i)) / LOG(10)
        Cha2 = INT(LOG(Wnx(ı)) / LOG(10))
        Char(ı) = 10 ^ (Cha1 - Cha2)
      END IF
    NEXT i
 END SUB

 SUB CurveMenu (xcount, owncolor)
      IF xcount > 6 THEN xcount = 6
      IF xcount < 1 THEN xcount = 1
      COLOR owncolor
      SELECT CASE xcount
           CASE 1
               CALL Posıtıon(2, 4, Menu$(1), owncolor)
           CASE 2
               CALL Posıtıon(2, 16, Menu$(2), owncolor)
           CASE 3
               CALL Position(2, 28, Menu$(3), owncolor)
           CASE 4
               CALL Posıtıon(2, 48, Menu$(4), owncolor)
           CASE 5
               CALL Position(2, 59, Menu$(5), owncolor)
           CASE 6
               CALL Posıtıon(2, 73, Menu$(6), owncolor)
      END SELECT
 END SUB

 SUB DLine (Xbegin, Ybegin, Xend, Yend, LineColor)
      LINE (Xbegin, Ybegin)-(Xend, Yend), LineColor
 END SUB

 SUB DrawSymol (YLP, XLP)
      XLP = XLP - 2
      SLP = 15.5
      XPıxelMn = XLP * 8
      XPıxelMnX = XLP * 8
      XPıxelRMn = XLP * 8
      XPixelRMnX = XLP * 8
      XPixelRMnR = XLP * 8
      XPıxelCMn = XLP * 8
      XPıxelCMnX = XLP * 8
      XPixelCMnC = XLP * 8
      XPixelCMnR = XLP * 8
      YPixelMn = (YLP + 0) * SLP
      YPıxelMnX = (YLP + 1) * SLP
      YPixelRMn = (YLP + 2) * SLP
      YPıxelRMnX = (YLP + 3) * SLP
      YPixelRMnR = (YLP + 4) * SLP
      YPixelCMn = (YLP + 5) * SLP
      YPıxelCMnX = (YLP + 6) * SLP
      YPıxelCMnC = (YLP + 8) * SLP
      YPıxelCMnR = (YLP + 7) * SLP
      IF YPıxelMn < Yo + ly THEN
           CIRCLE (XPixelMn, YPıxelMn), 4, 9
           LINE (XPıxelMn, YPıxelMn - 3)-(XPıxelMn,
 YPixelMn + 3), 9
           LINE (XPixelMn - 3, YPıxelMn)-(XPixelMn + 3,
 YPıxelMn), 9
      END IF
      IF YPixelMnX < Yo + ly THEN
           CIRCLE (XPixelMnX, YPixelMnX), 4, 9
           LINE (XPixelMnX - 3, YPixelMnX - 3)-
(XPixelMnX + 3, YPixelMnX + 3), 9
           LINE (XPixelMnX - 3, YPixelMnX + 3)-
(XPıxelMnX + 3, YPixelMnX - 3), 9
      END IF
      IF YPıxelRMn < Yo + ly THEN
           LINE (XPixelRMn - 3, YPixelRMn - 3)-
(XPixelRMn + 3, YPixelRMn + 3), 2
           LINE (XPıxelRMn - 3, YPıxelRMn + 3)-
(XPixelRMn + 3, YPixelRMn - 3), 2
           LINE (XPixelRMn, YPixelRMn - 4)-(XPıxelRMn,
 YPıxelRMn + 4), 2
           LINE (XPixelRMn - 4, YPıxelRMn)-(XPıxelRMn
 + 4, YPixelRMn), 2
      END IF
      IF YPixelRMnX < Yo + ly THEN LINE
(XPixelRMnX - 3, YPixelRMnX - 3)-(XPixelRMnX + 3,
YPixelRMnX + 3), 3, B
      IF YPixelRMnR < Yo + ly THEN LINE
(XPixelRMnR - 3, YPixelRMnR - 3)-(XPixelRMnR + 3,
YPixelRMnR + 3), 10, BF
      IF YPixelCMn < Yo + ly THEN
           LINE (XPixelCMn - 3, YPıxelCMn - 3)-
(XPıxelCMn + 3, YPixelCMn + 3), 14
           LINE (XPıxelCMn - 3, YPixelCMn + 3)-
(XPixelCMn + 3, YPixelCMn - 3), 14
      END IF
      IF YPixelCMnX < Yo + ly THEN
           LINE (XPixelCMnX, YPixelCMnX - 4)-
(XPixelCMnX, YPixelCMnX + 4), 15
           LINE (XPıxelCMnX - 4, YPixelCMnX)-
(XPixelCMnX + 4, YPixelCMnX), 15
      END IF
      IF YPixelCMnC < Yo + ly THEN
           CIRCLE (XPixelCMnC, YPixelCMnC), 4, 7
           LINE (XPixelCMnC - 4, YPixelCMnC)-
(XPıxelCMnC + 4, YPixelCMnC), 7
      END IF
      IF YPixelCMnR < Yo + ly THEN
           LINE (XPixelCMnR - 3, YPixelCMnR - 3)-
(XPixelCMnR + 3, YPixelCMnR + 3), 13, B
           LINE (XPixelCMnR, YPixelCMnR - 5)-
(XPixelCMnR, YPixelCMnR + 5), 13
      END IF
 END SUB

 SUB EraseScreen
    FOR i = 1 TO 27 - 5
      LOCATE 5 + i, 1: COLOR 0: PRINT STRING$(80, "
 ")
    NEXT i
 END SUB

 SUB FunckeyMain STATIC
 CALL center(1, "----------")          CALL
 center(2, "----------")
 CALL center(3, "----------")          CALL
 center(4, "----------")
 CALL center(5, "----------")
 CALL Position(2, 4, Menu$(1), 15)
      CALL Position(2, 16, Menu$(2), 15)
```

Copyright by Mahidol University

```
        CALL Position(2, 28, Menu$(3), 15)
        CALL Position(2, 48, Menu$(4), 15)
        CALL Position(2, 59, Menu$(5), 15)
        CALL Position(2, 73, Menu$(6), 15)
        CALL Position(4, 67, "Abort=[ESC]", 9)
END SUB

SUB FunckeyView
CALL center(1, "----------")          CALL
center(2, "----------")
CALL center(3, "----------")          CALL
center(4, "----------")
CALL center(5, "----------")
        CALL Position(2, 4, View$(1), 15)
        CALL Position(2, 16, View$(2), 15)
        CALL Position(2, 29, View$(3), 15)
        CALL Position(2, 42, View$(4), 15)
        CALL Position(2, 51, View$(5), 15)
        CALL Position(2, 63, View$(6), 15)
        CALL Position(2, 71, View$(7), 15)
        CALL Position(4, 67, "Abort=[ESC]", 9)

END SUB

SUB LabelPeak
    IF Nmax(1) > 0 THEN lgNmaxMn = LOG(Nmax(1)) /
LOG(10)
    IF Nmax(2) > 0 THEN lgNmaxMnX =
LOG(Nmax(2)) / LOG(10)
    IF Nmax(3) > 0 THEN lgNmaxRMn =
LOG(Nmax(3)) / LOG(10)
    IF Nmax(4) > 0 THEN lgNmaxRMnX =
LOG(Nmax(4)) / LOG(10)
    IF Nmax(5) > 0 THEN lgNmaxRMnR =
LOG(Nmax(5)) / LOG(10)
    IF Nmax(6) > 0 THEN lgNmaxCMn =
LOG(Nmax(6)) / LOG(10)
    IF Nmax(7) > 0 THEN lgNmaxCMnX =
LOG(Nmax(7)) / LOG(10)
    IF Nmax(8) > 0 THEN lgNmaxCMnC =
LOG(Nmax(8)) / LOG(10)
    IF Nmax(9) > 0 THEN lgNmaxCMnR =
LOG(Nmax(9)) / LOG(10)
    XPixelMn = Xo + (lx * lgNmaxMn / LGXM)
    XPixelMnX = Xo + (lx * lgNmaxMnX / LGXM)
    XPixelRMn = Xo + (lx * lgNmaxRMn / LGXM)
    XPixelRMnX = Xo + (lx * lgNmaxRMnX / LGXM)
    XPixelRMnR = Xo + (lx * lgNmaxRMnR / LGXM)
    XPixelCMn = Xo + (lx * lgNmaxCMn / LGXM)
    XPixelCMnX = Xo + (lx * lgNmaxCMnX / LGXM)
    XPixelCMnC = Xo + (lx * lgNmaxCMnC / LGXM)
    XPixelCMnR = Xo + (lx * lgNmaxCMnR / LGXM)
    YPixelMn = Yo + ly - (ly * Char(1) / Yscale)
    YPixelMnX = Yo + ly - (ly * Char(2) / Yscale)
    YPixelRMn = Yo + ly - (ly * Char(3) / Yscale)
    YPixelRMnX = Yo + ly - (ly * Char(4) / Yscale)
    YPixelRMnR = Yo + ly - (ly * Char(5) / Yscale)
    YPixelCMn = Yo + ly - (ly * Char(6) / Yscale)
    YPixelCMnX = Yo + ly - (ly * Char(7) / Yscale)
    YPixelCMnC = Yo + ly - (ly * Char(8) / Yscale)
    YPixelCMnR = Yo + ly - (ly * Char(9) / Yscale)
    IF YPixelMn < Yo + ly THEN
        CIRCLE (XPixelMn, YPixelMn), 4, 9
        LINE (XPixelMn, YPixelMn - 3)-(XPixelMn,
YPixelMn + 3), 9
        LINE (XPixelMn - 3, YPixelMn)-(XPixelMn + 3,
YPixelMn), 9
    END IF
    IF YPixelMnX < Yo + ly THEN
        CIRCLE (XPixelMnX, YPixelMnX), 4, 9
        LINE (XPixelMnX - 3, YPixelMnX - 3)-
(XPixelMnX + 3, YPixelMnX + 3), 9
        LINE (XPixelMnX - 3, YPixelMnX + 3)-
(XPixelMnX + 3, YPixelMnX - 3), 9
    END IF
    IF YPixelRMn < Yo + ly THEN
        LINE (XPixelRMn - 3, YPixelRMn - 3)-
(XPixelRMn + 3, YPixelRMn + 3), 2
        LINE (XPixelRMn - 3, YPixelRMn + 3)-
(XPixelRMn + 3, YPixelRMn - 3), 2
        LINE (XPixelRMn, YPixelRMn - 4)-(XPixelRMn,
YPixelRMn + 4), 2
        LINE (XPixelRMn - 4, YPixelRMn)-(XPixelRMn
+ 4, YPixelRMn), 2
    END IF
    IF YPixelRMnX < Yo + ly THEN LINE
(XPixelRMnX - 3, YPixelRMnX - 3)-(XPixelRMnX + 3,
YPixelRMnX + 3), 3, B
    IF YPixelRMnR < Yo + ly THEN LINE
(XPixelRMnR - 3, YPixelRMnR - 3)-(XPixelRMnR + 3,
YPixelRMnR + 3), 10, BF
    IF YPixelCMn < Yo + ly THEN
        LINE (XPixelCMn - 3, YPixelCMn - 3)-
(XPixelCMn + 3, YPixelCMn + 3), 14
        LINE (XPixelCMn - 3, YPixelCMn + 3)-
(XPixelCMn + 3, YPixelCMn - 3), 14
    END IF
    IF YPixelCMnX < Yo + ly THEN
        LINE (XPixelCMnX, YPixelCMnX - 4)-
(XPixelCMnX, YPixelCMnX + 4), 15
        LINE (XPixelCMnX - 4, YPixelCMnX)-
(XPixelCMnX + 4, YPixelCMnX), 15
    END IF
    IF YPixelCMnC < Yo + ly THEN
        CIRCLE (XPixelCMnC, YPixelCMnC), 4, 7
        LINE (XPixelCMnC - 4, YPixelCMnC)-
(XPixelCMnC + 4, YPixelCMnC), 7
    END IF
    IF YPixelCMnR < Yo + ly THEN
        LINE (XPixelCMnR - 3, YPixelCMnR - 3)-
(XPixelCMnR + 3, YPixelCMnR + 3), 13, B
        LINE (XPixelCMnR, YPixelCMnR - 5)-
(XPixelCMnR, YPixelCMnR + 5), 13
    END IF
END SUB

SUB LabelPeakCumul
    CalculateDPnAndPeakA
    CalMaxWn
    Chi = Wn(10, 1)
    FOR i = 2 TO number
        CWnx1 = Wn(10, i) + Chi
        Chi = CWnx1
    NEXT i
    CWnx = Chi
    Chi = 0
    FOR n = 1 TO number
        CWn = Wn(10, n) + Chi
        IF NCL(n) >= Nmax(10) THEN
            xxCumul = CWn
            EXIT FOR
        END IF
        Chi = CWn
    NEXT n
    lgNmax10 = LOG(Nmax(10)) / LOG(10)
    XP10 = Xo + (lx * lgNmax10 / LGXM)
    YP10C = (Yo + ly) - (ly * xxCumul / CWnx)
    LINE (XP10, YP10C)-(XP10 + 45, YP10C), 13
    LINE (XP10 + 45, YP10C)-(XP10 + 45 - 4, YP10C -
4), 13
    LINE (XP10 + 45, YP10C)-(XP10 + 45 - 4, YP10C +
4), 13
END SUB
```

Copyright by Mahidol University

```
SUB LabelPeakForReport
    CalculateDPnAndPeakA
    CalMaxWn
    Chi = Wn(10, 1)
    FOR i = 2 TO number
        CWnx1 = Wn(10, i) + Chi
        Chi = CWnx1
    NEXT i
    CWnx = Chi
    Chi = 0
    FOR n = 1 TO number
        CWn = Wn(10, n) + Chi
        IF NCL(n) >= Nmax(10) THEN
            xxCumul = CWn
            EXIT FOR
        END IF
        Chi = CWn
    NEXT n

    lgNmax10 = LOG(Nmax(10)) / LOG(10)
    XP10 = Xo + (lx * lgNmax10 / LGXM)
    YP10P = (Yo + ly) - (ly * Wnx(10) / WMAX)
    YP10C = (Yo + ly) - (ly * xxCumul / CWnx)
    LINE (XP10, YP10C)-(XP10 - 25, YP10C), 13
    LINE (XP10 - 25, YP10C)-(XP10 - 25 + 4, YP10C -
4), 13
    LINE (XP10, YP10P)-(XP10 + 25, YP10P), 15
    LINE (XP10 + 25, YP10P)-(XP10 + 25 - 4, YP10P -
4), 15
END SUB

SUB LabelPeakTT
    CalculateDPnAndPeakA
    CalMaxWn
    Chi = Wn(10, 1)
    FOR i = 2 TO number
        CWnx1 = Wn(10, i) + Chi
        Chi = CWnx1
    NEXT i
    CWnx = Chi
    Chi = 0
    FOR n = 1 TO number
        CWn = Wn(10, n) + Chi
        IF NCL(n) >= Nmax(10) THEN
            xxCumul = CWn
            EXIT FOR
        END IF
        Chi = CWn
    NEXT n
    lgNmax10 = LOG(Nmax(10)) / LOG(10)
    XP10 = Xo + (lx * (lgNmax10) / LGXM)
    YP10P = (Yo + ly) - (ly * Wnx(10) / WMAX)
    LINE (XP10, YP10P)-(XP10 - 35, YP10P), 15
    LINE (XP10 - 35, YP10P)-(XP10 - 35 + 4, YP10P -
4), 15
    LINE (XP10 - 35, YP10P)-(XP10 - 35 + 4, YP10P +
4), 15

END SUB

SUB LabelScale (WNaxis, WnALLaxis) STATIC
IF WNaxis = 1 THEN
REM Y-axis
    CALL Left(7, "Characteristic of Wn ", 13)
    CALL DLine(Xo, Yo, Xo, Yo + ly, 13)
    COLOR 13
    Yint = (ly / 10)
    PRESET (Xo, Yo + ly)
    FOR i = 1 TO 10 STEP 1
      PSET STEP(-4, -Yint)
      LINE -STEP(4, 0)
    NEXT i
    Yint = (ly / 5)
    PRESET (Xo, Yo + ly + Yint)
    FOR i = 0 TO 5 STEP 1
      PSET STEP(-8, -Yint)
      LINE -STEP(8, 0)
    NEXT i
    COLOR 13
    FOR i = 0 TO 5 STEP 1
        LOCATE 8 + 3 * i, 6
        PRINT USING "###.##"; (5 - i) * Yscale / 5
    NEXT i
END IF
IF WnALLaxis = 1 THEN
REM Y2-axis
    COLOR 15
    FOR i = 0 TO 5 STEP 1
        LOCATE 8 + 3 * i, 69
        PRINT USING "##.##^^^^"; (5 - i) * WMAX / 5
    NEXT i
    CALL Right(6, "Max.Weight Fraction", 15)
    CALL DLine(Xo + lx, Yo, Xo + lx, Yo + ly, 15)
    COLOR 15
    Yint = (ly / 10)
    PRESET (Xo + lx + 4, Yo + ly)
    FOR i = 1 TO 10 STEP 1
      PSET STEP(-4, -Yint)
      LINE -STEP(4, 0)
    NEXT i
    Yint = (ly / 5)
    PRESET (Xo + lx + 8, Yo + ly + Yint)
    FOR i = 0 TO 5 STEP 1
      PSET STEP(-8, -Yint)
      LINE -STEP(8, 0)
    NEXT i
END IF
END SUB

SUB Left (row, Text$, Tcolor)
  COLOR Tcolor
  LOCATE row, 2
  PRINT Text$;
END SUB

SUB Legends (YLP, XLP)
    COLOR 15
    LOCATE YLP - 1, XLP - 2. PRINT "SPECIES
Wn,MAX    Mp(GPC)"
    COLOR 10
    LOCATE YLP, XLP:    PRINT USING "Mn
##.##^^^^  ##.##^^^^"; Wnx(1); Nmax(1)
    LOCATE YLP + 1, XLP: PRINT USING "MnX
## ##^^^^  ##.##^^^^", Wnx(2); Nmax(2)
    LOCATE YLP + 2, XLP: PRINT USING "RMn
##.##^^^^  ##.##^^^^"; Wnx(3); Nmax(3)
    LOCATE YLP + 3, XLP: PRINT USING "RMnX
##.##^^^^  ##.##^^^^"; Wnx(4); Nmax(4)
    LOCATE YLP + 4, XLP: PRINT USING "RMnR
##.##^^^^  ## ##^^^^", Wnx(5), Nmax(5)
    LOCATE YLP + 5, XLP. PRINT USING "CTAMn
##.##^^^^  ##.##^^^^"; Wnx(6); Nmax(6)
    LOCATE YLP + 6, XLP· PRINT USING "CTAMnX
##.##^^^^  ## ##^^^^"; Wnx(7); Nmax(7)
    LOCATE YLP + 7, XLP: PRINT USING "CTAMnR
## ##^^^^  ##.##^^^^", Wnx(9); Nmax(9)
    LOCATE YLP + 8, XLP: PRINT USING
"CTAMnCTA ##.##^^^^  ## ##^^^^"; Wnx(8); Nmax(8)
    COLOR 15
    LOCATE YLP + 9, XLP: PRINT USING "TOTAL
##.##^^^^  ##.##^^^^"; Wnx(10), Nmax(10)
    CALL DrawSymol(YLP, XLP)
END SUB
```

Copyright by Mahidol University

```
SUB LoadAllParameter
    OPEN "CONST MTX" FOR INPUT AS #1
      INPUT #1, MoName$
      INPUT #1, IoName$
      INPUT #1, SoName$
      INPUT #1, Mo
      INPUT #1, Io
      INPUT #1, CTA
      INPUT #1, Temp
      INPUT #1, Time
      INPUT #1, Kp
      INPUT #1, Ktc
      INPUT #1, Ktd
      INPUT #1, Ktrm
      INPUT #1, Ki
      INPUT #1, Kd
      INPUT #1, f
      INPUT #1, Ktci
      INPUT #1, Ksr
      INPUT #1, Ktr
      INPUT #1, number
      INPUT #1, h
      INPUT #1, Xscale
      INPUT #1, Yscale
      INPUT #1, GelEffect$
      INPUT #1, XoCOOR
      INPUT #1, YoCOOR
      INPUT #1, XaxisLength
      INPUT #1, YaxisLength
      INPUT #1, ZaxisLength
      INPUT #1, LeftAngle
      INPUT #1, RightAngle
      INPUT #1, NCLstep
      INPUT #1, THrstep
    CLOSE #1
END SUB

SUB MaxW
    YLP = 10
    XLP = 47
    COLOR 15
    LOCATE YLP - 1, XLP      PRINT "SPECIES
Wn,MAX    Mp(GPC)"
    COLOR 10
    LOCATE YLP, XLP:    PRINT USING "  Mn
## ##^^^^  ##.##^^^^"; Wnx(1); Nmax(1)
    LOCATE YLP + 1, XLP: PRINT USING "  MnX
## ##^^^^  ##.##^^^^", Wnx(2); Nmax(2)
    LOCATE YLP + 2, XLP: PRINT USING "  RMn
##.##^^^^  ##.##^^^^"; Wnx(3); Nmax(3)
    LOCATE YLP + 3, XLP: PRINT USING "  RMnX
##.##^^^^  ##.##^^^^"; Wnx(4); Nmax(4)
    LOCATE YLP + 4, XLP: PRINT USING "  RMnR
##.##^^^^  ##.##^^^^", Wnx(5); Nmax(5)
    LOCATE YLP + 5, XLP: PRINT USING "  CTAMn
## ##^^^^  ##.##^^^^"; Wnx(6); Nmax(6)
    LOCATE YLP + 6, XLP: PRINT USING "
CTAMnX  ## ##^^^^  ## ##^^^^", Wnx(7), Nmax(7)
    LOCATE YLP + 7, XLP: PRINT USING "
CTAMnR  ##.##^^^^  ##.##^^^^"; Wnx(9); Nmax(9)
    LOCATE YLP + 8, XLP  PRINT USING "
CTAMnCTA ##.##^^^^  ##.##^^^^"; Wnx(8); Nmax(8)
    COLOR 15
    LOCATE YLP + 9, XLP: PRINT USING "  TOTAL
## ##^^^^  ##.##^^^^", Wnx(10); Nmax(10)
    CALL DrawSymol(YLP, XLP + 2)
END SUB

SUB Momt STATIC
    CalculateDPnAndPeakA
    ParametersBOX2
    i = 80 - 48
    j = 7
    COLOR 9
    LOCATE j + 5, i + 13: PRINT USING "##.##^^^^";
DPn(1)
    LOCATE j + 5, i + 13 * 2  PRINT USING
"##.##^^^^"; DPw(1)
    LOCATE j + 5, i + 13 * 3: PRINT USING "##.####";
PDI(1)
    COLOR 1
    LOCATE j + 6, i + 13: PRINT USING "##.##^^^^";
DPn(2)
    LOCATE j + 6, i + 13 * 2: PRINT USING
"##.##^^^^"; DPw(2)
    LOCATE j + 6, i + 13 * 3: PRINT USING "##.####";
PDI(2)
    COLOR 2
    LOCATE j + 7, i + 13. PRINT USING "## ##^^^^";
DPn(3)
    LOCATE j + 7, i + 13 * 2: PRINT USING
"## ##^^^^"; DPw(3)
    LOCATE j + 7, i + 13 * 3: PRINT USING "##.####";
PDI(3)
    COLOR 3
    LOCATE j + 8, i + 13: PRINT USING "##.##^^^^";
DPn(4)
    LOCATE j + 8, i + 13 * 2. PRINT USING
"##.##^^^^"; DPw(4)
    LOCATE j + 8, i + 13 * 3: PRINT USING "##.####";
PDI(4)
    COLOR 10
    LOCATE j + 9, i + 13: PRINT USING "##.##^^^^";
DPn(5)
    LOCATE j + 9, i + 13 * 2: PRINT USING
"##.##^^^^"; DPw(5)
    LOCATE j + 9, i + 13 * 3: PRINT USING "##.####",
PDI(5)
    COLOR 14
    LOCATE j + 10, i + 13: PRINT USING "##.##^^^^",
DPn(6)
    LOCATE j + 10, i + 13 * 2: PRINT USING
"##.##^^^^"; DPw(6)
    LOCATE j + 10, i + 13 * 3  PRINT USING
"## ####"; PDI(6)
    COLOR 12
    LOCATE j + 11, i + 13: PRINT USING "##.##^^^^";
DPn(7)
    LOCATE j + 11, i + 13 * 2. PRINT USING
"##.##^^^^", DPw(7)
    LOCATE j + 11, i + 13 * 3: PRINT USING
"##.####"; PDI(7)
    COLOR 7
    LOCATE j + 12, i + 13: PRINT USING "##.##^^^^";
DPn(8)
    LOCATE j + 12, i + 13 * 2: PRINT USING
"##.##^^^^", DPw(8)
    LOCATE j + 12, i + 13 * 3: PRINT USING
"## ####"; PDI(8)
    COLOR 13
    LOCATE j + 13, i + 13. PRINT USING "## ##^^^^";
DPn(9)
    LOCATE j + 13, i + 13 * 2: PRINT USING
"##.##^^^^"; DPw(9)
    LOCATE j + 13, i + 13 * 3: PRINT USING
"##.####"; PDI(9)
    COLOR 15
    LOCATE j + 15, i + 13: PRINT USING "##.##^^^^";
DPn(10)
    LOCATE j + 15, i + 13 * 2: PRINT USING
"##.##^^^^"; DPw(10)
    LOCATE j + 15, i + 13 * 3: PRINT USING
"##.####"; PDI(10)
END SUB
```

Copyright by Mahidol University

```
SUB Para STATIC
    ParametersBOX1
    i = 80 - 42
    j = 7
    COLOR 14
    LOCATE j + 5, i + 20: PRINT USING "### ##";
Temp
    LOCATE j + 6, i + 20  PRINT USING "####.#"; Time
    LOCATE j + 7, i + 20: PRINT USING "## #####";
Mo
    LOCATE j + 8, i + 20  PRINT USING "##.#####", Io
    LOCATE j + 9, i + 20: PRINT USING "##.#####";
CTA
    LOCATE j + 14, i + 20: PRINT USING "###.##"; 100
* (Mo - Mt(INT(FT / h) + 1)) / Mo
    IF CTA > 0 THEN
    LOCATE j + 15, i + 20  PRINT USING "### ##"; 100
* (CTA - CTAt(INT(FT / h) + 1)) / CTA
    END IF
END SUB

SUB ParametersBOX1
    x = 11
    i = 80 - 42
    j = 7

    CALL Position(j, i,----------",x)
CALL Position(j + 1, i, "³  Parameter and condition
reaction  ³", x)
    CALL Position(j + 2, i, ----------",x)
CALL Position(j + 3, i, "³ Parameter ³      Value    ³ Unit
³", x)
    CALL Position(j + 4, i, ----------",x)
CALL Position(j + 5, i, "³ Temp.  ³              ³ øC ³", x)
    CALL Position(j + 6, i, "³ Time   ³              ³ min.
³", x)
    CALL Position(j + 7, i, "³        ³              ³ mol/l ³",
x)
    CALL Position(j + 8, i, "³        ³              ³ mol/l ³",
x)
    CALL Position(j + 9, i, "³        ³              ³ mol/l ³",
x)
    CALL Position(j + 10, i, ----------,x)
    CALL Position(j + 11, i, ----------,x)
    CALL Position(j + 12, i, "³ Conversion³      Value    ³
Unit ³", x)
    CALL Position(j + 13, i, ----------",x)
CALL Position(j + 14, i, "³ Monomer ³              ³ %
³", x)
    CALL Position(j + 15, i, "³ CTA   ³              ³ %
³", x)
    CALL Position(j + 16, i, ----------",x)
A$ = "[" + UCASE$(MoName$) + "]"
    B$ = "[" + UCASE$(IoName$) + "]"
    C$ = "[" + UCASE$(SoName$) + "]"
    CALL Position(j + 7, i + 3, A$, 9)
    CALL Position(j + 8, i + 3, B$, 1)
    CALL Position(j + 9, i + 3, C$, 2)
END SUB

SUB ParametersBOX2
    x = 11
    i = 80 - 48
    j = 7
    CALL Position(j, i,----------",x)
CALL Position(j + 1, i, "³  Overall Molecular Weight
Distributions  ³", x)
    CALL Position(j + 2, i,----------",x)
CALL Position(j + 3, i, "³ Species ³  DPn  ³  DPw  ³
PD-Index ³", x)
    CALL Position(j + 4, i,----------",x)
CALL Position(j + 5, i, "³        ³        ³        ³
³", x)
    CALL Position(j + 6, i, "³        ³        ³        ³
³", x)
    CALL Position(j + 7, i, "³        ³        ³        ³
³", x)
    CALL Position(j + 8, i, "³        ³        ³        ³
³", x)
    CALL Position(j + 9, i, "³        ³        ³        ³
³", x)
    CALL Position(j + 10, i, "³       ³        ³        ³
³", x)
    CALL Position(j + 11, i, "³       ³        ³        ³
³", x)
    CALL Position(j + 12, i, "³       ³        ³        ³
³", x)
    CALL Position(j + 13, i, "³       ³        ³        ³
³", x)
    CALL Position(j + 14,i,----------",x)
CALL Position(j + 15, i, "³       ³        ³        ³
³", x)
    CALL Position(j + 16, i,----------",x)
CALL Position(j + 5, i + 3, "1 Mn", 9)
    CALL Position(j + 6, i + 3, "2 MnX", 1)
    CALL Position(j + 7, i + 3, "3 RMn", 2)
    CALL Position(j + 8, i + 3, "4 RMnX", 3)
    CALL Position(j + 9, i + 3, "5 RMnR", 10)
    CALL Position(j + 10, i + 3, "6 CMn", 14)
    CALL Position(j + 11, i + 3, "7 CMnX", 12)
    CALL Position(j + 12, i + 3, "8 CMnC", 7)
    CALL Position(j + 13, i + 3, "9 CMnR", 13)
    CALL Position(j + 15, i + 3, "TOTAL", 15)
END SUB

SUB ParametersBOXp
    CalculateDPnAndPeakA
    CalMaxWn
    TotalPeakArea = PeakArea(1) + PeakArea(2) +
PeakArea(3) + PeakArea(4) + PeakArea(5) + PeakArea(6)
+ PeakArea(7) + PeakArea(8) + PeakArea(9)
    assay1 = (PeakArea(1) / TotalPeakArea) * 100
    assay2 = (PeakArea(2) / TotalPeakArea) * 100
    assay3 = (PeakArea(3) / TotalPeakArea) * 100
    assay4 = (PeakArea(4) / TotalPeakArea) * 100
    assay5 = (PeakArea(5) / TotalPeakArea) * 100
    assay6 = (PeakArea(6) / TotalPeakArea) * 100
    assay7 = (PeakArea(7) / TotalPeakArea) * 100
    assay8 = (PeakArea(8) / TotalPeakArea) * 100
    assay9 = (PeakArea(9) / TotalPeakArea) * 100
    assay10 = 100
    IF DPn(1) > 0 THEN lgDPn1 = LOG(DPn(1)) /
LOG(10)
    IF DPn(2) > 0 THEN lgDPn2 = LOG(DPn(2)) /
LOG(10)
    IF DPn(3) > 0 THEN lgDPn3 = LOG(DPn(3)) /
LOG(10)
    IF DPn(4) > 0 THEN lgDPn4 = LOG(DPn(4)) /
LOG(10)
    IF DPn(5) > 0 THEN lgDPn5 = LOG(DPn(5)) /
LOG(10)
    IF DPn(6) > 0 THEN lgDPn6 = LOG(DPn(6)) /
LOG(10)
    IF DPn(7) > 0 THEN lgDPn7 = LOG(DPn(7)) /
LOG(10)
    IF DPn(8) > 0 THEN lgDPn8 = LOG(DPn(8)) /
LOG(10)
    IF DPn(9) > 0 THEN lgDPn9 = LOG(DPn(9)) /
LOG(10)
    IF DPn(10) > 0 THEN lgDPn10 = LOG(DPn(10)) /
LOG(10)
    i = 50
    j = 6
    COLOR 15
```

Copyright by Mahidol University

```
      LOCATE j + 3, 1.      PRINT "SPECIES    ASSAY(%)
 LOG DPn"
      COLOR 10
      LOCATE j + 4, i: PRINT USING "  Mn      ###.###
 ##.## "; assay1, lgDPn1
      LOCATE j + 5, 1. PRINT USING "  MnX     ###.###
 ## ## "; assay2; lgDPn2
      LOCATE j + 6, i. PRINT USING "  RMn     ### ###
 ##.## "; assay3; lgDPn3
      LOCATE j + 7, i· PRINT USING "  RMnX    ###.###
 ## ## "; assay4; lgDPn4
      LOCATE j + 8, i. PRINT USING "  RMnR    ###.###
 ##.## "; assay5; lgDPn5
      LOCATE j + 9, i: PRINT USING "  CTAMn
 ###.###   ##.## "; assay6; lgDPn6
      LOCATE j + 10, i: PRINT USING "  CTAMnX
 ### ###   ##.## ", assay7; lgDPn7
      LOCATE j + 11, i: PRINT USING "  CTAMnR
 ###.###   ##.## ", assay9; lgDPn9
      LOCATE J + 12, 1. PRINT USING "  CTAMnCTA
 ### ###   ##.## ", assay8; lgDPn8
      COLOR 15
      LOCATE J + 13, 1· PRINT USING "  TOTAL
 ###.###   ##.## "; assay10; lgDPn10
 END SUB

 SUB PeakA STATIC
      CalculateDPnAndPeakA
      YLP = 11
      XLP = 47
      COLOR 15
      LOCATE YLP - 2, XLP:    PRINT "Calculated Peak
 Area "
      COLOR 12
      LOCATE YLP - 1, XLP·      PRINT "Species    Peak
 area "
      COLOR 10
      LOCATE YLP, XLP:    PRINT USING "  Mn
 ##.##^^^^"; PeakArea(1)
      LOCATE YLP + 1, XLP: PRINT USING "  MnX
 ##.##^^^^"; PeakArea(2)
      LOCATE YLP + 2, XLP  PRINT USING "  RMn
 ##.##^^^^"; PeakArea(3)
      LOCATE YLP + 3, XLP· PRINT USING "  RMnX
 ##.##^^^^"; PeakArea(4)
      LOCATE YLP + 4, XLP: PRINT USING "  RMnR
 ## ##^^^^"; PeakArea(5)
      LOCATE YLP + 5, XLP  PRINT USING "  CTAMn
 ##.##^^^^"; PeakArea(6)
      LOCATE YLP + 6, XLP: PRINT USING "
 CTAMnX   ##.##^^^^"; PeakArea(7)
      LOCATE YLP + 7, XLP: PRINT USING "
 CTAMnR   ## ##^^^^"; PeakArea(9)
      LOCATE YLP + 8, XLP: PRINT USING "
 CTAMnCTA ##.##^^^^", PeakArea(8)
      CALL DrawSymol(YLP, XLP + 2)
 END SUB

 SUB PercentConversion
   Mx = Mt(0)
   CTAx = CTAt(0)
   FOR n = 0 TO INT(FT / h) + 1
      IF Mx < Mt(n) THEN Mx = Mt(n) ELSE Mx = Mx
      IF CTAx < CTAt(n) THEN CTAx = CTAt(n) ELSE
 CTAx = CTAx
   NEXT n
   MCONV = 100 * (Mo - Mx) / Mo
   IF CTA > 0 THEN
   CTACONV = 100 * (CTA - CTAx) / CTA
   END IF
 END SUB

 SUB PlotConver
 MCONV = 100 * (Mo - Mt(INT(FT / h) + 1)) / Mo
 IF CTA > 0 THEN
 CTACONV = 100 * (CTA - CTAt(INT(FT / h) + 1)) / CTA
 END IF
 REM X-axis
      CALL DLine(Xo - 4, Yo + ly, Xo + lx, Yo + ly, 14)
      CALL center(26, "Reaction Time(min)", 14)
      COLOR 14
      Xint = (lx / 10)
      PRESET (Xo - Xint, Yo + ly + 4)
      FOR i = 0 TO 10 STEP 1
        PSET STEP(Xint, -4)
        LINE -STEP(0, 4)
      NEXT i
      COLOR 14
      Xint = (lx / 5)
      PRESET (Xo - Xint, Yo + ly + 6)
      FOR 1 = 0 TO 5 STEP 1
        PSET STEP(Xint, -6)
        LINE -STEP(0, 6)
      NEXT 1
      FOR 1 = 0 TO 5
        COLOR 14
        LOCATE 24, 9 + 11 * i
        PRINT USING "####.#"; (i / 5) * Time
      NEXT 1
 REM Y-axis(Monomer)
      CALL Left(7, "Monomer Conversion(%)", 13)
      CALL DLine(Xo, Yo, Xo, Yo + ly, 13)
      COLOR 13
      Yint = (ly / 10)
      PRESET (Xo, Yo + ly)
      FOR i = 1 TO 10 STEP 1
        PSET STEP(-4, -Yint)
        LINE -STEP(4, 0)
      NEXT 1
      Yint = (ly / 5)
      PRESET (Xo, Yo + ly + Yint)
      FOR i = 0 TO 5 STEP 1
        PSET STEP(-8, -Yint)
        LINE -STEP(8, 0)
      NEXT i
      COLOR 13
      FOR i = 0 TO 5 STEP 1
          LOCATE 8 + 3 * 1, 6
          PRINT USING "###.##"; (5 - 1) * MCONV / 5
      NEXT 1
 IF CTA > 0 THEN
 REM Y-axis(CTA)
      COLOR 15
      FOR i = 0 TO 5 STEP 1
          LOCATE 8 + 3 * 1, 69
          PRINT USING "###.##"; (5 - 1) * CTACONV / 5
      NEXT i
      CALL Right(7, "CTA Conversion(%)", 15)
      CALL DLine(Xo + lx, Yo, Xo + lx, Yo + ly, 15)
      COLOR 15
      Yint = (ly / 10)
      PRESET (Xo + lx + 4, Yo + ly)
      FOR i = 1 TO 10 STEP 1
        PSET STEP(-4, -Yint)
        LINE -STEP(4, 0)
      NEXT i
      Yint = (ly / 5)
      PRESET (Xo + lx + 8, Yo + ly + Yint)
      FOR i = 0 TO 5 STEP 1
        PSET STEP(-8, -Yint)
        LINE -STEP(8, 0)
      NEXT i
 END IF
 REM Plot
```

Copyright by Mahidol University

```
    FOR n = 1 TO INT(FT / h) + 1
     LinTime = PTime(n)
     MCONVt = 100 * (Mo - Mt(n)) / Mo
     IF CTA > 0 THEN
     CCONVt = 100 * (CTA - CTAt(n)) / CTA
     END IF
     Xs = Xo + (lx * LinTime / FT)
     Y1 = Yo + ly - (ly * MCONVt / MCONV)
     IF CTA > 0 THEN
     Y2 = Yo + ly - (ly * CCONVt / CTACONV)
     END IF
     CALL PlotGraph(Xs, Y1, n, 13)
     IF CTA > 0 THEN
     CALL PlotGraph1(Xs, Y2, n, 15)
     END IF
    NEXT n
REM re-X-axis
CALL DLine(Xo - 4, Yo + ly, Xo + lx, Yo + ly, 14)
END SUB

SUB PlotCreateReport
     Xo = 100  Yo = 120: lx = 480: ly = 255
     ParametersBOXp
     AxesForReport
     PlotTotalForReport
     LabelPeakTT
     IF ReportResult$ = "Y" THEN
          PlotWeightDistribute
          LabelPeak
          CALL DrawSymol(10, 52)
     END IF
     AxesForReport
END SUB

SUB PlotCumulative STATIC
REM Y1-axis7
     CALL Right(7, "Cumulative Percentage", 13)
     CALL DLine(Xo + lx, Yo, Xo + lx, Yo + ly, 13)
     COLOR 13
     Yint = (ly / 10)
     PRESET (Xo + lx + 4, Yo + ly)
     FOR i = 1 TO 10 STEP 1
       PSET STEP(-4, -Yint)
       LINE -STEP(4, 0)
     NEXT i
     Yint = (ly / 5)
     PRESET (Xo + lx + 8, Yo + ly + Yint)
     FOR i = 0 TO 5 STEP 1
       PSET STEP(-8, -Yint)
       LINE -STEP(8, 0)
     NEXT i
     COLOR 13
     FOR i = 0 TO 5 STEP 1
          LOCATE 8 + 3 * i, 70
          PRINT USING "### ##", (5 - i) * 100 / 5
     NEXT i
     Chi = Wn(10, 1)
     FOR i = 2 TO number
          CWnx1 = Wn(10, i) + Chi
          Chi = CWnx1
     NEXT i
     CWnx = Chi
     Chi = 0
     FOR n = 1 TO number
          lgN = LOG(NCL(n)) / LOG(10)
          CWn = Wn(10, n) + Chi
          Xs = Xo + (lx * lgN / LGXM)
          Ys = Yo + ly - (ly * CWn / CWnx)
          CALL PlotGraph(Xs, Ys, n, 13)
          Chi = CWn
     NEXT n
     LabelPeakCumul
     Axes
END SUB

SUB PlotCumulativeForReport STATIC
REM Y1-axis
     CALL Left(7, "%Cumulative", 13)
     CALL DLine(Xo, Yo, Xo, Yo + ly, 13)
     COLOR 13
     Yint = (ly / 10)
     PRESET (Xo, Yo + ly)
     FOR i = 1 TO 10 STEP 1
       PSET STEP(-4, -Yint)
       LINE -STEP(4, 0)
     NEXT i
     Yint = (ly / 5)
     PRESET (Xo, Yo + ly + Yint)
     FOR i = 0 TO 5 STEP 1
       PSET STEP(-8, -Yint)
       LINE -STEP(8, 0)
     NEXT i
     COLOR 13
     FOR i = 0 TO 5 STEP 1
          LOCATE 8 + 3 * i, 2
          PRINT USING "###.##"; (5 - i) * 100 / 5
     NEXT i
     Chi = Wn(10, 1)
     FOR i = 2 TO number
          CWnx1 = Wn(10, i) + Chi
          Chi = CWnx1
     NEXT i
     CWnx = Chi
     Chi = 0
     FOR n = 1 TO number
          lgN = LOG(NCL(n)) / LOG(10)
          CWn = Wn(10, n) + Chi
          Xs = Xo + (lx * lgN / LGXM)
          Ys = Yo + ly - (ly * CWn / CWnx)
          CALL PlotGraph(Xs, Ys, n, 13)
          Chi = CWn
     NEXT n
END SUB

SUB PlotGraph (Xs, Zs, n, Line1) STATIC
   IF n > 1 THEN
     IF Zs > 0 THEN CALL DLine(Xs1, Zs1, Xs, Zs,
Line1)
   END IF
     Xs1 = Xs: Zs1 = Zs
END SUB

SUB PlotGraph1 (Xs, Zs, n, Line1) STATIC
   IF n > 1 THEN
     IF Zs > 0 THEN CALL DLine(Xs2, Zs2, Xs, Zs,
Line1)
   END IF
     Xs2 = Xs: Zs2 = Zs
END SUB

SUB PlotOverLay
     CLS
     Axes
     CALL Legends(10, 49)
     Replot
     TotalCurve
     LabelPeakTT
     Axes
END SUB

SUB PlotTotal
     Axes
     TotalCurve
     LabelPeakTT
```

Copyright by Mahidol University

```
    Axes
END SUB

SUB PlotTotalForReport
REM Y-axis
    COLOR 14
    FOR i = 0 TO 8 STEP 1
        LOCATE 8 + 2 * i, 3
        PRINT USING "##.##^^^^"; (8 - i) * WMAX / 8
    NEXT i
    LOCATE 7, 1  COLOR 15  PRINT "Weight Fraction
"
    COLOR 14
    Yint = (ly / 8)
    PRESET (Xo, Yo + ly)
    FOR i = 1 TO 8 STEP 1
      PSET STEP(-4, -Yint)
      LINE -STEP(4, 0)
    NEXT i
    Yint = (ly  4)
    PRESET (Xo, Yo + ly + Yint)
    FOR i = 0 TO 4 STEP 1
      PSET STEP(-6, -Yint)
      LINE -STEP(6, 0)
    NEXT i
    FOR n = 1 TO number
      lgN = LOG(NCL(n)) / LOG(10)
        Xs = Xo + (lx * lgN / LGXM)
        Yall = ((Yo + ly) - (ly * Wn(10, n) / WMAX))
      CALL PlotGraph(Xs, Yall, n, 15)
    NEXT n
END SUB

SUB PlotWeightDistribute
  FOR i = 1 TO 9
   FOR n = 1 TO number
    lgN = LOG(NCL(n)) / LOG(10)
    Xs = Xo + (lx * lgN / LGXM)
    IF Wnx(i) > 0 THEN Y1 = Yo + ly - (ly * (Char(i) /
Yscale) * (Wn(i, n) / Wnx(i)))
    SELECT CASE i
        CASE 1
          LineColor = 9
        CASE 2
          LineColor = 1
        CASE 3
          LineColor = 2
        CASE 4
          LineColor = 3
        CASE 5
          LineColor = 10
        CASE 6
          LineColor = 14
        CASE 7
          LineColor = 12
        CASE 8
          LineColor = 7
        CASE 9
          LineColor = 13
    END SELECT
    CALL PlotGraph(Xs, Y1, n, LineColor)
  NEXT n
 NEXT i
END SUB

SUB Position (row, col, Text$, Tcolor)
  COLOR Tcolor
  LOCATE row, col
  PRINT Text$;
END SUB

SUB ReadTransferFile

REM Read simulation file for Calculation routine
  count = 0
  OPEN "PLOTCONV.2D" FOR INPUT AS #10
    DO UNTIL EOF(10)
    count = count + 1
    INPUT #10, Mt(count), CTAt(count), PTime(count)
    LOOP
  CLOSE #10

  count = 0
  OPEN "PLOTW 2DN" FOR INPUT AS #11
    DO UNTIL EOF(11)
      count = count + 1
      INPUT #11, NCL(count)
    LOOP
  CLOSE #11
  TotalCount = count

  count = 0
  OPEN "PLOTW.2D1" FOR INPUT AS #12
    DO UNTIL EOF(12)
      count = count + 1
      INPUT #12, Wn(1, count)
    LOOP
  CLOSE #12

  count = 0
  OPEN "PLOTW 2D2" FOR INPUT AS #13
    DO UNTIL EOF(13)
      count = count + 1
      INPUT #13, Wn(2, count)
    LOOP
  CLOSE #13

  count = 0
  OPEN "PLOTW.2D3" FOR INPUT AS #14
    DO UNTIL EOF(14)
      count = count + 1
      INPUT #14, Wn(3, count)
    LOOP
  CLOSE #14

  count = 0
  OPEN "PLOTW.2D4" FOR INPUT AS #15
    DO UNTIL EOF(15)
      count = count + 1
      INPUT #15, Wn(4, count)
    LOOP
  CLOSE #15

  count = 0
  OPEN "PLOTW.2D5" FOR INPUT AS #16
    DO UNTIL EOF(16)
      count = count + 1
      INPUT #16, Wn(5, count)
    LOOP
  CLOSE #16

  count = 0
  OPEN "PLOTW.2D6" FOR INPUT AS #17
    DO UNTIL EOF(17)
      count = count + 1
      INPUT #17, Wn(6, count)
    LOOP
  CLOSE #17

  count = 0
  OPEN "PLOTW.2D7" FOR INPUT AS #18
    DO UNTIL EOF(18)
      count = count + 1
      INPUT #18, Wn(7, count)
    LOOP
```

Copyright by Mahidol University

```
    CLOSE #18

    count = 0
    OPEN "PLOTW 2D8" FOR INPUT AS #19
      DO UNTIL EOF(19)
        count = count + 1
        INPUT #19, Wn(8, count)
      LOOP
    CLOSE #19

    count = 0
    OPEN "PLOTW 2D9" FOR INPUT AS #20
      DO UNTIL EOF(20)
        count = count + 1
        INPUT #20, Wn(9, count)
      LOOP
    CLOSE #20

    count = 0
    OPEN "PLOTW.2DT" FOR INPUT AS #21
      DO UNTIL EOF(21)
        count = count + 1
        INPUT #21, Wn(10, count)
      LOOP
    CLOSE #21

END SUB

SUB Replot STATIC
      PlotWeightDistribute
      LabelPeak
END SUB

SUB Right (row, Text$, Tcolor)
    COLOR Tcolor
    LOCATE row, 80 - LEN(Text$)
    PRINT Text$,
END SUB

SUB TotalCurve
REM Y-axis
      COLOR 15
      FOR i = 0 TO 5 STEP 1
          LOCATE 8 + 3 * i, 3
          PRINT USING "##.##^^^^"; (5 - i) * WMAX / 5
      NEXT i
      CALL Left(7, "Weight Fraction          ", 15)
      LINE (Xo, Yo)-(Xo, Yo + ly), 15
      COLOR 15
      Yint = (ly / 10)
      PRESET (Xo, Yo + ly)
      FOR i = 1 TO 10 STEP 1
        PSET STEP(-4, -Yint)
        LINE -STEP(4, 0)
      NEXT i
      Yint = (ly / 5)
      PRESET (Xo, Yo + ly + Yint)
      FOR i = 0 TO 5 STEP 1
        PSET STEP(-8, -Yint)
        LINE -STEP(8, 0)
      NEXT i
      FOR n = 1 TO number
        lgN = LOG(NCL(n)) / LOG(10)
          Xs = Xo + (lx * lgN / LGXM)
          Yall = ((Yo + ly) - (ly * Wn(10, n) / WMAX))
        CALL PlotGraph(Xs, Yall, n, 15)
      NEXT n
END SUB

SUB ViewMenu (xcount, owncolor)
      IF xcount > 7 THEN xcount = 7
      IF xcount < 1 THEN xcount = 1
      COLOR owncolor
      SELECT CASE xcount
          CASE 1
              CALL Position(2, 4, View$(1), owncolor)
          CASE 2
              CALL Position(2, 16, View$(2), owncolor)
          CASE 3
              CALL Position(2, 29, View$(3), owncolor)
          CASE 4
              CALL Position(2, 42, View$(4), owncolor)
          CASE 5
              CALL Position(2, 51, View$(5), owncolor)
          CASE 6
              CALL Position(2, 63, View$(6), owncolor)
          CASE 7
              CALL Position(2, 71, View$(7), owncolor)
      END SELECT
END SUB

SUB ViewPlot
    DO: Xo = 100: Yo = 120: lx = 440: ly = 240
      FunckeyView
      count = 1
      DO
        CALL ViewMenu(count, 12)
      DO
        KBD$ = INKEY$
      LOOP UNTIL KBD$ <> ""
      SELECT CASE KBD$
          CASE CHR$(0) + "K"  'left
              count = count - 1
              IF count < 1 THEN count = 1
              CALL ViewMenu(count + 1, 15)
          CASE CHR$(0) + "M"  'right
              count = count + 1
              IF count > 7 THEN count = 7
              CALL ViewMenu(count - 1, 15)
      END SELECT
      inpcount = count
      LOOP UNTIL KBD$ = CHR$(13) OR KBD$ =
CHR$(27)
      IF KBD$ = CHR$(27) THEN EXIT SUB
      SELECT CASE inpcount
          CASE 1
              EraseScreen
              PlotConver
          CASE 2
              PlotTotal
          CASE 3
              EraseScreen
              AllSpecies
          CASE 4
              PlotOverLay
          CASE 5
              PlotCumulative
          CASE 6
              LOCATE 4, 3: COLOR 9: PRINT "Do you
want to report all species ? (<Y>..Show all species
<N> .Only Total)"
              DO
                  ReportResult$ = UCASE$(INPUT$(1))
              LOOP UNTIL ReportResult$ = "Y" OR
ReportResult$ = "N"
              EraseScreen
              PlotCreateReport
          CASE 7
              EraseScreen
      END SELECT
    LOOP UNTIL KBD$ = CHR$(27)
END SUB
```

Copyright by Mahidol University

7) Programmes for three dimension plot of the weight distribution of dead-end polymers (PLOTW-3D.BAS)

```
DECLARE SUB Labelscale ()
DECLARE SUB Funckey ()
DECLARE SUB CurveMenu (xcount!, owncolor!)
DECLARE SUB ReplotMenu ()
DECLARE SUB InitialHpoint (Xss!, Yss!, xcount!)
DECLARE SUB UpdateHpoint (Xss!, Yss!, xcount!)
DECLARE SUB LoadAllParameter ()
DECLARE SUB Axe3Dimension ()
DECLARE SUB PlotWeightDistribute ()
DECLARE SUB PlotGraphI (Xs!, Zs!, j!, Line1!)
DECLARE SUB PlotGraph (Xs!, Zs!, j!, Line1!)
DECLARE SUB DLine (Xbegin!, Ybegin!, Xend!, Yend!,
LineColor!)
DECLARE SUB CalculateMaxWmax ()
COMMON SHARED Mo, Io, CTA, Temp, Time, FT
COMMON SHARED number, h, Xscale
COMMON SHARED Xo, Yo, Ix, Iy, Iz, Seta, Beta, Treta,
Breta, SD
COMMON SHARED Wmax, Nmax, Tmax, Tox, Nox
LoadAllParameter
FT = 60 * Time
FTHcount = INT((FT / h) / Tox) + 1
DIM SHARED XoldPixel(0 TO FTHcount)
DIM SHARED YoldPixel(0 TO FTHcount)
DIM SHARED XnewPixel(0 TO FTHcount)
DIM SHARED YnewPixel(0 TO FTHcount)
DIM SHARED Menu$(1 TO 10)
Menu$(1) = "Left angle"
Menu$(2) = "Right angle"
Menu$(3) = "Line color"
Menu$(4) = "X-coordinate"
Menu$(5) = "Y-coordinate"
Menu$(6) = "3-D Plot"
CalculateMaxWmax
Seta = (3 14278 / 2) * (Treta / 90)
Beta = (3 14278 / 2) * (Breta / 90)
SD = 2
    CLS
    SCREEN 12
    PlotWeightDistribute
    Axe3Dimension
    IF Xo = 550 AND Yo = 380 AND Ix = 250 AND Iy =
250 AND Iz = 190 AND Breta = 30 AND Treta = 15
THEN
    Labelscale
    END IF
    LOCATE 1, 5. COLOR 12: PRINT "F10 <Menu>"
    DO
        DO
            kbd$ = INKEY$
        LOOP UNTIL kbd$ <> ""
        IF kbd$ = CHR$(0) + "D" THEN ReplotMenu
    LOOP UNTIL kbd$ = CHR$(27)
END

SUB Axe3Dimension
  LINE (Xo - Ix * COS(Seta) - Iy * COS(Beta), Yo - Iy *
SIN(Beta))-(Xo - Ix * COS(Seta), Yo), 9
  LINE -(Xo, Yo - Ix * SIN(Seta)), 9
  LINE -(Xo, Yo - Ix * SIN(Seta) - Iz), 9
END SUB

SUB CalculateMaxWmax STATIC
    SCREEN 12: COLOR 10: LOCATE 15, 30: PRINT
"Please wait . Now Calculate.!"
    InitialWmax = 0
    OPEN "PLOTW.3D" FOR INPUT AS #3
    DO UNTIL EOF(3)
    INPUT #3, Wn10x, nx, Tx
     IF InitialWmax > Wn10x THEN InitialWmax =
InitialWmax ELSE InitialWmax = Wn10x
    LOOP
    CLOSE #3
    Wmax = InitialWmax
    Tmax = FT
    Nmax = Xscale
    CLS
END SUB

SUB CurveMenu (xcount, owncolor)
    IF xcount > 6 THEN xcount = 6
    IF xcount < 1 THEN xcount = 1
    COLOR owncolor
    SELECT CASE xcount
        CASE 1
            LOCATE 2, 4: PRINT Menu$(1)
        CASE 2
            LOCATE 2, 16: PRINT Menu$(2)
        CASE 3
            LOCATE 2, 29: PRINT Menu$(3)
        CASE 4
            LOCATE 2, 41: PRINT Menu$(4)
        CASE 5
            LOCATE 2, 55: PRINT Menu$(5)
        CASE 6
            LOCATE 2, 69: PRINT Menu$(6)
    END SELECT
END SUB

SUB DLine (Xbegin, Ybegin, Xend, Yend, LineColor)
  LINE (Xbegin, Ybegin)-(Xend, Yend), LineColor
END SUB

SUB Funckey STATIC
    COLOR 12
    LOCATE 1, 1
    PRINT " ----------"
    LOCATE 2, 1
    PRINT "          "
    LOCATE 3, 1
    PRINT " ----------"
    LOCATE 4, 1
    PRINT "          "
    LOCATE 5, 1
    PRINT " ----------"
    COLOR 15
    LOCATE 2, 2
    PRINT " Left angle  Right angle  Line color  X-
coordinate  Y-coordinate  3-D Plot "
END SUB

SUB InitialHpoint (Xss, Yss, xcount) STATIC
  XoldPixel(xcount) = Xss
  YoldPixel(xcount) = Yss
END SUB

SUB Labelscale
  COLOR 9
  LOCATE 6, 66: PRINT "Weight fraction"
  FOR i = 0 TO 4
    LOCATE 8 + 3 * i, 72: PRINT USING "##.##^^^^";
(4 - i) * Wmax / 4
  NEXT i
  FOR i = 0 TO 1 STEP .25
```

Copyright by Mahidol University

```
      LINE (Xo, Yo - lx * SIN(Seta) - i * lz)-(Xo + 20 *
 COS(Seta), Yo - 20 * SIN(Seta) - lx * SIN(Seta) - i * lz), 9
    NEXT i
    COLOR 14
    LOCATE 25, 55: PRINT "log(Chain length)"
    FOR i = 0 TO 4
      LOCATE 25 - i, 40 + 8 * i: PRINT USING "## ##"; i
 * Nmax / 4
    NEXT i
    FOR i = 0 TO 1 STEP .25
      LINE (Xo - i * lx * COS(Seta), Yo - lx * SIN(Seta) + i
 * lx * SIN(Seta))-(Xo - i * lx * COS(Seta) + 10 *
 (COS(Beta) / SIN(Beta)), Yo - lx * SIN(Seta) + 10 + i * lx *
 SIN(Seta)), 9
    NEXT i
    COLOR 13
    LOCATE 22, 10: PRINT "Time(min)"
    FOR i = 0 TO 4
      LOCATE 17 + 2 * i, 4 + 7 * i: PRINT USING
 "####.#"; (4 - i) * Tmax / (4 * 60)
    NEXT i
    FOR i = 0 TO 1 STEP .25
      LINE (Xo - lx * COS(Seta) - ly * COS(Beta) + i * ly *
 COS(Beta), Yo - ly * SIN(Beta) + i * ly * SIN(Beta))-(Xo -
 lx * COS(Seta) - ly * COS(Beta) - 6 * (COS(Seta) /
 SIN(Seta)) + i * ly * COS(Beta), Yo + 6 - ly * SIN(Beta) + i
 * ly * SIN(Beta) _
 ), 9
    NEXT i
 END SUB

 SUB LoadAllParameter
      OPEN "CONST.MTX" FOR INPUT AS #1
        INPUT #1, MoName$
        INPUT #1, IoName$
        INPUT #1, SoName$
        INPUT #1, Mo
        INPUT #1, Io
        INPUT #1, CTA
        INPUT #1, Temp
        INPUT #1, Time
        INPUT #1, Kp
        INPUT #1, Ktc
        INPUT #1, Ktd
        INPUT #1, Ktrm
        INPUT #1, Ki
        INPUT #1, Kd
        INPUT #1, f
        INPUT #1, Ktci
        INPUT #1, Ksr
        INPUT #1, Ktr
        INPUT #1, number
        INPUT #1, h
        INPUT #1, Xscale
        INPUT #1, Yscale
        INPUT #1, GelEffect$
        INPUT #1, Xo
        INPUT #1, Yo
        INPUT #1, lx
        INPUT #1, ly
        INPUT #1, lz
        INPUT #1, Breta
        INPUT #1, Treta
        INPUT #1, Nox
        INPUT #1, Tox
      CLOSE #1
 END SUB

 SUB PlotGraph (Xs, Zs, j, Line1) STATIC
    IF j > 0 THEN
    CALL DLine(Xs1, Zs1, Xs, Zs, Line1)
    END IF
    Xs1 = Xs: Zs1 = Zs
 END SUB

 SUB PlotGraphI (Xs, Zs, j, Line1) STATIC
    IF j > 0 THEN
      CALL DLine(Xs2, Zs2, Xs, Zs, Line1)
    END IF
      Xs2 = Xs: Zs2 = Zs
 END SUB

 SUB PlotWeightDistribute
    lgN0 = 0
    count = 0
    OPEN "PLOTW.3D" FOR INPUT AS #4
    DO UNTIL EOF(4)
     INPUT #4, Wn10x, nx, Tx
     count = count + 1
          lgN = LOG(nx) / LOG(10)
          lx1 = lx * (Nmax - lgN) / Nmax
          ly1 = ly * Tx / Tmax
          lz1 = lz * Wn10x / Wmax
          Xs = Xo - (lx1 * COS(Seta)) - (ly1 * COS(Beta))
          Ys = Yo - ((lx - lx1) * SIN(Seta)) - (ly1 *
 SIN(Beta)) - lz1
          CALL PlotGraphI(Xs, Ys, Tx, SD)
          IF lgN = 0 THEN
               FTHcount = INT((FT / h) / Tox) + 1
               Sentcount = count MOD FTHcount
               IF Sentcount > 0 THEN CALL
 InitialHpoint(Xs, Ys, Sentcount)
          END IF
          IF lgN > 0 THEN
               FTHcount = INT((FT / h) / Tox) + 1
               Xsentcount = count MOD FTHcount
               IF Xsentcount > 0 THEN CALL
 UpdateHpoint(Xs, Ys, Xsentcount)
          END IF
    LOOP
    CLOSE #4
 END SUB

 SUB ReplotMenu STATIC
    DO
     Funckey
     count = 1
     DO
      CALL CurveMenu(count, 12)
     DO
      kbd$ = INKEY$
     LOOP UNTIL kbd$ <> ""
     SELECT CASE kbd$
          CASE CHR$(0) + "K"  'left
               count = count - 1
               IF count < 1 THEN count = 1
               CALL CurveMenu(count + 1, 15)
          CASE CHR$(0) + "M"  'right
               count = count + 1
               IF count > 6 THEN count = 6
               CALL CurveMenu(count - 1, 15)
     END SELECT
     inpcount = count
     LOOP UNTIL kbd$ = CHR$(13) OR kbd$ =
 CHR$(27)
     IF kbd$ = CHR$(27) THEN END
     COLOR 9
     SELECT CASE inpcount
          CASE 1
               LOCATE 4, 4: INPUT "Angle from left hand
 side : ", Breta
               IF Breta = 0 THEN Breta = 30
          CASE 2
```

Copyright by Mahidol University

```
            LOCATE 4, 4: INPUT "Angle from right hand
side . ", Treta
            IF Treta = 0 THEN Treta = 15
         CASE 3
            LOCATE 4, 4  INPUT "Line color of 3-D Plot
(Range 1..15) (enter to use defualt value=8) : ", SD
            IF SD < 1 OR SD > 15 THEN SD = 8
         CASE 4
            LOCATE 4, 4: INPUT "X-coordination of
origin (Range 1 to 639 in pixel unit, defualt=550): ", Xo
            IF Xo = 0 THEN Xo = 550
         CASE 5
            LOCATE 4, 4: INPUT "Y-coordination of
origin (Range 1 to 450 in pixel unit, defualt=380): ", Yo
            IF Yo = 0 THEN Yo = 380
         CASE 6
            Seta = (3.14278 / 2) * (Treta / 90)
            Beta = (3.14278 / 2) * (Breta / 90)
            CLS
            SCREEN 12
              PlotWeightDistribute
              Axe3Dimension
              IF Xo = 550 AND Yo = 380 AND lx = 250
AND ly = 250 AND lz = 190 AND Breta = 30 AND Treta
= 15 THEN
              Labelscale
              END IF
     END SELECT
  LOOP UNTIL kbd$ = CHR$(27)
END
END SUB

SUB UpdateHpoint (Xss, Yss, xcount) STATIC
  XnewPixel(xcount) = Xss
  YnewPixel(xcount) = Yss
  LINE (XoldPixel(xcount), YoldPixel(xcount))-
(XnewPixel(xcount), YnewPixel(xcount)), SD
  XoldPixel(xcount) = XnewPixel(xcount)
  YoldPixel(xcount) = YnewPixel(xcount)
END SUB
```

Copyright by Mahidol University

8) Programmes for displaying the plot of the Modified Mayo Equation (PLOTM-2D.BAS)

```
DECLARE SUB PlotGraphI (Xs!, Zs!, n!, Line1!)
DECLARE SUB PlotGraph (Xs!, Zs!, n!, Line1!)
DECLARE SUB CalMaxMin ()
DECLARE SUB ArrowMenu (Xcount!, owncolor!)
DECLARE SUB SelectPoint1 ()
DECLARE SUB SelectPoint2 ()
DECLARE SUB FunckeyForView ()
DECLARE SUB ViewGraphMenu (Xcount!, owncolor!)
DECLARE SUB PlotLine (i!, Zs!, ZZs!)
DECLARE SUB PlotRegressionLine ()
DECLARE SUB LinearRegression ()
DECLARE SUB Position (row!, col!, Text$, Tcolor!)
DECLARE SUB EraseScreen ()
DECLARE SUB FunckeyForTable1 ()
DECLARE SUB RegressionFromTable ()
DECLARE SUB FunckeyForTable2 ()
DECLARE SUB TableMenu1 (Xcount!, owncolor!)
DECLARE SUB Axes ()
DECLARE SUB LabelScale ()
DECLARE SUB PlotModifyMayo ()
DECLARE FUNCTION Xspixel! (Xval!)
DECLARE FUNCTION Yspixel! (Yval!)
DECLARE SUB PlotSymb (Xs!, Ys!)
DECLARE SUB PrintDATA (Anymore!)
DECLARE SUB TableMenu (Anymore!)
DECLARE SUB TableBox ()
DECLARE SUB PrintParameters ()
DECLARE SUB PrintDataTable ()
DECLARE SUB ViewGraph ()
DECLARE SUB ReadTransferFile ()
DECLARE SUB CurveMenu (Xcount!, owncolor!)
DECLARE SUB Center (row!, Text$, Tcolor!)
DECLARE SUB Left (row!, Text$, Tcolor!)
DECLARE SUB LoadAllParameter ()
DECLARE SUB Funckey ()
DECLARE SUB DLine (Xbegin!, Ybegin!, Xend!, Yend!,
LineColor!)
DECLARE SUB ParametersBOX ()
DECLARE SUB PrintRegressResult ()
DECLARE SUB ComparePlot ()
DECLARE SUB CalArrayOldMaYo ()
DECLARE SUB PlotRegressionLineOLD ()
DECLARE SUB PlotSymbOLD (Xs!, Ys!)
DECLARE SUB PlotLineOLD (i!, Zs!, ZZs!)
DECLARE SUB LinearRegressionOLD ()
DECLARE SUB PrintRegressResultOLD ()
DECLARE FUNCTION XspixelOLD! (Xval!)
DECLARE SUB PlotMayoOLD ()
COMMON SHARED MoName$, IoName$, SoName$, Mo,
Io, CTAo1, CTAo2, PercentSampling, Temp, Time
COMMON SHARED numberDATA, h, FT, Kp, Ktr
COMMON SHARED CurveSaving$, CurveDrive$,
CurveName$, EXPSaving$, EXPDrive$, EXPName$
COMMON SHARED ReciprocalTDPnMax,
ReciprocalTDPnMin, TDPnMax, TDPnMin,
RatioCTAtoMmax, RatioCTAtoMmin
COMMON SHARED RatioCTAtoMOLDmax,
RatioCTAtoMOLDmin
COMMON SHARED TotalCount
COMMON SHARED Xo, Yo, lx, ly
COMMON SHARED Group, Syx, eR2, eR, Slope, Intercept
COMMON SHARED SpecialKBD$, CP$, MK$, LC$,
RL$, Results$, ZZs1, Zs1, Xss1, Zss1, XXss1, ZZss1
COMMON SHARED SyxOLD, eR2OLD, eROLD,
SlopeOLD, InterceptOLD
SCREEN 12
COLOR 15
LOCATE 15, 25
PRINT "Wait !... Now System working."
LoadAllParameter
FT = Time * 60
MK$ = "Y"
LC$ = "N"
NumDATA = 1 + numberDATA
DIM SHARED RatioCTAtoM(1 TO NumDATA)
DIM SHARED RatioCTAtoMOLD(1 TO NumDATA)
DIM SHARED TDPn(1 TO NumDATA)
DIM SHARED DPEst(1 TO NumDATA)
DIM SHARED SamplingTime(1 TO NumDATA)
DIM SHARED Menu$(1 TO 5)
DIM SHARED Table$(1 TO 3)
DIM SHARED ViewG$(1 TO 8)
DIM SHARED Mark(1 TO 2)
ReadTransferFile
Mark(1) = 1
Mark(2) = TotalCount
Table$(1) = "First point"
Table$(2) = "Second point"
Table$(3) = "Linear regression"
ViewG$(1) = "Marker"
ViewG$(2) = "Line"
ViewG$(3) = "Regress"
ViewG$(4) = "Reset"
ViewG$(5) = "First"
ViewG$(6) = "Second"
ViewG$(7) = "Result"
ViewG$(8) = "Compare"
Menu$(1) = "Parameters"
Menu$(2) = "Data table"
Menu$(3) = "View graph"
Menu$(4) = "Clear screen"
CLS
SCREEN 12
  DO: Xo = 100: Yo = 120: lx = 480: ly = 240
    Funckey
    count = 1
    DO
     CALL CurveMenu(count, 12)
    DO
     KBD$ = INKEY$
    LOOP UNTIL KBD$ <> ""
    SELECT CASE KBD$
        CASE CHR$(0) + "K"  'left
           count = count - 1
           IF count < 1 THEN count = 1
           CALL CurveMenu(count + 1, 15)
        CASE CHR$(0) + "M"  'right
           count = count + 1
           IF count > 4 THEN count = 4
           CALL CurveMenu(count - 1, 15)
    END SELECT
    inpcount = count
    LOOP UNTIL KBD$ = CHR$(13) OR KBD$ =
CHR$(27)
    IF KBD$ = CHR$(27) THEN END
    SELECT CASE inpcount
        CASE 1
           PrintParameters
        CASE 2
           PrintDataTable
        CASE 3
           ViewGraph
        CASE 4
           EraseScreen

    END SELECT
  LOOP UNTIL KBD$ = CHR$(27)
END
```

Copyright by Mahidol University

```
SUB ArrowMenu (Xcount, owncolor)
    Xs = Xspixel(Xcount)
    Ys = Yspixel(Xcount)
    CALL DLine(Xs - 3, Ys - 3, Xs + 3, Ys + 3, owncolor)
    CALL DLine(Xs + 3, Ys - 3, Xs - 3, Ys + 3, owncolor)
    CALL DLine(Xs, Ys, Xs, Ys - 15, owncolor)
    CIRCLE (Xs, Ys - 20), 5, owncolor
END SUB

SUB Axes
SCREEN 12
    CALL DLine(Xo - 4, Yo + ly, Xo + lx, Yo + ly, 14)
    CALL Center(26, "    Ratio of [CTA] to [M]", 14)
    COLOR 14
    Xint = (lx / 10)
    PRESET (Xo - Xint, Yo + ly + 4)
    FOR i = 0 TO 10 STEP 1
      PSET STEP(Xint, -3)
      LINE -STEP(0, 3)
    NEXT i
    COLOR 14
    Xint = (lx / 5)
    PRESET (Xo - Xint, Yo + ly + 8)
    FOR i = 0 TO 5 STEP 1
      PSET STEP(Xint, -8)
      LINE -STEP(0, 8)
    NEXT i
    FOR i = 0 TO 5
      COLOR 14
      LOCATE 24, 9 + 12 * i
      PRINT USING "## #^^^^", RatioCTAtoMmin + ((i
/ 5) * (RatioCTAtoMmax - RatioCTAtoMmin))
    NEXT i
END SUB

SUB CalArrayOldMaYo
  count = 1
  FOR i = CTAo1 TO CTAo2 STEP (CTAo2 - CTAo1) /
numberDATA
    RatioCTAtoMOLD(count) = i / Mo
    count = count + 1
  NEXT i
END SUB

SUB CalMaxMin
  TDPnMax = TDPn(Mark(1))
  TDPnMin = TDPn(Mark(1))
  FOR i = Mark(1) TO Mark(2)
    IF TDPn(i) > TDPnMax THEN TDPnMax = TDPn(i)
ELSE TDPnMax = TDPnMax
    IF TDPn(i) < TDPnMin THEN TDPnMin = TDPn(i)
ELSE TDPnMin = TDPnMin
  NEXT i
  ReciprocalTDPnMax = 1 / (TDPnMin - 1)
  ReciprocalTDPnMin = 1 / (TDPnMax - 1)
  RatioCTAtoMmax = RatioCTAtoM(Mark(1))
  RatioCTAtoMmin = RatioCTAtoM(Mark(1))
  FOR i = Mark(1) TO Mark(2)
    IF RatioCTAtoM(i) > RatioCTAtoMmax THEN
RatioCTAtoMmax = RatioCTAtoM(i) ELSE
RatioCTAtoMmax = RatioCTAtoMmax
    IF RatioCTAtoM(i) < RatioCTAtoMmin THEN
RatioCTAtoMmin = RatioCTAtoM(i) ELSE
RatioCTAtoMmin = RatioCTAtoMmin
  NEXT i

END SUB

SUB Center (row, Text$, Tcolor)
  COLOR Tcolor
  LOCATE row, 41 - LEN(Text$) / 2
  PRINT Text$;
END SUB

SUB ComparePlot
  PlotModifyMayo
  PlotMayoOLD
END SUB

SUB CurveMenu (Xcount, owncolor)
    SELECT CASE Xcount
      CASE 1
        CALL Position(2, 4, Menu$(1), owncolor)
      CASE 2
        CALL Position(2, 17, Menu$(2), owncolor)
      CASE 3
        CALL Position(2, 30, Menu$(3), owncolor)
      CASE 4
        CALL Position(2, 43, Menu$(4), owncolor)
    END SELECT
END SUB

SUB DLine (Xbegin, Ybegin, Xend, Yend, LineColor)
    LINE (Xbegin, Ybegin)-(Xend, Yend), LineColor
END SUB

SUB EraseScreen
    LOCATE 6, 1
    FOR i = 1 TO 22
      PRINT STRING$(80, " ");
    NEXT i
END SUB

SUB Funckey STATIC
    CALL Center(1, PRINT " ----------", 12)
    CALL Center(2, PRINT "           ", 12)
    CALL Center(3, PRINT " ----------", 12)
    CALL Center(4, PRINT "           ", 12)
    CALL Center(5, PRINT " ----------", 12)
    CALL Position(2, 4, Menu$(1), 15)
    CALL Position(2, 17, Menu$(2), 15)
    CALL Position(2, 30, Menu$(3), 15)
    CALL Position(2, 43, Menu$(4), 15)
    CALL Position(2, 67, "Abort=[ESC]", 9)
END SUB

SUB FunckeyForTable1 STATIC
    CALL Center(1, PRINT " ----------", 12)
    CALL Center(2, PRINT "           ", 12)
    CALL Center(3, PRINT " ----------", 12)
    CALL Center(4, PRINT "           ", 12)
    CALL Center(5, PRINT " ----------", 12)
    CALL Center(2, "F10=[Table Menu]", 14)
    CALL Position(4, 55, "More=[PU&D],Abort=[ESC]",
9)
END SUB

SUB FunckeyForTable2 STATIC
    CALL Center(1, PRINT " ----------", 12)
    CALL Center(2, PRINT "           ", 12)
    CALL Center(3, PRINT " ----------", 12)
    CALL Center(4, PRINT "           ", 12)
    CALL Center(5, PRINT " ----------", 12)
    CALL Position(2, 4, Table$(1), 15)
    CALL Position(2, 17, Table$(2), 15)
    CALL Position(2, 31, Table$(3), 15)
    CALL Position(2, 50, Menu$(4), 15)
    CALL Position(2, 67, "Abort=[ESC]", 9)
END SUB


SUB FunckeyForView STATIC
    CALL Center(1, PRINT " ----------", 12)
```

Copyright by Mahidol University

```
        CALL Center(2, PRINT "                                ", 12)
        CALL Center(3, PRINT " ----------------------", 12)
        CALL Center(4, PRINT "                                ", 12)
        CALL Center(5, PRINT " ----------------------", 12)
        CALL Position(2, 4, ViewG$(1), 15)
        CALL Position(2, 12, ViewG$(2), 15)
        CALL Position(2, 18, ViewG$(3), 15)
        CALL Position(2, 27, ViewG$(4), 15)
        CALL Position(2, 34, ViewG$(5), 15)
        CALL Position(2, 41, ViewG$(6), 15)
        CALL Position(2, 49, ViewG$(7), 15)
        CALL Position(2, 57, ViewG$(8), 15)
        CALL Position(2, 67, "Abort=[ESC]", 9)
END SUB

SUB FunckeyUtil STATIC
        CALL Center(1, PRINT " ----------------------", 12)
        CALL Center(2, PRINT "                                ", 12)
        CALL Center(3, PRINT " ----------------------", 12)
        CALL Center(4, PRINT "                                ", 12)
        CALL Center(5, PRINT " ----------------------", 12)
        CALL Position(2, 4, Util$(1), 15)
        CALL Position(2, 20, Util$(2), 15)
        CALL Position(2, 40, Util$(3), 15)
        CALL Position(2, 67, "Abort=[ESC]", 9)
END SUB

SUB LabelScale STATIC
REM Y-axis
        CALL Left(6, "Reciprocal DPn", 15)
        CALL DLine(Xo, Yo, Xo, Yo + ly, 15)
        COLOR 15
        Yint = (ly / 10)
        PRESET (Xo, Yo + ly)
        FOR i = 1 TO 10 STEP 1
            PSET STEP(-4, -Yint)
            LINE -STEP(4, 0)
        NEXT i
        Yint = (ly / 5)
        PRESET (Xo, Yo + ly + Yint)
        FOR i = 0 TO 5 STEP 1
            PSET STEP(-8, -Yint)
            LINE -STEP(8, 0)
        NEXT i
        COLOR 15
        FOR i = 0 TO 5 STEP 1
            LOCATE 23 - 3 * i, 3
            PRINT USING "##.##^^^^",
ReciprocalTDPnMin + ((i / 5) * (ReciprocalTDPnMax -
ReciprocalTDPnMin))
        NEXT i
END SUB

SUB Left (row, Text$, Tcolor)
    COLOR Tcolor
    LOCATE row, 2
    PRINT Text$;
END SUB

SUB LinearRegression STATIC
        SumX = 0: SumY = 0  SumXY = 0: SumX2 = 0: STD
= 0: SRD = 0
        NP = ABS(Mark(2) - Mark(1)) + 1
        FOR i = Mark(1) TO Mark(2)
            SumX = SumX + RatioCTAtoM(i)
            IF TDPn(i) > 1 THEN
            SumY = SumY + (1 / (TDPn(i) - 1))
            SumXY = SumXY + RatioCTAtoM(i) * (1 /
(TDPn(i) - 1))
            END IF
            SumX2 = SumX2 + RatioCTAtoM(i) *
RatioCTAtoM(i)
        NEXT i
            Xmean = SumX / (NP)
            Ymean = SumY / (NP)
            Slp = ((NP * SumXY) - (SumX * SumY)) / ((NP *
SumX2) - (SumX * SumX))
            Intp = Ymean - (Slp * Xmean)
        FOR i = Mark(1) TO Mark(2)
            IF TDPn(i) > 0 THEN
            STD = STD + (((1 / (TDPn(i) - 1)) - Ymean) ^ 2)
            SRD = SRD + (((1 / (TDPn(i) - 1)) - (Slp *
RatioCTAtoM(i)) - Intp) ^ 2)
            END IF
        NEXT i
            IF NP - 2 > 0 THEN Syx = SQR(SRD / (NP - 2))
            IF STD <> 0 THEN eR2 = (STD - SRD) / STD
            IF eR2 >= 0 THEN eR = SQR(eR2)
            Slope = Slp
            Intercept = Intp
END SUB

SUB LinearRegressionOLD STATIC
        SumX = 0: SumY = 0  SumXY = 0: SumX2 = 0  STD
= 0: SRD = 0
        NP = ABS(Mark(2) - Mark(1)) + 1
        FOR i = Mark(1) TO Mark(2)
            SumX = SumX + RatioCTAtoMOLD(i)
            IF TDPn(i) > 1 THEN
            SumY = SumY + (1 / (TDPn(i) - 1))
            SumXY = SumXY + RatioCTAtoMOLD(i) * (1 /
(TDPn(i) - 1))
            END IF
            SumX2 = SumX2 + RatioCTAtoMOLD(i) *
RatioCTAtoMOLD(i)
        NEXT i
            Xmean = SumX / (NP)
            Ymean = SumY / (NP)
            Slp = ((NP * SumXY) - (SumX * SumY)) / ((NP *
SumX2) - (SumX * SumX))
            Intp = Ymean - (Slp * Xmean)
        FOR i = Mark(1) TO Mark(2)
            IF TDPn(i) > 0 THEN
            STD = STD + (((1 / (TDPn(i) - 1)) - Ymean) ^ 2)
            SRD = SRD + (((1 / (TDPn(i) - 1)) - (Slp *
RatioCTAtoMOLD(i)) - Intp) ^ 2)
            END IF
        NEXT i
            IF NP - 2 > 0 THEN SyxOLD = SQR(SRD / (NP
- 2))
            IF STD <> 0 THEN eR2 = (STD - SRD) / STD
            IF eR2 >= 0 THEN eROLD = SQR(eR2)
            SlopeOLD = Slp
            InterceptOLD = Intp
END SUB

SUB LoadAllParameter
    OPEN "CONST.MTY" FOR INPUT AS #1
        INPUT #1, MoName$
        INPUT #1, IoName$
        INPUT #1, SoName$
        INPUT #1, Mo
        INPUT #1, Io
        INPUT #1, CTAo1
        INPUT #1, CTAo2
        INPUT #1, PercentSampling
        INPUT #1, Temp
        INPUT #1, Time
        INPUT #1, Kp
        INPUT #1, Ktc
        INPUT #1, Ktd
        INPUT #1, Ktrm
        INPUT #1, Ki
        INPUT #1, Kd
```

Copyright by Mahidol University

```
        INPUT #1, f
        INPUT #1, Ktci
        INPUT #1, Ksr
        INPUT #1, Ktr
        INPUT #1, numberNCL
        INPUT #1, Xscale
        INPUT #1, h
        INPUT #1, numberDATA
        INPUT #1, GelEffect$
    CLOSE #1
END SUB

SUB PaintPattern (Xposit, Yposit, RowLength, ColLength,
Fcolor, Bcolor, Text$)
    FOR i = Yposit TO Yposit + RowLength
        COLOR Fcolor, Bcolor
        LOCATE i, Xposit + 1: PRINT
STRING$(ColLength, Text$)
    NEXT i
END SUB

SUB ParametersBOX
    X = 11
    CALL Center(10,"----------------------------", X)
    CALL Center(11, "                            ",X)
    CALL Center(12,"----------------------------", X)
    CALL Center(13, "                            ",X)
    CALL Center(14, "                            ",X)
    CALL Center(15, "                            ",X)
    CALL Center(16, "                            ",X)
    CALL Center(17, "                            ",X)
    CALL Center(18, "                            ",X)
    CALL Center(19, "                            ",X)
    CALL Center(20, "                            ",X)
    CALL Center(21, "                            ",X)
    CALL Center(22,"----------------------------", X)
END SUB

SUB PlotGraph (Xs, Zs, n, Line1) STATIC
  IF Xs >= Xo AND Xs <= Xo + lx AND Zs >= Yo AND
Zs <= Yo + ly THEN
      IF n  Mark(1) THEN
        IF Zs > 0 THEN CALL DLine(Xss1, Zss1, Xs, Zs,
Line1)
      END IF
        Xss1 = Xs: Zss1 = Zs
  END IF
END SUB

SUB PlotGraphI (Xs, Zs, n, Line1) STATIC
  IF Xs >= Xo AND Xs <= Xo + lx AND Zs >= Yo AND
Zs <= Yo + ly THEN
      IF n > Mark(1) THEN CALL DLine(XXss1, ZZss1,
Xs, Zs, Line1)
  XXss1 = Xs: ZZss1 = Zs
  END IF
END SUB

SUB PlotLine (i, Zs, ZZs)
    IF Zs >= Xo AND Zs <= Xo + lx AND ZZs >= Yo
AND ZZs <= Yo + ly THEN
        IF i > Mark(1) THEN CALL DLine(Zs1, ZZs1,
Zs, ZZs, 11)
        Zs1 = Zs: ZZs1 = ZZs
    END IF
END SUB

SUB PlotLineOLD (i, Zs, ZZs)
    IF Zs >= Xo AND Zs <= Xo + lx AND ZZs >= Yo
AND ZZs <= Yo + ly THEN
        IF i > Mark(1) THEN CALL DLine(Zs1, ZZs1,
Zs, ZZs, 13)
        Zs1 = Zs: ZZs1 = ZZs
    END IF

END SUB

SUB PlotMayoOLD
  IF RL$ = "Y" THEN CALL PlotRegressionLineOLD
  Axes
  LabelScale
  FOR n = Mark(1) TO Mark(2)
    Xs = XspixelOLD(n)
    Ys = Yspixel(n)      'OLD=MODIFY use same value
    IF MK$ = "Y" THEN CALL PlotSymbOLD(Xs, Ys)
    IF LC$ = "Y" THEN CALL PlotLineOLD(n, Xs, Ys)
  NEXT n
  IF Results$ = "Y" THEN LinearRegressionOLD:
ParametersBOX: PrintRegressResultOLD
  Axes
END SUB

SUB PlotModifyMayo
  CalMaxMin
  IF RL$ = "Y" THEN CALL PlotRegressionLine
  IF CP$ = "Y" THEN CALL PlotMayoOLD
  Axes
  LabelScale
  FOR n = Mark(1) TO Mark(2)
    Xs = Xspixel(n)
    Ys = Yspixel(n)
    IF MK$ = "Y" THEN CALL PlotSymb(Xs, Ys)
    IF LC$ = "Y" THEN CALL PlotLine(n, Xs, Ys)
  NEXT n
  IF Results$ = "Y" THEN LinearRegression:
ParametersBOX: PrintRegressResult
  Axes
END SUB

SUB PlotRegressionLine
  CalMaxMin
  LinearRegression
  FOR i = Mark(1) TO Mark(2)
    DPEst(i) = (Slope * RatioCTAtoM(i)) + Intercept
    Xrs = Xo + (lx * (RatioCTAtoM(i) -
RatioCTAtoMmin) / (RatioCTAtoMmax -
RatioCTAtoMmin))
    Yrs = Yo + ly - (ly * (DPEst(i) - ReciprocalTDPnMin)
/ (ReciprocalTDPnMax - ReciprocalTDPnMin))
    CALL PlotGraph(Xrs, Yrs, i, 9)
  NEXT i
END SUB

SUB PlotRegressionLineOLD
  LinearRegressionOLD
  FOR i = Mark(1) TO Mark(2)
    DPEst(i) = (SlopeOLD * RatioCTAtoMOLD(i)) +
InterceptOLD
    Xrs = Xo + (lx * (RatioCTAtoMOLD(i) -
RatioCTAtoMmin) / (RatioCTAtoMmax -
RatioCTAtoMmin))
    Yrs = Yo + ly - (ly * (DPEst(i) - ReciprocalTDPnMin)
/ (ReciprocalTDPnMax - ReciprocalTDPnMin))
    CALL PlotGraphI(Xrs, Yrs, i, 4)
  NEXT i
END SUB

SUB PlotSymb (Xs, Ys)
    IF Xs >= Xo AND Xs <= Xo + lx AND Ys >= Yo
AND Ys <= Yo + ly THEN
        CALL DLine(Xs - 3, Ys - 3, Xs + 3, Ys + 3, 11)
        CALL DLine(Xs + 3, Ys - 3, Xs - 3, Ys + 3, 11)
    END IF
END SUB
```

Copyright by Mahidol University

```
SUB PlotSymbOLD (Xs, Ys)
    IF Xs >= Xo AND Xs <= Xo + lx AND Ys >= Yo
AND Ys <= Yo + ly THEN
        CALL DLine(Xs - 2, Ys - 2, Xs + 2, Ys + 2, 13)
        CALL DLine(Xs + 2, Ys - 2, Xs - 2, Ys + 2, 13)
    END IF
END SUB

SUB Position (row, col, Text$, Tcolor)
  COLOR Tcolor
  LOCATE row, col
  PRINT Text$,
END SUB

SUB PrintDATA (Anymore)
    COLOR 14
    FOR i = 1 TO 16
      LOCATE 8 + i, 13
      PRINT i + (16 * fAnymore)
    NEXT i
    COLOR 9
    FOR i = 1 TO 16
      LOCATE 8 + i, 21
      IF i + (16 * Anymore) <= TotalCount THEN
      PRINT USING "##.####^^^^"; SamplingTime(i +
(16 * Anymore))
      ELSE
      LOCATE 8 + i, 21
      PRINT "          "
      END IF
    NEXT i
    COLOR 12
    FOR i = 1 TO 16
      LOCATE 8 + i, 39
      IF i + (16 * Anymore) <= TotalCount THEN
      PRINT USING "## ####^^^^"; RatioCTAtoM(i +
(16 * Anymore))
      ELSE
      LOCATE 8 + i, 39
      PRINT "          "
      END IF
    NEXT i
    COLOR 13
    FOR i = 1 TO 16
      LOCATE 8 + i, 56
      IF i + (16 * Anymore) <= TotalCount THEN
      PRINT USING "##.####^^^^"; TDPn(i + (16 *
Anymore))
      ELSE
      LOCATE 8 + i, 56
      PRINT "          "
      END IF
    NEXT i
END SUB

SUB PrintDataTable
    GroupCount = INT(TotalCount / 16)
    Group = GroupCount + 1
    count = 0
        DO
        TableBox
        FunckeyForTable1
        PrintDATA (count)
        TableMenu (count)
        SELECT CASE SpecialKBD$
            CASE CHR$(0) + "Q"
                count = count + 1
                IF count > Group THEN count = Group
            CASE CHR$(0) + "I"
                count = count - 1
                IF count < 0 THEN count = 0
            END SELECT
        LOOP UNTIL SpecialKBD$ = CHR$(27)
        EraseScreen
END SUB

SUB PrintParameters
    ParametersBOX
    i = 12: j = 28: jk = 42: kk = 52
    COLOR 11: LOCATE i - 1, j - 2: PRINT "Parameter
and condition reaction"
    COLOR 12
    LOCATE i + 1, j: PRINT "Temperature"
    LOCATE i + 2, j: PRINT "[" + UCASE$(MoName$)
+ "]"
    LOCATE i + 3, j: PRINT "[" + UCASE$(IoName$) +
"]"
    LOCATE i + 4, j: PRINT "[" + UCASE$(SoName$) +
"]" + " conc. range.."
    LOCATE i + 5, j + 2: PRINT "from"
    LOCATE i + 6, j + 2: PRINT "to"
    LOCATE i + 7, j: PRINT "Sampling at"
    COLOR 14
    LOCATE i + 1, jk + 2: PRINT USING "###.##";
Temp
    LOCATE i + 2, jk: PRINT USING "##.#####", Mo
    LOCATE i + 3, jk: PRINT USING "##.#####"; Io
    LOCATE i + 5, jk: PRINT USING "## #####",
CTAo1
    LOCATE i + 6, jk: PRINT USING "##.#####";
CTAo2
    LOCATE i + 7, jk + 2: PRINT USING "### ##";
PercentSampling
    COLOR 9
    LOCATE i + 1, kk: PRINT " øC  "
    LOCATE i + 2, kk  PRINT "mol/l"
    LOCATE i + 3, kk: PRINT "mol/l"
    LOCATE i + 5, kk: PRINT "mol/l"
    LOCATE i + 6, kk: PRINT "mol/l"
    LOCATE i + 7, kk: PRINT "%conv"
END SUB

SUB PrintRegressResult
    i = 12: j = 18: k = 50
    NP = ABS(Mark(2) - Mark(1)) + 1
    CALL Center(i - 1, "Linear Regression Results", 15)
    CALL Position(i + 1, j, "Number of data point on curve
", 12)
    CALL Position(i + 1, k, STR$(NP), 14)
    CALL Position(i + 2, j, "Slope (fit on curve ,Cs(Est.)):",
12)
    CALL Position(i + 2, k, STR$(Slope), 14)
    CALL Position(i + 3, j, "Cs (Ture Value =Ktr/Kp)
:", 12)
    s = Ktr / Kp
    CALL Position(i + 3, k, STR$(s), 14)
    CALL Position(i + 4, j, "Y-Intercept (Yo)          :",
12)
    CALL Position(i + 4, k, STR$(Intercept), 14)
    CALL Position(i + 5, j, "Standard Error            :",
12)
    CALL Position(i + 5, k, STR$(Syx), 14)
    CALL Position(i + 6, j, "Correlation Coefficient (R)
", 12)
    CALL Position(i + 6, k, STR$(eR), 14)
    CALL Position(i + 7, j, "Coefficient
Determination(R^2):", 12)
    CALL Position(i + 7, k, STR$(eR ^ 2), 14)
END SUB

SUB PrintRegressResultOLD
    i = 12: j = 18: k = 50
    NP = ABS(Mark(2) - Mark(1)) + 1
```

Copyright by Mahidol University

```
      CALL Center(i - 1, "Linear Regression Results", 15)
      CALL Position(i + 1, j, "Number of data point on curve
", 12)
      CALL Position(i + 1, k, STR$(NP), 14)
      CALL Position(i + 2, j, "Slope (fit on curve .Cs(Est )) ",
12)
      CALL Position(i + 2, k, STR$(SlopeOLD), 14)
      CALL Position(i + 3, j, "Cs (Ture Value =Ktr/Kp)
.", 12)
      s = Ktr / Kp
      CALL Position(i + 3, k, STR$(s), 14)
      CALL Position(i + 4, j, "Y-Intercept (Yo)          ",
12)
      CALL Position(i + 4, k, STR$(InterceptOLD), 14)
      CALL Position(i + 5, j, "Standard Error            ",
12)
      CALL Position(i + 5, k, STR$(SyxOLD), 14)
      CALL Position(i + 6, j, "Correlation Coefficient  (R)
", 12)
      CALL Position(i + 6, k, STR$(eROLD), 14)
      CALL Position(i + 7, j, "Coefficient
Determination(R^2) ", 12)
      CALL Position(i + 7, k, STR$(eROLD ^ 2), 14)

END SUB

SUB ReadTransferFile
REM Read simulation file for Calculation routine
   count = 0
   OPEN "T-R-DP.MMR" FOR INPUT AS #10
      DO UNTIL EOF(10)
      count = count + 1
      INPUT #10, SamplingTime(count),
RatioCTAtoM(count), TDPn(count)
      LOOP
      TotalCount = count
   CLOSE #10
  'Routine make array value for old Mayo plot
   CalArrayOldMaYo
END SUB

SUB RegressionFromTable
   DO
      FunckeyForTable2
      count = 1
      DO
       CALL TableMenu1(count, 12)
      DO
       KBD$ = INKEY$
      LOOP UNTIL KBD$ <> ""
      SELECT CASE KBD$
          CASE CHR$(0) + "K"  'left
             count = count - 1
             IF count < 1 THEN count = 1
             CALL TableMenu1(count + 1, 15)
          CASE CHR$(0) + "M"  'right
             count = count + 1
             IF count > 4 THEN count = 4
             CALL TableMenu1(count - 1, 15)
          CASE CHR$(13)  'right
             inpcount = count
           SELECT CASE inpcount
              CASE 1
                    COLOR 15
                    LOCATE 4, 4: INPUT "First point
No.  ", Firstpoint
                    IF Firstpoint = 0 THEN Firstpoint = 1
                    IF Firstpoint > TotalCount THEN
Firstpoint = TotalCount
                    IF Firstpoint = Mark(2) THEN
Firstpoint = 1
                    Mark(1) = Firstpoint
                    LOCATE 4, 4: PRINT "
"
              CASE 2
                    COLOR 15
                    LOCATE 4, 4: INPUT "Second point
No   ", Secondpoint
                    IF Secondpoint = 0 THEN
Secondpoint = TotalCount
                    IF Secondpoint > TotalCount THEN
Secondpoint = TotalCount
                    IF Secondpoint = Mark(1) THEN
Secondpoint = 50
                    Mark(2) = Secondpoint
                    LOCATE 4, 4: PRINT "
"
              CASE 3
                 LinearRegression
                 ParametersBOX
                 PrintRegressResult
              CASE 4
                 EraseScreen
           END SELECT
      END SELECT
      LOOP UNTIL KBD$ = CHR$(13) OR KBD$ =
CHR$(27)
   LOOP UNTIL KBD$ = CHR$(27)
END SUB

SUB Right (row, Text$, Tcolor)
   COLOR Tcolor
   LOCATE row, 80 - LEN(Text$)
   PRINT Text$,
END SUB

SUB SaveSim

END SUB

SUB SelectLineColor
END SUB

SUB SelectMarker
END SUB

SUB SelectPoint1
      count = Mark(1)
      DO
          CALL ArrowMenu(count, 14)
            DO
              KBD$ = INKEY$
            LOOP UNTIL KBD$ <> ""
            SELECT CASE KBD$
                CASE CHR$(0) + "K"  'left
                   count = count - 1
                   IF count < Mark(1) THEN count =
Mark(1)
                   CALL ArrowMenu(count + 1, 11)
                   Xs = Xspixel(count + 1)
                   Ys = Yspixel(count + 1)
                   CALL DLine(Xs, Ys, Xs, Ys - 15, 0)
                   CIRCLE (Xs, Ys - 20), 5, 0
                CASE CHR$(0) + "M"  'right
                   count = count + 1
                   IF count > Mark(2) THEN count =
Mark(2)
                   CALL ArrowMenu(count - 1, 11)
                   Xs = Xspixel(count - 1)
                   Ys = Yspixel(count - 1)
                   CALL DLine(Xs, Ys, Xs, Ys - 15, 0)
                   CIRCLE (Xs, Ys - 20), 5, 0
           END SELECT
      LOOP UNTIL KBD$ = CHR$(13)
```

Copyright by Mahidol University

```
    Mark(1) = count
END SUB

SUB SelectPoint2
    count = Mark(2)
    DO
        CALL ArrowMenu(count, 14)
        DO
            KBD$ = INKEY$
        LOOP UNTIL KBD$ <> ""
        SELECT CASE KBD$
            CASE CHR$(0) + "K"  'left
                count = count - 1
                IF count < Mark(1) THEN count =
Mark(1)
                CALL ArrowMenu(count + 1, 11)
                Xs = Xspixel(count + 1)
                Ys = Yspixel(count + 1)
                CALL DLine(Xs, Ys, Xs, Ys - 15, 0)
                CIRCLE (Xs, Ys - 20), 5, 0
            CASE CHR$(0) + "M"  'right
                count = count + 1
                IF count > Mark(2) THEN count =
Mark(2)
                CALL ArrowMenu(count - 1, 11)
                Xs = Xspixel(count - 1)
                Ys = Yspixel(count - 1)
                CALL DLine(Xs, Ys, Xs, Ys - 15, 0)
                CIRCLE (Xs, Ys - 20), 5, 0
        END SELECT
    LOOP UNTIL KBD$ = CHR$(13)
    Mark(2) = count
END SUB

SUB TableBox
    X = 11
    CALL Center(6,"------------------------------", X)
    CALL Center(7,"                              ",X)
    CALL Center(8,"------------------------------", X)
    COLOR 14
    LOCATE 7, 14  PRINT "No."
    COLOR 9
    LOCATE 7, 19  PRINT "Sampling time,min"
    COLOR 12
    LOCATE 7, 41  PRINT "[CTA]/[M]"
    COLOR 13
    LOCATE 7, 61  PRINT "DP"
    FOR i = 9 TO 24
    CALL Center(i, "                              ",X)
    NEXT i
    CALL Center(25,"------------------------------", X)
    CALL Center(26,"                              ",X)
    CALL Center(27,"------------------------------", X)
END SUB

SUB TableMenu (Anymore)
        COLOR 14
        LOCATE 26, 14
        PRINT USING " View graph with data from
number ### to", Mark(1);
        PRINT USING " ###"; Mark(2)
        DO
          KBD$ = INKEY$
        LOOP UNTIL KBD$ <> ""
        SELECT CASE KBD$
            CASE CHR$(0) + "D"
                RegressionFromTable
        END SELECT
SpecialKBD$ = KBD$
'LOOP UNTIL SpecialKBD$ = CHR$(0) + "Q" OR
SpecialKBD$ = CHR$(0) + "I" OR SpecialKBD$ =
CHR$(13) OR SpecialKBD$ = CHR$(27)
END SUB

SUB TableMenu1 (Xcount, owncolor)
    SELECT CASE Xcount
        CASE 1
            CALL Position(2, 4, Table$(1), owncolor)
        CASE 2
            CALL Position(2, 17, Table$(2), owncolor)
        CASE 3
            CALL Position(2, 31, Table$(3), owncolor)
        CASE 4
            CALL Position(2, 50, Menu$(4), owncolor)
    END SELECT

END SUB

SUB ViewGraph
  DO
    EraseScreen
    PlotModifyMayo
    FunckeyForView
    LOCATE 4, 67: COLOR 13: PRINT "["; Mark(1),
"  ", Mark(2), "]"
    count = 1
    DO
     CALL ViewGraphMenu(count, 12)
    DO
     KBD$ = INKEY$
    LOOP UNTIL KBD$ <> ""
    SELECT CASE KBD$
        CASE CHR$(0) + "K"  'left
            count = count - 1
            IF count < 1 THEN count = 1
            CALL ViewGraphMenu(count + 1, 15)
        CASE CHR$(0) + "M"  'right
            count = count + 1
            IF count > 8 THEN count = 8
            CALL ViewGraphMenu(count - 1, 15)
        CASE CHR$(13)
          inpcount = count
    END SELECT
    LOOP UNTIL KBD$ = CHR$(27) OR KBD$ =
CHR$(13)
    IF KBD$ = CHR$(27) THEN EXIT DO
    SELECT CASE inpcount
        CASE 1
            COLOR 15
            LOCATE 4, 4: PRINT "Marker (Y/N)"
            DO
                MK$ = UCASE$(INPUT$(1))
            LOOP UNTIL MK$ = "Y" OR MK$ = "N"
            LOCATE 4, 4: PRINT "                "
        CASE 2
            COLOR 15
            LOCATE 4, 4: PRINT "Line (Y/N)"
            DO
                LC$ = UCASE$(INPUT$(1))
            LOOP UNTIL LC$ = "Y" OR LC$ = "N"
            LOCATE 4, 4: PRINT "                "
        CASE 3
            COLOR 15
            LOCATE 4, 4: PRINT "Plot Regression line
(Y/N)"
            DO
                RL$ = UCASE$(INPUT$(1))
            LOOP UNTIL RL$ = "Y" OR RL$ = "N"
            LOCATE 4, 4: PRINT "                "
        CASE 4
            COLOR 15
            LOCATE 4, 4: PRINT "You want to reset
point? (Y/N)"
            DO
```

Copyright by Mahidol University

```
                Reset$ = UCASE$(INPUT$(1))
            LOOP UNTIL Reset$ = "Y" OR Reset$ =
"N"
            IF Reset$ = "Y" THEN Mark(1) = 1:
Mark(2) = TotalCount
            LOCATE 4, 4: PRINT "                    "
        CASE 5
            SelectPoint1
        CASE 6
            SelectPoint2
        CASE 7
            COLOR 15
            LOCATE 4, 4: PRINT "Print regression
result (Y/N)"
            DO
                Results$ = UCASE$(INPUT$(1))
            LOOP UNTIL Results$ = "Y" OR Results$ =
"N"
            LOCATE 4, 4: PRINT "                    "
        CASE 8
            COLOR 15
            LOCATE 4, 4: PRINT "Do you want to
compare between Moyo VS Modify Mayo? (Y/N)"
            DO
                CP$ = UCASE$(INPUT$(1))
            LOOP UNTIL CP$ = "Y" OR CP$ = "N"
            LOCATE 4, 4: PRINT "
"

    END SELECT
    EraseScreen
  LOOP UNTIL KBD$ = CHR$(27)
  EraseScreen
END SUB

SUB ViewGraphMenu (Xcount, owncolor)
    SELECT CASE Xcount
        CASE 1
            CALL Position(2, 4, ViewG$(1), owncolor)
        CASE 2
            CALL Position(2, 12, ViewG$(2), owncolor)
        CASE 3
            CALL Position(2, 18, ViewG$(3), owncolor)
        CASE 4
            CALL Position(2, 27, ViewG$(4), owncolor)
        CASE 5
            CALL Position(2, 34, ViewG$(5), owncolor)
        CASE 6
            CALL Position(2, 41, ViewG$(6), owncolor)
        CASE 7
            CALL Position(2, 49, ViewG$(7), owncolor)
        CASE 8
            CALL Position(2, 57, ViewG$(8), owncolor)
    END SELECT

END SUB

FUNCTION Xspixel (Xval)
    Xspixel = Xo + (lx * (RatioCTAtoM(Xval) -
RatioCTAtoMmin) / (RatioCTAtoMmax -
RatioCTAtoMmin))
END FUNCTION

FUNCTION XspixelOLD (Xval)
    XspixelOLD = Xo + (lx * (RatioCTAtoMOLD(Xval) -
RatioCTAtoMmin) / (RatioCTAtoMmax -
RatioCTAtoMmin))
END FUNCTION

FUNCTION Yspixel (Yval)
    Intermediate = 1 / (TDPn(Yval) - 1)
    Yspixel = Yo + ly - (ly * (Intermediate -
ReciprocalTDPnMin) / (ReciprocalTDPnMax -
ReciprocalTDPnMin))
END FUNCTION
```

9) Programmes for editing rate coefficients (ED.BAS)

```
DECLARE SUB center (text$)
DECLARE SUB SentRateBackToMian ()
DECLARE SUB LoadParameter ()
DECLARE SUB LoadInitiatorFileArhenius ()
DECLARE SUB LoadRateFile ()
DECLARE SUB LoadMonomerFileArhenius ()
DECLARE SUB LoadCTransferFileArhenius ()
DECLARE SUB LoadMonomerFileCustom ()
DECLARE SUB LoadInitiatorFileCustom ()
DECLARE SUB LoadCtransferFileCustom ()
DECLARE SUB ScanFile ()
DECLARE SUB OpenMonomerFileCustom ()
DECLARE SUB OpenInitiatorFileCustom ()
DECLARE SUB OpenCtransferFileCustom ()
DECLARE SUB OpenMonomerFileArhenius ()
DECLARE SUB OpenInitiatorFileArhenius ()
DECLARE SUB OpenCTransferFileArhenius ()
DECLARE SUB ReteiveFile ()
DECLARE SUB SelectType ()
DECLARE SUB SelectInputFile ()
DECLARE SUB SaveFile ()
DECLARE SUB InputText (xcount!)
DECLARE SUB FileManage ()
DECLARE SUB PrintSubBar1 ()
DECLARE SUB EditRateSubMenu1 (xcount!, Fcolor!,
Bcolor!)
DECLARE SUB EditSelecFile (xcount!, Fcolor!, Bcolor!)
DECLARE FUNCTION CalK! (a!, b!, T!)
DECLARE SUB PaintT (Xposit!, Yposit!, ColLength!,
Fcolor!, Bcolor!, text$)
DECLARE SUB DBox1 (fc!, bc!)
DECLARE SUB DBox2 (fc!, bc!)
DECLARE SUB DrawB ()
DECLARE SUB PrintText (Xposit!, Yposit!, Fcolor!,
Bcolor!, text$)
DECLARE SUB DBox3 (fc!, bc!)
DECLARE SUB PrintBar1 ()
DECLARE SUB PrintBar2 ()
DECLARE SUB Bar2 ()
DECLARE SUB Bar1 ()
DECLARE SUB EditRateMenu1 (xcount!, Fcolor!,
Bcolor!)
DECLARE SUB EditRateMenu2 (xcount!, Fcolor!,
Bcolor!)
DECLARE SUB Bar3MoC ()
DECLARE SUB Bar3InC ()
DECLARE SUB Bar3CTC ()
DECLARE SUB Bar3MoA ()
DECLARE SUB Bar3InA ()
DECLARE SUB Bar3CTA ()
DECLARE SUB ClearBox3 ()
DECLARE SUB EditKMoA (xcount!, Fcolor!, Bcolor!)
DECLARE SUB EditKInA (xcount!, Fcolor!, Bcolor!)
DECLARE SUB EditKCTA (xcount!, Fcolor!, Bcolor!)
DECLARE SUB PrintBar3Mo ()
DECLARE SUB PrintBar3In ()
DECLARE SUB PrintBar3CT ()
DECLARE SUB Kcomp (ycount!)
DECLARE SUB PrintBar3Kcomp ()
DECLARE SUB EditKcomp (zcount!, xcount!, Fcolor!,
Bcolor!)
```

Copyright by Chulalongkorn University

```
DECLARE SUB KcompC (xcount!)
DECLARE SUB InputKparameter (zcount!, xcount!,
Fcolor!, Bcolor!)
DECLARE FUNCTION InputKcustom! ()
DECLARE FUNCTION InputAarhenius! ()
DECLARE FUNCTION InputBarhenius! ()
DECLARE FUNCTION InputTarhenius! ()
COMMON SHARED ArheCount, MICCount, SelectCount,
Temperature
COMMON SHARED aKp, bKp, tKp, aKtd, bKtd, tKtd,
aKtc, bKtc, tKtc, aKtrm, bKtrm, tKtrm
COMMON SHARED aKi, bKi, tKi, aKd, bKd, tKd, af, bf,
tf, aKtci, bKtci, tKtci
COMMON SHARED aKsr, bKsr, tKsr, aKtr, bKtr, tKtr
COMMON SHARED Kp, Ktd, Ktc, Ktrm, Ki, Kd, f, Ktci,
Ksr, Ktr, KCinput, KAinput
COMMON SHARED CustomName$, CustomPath$,
OpenRateName$
DIM SHARED EdMIC$(1 TO 4)
DIM SHARED EdRT$(1 TO 2)
DIM SHARED EdMo$(1 TO 4)
DIM SHARED EdIn$(1 TO 4)
DIM SHARED EdCT$(1 TO 2)
DIM SHARED Kparameter$(1 TO 3)
DIM SHARED FileCommand$(1 TO 3)
DIM SHARED SelectText$(1 TO 5)
    EdMIC$(1) = "Monomer"
    EdMIC$(2) = "Initiator"
    EdMIC$(3) = "Chain transfer agent"
    EdMIC$(4) = "File!"
    EdRT$(2) = "Custom define value"
    EdRT$(1) = "Arhenious Expression"
    EdMo$(1) = "Propagation,Kp"
    EdMo$(2) = "Disproportionat. terminat.,Ktd"
    EdMo$(3) = "Combination termination,Ktc"
    EdMo$(4) = "Chain transfer to monomer,Ktrm"
    EdIn$(1) = "Initiation,Ki"
    EdIn$(2) = "Dissociation,Kd"
    EdIn$(3) = "Initiator efficiency,f"
    EdIn$(4) = "Primary radical terminat.,Ktci"
    EdCT$(1) = "Prepare CTA radical,Ksr"
    EdCT$(2) = "Chain transfer to CTA,Ktr"
    Kparameter$(1) = "a="
    Kparameter$(2) = "b= "
    Kparameter$(3) = "T=;"
    FileCommand$(1) = "Retrive"
    FileCommand$(2) = "Save"
    FileCommand$(3) = "Quit!"
    SelectText$(1) = "Monomer"
    SelectText$(2) = "Initiator"
    SelectText$(3) = "Chain transfer agent"
    SelectText$(4) = "File name (maximum 8 character) : "
    SelectText$(5) = "Directory ( Drive:\Path name )  "
    CustomName$ = "no-name"
    CustomPath$ = "C:\simulate\package"
    ArheCount = 1
    MICCount = 1
    SelectCount = 1
    LoadParameter
    tKp = Temperature
    tKtd = tKp
    tKtc = tKp
    tKtrm = tKp
    tKi = tKp
    tKd = tKp
    tf = tKp
    tKtci = tKp
    tKsr = tKp
    tKtr = tKp
    COLOR 0, 6  CLS
    DrawB
    PrintBar1
    PrintBar2
  DO
    REM Dialog box 1
    CALL DBox1(15, 11)  Bar1  CALL DBox1(0, 11)
    REM Dialog box 2
    CALL DBox2(15, 5): Bar2: CALL DBox2(0, 5)
    REM Dialog box 3 shown and edit rate coefficient
    ClearBox3
    CALL DBox3(15, 1)
      IF ArheCount = 1 THEN
        COLOR 13
        Define$ = "Definition   k =
a.exp(b/(Temp.+273 3))"
        LOCATE 12, (80 - LEN(Define$)) / 2  PRINT
Define$
      ELSE
        COLOR 13
        Define$ = "Definition   none(k=:manual key)"
        LOCATE 12, (80 - LEN(Define$)) / 2  PRINT
Define$
      END IF
      REM select routine
      CALL center("Press up or down arrow key to move ..
and select by enter")
      SELECT CASE MICCount
        CASE 1
          IF ArheCount = 1 THEN Bar3MoA
          IF ArheCount = 2 THEN Bar3MoC
        CASE 2
          IF ArheCount = 1 THEN Bar3InA
          IF ArheCount = 2 THEN Bar3InC
        CASE 3
          IF ArheCount = 1 THEN Bar3CTA
          IF ArheCount = 2 THEN Bar3CTC
      END SELECT
      CALL DBox3(0, 1)
  LOOP
END

SUB Bar1
  count = MICCount
  DO
    CALL center("Press left or right arrow key. and TAB..to
move..and select by enter")
        CALL EditRateMenu1(count, 15, 3)
    DO
        kbd$ = INKEYS
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
      CASE CHR$(0) + "K"  'left
        count = count - 1
        IF count < 1 THEN count = 1
        CALL EditRateMenu1(count + 1, 0, 3)
      CASE CHR$(0) + "M"  'right
        count = count + 1
        IF count > 4 THEN count = 4
        CALL EditRateMenu1(count - 1, 0, 3)
      CASE CHR$(13)  'select
        MICCount = count
        IF MICCount = 4 THEN FileManage
    END SELECT
  LOOP UNTIL kbd$ = CHR$(9)
  MICCount = count
END SUB

SUB Bar2
  count = ArheCount
  DO
    CALL center("Press left or right arrow key..and TAB .to
move..and select by enter")
        CALL EditRateMenu2(count, 15, 5)
    DO
```

Copyright by Mahidol University

```
        kbd$ = INKEY$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "K"  'left
            count = count - 1
            IF count < 1 THEN count = 1
            CALL EditRateMenu2(count + 1, 0, 5)
        CASE CHR$(0) + "M"  'right
            count = count + 1
            IF count > 2 THEN count = 2
            CALL EditRateMenu2(count - 1, 0, 5)
    END SELECT
  LOOP UNTIL kbd$ = CHR$(9)
  ArheCount = count
END SUB

SUB Bar3CTA
  PrintBar3CT
  count = 1
  DO
        CALL EditKCTA(count, 14, 1)
    DO
        kbd$ = INKEY$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"  'left
            count = count - 1
            IF count < 1 THEN count = 1
            CALL EditKCTA(count + 1, 7, 1)
        CASE CHR$(0) + "P"  'right
            count = count + 1
            IF count > 2 THEN count = 2
            CALL EditKCTA(count - 1, 7, 1)
        CASE CHR$(13)  'select
            CALL center("Press up or down arrow key to
move .and select by enter .ESC to abort")
            CALL Kcomp(count)
    END SELECT
  LOOP UNTIL kbd$ = CHR$(9)
END SUB

SUB Bar3CTC
  PrintBar3CT
  count = 1
  DO
        CALL EditKCTA(count, 14, 1)
    DO
        kbd$ = INKEY$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"  'left
            count = count - 1
            IF count < 1 THEN count = 1
            CALL EditKCTA(count + 1, 7, 1)
        CASE CHR$(0) + "P"  'right
            count = count + 1
            IF count > 2 THEN count = 2
            CALL EditKCTA(count - 1, 7, 1)
        CASE CHR$(13)  'select
            CALL KcompC(count)
    END SELECT
  LOOP UNTIL kbd$ = CHR$(9)
END SUB

SUB Bar3InA
  PrintBar3In
  count = 1
  DO
        CALL EditKInA(count, 14, 1)
    DO
        kbd$ = INKEY$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"  'left
            count = count - 1
            IF count < 1 THEN count = 1
            CALL EditKInA(count + 1, 7, 1)
        CASE CHR$(0) + "P"  'right
            count = count + 1
            IF count > 4 THEN count = 4
            CALL EditKInA(count - 1, 7, 1)
        CASE CHR$(13)  'select
            CALL Kcomp(count)
    END SELECT
  LOOP UNTIL kbd$ = CHR$(9)
END SUB

SUB Bar3InC
  PrintBar3In
  count = 1
  DO
        CALL EditKInA(count, 14, 1)
    DO
        kbd$ = INKEY$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"  'left
            count = count - 1
            IF count < 1 THEN count = 1
            CALL EditKInA(count + 1, 7, 1)
        CASE CHR$(0) + "P"  'right
            count = count + 1
            IF count > 4 THEN count = 4
            CALL EditKInA(count - 1, 7, 1)
        CASE CHR$(13)  'select
            CALL KcompC(count)
    END SELECT
  LOOP UNTIL kbd$ = CHR$(9)
END SUB

SUB Bar3MoA
  PrintBar3Mo
  count = 1
  DO
  CALL center("Press up or down arrow key  or TAB. to
move..and select by enter")
        CALL EditKMoA(count, 14, 1)
    DO
        kbd$ = INKEY$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"  'left
            count = count - 1
            IF count < 1 THEN count = 1
            CALL EditKMoA(count + 1, 7, 1)
        CASE CHR$(0) + "P"  'right
            count = count + 1
            IF count > 4 THEN count = 4
            CALL EditKMoA(count - 1, 7, 1)
        CASE CHR$(13)  'select
            CALL Kcomp(count)
    END SELECT
  LOOP UNTIL kbd$ = CHR$(9)
END SUB

SUB Bar3MoC
  PrintBar3Mo
  count = 1
  DO
        CALL EditKMoA(count, 14, 1)
    DO
        kbd$ = INKEY$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
```

Copyright by Mahidol University

```
            CASE CHR$(0) + "H"  'left
               count = count - 1
               IF count < 1 THEN count = 1
               CALL EditKMoA(count + 1, 7, 1)
            CASE CHR$(0) + "P"  'right
               count = count + 1
               IF count > 4 THEN count = 4
               CALL EditKMoA(count - 1, 7, 1)
            CASE CHR$(13)  'select
               CALL KcompC(count)
         END SELECT
      LOOP UNTIL kbd$ = CHR$(9)
END SUB

FUNCTION CalK (a, b, T) STATIC
      CalK = a * EXP(b / (T + 273.3))
END FUNCTION

SUB center (text$)
   COLOR 15, 1
   LOCATE 22, 1: PRINT STRING$(79, " ")
   LOCATE 22, (80 - LEN(text$)) / 2: PRINT text$
END SUB

SUB ClearBox3
      x = 10
      y = 5
      l = 60
      FOR i = 0 TO 8
      CALL PaintT(x, 11 + i, l, 0, 1, " ")
      NEXT i
      CALL DBox3(15, 1)
END SUB

SUB DBox1 (fc, bc)
      x = 10
      y = 5
      l = 60
      CALL PaintT(x, y, l, fc, bc, "Ä")
      CALL PaintT(x, y + 2, l, fc, bc, "Ä")
      CALL PaintT(x, y + 1, 1, fc, bc, "³")
      CALL PaintT(x + l - 1, y + 1, 1, fc, bc, "³")
      CALL PaintT(x, y, 1, fc, bc, "Ú")
      CALL PaintT(x + l - 1, y, 1, fc, bc, "¿")
      CALL PaintT(x, y + 2, 1, fc, bc, "À")
      CALL PaintT(x + l - 1, y + 2, 1, fc, bc, "Ù")
END SUB

SUB DBox2 (fc, bc)
      x = 10
      y = 8
      l = 60
      CALL PaintT(x, y, l, fc, bc, "Ä")
      CALL PaintT(x, y + 2, l, fc, bc, "Ä")
      CALL PaintT(x, y + 1, 1, fc, bc, "³")
      CALL PaintT(x + l - 1, y + 1, 1, fc, bc, "³")
      CALL PaintT(x, y, 1, fc, bc, "Ú")
      CALL PaintT(x + l - 1, y, 1, fc, bc, "¿")
      CALL PaintT(x, y + 2, 1, fc, bc, "À")
      CALL PaintT(x + l - 1, y + 2, 1, fc, bc, "Ù")

END SUB

SUB DBox3 (fc, bc)
      x = 10
      y = 11
      l = 60
      CALL PaintT(x, y, l, fc, bc, "Ä")
      CALL PaintT(x, y + 8, l, fc, bc, "Ä")
      FOR i = 0 TO 7
      CALL PaintT(x, y + 1 + i, 1, fc, bc, "³")
      CALL PaintT(x + l - 1, y + 1 + i, 1, fc, bc, "³")
      NEXT
      FOR i = 0 TO 4
      CALL PaintT(x + l - 27, y + 3 + i, 1, fc, bc, "³")
      NEXT i
      CALL PaintT(x, y, 1, fc, bc, "Ú")
      CALL PaintT(x + l - 1, y, 1, fc, bc, "¿")
      CALL PaintT(x, y + 8, 1, fc, bc, "À")
      CALL PaintT(x + l - 1, y + 8, 1, fc, bc, "Ù")
      LOCATE y + 3, x + 1: PRINT STRING$(58, "Ä")
END SUB

SUB DBoxB
      x = 12
      y = 6
      l = 60
      fc = 1
      bc = 3
      FOR i = 0 TO 14
         CALL PaintT(x, y + i, l, fc, bc, CHR$(219))
      NEXT i
END SUB

SUB DrawB
      x = 10
      y = 5
      l = 60
      FOR i = 0 TO 19
         CALL PaintT(1, 3 + i, 80, 15, 0, CHR$(219))
      NEXT i
      FOR i = 0 TO 14
         CALL PaintT(x + 2, 6 + i, l, 11, 8, " ")
      NEXT i

      CALL PaintT(1, 1, 80, 0, 4, " ")
      d1$ = "EDIT RATE COEFFICIENT"
      LOCATE 1, (80 - LEN(d1$)) / 2: COLOR 15  PRINT
d1$
      CALL PaintT(1, 2, 80, 0, 15, " ")
      CALL PaintT(1, 22, 80, 0, 9, " ")
      CALL PaintT(1, 23, 80, 0, 3, " ")
      d2$ = "  ACTIONS  "
      d3$ = " [ARROW,TAB=MOVE] "
      d4$ = " [ENTER=SELECT] "
      d5$ = " [ESC=ABORT (or SUB-ACTIONs)] "
      LOCATE 23, 2
      COLOR 1: PRINT d2$;
      COLOR 0  PRINT d3$;
      COLOR 0: PRINT d4$;
      COLOR 0: PRINT d5$;
      FOR i = 0 TO 2
         CALL PaintT(x, y + i, l, 0, 11, " ")
      NEXT i
      CALL DBox1(0, 11)
      FOR i = 0 TO 2
         CALL PaintT(x, 8 + i, l, 0, 5, " ")
      NEXT i
      CALL DBox2(0, 5)
      FOR i = 0 TO 7
      CALL PaintT(x, 11 + i, l, 0, 1, " ")
      NEXT i
      CALL DBox3(0, 1)
END SUB

SUB EditKcomp (zcount, xcount, Fcolor, Bcolor)
COLOR Fcolor, Bcolor
REM Kp
   IF MICCount = 1 AND ArheCount = 1 AND zcount = 1
AND xcount = 1 THEN LOCATE 16, 11 + 59 - 26: PRINT
Kparameter$(1); : PRINT USING "##.##^^^^"; aKp
   IF MICCount = 1 AND ArheCount = 1 AND zcount = 1
AND xcount = 2 THEN LOCATE 17, 11 + 59 - 26: PRINT
Kparameter$(2); : PRINT USING "##.##^^^^"; bKp
```

Copyright by Mahidol University

```
   IF MICCount = 1 AND ArheCount = 1 AND zcount = 1
AND xcount = 3 THEN LOCATE 18, 11 + 59 - 26: PRINT
Kparameter$(3); · PRINT USING "##.##^^^^"; tKp
REM Ktd
   IF MICCount = 1 AND ArheCount = 1 AND zcount = 2
AND xcount = 1 THEN LOCATE 16, 11 + 59 - 26: PRINT
Kparameter$(1);  PRINT USING "## ##^^^^", aKtd
   IF MICCount = 1 AND ArheCount = 1 AND zcount = 2
AND xcount = 2 THEN LOCATE 17, 11 + 59 - 26  PRINT
Kparameter$(2); · PRINT USING "## ##^^^^", bKtd
   IF MICCount = 1 AND ArheCount = 1 AND zcount = 2
AND xcount = 3 THEN LOCATE 18, 11 + 59 - 26  PRINT
Kparameter$(3); . PRINT USING "##.##^^^^"; tKtd
REM Ktc
   IF MICCount = 1 AND ArheCount = 1 AND zcount = 3
AND xcount = 1 THEN LOCATE 16, 11 + 59 - 26. PRINT
Kparameter$(1), : PRINT USING "##.##^^^^"; aKtc
   IF MICCount = 1 AND ArheCount = 1 AND zcount = 3
AND xcount = 2 THEN LOCATE 17, 11 + 59 - 26· PRINT
Kparameter$(2); . PRINT USING "##.##^^^^"; bKtc
   IF MICCount = 1 AND ArheCount = 1 AND zcount = 3
AND xcount = 3 THEN LOCATE 18, 11 + 59 - 26  PRINT
Kparameter$(3); : PRINT USING "## ##^^^^"; tKtc
REM Ktrm
   IF MICCount = 1 AND ArheCount = 1 AND zcount = 4
AND xcount = 1 THEN LOCATE 16, 11 + 59 - 26. PRINT
Kparameter$(1);  PRINT USING "## ##^^^^"; aKtrm
   IF MICCount = 1 AND ArheCount = 1 AND zcount = 4
AND xcount = 2 THEN LOCATE 17, 11 + 59 - 26. PRINT
Kparameter$(2),  PRINT USING "##.##^^^^"; bKtrm
   IF MICCount = 1 AND ArheCount = 1 AND zcount = 4
AND xcount = 3 THEN LOCATE 18, 11 + 59 - 26. PRINT
Kparameter$(3),  PRINT USING "## ##^^^^"; tKtrm
REM K1
   IF MICCount = 2 AND ArheCount = 1 AND zcount = 1
AND xcount = 1 THEN LOCATE 16, 11 + 59 - 26  PRINT
Kparameter$(1), : PRINT USING "## ##^^^^", aK1
   IF MICCount = 2 AND ArheCount = 1 AND zcount = 1
AND xcount = 2 THEN LOCATE 17, 11 + 59 - 26· PRINT
Kparameter$(2); : PRINT USING "## ##^^^^", bK1
   IF MICCount = 2 AND ArheCount = 1 AND zcount = 1
AND xcount = 3 THEN LOCATE 18, 11 + 59 - 26  PRINT
Kparameter$(3); . PRINT USING "## ##^^^^", tK1
REM Kd
   IF MICCount = 2 AND ArheCount = 1 AND zcount = 2
AND xcount = 1 THEN LOCATE 16, 11 + 59 - 26  PRINT
Kparameter$(1),  PRINT USING "##.##^^^^", aKd
   IF MICCount = 2 AND ArheCount = 1 AND zcount = 2
AND xcount = 2 THEN LOCATE 17, 11 + 59 - 26: PRINT
Kparameter$(2),  PRINT USING "##.##^^^^"; bKd
   IF MICCount = 2 AND ArheCount = 1 AND zcount = 2
AND xcount = 3 THEN LOCATE 18, 11 + 59 - 26  PRINT
Kparameter$(3);  PRINT USING "##.##^^^^"; tKd
REM f
   IF MICCount = 2 AND ArheCount = 1 AND zcount = 3
AND xcount = 1 THEN LOCATE 16, 11 + 59 - 26. PRINT
Kparameter$(1); : PRINT USING "##.##^^^^"; af
   IF MICCount = 2 AND ArheCount = 1 AND zcount = 3
AND xcount = 2 THEN LOCATE 17, 11 + 59 - 26: PRINT
Kparameter$(2); : PRINT USING "##.##^^^^"; bf
   IF MICCount = 2 AND ArheCount = 1 AND zcount = 3
AND xcount = 3 THEN LOCATE 18, 11 + 59 - 26: PRINT
Kparameter$(3); · PRINT USING "##.##^^^^"; tf
REM Ktc1
   IF MICCount = 2 AND ArheCount = 1 AND zcount = 4
AND xcount = 1 THEN LOCATE 16, 11 + 59 - 26  PRINT
Kparameter$(1); . PRINT USING "## ##^^^^", aKtc1
   IF MICCount = 2 AND ArheCount = 1 AND zcount = 4
AND xcount = 2 THEN LOCATE 17, 11 + 59 - 26  PRINT
Kparameter$(2); . PRINT USING "##.##^^^^", bKtc1
   IF MICCount = 2 AND ArheCount = 1 AND zcount = 4
AND xcount = 3 THEN LOCATE 18, 11 + 59 - 26: PRINT
Kparameter$(3); : PRINT USING "##.##^^^^"; tKtc1
REM Ksr
   IF MICCount = 3 AND ArheCount = 1 AND zcount = 1
AND xcount = 1 THEN LOCATE 16, 11 + 59 - 26: PRINT
Kparameter$(1); · PRINT USING "##.##^^^^"; aKsr
   IF MICCount = 3 AND ArheCount = 1 AND zcount = 1
AND xcount = 2 THEN LOCATE 17, 11 + 59 - 26: PRINT
Kparameter$(2); : PRINT USING "##.##^^^^"; bKsr
   IF MICCount = 3 AND ArheCount = 1 AND zcount = 1
AND xcount = 3 THEN LOCATE 18, 11 + 59 - 26· PRINT
Kparameter$(3), . PRINT USING "##.##^^^^"; tKsr
REM Ktr
   IF MICCount = 3 AND ArheCount = 1 AND zcount = 2
AND xcount = 1 THEN LOCATE 16, 11 + 59 - 26· PRINT
Kparameter$(1); : PRINT USING "##.##^^^^"; aKtr
   IF MICCount = 3 AND ArheCount = 1 AND zcount = 2
AND xcount = 2 THEN LOCATE 17, 11 + 59 - 26: PRINT
Kparameter$(2); : PRINT USING "##.##^^^^"; bKtr
   IF MICCount = 3 AND ArheCount = 1 AND zcount = 2
AND xcount = 3 THEN LOCATE 18, 11 + 59 - 26· PRINT
Kparameter$(3); . PRINT USING "##.##^^^^"; tKtr

END SUB

SUB EditKCTA (xcount, Fcolor, Bcolor)
    IF xcount > 2 THEN xcount = 2
    IF xcount < 1 THEN xcount = 1
    COLOR Fcolor, Bcolor
    SELECT CASE xcount
        CASE 1
            IF ArheCount = 1 THEN Ksr = CalK(aKsr,
bKsr, tKsr)
            LOCATE 15, 12: PRINT EdCT$(1)
            LOCATE 13, 11· PRINT STRING$(58, " ")
            COLOR 12
            Define$ = "Temperature  " + STR$(tKsr) + "
(øC)  ·  Ksr =:" + STR$(Ksr) + "  l/mol.sec"
            LOCATE 13, (80 - LEN(Define$)) / 2.
PRINT Define$
        CASE 2
            IF ArheCount = 1 THEN Ktr = CalK(aKtr,
bKtr, tKtr)
            LOCATE 16, 12  PRINT EdCT$(2)
            LOCATE 13, 11· PRINT STRING$(58, " ")
            COLOR 12
            Define$ = "Temperature  " + STR$(tKtr) + "
(øC)  :  Ktr =:" + STR$(Ktr) + "  l/mol.sec"
            LOCATE 13, (80 - LEN(Define$)) / 2.
PRINT Define$
    END SELECT
END SUB

SUB EditKInA (xcount, Fcolor, Bcolor)
    IF xcount > 4 THEN xcount = 4
    IF xcount < 1 THEN xcount = 1
    COLOR Fcolor, Bcolor
    SELECT CASE xcount
        CASE 1
            IF ArheCount = 1 THEN K1 = CalK(aK1,
bKi, tKi)
            LOCATE 15, 12: PRINT EdIn$(1)
            LOCATE 13, 11  PRINT STRING$(58, " ")
            COLOR 12
            Define$ = "Temperature  " + STR$(tK1) + "
(øC)    K1 =:" + STR$(K1) + "  l/mol.sec"
            LOCATE 13, (80 - LEN(Define$)) / 2.
PRINT Define$
        CASE 2
            IF ArheCount = 1 THEN Kd = CalK(aKd,
bKd, tKd)
```

Copyright by Mahidol University

```
            LOCATE 16, 12: PRINT EdIn$(2)
            LOCATE 13, 11. PRINT STRING$(58, " ")
            COLOR 12
            Define$ = "Temperature  " + STR$(tKd) + "
(øC)    Kd =:" + STR$(Kd) + "  1/sec"
            LOCATE 13, (80 - LEN(Define$)) / 2
PRINT Define$
        CASE 3
            IF ArheCount = 1 THEN f = CalK(af, bf, tf)
            LOCATE 17, 12: PRINT EdIn$(3)
            LOCATE 13, 11· PRINT STRING$(58, " ")
            COLOR 12
            Define$ = "Temperature  " + STR$(tf) + "
(øC)  .  f =." + STR$(f) + "  unitless"
            LOCATE 13, (80 - LEN(Define$)) / 2:
PRINT Define$
        CASE 4
            IF ArheCount = 1 THEN Ktci = CalK(aKtci,
bKtci, tKtci)
            LOCATE 18, 12: PRINT EdIn$(4)
            LOCATE 13, 11: PRINT STRING$(58, " ")
            COLOR 12
            Define$ = "Temperature  " + STR$(tKtci) + "
(øC)  :  Ktci =:" + STR$(Ktci) + "  l/mol.sec"
            LOCATE 13, (80 - LEN(Define$)) / 2·
PRINT Define$
    END SELECT
END SUB

SUB EditKMoA (xcount, Fcolor, Bcolor)
    IF xcount > 4 THEN xcount = 4
    IF xcount < 1 THEN xcount = 1
    COLOR Fcolor, Bcolor
    SELECT CASE xcount
        CASE 1
            IF ArheCount = 1 THEN Kp = CalK(aKp,
bKp, tKp)
            LOCATE 15, 12· PRINT EdMo$(1)
            LOCATE 13, 11: PRINT STRING$(58, " ")
            COLOR 12
            Define$ = "Temperature  " + STR$(tKp) + "
(øC)  :  Kp =:" + STR$(Kp) + "  l/mol.sec"
            LOCATE 13, (80 - LEN(Define$)) / 2:
PRINT Define$
        CASE 2
            IF ArheCount = 1 THEN Ktd = CalK(aKtd,
bKtd, tKtd)
            LOCATE 16, 12: PRINT EdMo$(2)
            LOCATE 13, 11: PRINT STRING$(58, " ")
            COLOR 12
            Define$ = "Temperature  " + STR$(tKtd) + "
(øC)  :  Ktd =:" + STR$(Ktd) + "  l/mol sec"
            LOCATE 13, (80 - LEN(Define$)) / 2·
PRINT Define$
        CASE 3
            IF ArheCount = 1 THEN Ktc = CalK(aKtc,
bKtc, tKtc)
            LOCATE 17, 12: PRINT EdMo$(3)
            LOCATE 13, 11: PRINT STRING$(58, " ")
            COLOR 12
            Define$ = "Temperature  " + STR$(tKtc) + "
(øC)  :  Ktc =:" + STR$(Ktc) + "  l/mol.sec"
            LOCATE 13, (80 - LEN(Define$)) / 2·
PRINT Define$
        CASE 4
            IF ArheCount = 1 THEN Ktrm =
CalK(aKtrm, bKtrm, tKtrm)
            LOCATE 18, 12: PRINT EdMo$(4)
            LOCATE 13, 11: PRINT STRING$(58, " ")
            COLOR 12
            Define$ = "Temperature  " + STR$(tKtrm) +
" (øC)  .  Ktrm =:" + STR$(Ktrm) + "  l/mol.sec"
            LOCATE 13, (80 - LEN(Define$)) / 2·
PRINT Define$
    END SELECT
END SUB

SUB EditRateMenu1 (xcount, Fcolor, Bcolor)
    IF xcount > 4 THEN xcount = 4
    IF xcount < 1 THEN xcount = 1
    COLOR Fcolor, Bcolor
    SELECT CASE xcount
        CASE 1
            LOCATE 6, 13  PRINT EdMIC$(1)
        CASE 2
            LOCATE 6, 24  PRINT EdMIC$(2)
        CASE 3
            LOCATE 6, 38  PRINT EdMIC$(3)
        CASE 4
            LOCATE 6, 62: PRINT EdMIC$(4)
    END SELECT
END SUB

SUB EditRateMenu2 (xcount, Fcolor, Bcolor)
    IF xcount > 2 THEN xcount = 2
    IF xcount < 1 THEN xcount = 1
    COLOR Fcolor, Bcolor
    SELECT CASE xcount
        CASE 1
            LOCATE 9, 18. PRINT EdRT$(1)
        CASE 2
            LOCATE 9, 43· PRINT EdRT$(2)
    END SELECT
END SUB

SUB EditRateSubMenu1 (xcount, Fcolor, Bcolor)
    COLOR Fcolor, Bcolor
    SELECT CASE xcount
        CASE 1
            LOCATE 6, 13. PRINT FileCommand$(1)
        CASE 2
            LOCATE 6, 22  PRINT FileCommand$(2)
        CASE 3
            LOCATE 6, 29: PRINT FileCommand$(3)
    END SELECT
END SUB

SUB EditSelecFile (xcount, Fcolor, Bcolor)
    COLOR Fcolor, Bcolor
    SELECT CASE xcount
        CASE 1
            LOCATE 13, 12: PRINT SelectText$(1)
        CASE 2
            LOCATE 14, 12  PRINT SelectText$(2)
        CASE 3
            LOCATE 15, 12: PRINT SelectText$(3)
        CASE 4
            LOCATE 17, 12· PRINT SelectText$(4)
        CASE 5
            LOCATE 18, 12: PRINT SelectText$(5)
    END SELECT
END SUB

SUB FileManage
  PrintSubBar1
  CALL DBox1(15, 11)
  count = 1
  DO
  CALL center("Press left or right arrow key to move .and
select by enter  Escape to abort")
        CALL EditRateSubMenu1(count, 15, 3)
    DO
        kbd$ = INKEY$
    LOOP UNTIL kbd$ <> ""
```

Copyright by Mahidol University

```
    SELECT CASE kbd$
        CASE CHR$(0) + "K"  'left
            count = count - 1
            IF count < 1 THEN count = 1
            CALL EditRateSubMenu1(count + 1, 0, 3)
        CASE CHR$(0) + "M"  'right
            count = count + 1
            IF count > 3 THEN count = 3
            CALL EditRateSubMenu1(count - 1, 0, 3)
        CASE CHR$(13)  'select
            IF count = 1 THEN ReteiveFile
            IF count = 2 THEN SaveFile
            IF count = 3 THEN SentRateBackToMian
SCREEN 12: SCREEN 0, 0  END
    END SELECT
  LOOP UNTIL kbd$ = CHR$(27)
    LOCATE 6, 13: COLOR 0, 11: PRINT STRING$(55,
" ")
    PrintBar1
END SUB

FUNCTION InputAarhenius STATIC
    CALL center("Input! Numberic value")
    COLOR 27
    LOCATE 16, 11 + 59 - 14: PRINT ">>";
    COLOR 12
    LOCATE 16, 11 + 59 - 12: PRINT "__________";
    LOCATE 16, 11 + 59 - 11
    COLOR 11
    INPUT "", KAinput
    LOCATE 16, 11 + 59 - 14: PRINT "          "
    InputAarhenius = KAinput
END FUNCTION

FUNCTION InputBarhenius STATIC
    CALL center("Input! Numberic value")
    COLOR 27
    LOCATE 17, 11 + 59 - 14: PRINT ">>";
    COLOR 12
    LOCATE 17, 11 + 59 - 12  PRINT "__________";
    LOCATE 17, 11 + 59 - 11
    COLOR 11
    INPUT "", KAinput
    LOCATE 17, 11 + 59 - 14. PRINT "          "
    InputBarhenius = KAinput
END FUNCTION

FUNCTION InputKcustom STATIC
    COLOR 27
    LOCATE 15, 11 + 59 - 25. PRINT ">>>";
    COLOR 11
    INPUT "", KCinput
    InputKcustom = KCinput
      FOR i = 0 TO 3
      CALL PaintT(11 + 59 - 26, 15 + i, 25, 0, 1, " ")
      NEXT i
END FUNCTION

SUB InputKparameter (zcount, xcount, Fcolor, Bcolor)
REM Kp
  IF MICCount = 1 AND ArheCount = 1 AND zcount = 1
AND xcount = 1 THEN aKp = InputAarhenius
  IF MICCount = 1 AND ArheCount = 1 AND zcount = 1
AND xcount = 2 THEN bKp = InputBarhenius
  IF MICCount = 1 AND ArheCount = 1 AND zcount = 1
AND xcount = 3 THEN tKp = InputTarhenius
REM Ktd
  IF MICCount = 1 AND ArheCount = 1 AND zcount = 2
AND xcount = 1 THEN aKtd = InputAarhenius
  IF MICCount = 1 AND ArheCount = 1 AND zcount = 2
AND xcount = 2 THEN bKtd = InputBarhenius
  IF MICCount = 1 AND ArheCount = 1 AND zcount = 2
AND xcount = 3 THEN tKtd = InputTarhenius
REM Ktc
  IF MICCount = 1 AND ArheCount = 1 AND zcount = 3
AND xcount = 1 THEN aKtc = InputAarhenius
  IF MICCount = 1 AND ArheCount = 1 AND zcount = 3
AND xcount = 2 THEN bKtc = InputBarhenius
  IF MICCount = 1 AND ArheCount = 1 AND zcount = 3
AND xcount = 3 THEN tKtc = InputTarhenius
REM Ktrm
  IF MICCount = 1 AND ArheCount = 1 AND zcount = 4
AND xcount = 1 THEN aKtrm = InputAarhenius
  IF MICCount = 1 AND ArheCount = 1 AND zcount = 4
AND xcount = 2 THEN bKtrm = InputBarhenius
  IF MICCount = 1 AND ArheCount = 1 AND zcount = 4
AND xcount = 3 THEN tKtrm = InputTarhenius
REM Ki
  IF MICCount = 2 AND ArheCount = 1 AND zcount = 1
AND xcount = 1 THEN aKi = InputAarhenius
  IF MICCount = 2 AND ArheCount = 1 AND zcount = 1
AND xcount = 2 THEN bKi = InputBarhenius
  IF MICCount = 2 AND ArheCount = 1 AND zcount = 1
AND xcount = 3 THEN tKi = InputTarhenius
REM Kd
  IF MICCount = 2 AND ArheCount = 1 AND zcount = 2
AND xcount = 1 THEN aKd = InputAarhenius
  IF MICCount = 2 AND ArheCount = 1 AND zcount = 2
AND xcount = 2 THEN bKd = InputBarhenius
  IF MICCount = 2 AND ArheCount = 1 AND zcount = 2
AND xcount = 3 THEN tKd = InputTarhenius
REM f
  IF MICCount = 2 AND ArheCount = 1 AND zcount = 3
AND xcount = 1 THEN af = InputAarhenius
  IF MICCount = 2 AND ArheCount = 1 AND zcount = 3
AND xcount = 2 THEN bf = InputBarhenius
  IF MICCount = 2 AND ArheCount = 1 AND zcount = 3
AND xcount = 3 THEN tf = InputTarhenius
REM Ktci
  IF MICCount = 2 AND ArheCount = 1 AND zcount = 4
AND xcount = 1 THEN aKtci = InputAarhenius
  IF MICCount = 2 AND ArheCount = 1 AND zcount = 4
AND xcount = 2 THEN bKtci = InputBarhenius
  IF MICCount = 2 AND ArheCount = 1 AND zcount = 4
AND xcount = 3 THEN tKtci = InputTarhenius
REM Ksr
  IF MICCount = 3 AND ArheCount = 1 AND zcount = 1
AND xcount = 1 THEN aKsr = InputAarhenius
  IF MICCount = 3 AND ArheCount = 1 AND zcount = 1
AND xcount = 2 THEN bKsr = InputBarhenius
  IF MICCount = 3 AND ArheCount = 1 AND zcount = 1
AND xcount = 3 THEN tKsr = InputTarhenius
REM Ktr
  IF MICCount = 3 AND ArheCount = 1 AND zcount = 2
AND xcount = 1 THEN aKtr = InputAarhenius
  IF MICCount = 3 AND ArheCount = 1 AND zcount = 2
AND xcount = 2 THEN bKtr = InputBarhenius
  IF MICCount = 3 AND ArheCount = 1 AND zcount = 2
AND xcount = 3 THEN tKtr = InputTarhenius

END SUB

FUNCTION InputTarhenius STATIC
    CALL center("Input! Numberic value")
    COLOR 27
    LOCATE 18, 11 + 59 - 14  PRINT ">>",
    COLOR 12
    LOCATE 18, 11 + 59 - 12  PRINT "__________",
    LOCATE 18, 11 + 59 - 11
    COLOR 11
    INPUT "", KAinput
    LOCATE 18, 11 + 59 - 14: PRINT "          "
    InputTarhenius = KAinput
```

Copyright by Mahidol University

```
END FUNCTION

SUB InputText (xcount)
   CALL center("Input!  file name (Max.8)..and directory")

   COLOR 8, 15
   LOCATE 17, 12 + LEN(SelectText$(4)) + 1: PRINT
UCASE$(CustomName$)
   LOCATE 18, 12 + LEN(SelectText$(5)) + 1: PRINT
UCASE$(CustomPath$)
   COLOR 0
   DO
   IF xcount = 4 THEN LOCATE 17, 12 +
LEN(SelectText$(4)) + 10: INPUT ">>", CustomName$
   IF CustomName$ = "" THEN CustomName$ = "MTEC"
   LOOP UNTIL LEN(CustomName$) <= 8 AND
LEN(CustomName$) > 0
   COLOR 8
   LOCATE 17, 12 + LEN(SelectText$(4)) + 1: PRINT "
"
   LOCATE 17, 12 + LEN(SelectText$(4)) + 10: PRINT "
"
   LOCATE 17, 12 + LEN(SelectText$(4)) + 1: PRINT
UCASE$(CustomName$)
   COLOR 15, 1
   DO
   LOCATE 19, LEN(SelectText$(5)) + 10: PRINT "
"
   IF xcount = 5 THEN LOCATE 19, LEN(SelectText$(5))
+ 10: INPUT ".>> ", CustomPath$
   IF CustomPath$ = "" THEN CustomPath$ =
"C:\simulate\package"
   LOOP UNTIL CustomPath$ <> ""
   COLOR 15 + 15
   LOCATE 19, 12 + LEN(SelectText$(5)) + 1  PRINT "
"
   LOCATE 19, 12 + LEN(SelectText$(5)) + 10: PRINT "
"
   LOCATE 19, 12 + LEN(SelectText$(5)) + 1: PRINT
UCASE$(CustomPath$)
   COLOR 8, 7
   LOCATE 18, 12 + LEN(SelectText$(5)) + 1  PRINT "
"
   LOCATE 18, 12 + LEN(SelectText$(5)) + 10. PRINT "
"
   LOCATE 18, 12 + LEN(SelectText$(5)) + 1: PRINT
UCASE$(CustomPath$)
END SUB

SUB Kcomp (ycount)
  PrintBar3Kcomp
  count = 1
  DO
  CALL center("Press up or down arrow key to move..and
select by enter. ESC to abort")
  CALL EditKcomp(ycount, 1, 7, 1)
  CALL EditKcomp(ycount, 2, 7, 1)
  CALL EditKcomp(ycount, 3, 7, 1)

        CALL EditKcomp(ycount, count, 14, 1)
    DO
        kbd$ = INKEY$
    LOOP UNTIL kbd$ <> ""
    SELECT CASE kbd$
        CASE CHR$(0) + "H"  'left
           count = count - 1
           IF count < 1 THEN count = 1
           CALL EditKcomp(ycount, count + 1, 7, 1)
        CASE CHR$(0) + "P"  'right
           count = count + 1
           IF count > 3 THEN count = 3
           CALL EditKcomp(ycount, count - 1, 7, 1)
        CASE CHR$(13)  'select
           CALL InputKparameter(ycount, count, 11, 1)
    END SELECT
  LOOP UNTIL kbd$ = CHR$(27)
  FOR i = 0 TO 3
  CALL PaintT(11 + 59 - 26, 15 + i, 25, 0, 1, " ")
  NEXT i
END SUB

SUB KcompC (xcount)
  IF MICCount = 1 AND ArheCount = 2 AND xcount = 1
THEN Kp = InputKcustom
  IF MICCount = 1 AND ArheCount = 2 AND xcount = 2
THEN Ktd = InputKcustom
  IF MICCount = 1 AND ArheCount = 2 AND xcount = 3
THEN Ktc = InputKcustom
  IF MICCount = 1 AND ArheCount = 2 AND xcount = 4
THEN Ktrm = InputKcustom
  IF MICCount = 2 AND ArheCount = 2 AND xcount = 1
THEN Ki = InputKcustom
  IF MICCount = 2 AND ArheCount = 2 AND xcount = 2
THEN Kd = InputKcustom
  IF MICCount = 2 AND ArheCount = 2 AND xcount = 3
THEN f = InputKcustom
  IF MICCount = 2 AND ArheCount = 2 AND xcount = 4
THEN Ktci = InputKcustom
  IF MICCount = 3 AND ArheCount = 2 AND xcount = 1
THEN Ksr = InputKcustom
  IF MICCount = 3 AND ArheCount = 2 AND xcount = 2
THEN Ktr = InputKcustom
END SUB

SUB LoadCTransferFileArhenius
   OPEN OpenRateName$ FOR INPUT AS #3
   INPUT #3, aKsr
   INPUT #3, bKsr
   INPUT #3, aKtr
   INPUT #3, bKtr
   CLOSE #3


END SUB

SUB LoadCtransferFileCustom
   OPEN OpenRateName$ FOR INPUT AS #6
   INPUT #6, Ksr
   INPUT #6, Ktr
   CLOSE #6

END SUB

SUB LoadInitiatorFileArhenius
   OPEN OpenRateName$ FOR INPUT AS #2
   INPUT #2, aKi
   INPUT #2, bKi
   INPUT #2, aKd
   INPUT #2, bKd
   INPUT #2, af
   INPUT #2, bf
   INPUT #2, tf
   INPUT #2, aKtci
   INPUT #2, bKtci
   CLOSE #2
END SUB

SUB LoadInitiatorFileCustom
   OPEN OpenRateName$ FOR INPUT AS #5
   INPUT #5, Ki
   INPUT #5, Kd
   INPUT #5, f
   INPUT #5, Ktci
   CLOSE #5
```

Copyright by Mahidol University

```
END SUB

SUB LoadMonomerFileArhenius
   OPEN OpenRateName$ FOR INPUT AS #1
   INPUT #1, aKp
   INPUT #1, bKp
   INPUT #1, aKtd
   INPUT #1, bKtd
   INPUT #1, aKtc
   INPUT #1, bKtc
   INPUT #1, aKtrm
   INPUT #1, bKtrm
   CLOSE #1

END SUB

SUB LoadMonomerFileCustom
   OPEN OpenRateName$ FOR INPUT AS #4
   INPUT #4, Kp
   INPUT #4, Ktd
   INPUT #4, Ktc
   INPUT #4, Ktrm
   CLOSE #4

END SUB
SUB LoadParameter
  OPEN "SENTED.CTN" FOR INPUT AS #14
     INPUT #14, Temperature
  CLOSE #14
END SUB

SUB LoadRateFile
    IF ArheCount = 1 AND SelectCount = 1 THEN
LoadMonomerFileArhenius
    IF ArheCount = 1 AND SelectCount = 2 THEN
LoadInitiatorFileArhenius
    IF ArheCount = 1 AND SelectCount = 3 THEN
LoadCTransferFileArhenius
    IF ArheCount = 2 AND SelectCount = 1 THEN
LoadMonomerFileCustom
    IF ArheCount = 2 AND SelectCount = 2 THEN
LoadInitiatorFileCustom
    IF ArheCount = 2 AND SelectCount = 3 THEN
LoadCtransferFileCustom

  'OPEN OpenRateName$ FOR INPUT AS #1
END SUB

SUB OpenCTransferFileArhenius
   OPEN CustomPath$ + "\" + CustomName$ + " ARC"
FOR OUTPUT AS #3
   PRINT #3, aKsr
   PRINT #3, bKsr
   PRINT #3, aKtr
   PRINT #3, bKtr
   CLOSE #3

END SUB

SUB OpenCtransferFileCustom
   OPEN CustomPath$ + "\" + CustomName$ + " CTC"
FOR OUTPUT AS #6
   PRINT #6, Ksr
   PRINT #6, Ktr
   CLOSE #6
END SUB

SUB OpenInitiatorFileArhenius
   OPEN CustomPath$ + "\" + CustomName$ + " ARI"
FOR OUTPUT AS #2
   PRINT #2, aKi
   PRINT #2, bKi
   PRINT #2, aKd
   PRINT #2, bKd
   PRINT #2, af
   PRINT #2, bf
   PRINT #2, tf
   PRINT #2, aKtci
   PRINT #2, bKtci
   CLOSE #2
END SUB

SUB OpenInitiatorFileCustom
   OPEN CustomPath$ + "\" + CustomName$ + " CTI"
FOR OUTPUT AS #5
   PRINT #5, Ki
   PRINT #5, Kd
   PRINT #5, f
   PRINT #5, Ktci
   CLOSE #5
END SUB

SUB OpenMonomerFileArhenius
   OPEN CustomPath$ + "\" + CustomName$ + ".ARM"
FOR OUTPUT AS #1
   PRINT #1, aKp
   PRINT #1, bKp
   PRINT #1, aKtd
   PRINT #1, bKtd
   PRINT #1, aKtc
   PRINT #1, bKtc
   PRINT #1, aKtrm
   PRINT #1, bKtrm
   CLOSE #1
END SUB

SUB OpenMonomerFileCustom
   OPEN CustomPath$ + "\" + CustomName$ + " CTM"
FOR OUTPUT AS #4
   PRINT #4, Kp
   PRINT #4, Ktd
   PRINT #4, Ktc
   PRINT #4, Ktrm
   CLOSE #4
END SUB

SUB PauntT (Xposit, Yposit, ColLength, Fcolor, Bcolor,
text$)
    COLOR Fcolor, Bcolor
    LOCATE Yposit, Xposit  PRINT
STRING$(ColLength, text$)
END SUB

SUB PrintBar1
    x = 12
    y = 6
    ac = 0
    CALL PrintText(x + 1, y, ac, 11, EdMIC$(1))
    CALL PrintText(x + 2 + 10, y, ac, 11, EdMIC$(2))
    CALL PrintText(x + 2 + 24, y, ac, 11, EdMIC$(3))
    CALL PrintText(x + 2 + 48, y, ac, 11, EdMIC$(4))
END SUB

SUB PrintBar2
    x = 12
    y = 6
    ac = 0
    CALL PrintText(x + 6, y + 3, ac, 5, EdRT$(1))
    CALL PrintText(x + 31, y + 3, ac, 5, EdRT$(2))
END SUB

SUB PrintBar3CT
```

Copyright by Mahidol University

```
    COLOR 7, 1
    LOCATE 15, 12: PRINT EdCT$(1)
    LOCATE 16, 12: PRINT EdCT$(2)
END SUB

SUB PrintBar3In
    COLOR 7, 1
    LOCATE 15, 12  PRINT EdIn$(1)
    LOCATE 16, 12. PRINT EdIn$(2)
    LOCATE 17, 12  PRINT EdIn$(3)
    LOCATE 18, 12: PRINT EdIn$(4)
END SUB

SUB PrintBar3Kcomp
    COLOR 13
    LOCATE 15, 46: PRINT "Old Value"
    LOCATE 15, 58: PRINT "New Value"
    COLOR 7, 1
    LOCATE 16, 11 + 59 - 26: PRINT Kparameter$(1); :
PRINT USING "##.##^^^^"; Kparameter1
    LOCATE 17, 11 + 59 - 26: PRINT Kparameter$(2),
PRINT USING "##.##^^^^"; Kparameter2
    LOCATE 18, 11 + 59 - 26. PRINT Kparameter$(3), :
PRINT USING "##.##^^^^"; Kparameter3
END SUB

SUB PrintBar3Mo
    COLOR 7, 1
    LOCATE 15, 12: PRINT EdMo$(1)
    LOCATE 16, 12: PRINT EdMo$(2)
    LOCATE 17, 12: PRINT EdMo$(3)
    LOCATE 18, 12. PRINT EdMo$(4)
END SUB

SUB PrintSubBar1
    x = 12
    y = 6
    ac = 0
    LOCATE y, x + 1: COLOR ac, 11: PRINT
STRING$(55, " ")
    CALL PrintText(x + 1, y, ac, 11, FileCommand$(1))
    CALL PrintText(x + 2 + 8, y, ac, 11,
FileCommand$(2))
    CALL PrintText(x + 2 + 15, y, ac, 11,
FileCommand$(3))

END SUB

SUB PrintText (Xposit, Yposit, Fcolor, Bcolor, text$)
    COLOR Fcolor, Bcolor
    LOCATE Yposit, Xposit: PRINT text$
END SUB

SUB ReteiveFile
  CALL center("Press up or down arrow key to move..and
select by enter..Escape to abort")
  FOR i = 0 TO 6
    LOCATE 12 + i, 11  COLOR , 15: PRINT
STRING$(58, " ")
  NEXT
  Define$ = "Select type of reagent which you want to
reteive file ?"
  COLOR 0
  LOCATE 12, (80 - LEN(Define$)) / 2. PRINT Define$
  COLOR 1
  LOCATE 13, 12: COLOR 1. PRINT SelectText$(1)
  LOCATE 14, 12: COLOR 1: PRINT SelectText$(2)
  LOCATE 15, 12· COLOR 1: PRINT SelectText$(3)
  CALL DBox1(15, 11)
  PrintSubBar1
  count = 1
  DO
    CALL EditSelecFile(count, 11, 15)
  DO
    kbd$ = INKEY$
  LOOP UNTIL kbd$ <> ""
  SELECT CASE kbd$
    CASE CHR$(0) + "H"  'left
      count = count - 1
      IF count < 1 THEN count = 1
      CALL EditSelecFile(count + 1, 1, 15)
    CASE CHR$(0) + "P"  'right
      count = count + 1
      IF count > 3 THEN count = 3
      CALL EditSelecFile(count - 1, 1, 15)
    CASE CHR$(13)  'select
      SelectCount = count
      SelectInputFile
      ScanFile
      CALL center("Press up or down arrow key to
move. and select by enter  Escape to abort")
      CALL DBox1(15, 11)
      PrintSubBar1
  END SELECT
 LOOP UNTIL kbd$ = CHR$(27)
  x = 10
  y = 5
  l = 60
  FOR i = 0 TO 7
  CALL PaintT(x, 11 + i, l, 0, 1, " ")
  NEXT i
 CALL DBox3(0, 1)
END SUB

SUB SaveFile
    SelectType
    IF ArheCount = 1 AND SelectCount = 1 THEN
OpenMonomerFileArhenius
    IF ArheCount = 1 AND SelectCount = 2 THEN
OpenInitiatorFileArhenius
    IF ArheCount = 1 AND SelectCount = 3 THEN
OpenCTransferFileArhenius
    IF ArheCount = 2 AND SelectCount = 1 THEN
OpenMonomerFileCustom
    IF ArheCount = 2 AND SelectCount = 2 THEN
OpenInitiatorFileCustom
    IF ArheCount = 2 AND SelectCount = 3 THEN
OpenCtransferFileCustom

END SUB

SUB ScanFile
    ImageDrive$ = LEFT$(CustomPath$, 2)
    SHELL ImageDrive$
    IF ArheCount = 1 AND SelectCount = 1 THEN
SHELL "dir *.ARM /B/S/O" + " > " + "ROOT.SCN"
    IF ArheCount = 1 AND SelectCount = 2 THEN
SHELL "dir *.ARI /B/S/O" + " > " + "ROOT.SCN"
    IF ArheCount = 1 AND SelectCount = 3 THEN
SHELL "dir * ARC /B/S/O" + " > " + "ROOT.SCN"
    IF ArheCount = 2 AND SelectCount = 1 THEN
SHELL "dir * CTM /B/S/O" + " > " + "ROOT.SCN"
    IF ArheCount = 2 AND SelectCount = 2 THEN
SHELL "dir *.CTI /B/S/O" + " > " + "ROOT.SCN"
    IF ArheCount = 2 AND SelectCount = 3 THEN
SHELL "dir *.CTC /B/S/O" + " > " + "ROOT SCN"
    OPEN "Mdrive.DV" FOR OUTPUT AS #12
        PRINT #12, ImageDrive$
    CLOSE #12
    SHELL "MTF.EXE"
    SCREEN 12
    SCREEN 0, 0
    DrawB
    PrintBar1
```

Copyright by Mahidol University

```
      PrintBar2
      PrintSubBar1
   FOR i = 0 TO 6
      LOCATE 12 + i, 11: COLOR , 15: PRINT
STRING$(58, " ")
   NEXT
   Define$ = "Select type of reagent which you want to
retetve file ?"
   COLOR 6
   LOCATE 12, (80 - LEN(Define$)) / 2: PRINT Define$
   COLOR 1
   LOCATE 13, 12. COLOR 1: PRINT SelectText$(1)
   LOCATE 14, 12: COLOR 1: PRINT SelectText$(2)
   LOCATE 15, 12: COLOR 1: PRINT SelectText$(3)
      OPEN "Ldrive.DV" FOR INPUT AS #13
          INPUT #13, OCN$
      CLOSE #13
      IF OCN$ <> "" THEN
          OpenRateName$ = OCN$
          LoadRateFile
      END IF
END SUB

SUB SelectInputFile
   COLOR 0
   Define$ = "Input your reagent name ?"
   LOCATE 16, (80 - LEN(Define$)) / 2: PRINT Define$
   COLOR 1
   LOCATE 17, 12: COLOR 1: PRINT SelectText$(4)
   LOCATE 18, 12: COLOR 1: PRINT SelectText$(5)
   count = 5
   DO
          CALL EditSelecFile(count, 11, 15)
      DO
          kbd$ = INKEY$
      LOOP UNTIL kbd$ <> ""
      SELECT CASE kbd$
          CASE CHR$(0) + "H"  'left
              count = count - 1
              IF count < 4 THEN count = 4
              CALL EditSelecFile(count + 1, 1, 15)
          CASE CHR$(0) + "P"  'right
              count = count + 1
              IF count > 5 THEN count = 5
              CALL EditSelecFile(count - 1, 1, 15)
          CASE CHR$(13)  'select
              CALL InputText(count)
              CALL center("Now!  you have to press ESC
key to actions..or ARROW and Enter to edit")
      END SELECT
   LOOP UNTIL kbd$ = CHR$(27)
   FOR i = 0 TO 2
      LOCATE 16 + i, 11: COLOR , 15. PRINT
STRING$(58, " ")
   NEXT
END SUB

SUB SelectType
   FOR i = 0 TO 6
      LOCATE 12 + i, 11: COLOR , 15: PRINT
STRING$(58, " ")
   NEXT
   Define$ = "Select type of reagent which you want to
create file ?"
   COLOR 6
   LOCATE 12, (80 - LEN(Define$)) / 2: PRINT Define$
   COLOR 1
   LOCATE 13, 12: COLOR 1: PRINT SelectText$(1)
   LOCATE 14, 12: COLOR 1: PRINT SelectText$(2)
   LOCATE 15, 12: COLOR 1: PRINT SelectText$(3)
   count = 1
   DO
          CALL EditSelecFile(count, 11, 15)
      DO
          kbd$ = INKEY$
      LOOP UNTIL kbd$ <> ""
      SELECT CASE kbd$
          CASE CHR$(0) + "H"  'left
              count = count - 1
              IF count < 1 THEN count = 1
              CALL EditSelecFile(count + 1, 1, 15)
          CASE CHR$(0) + "P"  'right
              count = count + 1
              IF count > 3 THEN count = 3
              CALL EditSelecFile(count - 1, 1, 15)
          CASE CHR$(13)  'select
              SelectCount = count
              SelectInputFile
      END SELECT
   LOOP UNTIL kbd$ = CHR$(27)
      x = 10
      y = 5
      l = 60
      FOR i = 0 TO 7
      CALL PaintT(x, 11 + i, l, 0, 1, " ")
      NEXT i
     CALL DBox3(0, 1)
END SUB

SUB SentRateBackToMian
   CALL center("Press <Y> ..accept  or  Press
<N>...except")
   LOCATE 9, 11  COLOR , 15: PRINT STRING$(58, " ")
   FOR i = 0 TO 6
      LOCATE 12 + i, 11  COLOR , 15: PRINT
STRING$(58, " ")
   NEXT
   Q1$ = "Now! before exit EDIT Rate routine , please
confirm... "
   COLOR 0
   LOCATE 9, (80 - LEN(Q1$)) / 2  PRINT Q1$
DO
   Define1$ = "Transmit !  "
   Define2$ = " (edited values)"
   COLOR 0
   LOCATE 12, (80 - LEN(Define1$ + SelectText$(1) +
Define2$)) / 2: PRINT Define1$ + SelectText$(1) +
Define2$,
   LOCATE 14, (80 - LEN(Define1$ + SelectText$(2) +
Define2$)) / 2  PRINT Define1$ + SelectText$(2) +
Define2$;
   LOCATE 16, (80 - LEN(Define1$ + SelectText$(3) +
Define2$)) / 2: PRINT Define1$ + SelectText$(3) +
Define2$;
   COLOR 15
   LOCATE 12, (80 - LEN(Define1$ + SelectText$(1) +
Define2$)) / 2: PRINT Define1$ + SelectText$(1) +
Define2$;
   COLOR 16
   LOCATE 13, (80 - LEN("<Yes/No!>")) / 2: PRINT
"<Yes/No!>"
   DO
   MacceptText$ = UCASE$(INPUT$(1))
   LOOP UNTIL MacceptText$ = "Y" OR MacceptText$ =
"N"
   IF MacceptText$ = "Y" THEN
        COLOR 15
        LOCATE 13, (80 - LEN("   Yes   ")) / 2: PRINT "
Yes   "
   ELSE
        COLOR 15
        LOCATE 13, (80 - LEN("   No!   ")) / 2: PRINT "
No!   "
   END IF
```

©Copyright by Mahidol University

```
   COLOR 15
   LOCATE 14, (80 - LEN(Define1$ + SelectText$(2) +
Define2$)) / 2: PRINT Define1$ + SelectText$(2) +
Define2$,
   COLOR 16
   LOCATE 15, (80 - LEN("<Yes/No!>")) / 2: PRINT
"<Yes/No!>"
   DO
   IacceptText$ = UCASE$(INPUT$(1))
   LOOP UNTIL IacceptText$ = "Y" OR IacceptText$ =
"N"
   IF IacceptText$ = "Y" THEN
        COLOR 15
        LOCATE 15, (80 - LEN("   Yes   ")) / 2: PRINT "
Yes   "
   ELSE
        COLOR 15
        LOCATE 15, (80 - LEN("   No!   ")) / 2: PRINT "
No!   "
   END IF
   COLOR 15
   LOCATE 16, (80 - LEN(Define1$ + SelectText$(3) +
Define2$)) / 2: PRINT Define1$ + SelectText$(3) +
Define2$,
   COLOR 16
   LOCATE 17, (80 - LEN("<Yes/No!>")) / 2: PRINT
"<Yes/No!>"
   DO
   CacceptText$ = UCASE$(INPUT$(1))
   LOOP UNTIL CacceptText$ = "Y" OR CacceptText$ =
"N"
   IF CacceptText$ = "Y" THEN
        COLOR 15
        LOCATE 17, (80 - LEN("   Yes   ")) / 2: PRINT "
Yes   "
   ELSE
        COLOR 15
        LOCATE 17, (80 - LEN("   No!   ")) / 2: PRINT "
No!   "
   END IF
   DO
        LOCATE 18, (80 - LEN("<<<< You want do
change ! >>>>")) / 2
        COLOR 8. PRINT "<<<< You want do change !
>>>>"
        LOCATE 18, (80 - LEN("<<<< You want do
change ! >>>>")) / 2
        INPUT "<<<< You want do change ! >>>>",
QuitText$
   LOOP UNTIL UCASE$(QuitText$) = "N" OR
UCASE$(QuitText$) = "Y"
   LOCATE 18, (80 - LEN("                    ")) / 2
   PRINT "                    "
   IF UCASE$(QuitText$) = "N" THEN EXIT DO
LOOP
KILL "ACEDIT.AV"
KILL "SENTMIAN.CTN"
   OPEN "SENTMIAN.CTN" FOR OUTPUT AS #21
   SELECT CASE MacceptText$
     CASE "Y"
     IF ArheCount = 1 THEN Kp = CalK(aKp, bKp, tKp)
     IF ArheCount = 1 THEN Ktd = CalK(aKtd, bKtd,
tKtd)
     IF ArheCount = 1 THEN Ktc = CalK(aKtc, bKtc, tKtc)
     IF ArheCount = 1 THEN Ktrm = CalK(aKtrm, bKtrm,
tKtrm)
     IF ArheCount = 2 THEN Kp = Kp
     IF ArheCount = 2 THEN Ktd = Ktd
     IF ArheCount = 2 THEN Ktc = Ktc
     IF ArheCount = 2 THEN Ktrm = Ktrm
     PRINT #21, Kp
     PRINT #21, Ktd
     PRINT #21, Ktc
     PRINT #21, Ktrm
   END SELECT
   SELECT CASE IacceptText$
     CASE "Y"
     IF ArheCount = 1 THEN Ki = CalK(aKi, bKi, tKi)
     IF ArheCount = 1 THEN Kd = CalK(aKd, bKd, tKd)
     IF ArheCount = 1 THEN f = CalK(af, bf, tf)
     IF ArheCount = 1 THEN Ktci = CalK(aKtci, bKtci,
tKtci)
     IF ArheCount = 2 THEN Ki = Ki
     IF ArheCount = 2 THEN Kd = Kd
     IF ArheCount = 2 THEN f = f
     IF ArheCount = 2 THEN Ktci = Ktci
     PRINT #21, Ki
     PRINT #21, Kd
     PRINT #21, f
     PRINT #21, Ktci
   END SELECT
   SELECT CASE CacceptText$
     CASE "Y"
     IF ArheCount = 1 THEN Ksr = CalK(aKsr, bKsr, tKsr)
     IF ArheCount = 1 THEN Ktr = CalK(aKtr, bKtr, tKtr)
     IF ArheCount = 2 THEN Ksr = Ksr
     IF ArheCount = 2 THEN Ktr = Ktr
     PRINT #21, Ksr
     PRINT #21, Ktr
   END SELECT
   CLOSE #21
   OPEN "ACEDIT.AV" FOR OUTPUT AS #22
     PRINT #22, MacceptText$
     PRINT #22, IacceptText$
     PRINT #22, CacceptText$
   CLOSE #22
END SUB
```

10) Programmes for retriving previous simulation results (MTF.BAS)

```
DECLARE SUB DrawScreen ()
DECLARE SUB Hint ()
DECLARE SUB PaintPattern (Xposit!, Yposit!,
RowLength!, ColLength!, Fcolor!, Bcolor!, Text$)
DECLARE SUB FL (Anymore!)
DECLARE SUB Openfile ()
DECLARE SUB PrintFile (count)
COMMON SHARED TotalCount, Group
COMMON SHARED SpecialKBD$, Choice$, Drive$
CLS
DIM SHARED Soul$(1 TO 5000)
Openfile
DrawScreen
        GroupCount = INT(TotalCount / 16)
        Group = GroupCount + 1
        count = 0
        DO
        CALL PaintPattern(1, 4, 16, 79, 0, 0, " ")
        PrintFile (count)
        FL (count)
        SELECT CASE SpecialKBD$
            CASE CHR$(0) + "Q"
                count = count + 1
                IF count > Group THEN count = Group
            CASE CHR$(0) + "I"
                count = count - 1
                IF count < 0 THEN count = 0
            END SELECT
```

Copyright by Mahidol University

```
        LOOP UNTIL SpecialKBD$ = CHR$(13) AND
Choice$ <> "" OR SpecialKBD$ = CHR$(27)
    IF SpecialKBD$ = CHR$(13) THEN
        OPEN "Ldrive.DV" FOR OUTPUT AS #3
            PRINT #3, Choice$
        CLOSE #3
    ELSEIF SpecialKBD$ = CHR$(27) THEN
        OPEN "Ldrive DV" FOR OUTPUT AS #3
            PRINT #3, None$
        CLOSE #3
    END IF
    KILL "ROOT.SCN"

SUB DrawScreen
    LOCATE 1, 1: COLOR 0, 1: PRINT STRING$(80, "
")
    abc1$ = "FILE LISTING"
    LOCATE 1, (80 - LEN(abc1$)) / 2: COLOR 15:
PRINT abc1$
    LOCATE 2, 1: COLOR 0, 11: PRINT STRING$(80, "
")
    LOCATE 2, 5: COLOR 0, 11: PRINT "File Name"
    Hint
END SUB

SUB FL (Anymore)
        RowCount = 0
        DO
        CALL PaintPattern(1, 4 + RowCount, 0, 79, 10, 1,
" ")
        LOCATE 4 + RowCount, 5: COLOR 15: PRINT
Soul$(1 + RowCount + (16 * Anymore))
        IF Soul$(1 + RowCount + (16 * Anymore)) = ""
THEN
        LOCATE 4 + RowCount, 5: COLOR 15. PRINT
"NO FILE"
        END IF
        Choice$ = Soul$(1 + RowCount + (16 *
Anymore))
        DO
            kbd$ = INKEY$
        LOOP UNTIL kbd$ <> ""
        SELECT CASE kbd$
        CASE CHR$(0) + "P"
            RowCount = RowCount + 1
            IF RowCount > 15 THEN RowCount = 15
            IF RowCount > 0 AND RowCount <= 15
THEN
            CALL PaintPattern(1, 4 + RowCount - 1, 0,
79, 0, 0, " ")
            LOCATE 4 + RowCount - 1, 5: COLOR 10:
PRINT Soul$(1 + RowCount - 1 + (16 * Anymore))
            END IF
        CASE CHR$(0) + "H"
            RowCount = RowCount - 1
            IF RowCount < 0 THEN RowCount = 0
            CALL PaintPattern(1, 4 + RowCount + 1, 0,
79, 0, 0, " ")
            LOCATE 4 + RowCount + 1, 5: COLOR 10:
PRINT Soul$(1 + RowCount + 1 + (16 * Anymore))
        END SELECT
        SpecialKBD$ = kbd$
        LOOP UNTIL SpecialKBD$ = CHR$(0) + "Q"
OR SpecialKBD$ = CHR$(0) + "I" OR SpecialKBD$ =
CHR$(13) OR SpecialKBD$ = CHR$(27)
END SUB

SUB Hint
    CALL PaintPattern(1, 22, 0, 79, 0, 11, " ")
    CALL PaintPattern(1, 23, 0, 79, 0, 15, " ")
    LOCATE 23, 5
    COLOR 1.
    PRINT " More";
    COLOR 0
    PRINT " <Down,Up Arrow> or <Page Down,Up> ";
    COLOR 1:
    PRINT " Select";
    COLOR 0
    PRINT " <Enter> ";
    COLOR 1.
    PRINT " EXIT";
    COLOR 0
    PRINT " <Escape> "

END SUB

SUB Openfile
        CLS
        OPEN "Mdrive.DV" FOR INPUT AS #1
            INPUT #1, Drive$
        CLOSE #1
        OPEN "ROOT.SCN" FOR INPUT AS #2
        count = 0
        DO UNTIL EOF(2)
            INPUT #2, ScanRoot$
            count = count + 1
            Soul$(count) = ScanRoot$
        LOOP
        TotalCount = count
        CLOSE #2
END SUB

SUB PaintPattern (Xposit, Yposit, RowLength, ColLength,
Fcolor, Bcolor, Text$)
    FOR i = Yposit TO Yposit + RowLength
        COLOR Fcolor, Bcolor
        LOCATE i, Xposit + 1  PRINT
STRING$(ColLength, Text$)
    NEXT i
END SUB

SUB PrintFile (Anymore)
    FOR i = 1 TO 16
      LOCATE 3 + i, 5
      IF Soul$(i + (16 * Anymore)) <> "" THEN
        COLOR 10, 0: PRINT Soul$(i + (16 * Anymore))
      ELSEIF Soul$(i + (16 * Anymore)) = "" THEN
        COLOR 12, 0: PRINT "NO FILE"
      END IF
    NEXT i
    IF TotalCount = 0 THEN COLOR 12  LOCATE 4, 4:
PRINT "File Not Found"

END SUB
```

Copyright by Mahidol University

Appendix IV

Derivation of Calculation of Chain Transfer Agent Conversion with Effect of Volume of Reaction Mixture Change.

From Table 3-1 (Chapter 3), at initial reaction time, total volume of substances in reaction mixture tubes (V_0) :-

$$V_0 = V_{styrene} + V_{cyclohexane} + V_{CCl4} + V_{AIBN} \quad \text{(AIV-1)}$$

And at specific reaction time, t, total volume of substances in reaction mixture tubes (V_t) which have the effect of volume of polymers ($V_{poly(styrene)}$):-

$$V_t = V_{styrene,t} + V_{poly(styrene),t} + V_{cyclohexane} + V_{CCl4} + V_{AIBN} \quad \text{(AIV-2)}$$

Density of cyclohexane, carbon tetrachloride and AIBN do not change. It is generally known that Volume = Mass/Density, also V_0 and V_t can be obtained as in Equations (IV-3) and (IV-4).

$$V_0 = \frac{M_{Styrene}}{\rho_{Styrene}} + C \quad \text{(AIV-3)}$$

$$V_t = \frac{100.M_{Styrene}}{(100-x).\rho_{Styrene} + x.\rho_{poly(styrene)}} + C \quad \text{(AIV-4)}$$

Where C is $V_{cyclohexane} + V_{CCl4} + V_{AIBN}$.

$V_{styrene}$ is Density of styrene monomer.

$V_{poly(styrene)}$ is Density of poly(styrene).

$M_{styrene}$ is Mass of styrene monomer in reaction mixture.

x is percentage of monomer conversion.

Copyright by Mahidol University

From the generally known that $M_t.V_t=M_0.V_0$, substitute M_t with $H_{t,0}$, M_0 (or $H_{t,t}$) can be obtained as

$$H_{t,t} = H_t.\frac{V_t}{V_0} \qquad \text{(AIV-5)}$$

Where $H_{t,t}$ is the peak height which treats effect of volume change. Therefore, the modified calculation of percentage of chain transfer agent conversion which addition of volume change effect can be obtained as below.

$$\text{CTA conversion (\%)} = 100.\frac{H_0 - H_t.\frac{V_t}{V_0}}{H_0} \qquad \text{(AIV-6)}$$

Copyright by Mahidol University

Appendix V

The Modified Mayo Equation

The number-average degree of polymerization, DP_n is the mean number of monomer units per polymer chain and is readily calculated from

$$DP_n = \frac{\text{Rate of consumption of monomer}}{\text{Rate of formation of polymer}} \qquad \text{(AV-1)}$$

from kinetics mechanism, DP_n can be obtained as below.

$$DP_n = \frac{d[M]}{\Gamma dt} \qquad \text{(AV-2)}$$

where Γ is the rate of formation of polymer

$$\Gamma = \frac{d[M_n]}{dt} + \frac{d[M_nX]}{dt} + \frac{d[RM_n]}{dt} + \frac{d[RM_nX]}{dt} + \frac{d[RM_nR]}{dt} + \frac{d[CTAM_n]}{dt} + \frac{d[CTAM_nX]}{dt} + \frac{d[CTAM_nCTA]}{dt} + \frac{d[CTAM_nR]}{dt}$$

From equations (AI-39)-(AI-47)

$$\Gamma = 0.5k_{tc}\{P_m^2+P_r^2+P_c^2\} + k_{tc}\{P_rP_m+P_cP_m+P_rP_c\}+ (P_r + P_c + P_m)\{k_{tci}[R^*] + k_{tr}[CTA] + k_{trm}[M] + k_{td}(P_r + P_c + P_m)\} \qquad \text{(AV-3)}$$

From Equation (AI-12)

$$-\frac{d[M]}{dt} = k_i[R^*][M] + k_{i,CTA}[CTA^*][M] + k_p[M]\{P_m+P_r+P_c\} + k_{trm}[M]\{P_m+P_r+P_c\}$$

Copyright by Mahidol University

Substitution of $k_{i,CTA}[CTA^*][M]$, $k_i[R^*][M]$, and $[R^*]$ into Equation (AI-12), Equation (AV-4) can be obtained as below.

$$-\frac{d[M]}{dt} = 2k_df[I] + (P_m+P_r+P_c)\{k_p[M] + k_{trm}[M] + k_{tr}[CTA] - k_{tci}[R^*]\} \quad \text{(AV-4)}$$

From QSSA that $dP_r/dt=0$, $dP_c/dt=0$ and $dP_m/dt=0$,therefore, $dP_r/dt + dP_c/dt + dP_m/dt =0$, so

$$\frac{d}{dt}P_r + \frac{d}{dt}P_c + \frac{d}{dt}P_m = 2k_df[I] + 0.5k_{tc}\{P_m^2+P_r^2+P_c^2\} + 2k_{tc}\{P_rP_m+P_cP_m+P_rP_c\} + (P_r + P_c +P_m)\{2k_{tr}[CTA] + 2k_{trm}[M] + k_{td}(P_r + P_c + P_m)\} = 0 \quad \text{(AV-5)}$$

Substitution of $0.5k_{tc}\{P_m^2+P_r^2+P_c^2\} + k_{tc}\{P_rP_m+P_cP_m+P_rP_c\}$ into equation (AV-3), Γ can be written as shown in equation (AV-6),

$$-\Gamma = 2k_df[I] + k_{tc}\{P_rP_m+P_cP_m+P_rP_c\}+ (P_r+P_c+P_m)\{k_{tci}[R^*] + k_{tr}[CTA] + k_{trm}[M]\} \quad \text{(AV-6)}$$

DP_n can be simplified as follow,

$$\frac{1}{DP_n} = \frac{\frac{d}{dt}[M] + k_{tc}(P_rP_m + P_rP_c + P_cP_m) + (2k_{tci} - k_p)(P_r + P_c + P_m)}{\frac{d}{dt}[M]} = 1 + \frac{k_{tc}(P_rP_m + P_rP_c + P_cP_m) + (2k_{tci}[R^*] - k_p[M])(P_r + P_c + P_m)}{\frac{d}{dt}[M]} \quad \text{(AV-7)}$$

Substitution of d[M]/dt (Equation (AV-4)) in to Equation (AV-7) give:-

Copyright by Mahidol University

$$\frac{1-DP_n}{DP_n} = \frac{\frac{k_{tc}(P_rP_m + P_rP_c + P_cP_m)}{P_m + P_r + P_c} + 2k_{tci}[R^*] - k_p[M]}{\frac{2k_df[I]}{P_m + P_r + P_c} + k_p[M] + k_{trm}[M] + k_{tr}[CTA] - k_{tci}[R^*]} \quad \text{(AV-8)}$$

LIBRARY AND INFORMATION CENTER
MAHIDOL UNIVERSITY

All term on the right hand side of equation (AV-8) divide with $k_p[M]$, it can get,

$$\frac{DP_n - 1}{DP_n} = \frac{1 - \frac{k_{tc}(P_rP_m + P_rP_c + P_cP_m)}{k_p[M](P_m + P_r + P_c)} - \frac{2k_{tci}[R^*]}{k_p[M]}}{1 + C_m + C_s\frac{[CTA]}{[M]} - \frac{k_{tci}[R^*]}{k_p[M]} + \frac{2k_df[I]}{k_p[M](P_m + P_r + P_c)}} \qquad \text{(AV-9)}$$

and then, rearrange equation (AV-9), can write as

$$\frac{1 + \omega DP_n}{DP_n - 1} = C_m + C_s\frac{[CTA]}{[M]} - \frac{k_{tci}[R^*]}{k_p[M]} + \frac{2k_df[I]}{k_p[M](P_m + P_r + P_c)} \qquad \text{(AV-10)}$$

where

$$\omega = -\frac{k_{tc}(P_rP_m + P_rP_c + P_cP_m)}{k_p[M](P_m + P_r + P_c)} - \frac{2k_{tci}[R^*]}{k_p[M]} \qquad \text{(AV-11)}$$

Parameter ω is low value compared to 1, also the synthetic conditions are often used in small amount of initiator, also, ω can be ignored. Therefore, the modified Mayo equation can be obtained as follow,

$$\frac{1}{DP_n - 1} = \left(\frac{1}{DP_n}\right)_0 + C_m + C_s\frac{[CTA]}{[M]} \qquad \text{(AV-12)}$$

where

$$\left(\frac{1}{DP_n}\right)_0 = \frac{2k_df[I]}{k_p[M](P_m + P_r + P_c)} - \frac{2k_{tci}[R^*]}{k_p[M]} \qquad \text{(AV-13)}$$

Copyright by Mahidol University

27350