

**FIXED BED REACTOR MODELING
AND KINETICS EVALUATION**

By

ROBERT EI WADE

Bachelor of Science

Montana State University

Bozeman, Montana

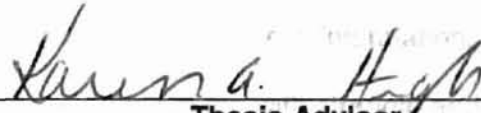
1992

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 1999

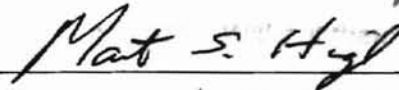
FIXED BED REACTOR MODELING

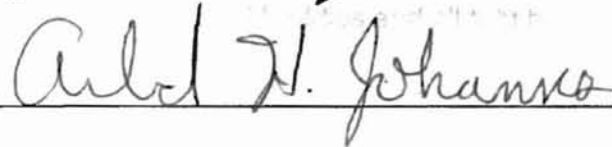
AND KINETICS EVALUATION

Thesis Approved:



Thesis Advisor







Dean of the Graduate College

ACKNOWLEDGMENTS

I wish to express my appreciation to my advisor Dr. Karen High for her patient guidance and encouragement throughout my masters program. Many thanks go also to Dr. Arland Johannes, Dr. Alan Tree and Dr. Marty High for serving on my graduate committee. Their advice, insight and direction were very helpful throughout the study.

Special appreciation goes out to Doug Jack, Prakash Karpe, and Daryl Dunham of Conoco for providing much needed process information, and funding for this project.

My wife and parents, provided me with strength and determination to constantly improve myself and achieve my goals. Without them, this would not have been possible. Thanks also goes out to the friends, travel buddies and all that helped in the preparation of the final thesis draft.

TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION	1
2 BACKGROUND	3
2.1 NON-IDEAL METHODS	3
2.1.1 Residence time distributions (empirical)	4
2.1.2 Population balances	4
2.1.3 Effective transport models (semi-empirical).....	5
2.1.4 Recent advances	6
2.2 NUMERICAL SOLUTION METHODS.....	7
2.2.1 Finite difference.....	8
2.2.2 Method of lines.....	8
2.2.3 Adams-Moulton.....	9
2.3 SELECTED TEST REACTIONS.....	9
2.3.1 Ethane cracking reaction.....	10
2.3.2 MTBE synthesis reaction.....	11
3 MODEL DEVELOPMENT	15
3.1 GENERIC FIXED BED REACTOR.....	15
3.1.1 Model assumption.....	17
3.2 EQUATION DEVELOPMENT AND NON-DIMENSIONALIZATION	19
3.2.1 Species mass balance.....	19
3.2.2 Overall continuity equation	20
3.2.3 Momentum balances.....	21
3.2.4 Thermal energy balance.....	22
3.2.5 Non-dimensionalization.....	24
3.3 EFFECT OF DIMENSIONLESS GROUPS	25
3.4 BOUNDARY CONDITIONS	25
3.5 SUPPORT EQUATIONS.....	29
3.5.1 Pressure drop	29
3.5.2 Effective transport properties	30
3.5.3 Heat capacity calculations	32
3.5.4 Enthalpy of reaction calculations.....	32
3.6 COMPLETE DEFINITION OF SYSTEM.....	33
4 NUMERICAL METHODS	35
4.1 PDETWO CODE.....	35
4.2 ALGORITHM FLOW.....	36
4.3 MAIN ROUTINE	37
4.4 BOUNDARY CONDITIONS	39
4.4.1 Diffusion coefficients	40
4.5 EQUATION DEFINITION, KINETICS AND CALCULATIONS	41

5 RESULTS & DISCUSSION	43
5.1 PDETWO SOLUTION TEST (ANALYTIC COMPARISON)	43
5.2 REACTOR MODEL TESTING.....	44
5.2.1 Plug flow reactor benchmark.....	46
5.2.2 Asymptotic comparison.....	49
5.2.3 Effect of domain step size.....	53
5.2.4 Mass conservation test.....	57
5.3 PARAMETRIC ANALYSIS.....	57
5.3.1 Effect of inlet temperature.....	59
5.3.2 Reactor length analysis.....	61
5.3.3 Flow rate analysis.....	64
5.4 SUMMARY OF RESULTS	66
6 CONCLUSIONS AND RECOMMENDATIONS.....	71
6.1 CONCLUSIONS	71
6.2 RECOMMENDATIONS	72
BIBLIOGRAPHY.....	75
APPENDICIES.....	78
APPENDIX A DETERMINATION OF RTD AND EFFECTIVE DIFFUSION PARAMETERS.....	79
APPENDIX B IMPLEMENTING THE SOFTWARE INTO SIMULATION ENGINES USING DYNAMIC LINK LIBRARIES	84
APPENDIX C FORTRAN CODE.....	90
C.1 MAIN SUBROUTINE	91
C.2 Horizontal boundary condition routine.....	99
C.3 Vertical boundary condition routine.....	101
C.4 SUBROUTINE DIFFH (T,X,Y,U,DH,NPDE).....	103
C.5 Vertical diffusion routine.....	104
C.6 Function evaluation routine	105
C.7 Ethane kinetics routine	105
C.8 MTBE kinetics routine	109
C.9 Physical property calculation routine	111
C.10 Date module.....	112
C.11 Common blocks.....	113
APPENDIX D EXCEL™ VISUAL BASIC INTERFACE ROUTINES	115

LIST OF TABLES
LIST OF FIGURES

Table	Page
TABLE 2-1 MTBE SYNTHESIS REACTION KINETICS (AL-JARALLAH ET AL. (1988)).....	13
TABLE 3-1 GENERALIZED FIXED BED REACTOR EQUATIONS.....	23
TABLE 3-2 DIMENSIONLESS GROUPS USED IN NON-DIMENSIONALIZATION	24
TABLE 3-3 BOUNDARY CONDITIONS.....	28
TABLE 4-1 EQUATION IMPLEMENTATION.....	42
TABLE 5-1 ANALYTICAL COMPARISON TO PDETWO RESULTS	44
TABLE 5-2 TEST CASE SPECIFICATIONS AND CONDITIONS.....	46
TABLE 5-3 HYSYST™ SIMULATION RESULTS, ETHANE CRACKING PFR REACTION	47
TABLE 5-4 MODEL SIMULATION RESULTS, ETHANE CRACKING PFR REACTION	47
TABLE 5-5 ASYMPTOTIC COMPARISON OF MIXING CUP CONCENTRATION PROFILES FOR THE ETHANE CRACKING REACTION AT HIGH PECKET NUMBERS.....	50
TABLE 5-6 ASYMPTOTIC COMPARISON OF MIXING CUP CONCENTRATION PROFILES FOR THE ETHANE CRACKING REACTION AT LOW PECKET NUMBERS	51
TABLE 5-7 EFFECT OF STEP SIZE ON THE NUMERICAL RESULT IN THE ETHANE CRACKING REACTION.....	53
TABLE 5-8 EFFECT OF STEP SIZE ON THE NUMERICAL RESULTS IN THE MTBE SYNTHESIS REACTION	55
TABLE 5-9 MASS BALANCE DATA FOR THE ETHANE CRACKING REACTION.....	58
TABLE 5-10 MASS BALANCE DATA FOR THE MTBE SYNTHESIS REACTION.....	58
TABLE 5-11 PARAMETRIC ANALYSES - VARIABLE RANGE	57
TABLE 5-12 COMPARISONS OF MTBE SYNTHESIS KINETIC EXPRESSIONS TO EXPERIMENTAL VALUES..	66
TABLE 5-13 SUMMARY OF RESULTS	68

NOMENCLATURE
LIST OF FIGURES

FIGURE		Page
FIGURE 3-1	SCHEMATIC OF A TYPICAL FIXED BED REACTOR.	15
FIGURE 3-2	VISUALIZATION OF FLOW AROUND A CATALYST PARTICLE.....	16
FIGURE 3-3	BOUNDARY CONDITIONS & PROFILES (SLIP CONDITIONS).....	27
FIGURE 3-4	BOUNDARY CONDITIONS & PROFILES (NO SLIP CONDITIONS).	27
FIGURE 4-1	PDETWO FIXED BED SOLUTION FLOW CHART.	38
FIGURE 4-2	DOMAIN GRID LAYOUT.	39
FIGURE 5-1	COMPARISON BETWEEN PDETWO AND ANALYTIC SOLUTION.....	45
FIGURE 5-2	MOLE FRACTION COMPARISONS OF HYSYS™ SIMULATION MODEL FOR THE ETHANE CRACKING REACTION.	48
FIGURE 5-3	ASYMPTOTIC COMPARISON OF REACTOR TYPES AT HIGH PECKET NUMBERS FOR THE ETHANE CRACKING REACTION.	52
FIGURE 5-4	ASYMPTOTIC COMPARISON OF REACTOR TYPES AT LOW PECKET NUMBERS FOR THE ETHANE CRACKING REACTION.	54
FIGURE 5-5	GAS PHASE HEAT CAPACITY VERSUS TEMPERATURE PROFILES FOR THE ETHANE CRACKING REACTION.....	56
FIGURE 5-6	EFFECT OF INLET TEMPERATURE ON CONVERSION IN THE ETHANE CRACKING REACTION UNDER ISOTHERMAL AND ADIABATIC CONDITIONS.....	60
FIGURE 5-7	EFFECT OF INLET TEMPERATURE ON CONVERSION IN THE MTBE SYNTHESIS REACTION UNDER ISOTHERMAL AND ADIABATIC CONDITIONS.....	62
FIGURE 5-8	EFFECT OF REACTOR LENGTH ON CONVERSION IN THE ETHANE CRACKING REACTION UNDER ISOTHERMAL AND ADIABATIC CONDITIONS.....	63
FIGURE 5-9	EFFECT OF REACTOR LENGTH ON CONVERSION IN THE MTBE SYNTHESIS REACTION UNDER ISOTHERMAL AND ADIABATIC CONDITIONS.....	65
FIGURE 5-10	COMPARISON OF MODEL PERFORMANCE TO ZHANG AND DATTA (1995) EXPERIMENTAL DATA FOR THE MTBE SYNTHESIS REACTION.	67

NOMENCLATURE

Roman Letters

a_i	PDETWO initial x boundary condition in PDETWO
AH_i	PDETWO horizontal boundary coefficient on variable u_i
AV_i	PDETWO vertical boundary coefficient on variable u_i
b_i	PDETWO initial y boundary condition in PDETWO
BH_i	PDETWO horizontal boundary coefficient on variable du_i/dr
BV_i	PDETWO vertical boundary coefficient on variable du_i/dz
CH_i	PDETWO horizontal boundary coefficient on variable
CV_i	PDETWO vertical boundary coefficient on variable
\bar{C}_i	Dimensionless concentration for component i
C_a	Concentration of component a (basis component)
C_{ao}	Initial concentration of component a
C_i	Concentration of component i
d_p	Catalyst particle diameter
\tilde{D}_{er}	Effective radial diffusion coefficient
\tilde{D}_{ez}	Effective axial diffusion coefficient
$\tilde{D}_{e\theta}$	Effective θ diffusion coefficient
E_a	Activation Energy
F_i	Total fluid flow rate
F_o	Initial fluid flow rate
C_p	Average heat capacity

C_{p_i}	Component heat capacity
C	Dimensionless concentration
g_r	Gravity term in the radial direction
g_z	Gravity term in the axial direction
g_θ	Gravity term in the θ direction
ΔH_{rxn}	Heat of reaction
ΔH_{rxn}°	Heat of reaction at the reference temperature
k	Rate constant
k_0	Arrhenius coefficient
K_{eq}	Equilibrium constant
\bar{k}_{er}	Radial effective thermal conductivity
\bar{k}_{ez}	Axial effective thermal conductivity
$\bar{k}_{e\theta}$	θ effective thermal conductivity
L	Reactor length
M_c	Mass of catalyst charge to reactor
mw_i	Molecular weight of component i
N_{cc}	Number of components present
N_{rxn}	Number of reactions present
N_{vel}	Number of velocity equations
N_{heat}	Number of heat equations
N_{eq}	Total number of equation required
N_{pde}	Number of partial differential equations
N_{tubes}	Number of reactor tubes present

N_{fr}	Froment number
N_{re}	Reynolds number
N_{sc}	Schmidt Number
ρ	Gunn's geometric correction
\bar{P}_{er}	Radial Peclet number
\bar{P}_{ez}	Axial Peclet number
$\bar{P}_{e\theta}$	θ Peclet number
P	Pressure
P_0	Inlet pressure
\bar{P}	Dimensionless pressure
R	Gas constant
R	Current radius
\bar{r}	Dimensionless radius
R_{wall}	Reactor radius at wall
R_{xn}	Intrinsic reaction rate
S_r	Temperature generation term
v_r	Velocity in the r direction
v_z	Velocity in the z direction
v_{z0}	Initial velocity in the z direction
v_θ	Velocity in the θ direction
\bar{v}_z	Dimensionless velocity in the z direction
T_0	Reactor inlet temperature
T	Temperature

\bar{T} Dimensionless temperature

Abbreviat t Time

t_0 Initial time value

w Dummy variable for equation transformation

X_a Conversion of component a

Z Current position z down reactor

\bar{z} Dimensionless length

Greek Letters

α_i Coefficient of the heat capacity equation

β_i Coefficient of the heat capacity equation

γ_i Coefficient of the heat capacity equation

δ_i Coefficient of the heat capacity equation

ϵ Bed porosity

σ Ratio of length to particle diameter

η Ratio of length to radius

M Ratio of Radius to particle diameter

τ Catalyst pore porosity

ρ Bulk fluid density

ρ_f Fluid density

ρ_b Catalyst bulk density

μ_{ez} Effective axial viscosity

μ_{er} Effective Radial viscosity

$\mu_{e\theta}$ Effective θ viscosity

Chapter 1

Introduction

Abbreviations

CSTR Constantly stirred tank reactor

DLL Dynamic link library

LHHW Langmuir–Henshelwood–Houghen-Watson kinetics

MTBE Methyl tert-butyl-ether

MOL Method of lines

ODE Ordinary differential equation

PDE Partial differential equation

PFR Plug flow reactor

RTD Residence time distribution

Chapter 1

Introduction

The developed model must be able to handle a wide range of reaction conditions and be flexible to ideal and non-ideal assumptions. This allows the model to be useful in evaluation and testing at all levels of development including pilot plant and full-scale.

Incorporation of laboratory methods into the model will allow the model to

The chemicals and refining industry is under constant pressure to develop and implement new cost saving technologies. New technologies must undergo extensive testing and development before full-scale implementation. Testing begins in labs and research centers with the highest potential technologies evaluated at pilot scales. These evaluations are both costly and time consuming, but are a vital part of the successful scale up and development process.

In an effort to decrease costs, companies sometimes use outside labs to test and evaluate new technologies. Following in this practice Conoco requested that Oklahoma State University develop kinetics and detailed computer models for the evaluation of the Nafion™ catalyst. Initial research provided by Conoco suggests that Nafion™ could potentially serve as a powerful catalyst in processes such as; catalytic polymerization, alkylation, MTBE synthesis and butene isomerization.

The Nafion™ project was divided into a computer modeling project and an experimental testing project. This work focuses on the model development and simulation, with experimental kinetic evaluations to follow. Detailed research was conducted on the processes of interest, and a generic fixed bed computer model was developed.

In an effort to provide Conoco with a model that could accurately handle simulations from bench to full scale, the model must be able to handle non-ideal condition and be as generic as possible. In addition, the developed code must be able to interface with current simulation software such as HYSYS™, ASPEN PLUS™ and Excel™.

The developed model must be able to hand a wide range of reaction conditions and be flexible to ideal and non-ideal assumptions. This allows the model to be useful in evaluation and testing at all levels of development including pilot plant and full-scale operation. Incorporation of non-ideal methods into the model will allow the model to handle inline optimization, scale up and prediction during and after process development.

In order to determine the accuracy and flexibility of the developed code, testing and parametric analysis on two test reactions will take place. Test reactions include the gas phase ethane cracking reaction and liquid phase MTBE synthesis reaction. The chosen reactions provide two extremes of conditions. The ethane crack reaction is used to determine how effectively the model handles highly endothermic gas phase reactions and represents the upper extreme of anticipated operating conditions. The MTBE synthesis reaction represents typical simulation conditions expected for Nafion™ systems considered.

Code development began with a review of historical and state of the art fixed bed modeling techniques to determine the final techniques used in this work. Chapter 2 also presents the two reaction systems used to test model flexibility and limitations. Chapter 3 and 4 provide detailed information on the assumptions and steps taken to develop the generalized model. These chapters also provide supporting equations and general description of the overall code framework. Chapter 5 provides an analysis of model testing and evaluation. Evaluation includes studies on feasibility, accuracy, asymptotic limits and parametric sensitivity for each test reaction. Finally, chapter 6 summarizes results and outlines future recommendations.

Chapter 2

Background

This chapter reviews the fundamental concepts critical to advanced chemical reactor modeling. Review of non-ideal modeling, state of the art physical approximations, and numerical methods provide the basis for this work. Once these techniques are reviewed and methods are chosen, the model test reactions are presented. We begin with a discussion of traditional non-ideal reactor modeling techniques.

2.1 Non-Ideal methods

Most reactor design and analysis focuses on ideal versions of the actual reactor, ignoring non-ideal behavior such as turbulence, catalyst deactivation and changing physical properties. Resulting solutions from ideal methods provide rough approximations and do not account for many performance effecting phenomena (Levenspiel 1972; Fogler 1986; Nauman 1987). Critical phenomena include fluctuations in velocity profiles due to turbulence, uneven catalyst distribution and potentially catalyst deactivation (Himmelblau and Bischoff 1968; Fogler 1986; Levenspiel 1989). Better simulation results over a wide range of conditions can only be achieved by taking the above mentioned factors into account.

There is currently a narrow range of techniques developed to account for non-ideal and highly complex deviations from ideal conditions in fixed bed reactors. Methods include residence time distributions (RTD), population balances and effective diffusion models (Levenspiel 1972; Froment and Bischoff 1979; Fogler 1986). The current project starts with the simplest of the above stated methods, the RTD approach.

2.1.1 Residence time distributions (empirical)

One of the most common methods, residence time distributions (RTD) (Himmelblau and Bischoff 1968; Froment and Bischoff 1979; Nauman 1987) uses empirical testing to determine reactor performance. The RTD modeling method requires experimental results from the actual reactor modeled. The RTD method is computationally very fast, but has no theoretical foundations, and is thus only applicable for a narrow range of predictions. The method can only accurately predict performance or scale up problems on the reactor that experimental data was obtained. To further emphasize limitations, the RTD method can only be used for optimization at or very close to the test conditions.

From a design stand point this method appears to offer no cost saving advantages such as reducing the number of pilot plants built or even wide range optimization. A method that better approximates the physical phenomena involved is required to effectively reduce cost and provide predictions outside the initial testing domain.

2.1.2 Population balances

The population balance approach suggests that age distributions can be applied to the physical characteristics with the reactor. Like the RTD method, experimental data is required to determine probability distribution functions for concentration, adsorption and other required reactor characteristics. The population balance method relies heavily on probability and statistical distributions (Himmelblau and Bischoff 1968). The approach suffers from the same major problems that the RTD method suffers in that detailed testing on the reactor is required for even rough approximations. This, along with complexity issues makes this method less than desirable.

2.1.3 Effective transport models (semi-empirical)

The effective transport method combines traditional mass and energy balances with experimental RTD data. The combination of these methods provides strong theoretical techniques with the adaptability of 'empirical' fine-tuning. The key to the effective transport method is accurate approximation or determination of transport properties.

The effective properties account for flow non-idealities such as uneven void fractions, dead spaces, and other real-life phenomena encountered. Due to their empirical nature, the resulting diffusivities are applicable primarily to the conditions at which they were determined. However, the physical nature of this method allows better predictions over a wider range than does the RTD method.

Determination of effective properties begins with RTD tracer experiments or semi-empirical correlation. Once reactor performance and plot feasibility has been determined, least squares or other error minimization techniques determine the effective diffusivities as dictated by the governing partial differential equations. Resulting effective terms are considered identical (Froment and Bischoff 1979) for all chemical species present as the physical effects of turbulent flow greatly dominate any molecular effects.

Effective transport models also offer the ability to be updated as new prediction and theoretical methods become available. This plays a critical role in model longevity and is very attractive, as many significant advances have occurred over the years. The effective transport method efficiently combines the adaptability of RTD and physical significance of classical methods. This combination allows prediction of non-ideal conditions and reactor performance well outside the initial testing ranges. Wide prediction range and ease of tuning makes effective transport models a viable tool to

both help reduce pilot plant testing and predict scale up problems prior to construction, providing a powerful cost savings tool.

2.1.4 Recent advances

In the past 15 years, effective diffusion methods have seen a large number of advances and changes. Advances in computer speed and efficiency have played a major role, allowing relatively fast solution of 2 and 3 dimensional models. It is now common place for accurate models to account for both axial and radial variations when plug flow assumptions do not sufficiently predict reactor performance.

Researchers such as Gunn (Gunn 1987; Gunn et al. 1987) and Vortmeyer (Vortmeyer and Schuster 1983) have used modern computers to model reactors by taking into account variations of much higher complexity than ever before. One of the most useful advances came from Gunn et al. in 1987. Gunn was able to model and predict effective axial and radial transport coefficients for both heat and mass transfer in packed beds. Gunn's work allows for good estimation of effective transport properties prior to reactor tracer testing, making non-ideal modeling using effective transport properties a much more valuable and flexible tool.

Vortmeyer (Vortmeyer and Schuster 1983) introduced equations to approximate and account for radial variations in bed void fraction and axial velocity. Though not accounted for in this work, it is mentioned because the model developed in this work is designed to handle these upgrade approximations in future studies. Advances by Vortmeyer, Gunn, and the rise of computational fluid dynamics have greatly enhanced the accuracy and prediction of packed bed reactors.

Reactor modeling methods and recent advances provide a powerful tool, but are useless without a means for solving the resulting equations. We now look at the numerical methods required for the solution of the reactor model.

2.2 Numerical solution methods

The model developed in this work will involve very complex kinetics, requiring numerical methods. Analytical solutions for partial differential equations are possible for only the simplest of cases (Rice and Do 1995) and are not possible for highly non-linear systems with complex kinetics (Levenspiel 1972; Rice and Do 1995). Since the model being developed includes second order partial derivatives in at least 2 dimensions as well as potentially non-linear terms, analytical solutions are not possible. Generalized numerical solution methods will be required.

Many numerical programs are available for the solution of partial differential equations, with most methods designed for specific equation solutions. For solution of model equations under a wide range of assumptions, a much more generalized code must be implemented. PDETWO (Melgaard and Sincovec 1981; Melgaard and Sincovec 1981) is a widely used publicly available set of routines for the solution of a wide range of partial differential equations. PDETWO was originally developed as a generalized tool capable of handling a very wide range of coupled partial differential equations in three spatial dimensions and one time dimension. Because of its generality, PDETWO has been used for the solution of partial differential equations ranging from the very simple (Melgaard and Sincovec 1981; Melgaard and Sincovec 1981) to highly coupled nonlinear magneto-hydrodynamic problems.

PDETWO, though very general, is limited to systems that do not contain cross derivatives. Cross derivatives are of the generalized form:

$$\frac{\partial C}{\partial x \partial y}, \quad (2-1)$$

and are not features of this work. Using finite difference techniques and the method of lines, PDETWO converts the partial differential equations into coupled first order ODE.

Subsequent solution is completed using traditional finite difference methods and techniques.

2.2.1 Finite difference

Finite difference methods approximate partial differential equation derivatives by dividing the area of interest into an evenly spaced system of grid points and application of Taylor series expansions to approximate the solution at each grid point. This method of approximation is by far the most commonly used and extensively covered in numerical methods texts. An extraordinary amount of research and development in numerical approximation has led to the development of many methods based on finite difference principles. Methods include Euler, Runge-Kutta, Adams-Moulton, Gears, etc; each with many different forms and variations. For the purpose of this document, discussion will be limited to the Adams-Moulton, since it is bundled with the PDETWO algorithm (Melgaard and Sincovec 1981; Melgaard and Sincovec 1981).

2.2.2 Method of lines

One of the most common methods used for solving second order ordinary differential equations or partial differential equations is the method of lines (MOL). Using a transformation of variables solution approach (Carnahan et al. ; Himmelblau and Bischoff 1968; Riggs 1994), the method of lines converts second order or higher partial differential equations into coupled first order ordinary differential equations. Once the equations have been reduced and the system Jacobian generated, integration is completed using an appropriate integration routine. PDETWO implements a variable order Adams-Moulton method for equation solution.

2.2.3 Adams-Moulton

The Adams-Moulton methods use backward finite difference techniques to approximate the solution of ordinary differential equations (Hornbeck 1975). The simplest Adams-Moulton method incorporates an iterative solution approach that includes the point being predicted. This first order Adams-Moulton formula:

$$y_{i+1} = y_i + hf(t_{i+1}, y_{i+1}) \quad (2-2)$$

is commonly referred to as the backward difference Euler formula. Higher order Adams-Moulton methods are more typically employed. The high order methods take into account previously computed solutions and provide solutions of much higher accuracy.

PDE TWO uses a variable order Adams-Moulton method for equation integration. Adams-Moulton methods, due to the backward difference techniques, offer very high accuracy at the expense of iteration time. It should be noted that integration with Adams-Moulton techniques, though slow, offer an actual error that is an order of magnitude lower than techniques using forward difference techniques such as Gears method (Hornbeck 1975). The Adams-Moulton method is used throughout all phases of solution due to its stability and accuracy.

2.3 Selected test reactions

Test reactions were selected to test the limitations of the model over a wide range of reactions, including the primary reaction presented by Conoco. It was determined early in the project that availability of catalyst and reactor operation issues would result in the model phase of the project being completed prior to receiving Nafion™ experimental data.

Since the first phase of this project (model development) will be completed prior to the availability of the Nafion™ reactor data, model performance is compared to a

similar reaction. The ethane cracking reaction and the MTBE synthesis reaction were chosen to determine if the developed model could feasibly be applied to the Nafion™ reactions. The MTBE synthesis reaction (uses an Amberlest-15 catalyst) was chosen because of similarities to the Nafion™ in catalyst type and anticipated reaction complexity. The ethane cracking reaction was chosen to determine model flexibility and generality under conditions very different from the MTBE synthesis reaction.

2.3.1 Ethane cracking reaction

The ethane cracking reaction (Fogler 1986) was used to evaluate the effectiveness of the model in gas systems at high temperature. The ethane cracking reaction was chosen because of its extensive use for plug flow calculations in chemical kinetics textbooks (Froment and Bischoff 1979; Fogler 1986).

The irreversible endothermic ethane cracking reaction is:



and is described by elementary first order kinetics:

$$r_a = kC_a \quad (2-4)$$

Shah et al. (1967) experimental results determined an average activation energy of 82,000 cal/g, and an Arrhenius type pre-exponential (k_0) of 6.04×10^{16} /s, with k defined as:

$$k = 6.04 \times 10^{16} \exp\left(\frac{-82000}{RT}\right) \quad (2-5)$$

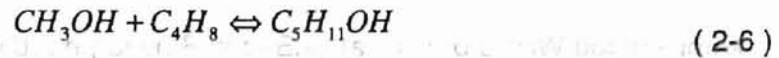
The ethane cracking reaction allows testing of the model on a gas phase system with varying velocity and density. Since the reaction is gas phase, concentrations need to be corrected under non-isothermal conditions, potentially resulting in significant stiffness and nonlinear solution problems. In addition, analytical solutions are known for

ideal PFR conditions, the ethane cracking reaction provides a good benchmark for near ideal model simulations.

2.3.2 MTBE synthesis reaction

The second test reaction (MTBE synthesis) serves as a focus of this research. The MTBE synthesis reaction is a liquid phase reaction (much like the anticipated Nafion™ reactions) and uses a catalyst similar to the Nafion™ catalyst. It is anticipated that if the developed model successfully predicts MTBE synthesis it will also prove effective in predicting other non-ideal reactions such as those involve with the Nafion™ catalyst. In addition, the MTBE synthesis reaction may also prove a viable application for the Nafion™ catalyst.

The MTBE synthesis reaction:



is a mildly exothermic liquid phase reaction and plays a major role in the production of gasoline as a fuel octane enhancement. Since the passage of the Clean Air Act Amendment of 1990, MTBE usage has grown (Zhang and Datta 1995), due to its high octane rating. As early as 1993, MTBE production was second only to ethylene (Zhang and Datta 1995). MTBE now plays a vital role in motor vehicle fuel sales as an additive for increasing the octane number of commercial gasoline (Zhang and Datta 1995; Gomez et al. 1997; Shikata et al. 1997; Sola and Pericas 1997).

MTBE is synthesized by reacting liquid methanol and isobutylene at temperatures ranging from 313 K to 353 K (Rehfinger and Hoffman 1990). Reaction pressure is maintained sufficiently high enough to ensure that the isobutylene stays in the liquid phase (typically around 17 atm). The mildly exothermic reaction typically progresses in fixed bed reactors to allow control of reaction temperatures. The reaction is equilibrium limited, as higher temperatures tend to favor production of the reactants

and not MTBE. Extensive documentation of the thermodynamic limitations of this reaction can be readily found in the literature (Kleir et al. 1982; Al-Jarallah et al. 1988; Rehfinger and Hoffman 1990; Rehfinger and Hoffmann 1990; Zhang and Datta 1995; Gomez et al. 1997; Shikata, Natakata et al. 1997; Sola and Pericas 1997).

Research on the kinetics for MTBE synthesis has been extensively published since the 1970's. Most research has focussed on developing Langmuir–Henshelwood–Houghen-Watson (LHHW) type kinetics with non-idealities (activity coefficients) modeled by the UNIFAC method. Non-idealities with the UNIFAC method were primarily used due to the large polarity difference between the reacting species. Excluding (Zhang and Datta 1995), all published experimental kinetic evaluations have been conducted in batch and CSTRs only and does not evaluate the feasibility of most of these kinetic expressions under industry fixed bed conditions.

Al-Jarallah et al. (1988) proposed Rideal-Eley (similar to LHHW but assumes different absorption methods) kinetics based on component concentrations with apparent success in batch reactors. Rideal-Eley kinetics differ from LHHW kinetics only in where the reactions are assumed to occur. Rideal-Eley mechanisms assumes that the reaction occurs between the absorbed component and a non-absorbed component, where as LHHW kinetics assumes the reaction occurs between sorbed components only (Sattersfield 1980).

Table 2-1 lists resulting Al-Jarallah et al. kinetic expressions. Studies were conducted in internally cooled batch reactors or continuously stirred tank reactors (CSTRs), and very little has been published as to the applicability of the resulting kinetic expressions to industrial scale MTBE synthesis systems. Application of the Rideal-Eley expressions proposed by Al-Jarallah et al. though acknowledged in the literature has not been experimentally or computer simulated in industrial type reactor conditions. This work will look at and evaluate the applicability of the rate expressions obtained by Al-

Jarallah to conditions used by (Zhang and Datta 1995) in the determination of kinetic feasibility: against the experimental data of (Zhang and Datta 1995) the effective

Table 2-1

MTBE Synthesis Reaction Kinetics (Al-Jarallah et al. (1988))

Rate law:

$$r_s = k_s K_A \left(\frac{C_A C_B^{0.5} - \frac{C_C^{1.5}}{K_{eq}}}{(1 + K_A C_A + K_C C_C)} \right)$$

**Surface reaction rate:
(forward)**

$$k_s = 1.2 \times 10^{13} \exp\left(\frac{-87000}{RT}\right)$$

**Adsorption constant:
(Methanol)**

$$K_a = 5.1 \times 10^{-13} \exp\left(\frac{97500}{RT}\right)$$

**Adsorption constant:
(MTBE)**

$$K_c = 1.6 \times 10^{-16} \exp\left(\frac{119000}{RT}\right)$$

**Reaction equilibrium
constant:**

$$\ln K_{eq} = -13.482 + \frac{4388.7}{T} + 1.2353 \ln T - 0.013849T + 2.5923 \times 10^{-5} T^2 - 3.1881 \times 10^{-8} T^3$$

Note: definition and units for all variables listed can be found in the nomenclature section.

Zhang and Datta evaluated the usability of the kinetic expressions, using the UNIFAC method of predicting activity coefficients, in fixed bed reactors. Extensive evaluation of equilibrium expressions and parametric analysis was done. Activity-based kinetics is indeed applicable to industrial MTBE conditions. Note that Zhang and Datta limited the testing to near isothermal conditions, and did not explore the adiabatic operations.

In order to test the applicability of the Rideal-Eley mechanism proposed by Al-Jarallah et al. against the experimental data of (Zhang and Datta 1995) the effective diffusion model must first be developed. Chapter 3 describes and implements typical assumptions for the fixed bed reactor. Additionally, required support equations for physical properties are presented.

Four sections, which are ordered to follow the logical flow and development of reaction models commonly outlined in textbooks such as Bird et al. (1999). Model development starts with the description and review of general fixed bed reactor models. It is followed by formulation and application of model assumptions along with adequate dimensional relations are presented. Finally, support

relations are presented which will be a degree of freedom in the system to be modeled. Initial and boundary conditions are also

3.1 General fixed bed reactor

Figure 3.1.1

Figure 3.1.2

Figure 3.1.3

Figure 3.1.4

Figure 3.1.5



the catalyst particles. The reactor is symmetric about ($r=0$), and it is assumed that the reactor can be divided into axial (A) & radial (B) directions for numerical solution.

Chapter 3

Model development

One of the most important features of reactor non-ideality involves how the fluid flows. This Chapter is divided into four sections, which are ordered to follow the logical flow and development of reactor models commonly outlined in textbooks such as Bird et al. (1960). Model development starts with the designation and review of generic fixed bed reactor conditions and geometry. Formulation and application of model assumptions along with equation non-dimensionalization are presented. Finally, support equations for physical property calculations are presented, along with a degree of freedom analysis to determine if the system is completely defined and ready for solution.

3.1 Generic fixed bed reactor

Many different types of physical phenomena occur that effects the efficiency of a fixed bed reactor. Figure 3-1 depicts a cross section of a single tubular fixed bed reactor packed with porous catalyst particles. Fluid into the reactor from the left, and between

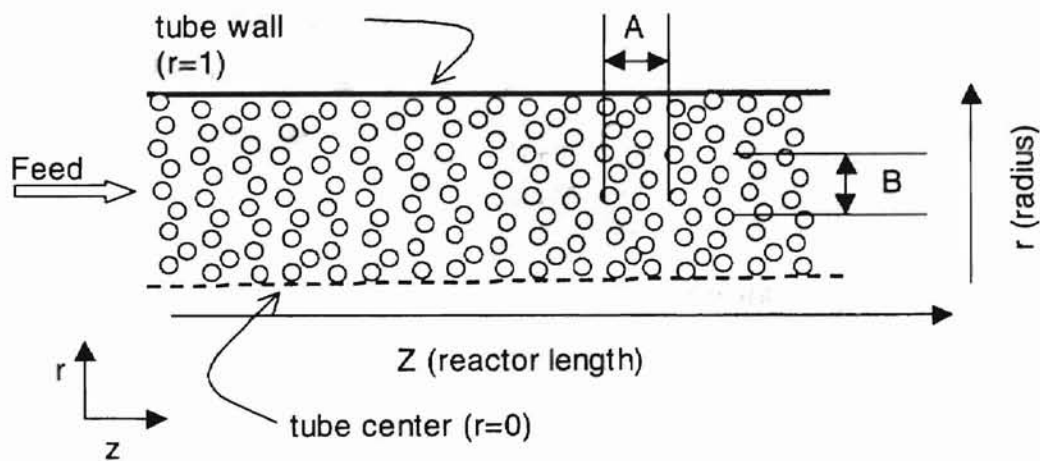


Figure 3-1 Schematic of a typical fixed bed reactor.

the catalyst particles. The reactor is symmetric about ($r=0$), and it is assumed that the reactor can be divided into small differential sections (axial (A) & radial (B)) for numerical solution.

One of the most prominent causes of reactor non-ideality involves how the fluid flows through the reactor. Reacting fluids enter the reactor (from the left) and flow through the spaces between the catalyst particles. The flow through the reactor can be laminar or turbulent, with turbulent flow being the most common in industrial fixed bed reactors. To best understand the flow characteristics of a reactor, a look at turbulent flow around individual catalyst particles is required.

Turbulent flow through a fixed bed can occur in all directions. For example (Figure 3-2), as the flowing fluid approaches the catalyst particle some of the fluid sticks to the surface, slowing almost to a stop, resulting in a laminar layer encapsulating the wall or catalyst and will be discussed later. Fluid away from the catalyst or wall surface maintains a higher velocity, resulting in vortices and eddy currents.

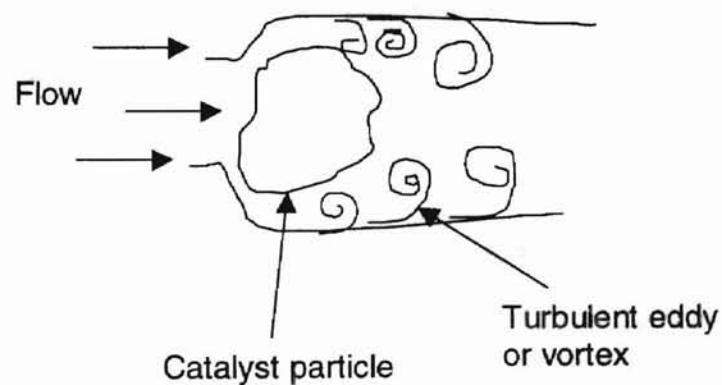


Figure 3-2 Visualization of flow around a catalyst particle.

Turbulent velocity gradients and eddy currents dissipate energy through the fluid, commonly referred to as viscous dissipation. The magnitude of energy loss associated with viscous dissipation is proportional to the type of fluid (Newtonian or other) and its velocity with highest energy losses occurring in non-Newtonian fluids.

Having provided a general description of flow around the catalyst, we now look at flow through the catalyst particle. A laminar boundary layer forms around the catalyst particle. Within this layer, convective flow is very slow, by boundary layer theory and approaches zero at the exact particle surface. Movement of the fluid is diffusion controlled and fluid particles must diffuse through the boundary to sorb onto the catalyst surface. The sorbed fluid may then react on the surface or diffuses into the pores of the catalyst particle.

Flow geometry and radial variations cause strong deviations from ideal reactors, increase computation time and require simplifying assumptions. Isothermal, adiabatic and non-isothermal conditions encompass the range of reactor operations. Throughout the reactor, diffusion of chemical species occurs in all directions, resulting in departure from ideal conditions. Velocities, concentrations and temperatures vary in all directions within a real reactor, leading to a large number of equations and calculations for accurate representation.

3.1.1 Model assumption

Simplified assumptions were implemented to minimize computation time and decrease overall complexity. An analytical solution of the most complex reactors that takes into account all the features of turbulent reacting flow is not possible. Numerical approximation of complex equations can be time consuming and in many cases is not currently feasible with modern computers (Szukiewicz et al. 1998). Therefore, common

simplifying assumptions (Bird et al. 1960; Froment 1967; Himmelblau 1996) are applied to implemented as follows:

1. Physical properties are constant for a given differential element in the system (see item A and B of figure 3-1). Physical properties are updated between elements using experimental and thermophysical correlations (Himmelblau 1996).
2. All fluids considered are Newtonian (Himmelblau 1996).
3. Viscous dissipation is negligible (Himmelblau 1996).
4. Effective turbulent transport properties are used, accounting for variations from ideal conditions in turbulent flow systems. Effective properties also correct for errors incurred through time averaging of dependent variables (Levenspiel 1972; Fogler 1986; Himmelblau 1996).
5. The effects of external forces (i.e., gravity) are identical for all components (Himmelblau 1996).
6. Velocities in the axial direction (v_z) of the tubes are significantly greater in comparison to those in the r (v_r) or θ (v_θ) directions. High flow rates and the nature of packed bed systems tend to limit the ability of a fluid or gas to flow in the r or θ direction (Froment and Bischoff 1979; Nauman 1987).

Additionally, pseudo-homogenous reaction conditions are applied. The pseudo-homogenous condition assumes that all reactions considered occur at the surface of the catalyst particle and not within the particle. Pseudo-homogenous conditions are considered accurate for catalyst systems in which pore diffusion is not rate limiting (Froment and Bischoff 1979; Cussler 1997). Pseudo-homogenous models assume that reactions occur only at the catalyst surface and are therefore less accurate, but solution times are much more reasonable. The assumptions outlined above are widely published

and reviewed (Bird et al. 1960; Nauman 1987; Himmelblau 1996) and are now applied to the general multiple gradient equations.

3.2 Equation development and non-dimensionalization

The generic steady state heat and mass transfer equations (commonly referred to as the multiple gradient equations) are the starting point for the development of most reactor models (Bird et al. 1960; Himmelblau 1996). Steady state conditions are used as a simplification to determine if the base model is applicable. Once the model proves feasible, only simple modifications are required to extend it to non-steady state. The multiple gradient equations assume time averaging, Newtonian fluid behavior, no viscous dissipation, and effective transport properties. This work starts with the most generalized steady state form of the multiple gradient equations of mass, energy, and momentum.

3.2.1 Species mass balance

Development of the model began with the mass balance equation written in cylindrical coordinates:

$$\frac{\partial}{\partial r}(v_r C_a) + \frac{\partial}{\partial \theta}(v_\theta C_a) + \frac{\partial}{\partial z}(v_z C_a) = \tilde{D}_{ar} \left(\frac{\partial^2}{\partial r^2} (C_a) + \frac{1}{r} \frac{\partial}{\partial r} (C_a) \right) + \tilde{D}_{a\theta} \left(\frac{1}{r^2} \frac{\partial^2 C_a}{\partial \theta^2} \right) + \tilde{D}_{az} \left(\frac{\partial^2}{\partial z^2} (C_a) \right) + R_a \quad (3-1)$$

Assuming physical properties and velocities are constant over the differential element (a representative element was shown in Figure 3-1), the velocity (v) terms are taken outside of the differentials. Further assumption of no radial or theta velocities eliminates those terms on the left side of the equations, resulting in:

$$v_z \frac{\partial C_a}{\partial z} = \tilde{D}_{ar} \left(\frac{\partial^2 C_a}{\partial r^2} + \frac{1}{r} \frac{\partial C_a}{\partial r} \right) + \tilde{D}_{a\theta} \left(\frac{1}{r^2} \frac{\partial^2 C_a}{\partial \theta^2} \right) + \tilde{D}_{az} \left(\frac{\partial^2 C_a}{\partial z^2} \right) + R_a \quad (3-2)$$

Application of assumption 7 and cylindrical symmetry eliminates all terms involving θ : (3-7)

$$v_z \frac{\partial C_a}{\partial z} = \tilde{D}_{ar} \left(\frac{\partial^2 C_a}{\partial r^2} + \frac{1}{r} \frac{\partial C_a}{\partial r} \right) + \tilde{D}_{az} \left(\frac{\partial^2 C_a}{\partial z^2} \right) + R_a. \quad (3-3)$$

Equation (3-3) defines the basic species mass balance equation used throughout this work. The mass balance equations must now be coupled with the momentum and continuity equations. (3-9)

3.2.2 Overall continuity equation

The overall mass continuity equation was defined using the generalized equation

$$\frac{1}{r} \frac{\partial}{\partial r} (\rho r v_r) + \frac{1}{r} \frac{\partial}{\partial \theta} (\rho v_\theta) + \frac{\partial}{\partial z} (\rho v_z) = 0, \quad (3-4)$$

as outlined by (Bird et al. 1960). Application of symmetry and assumption 7 leaves only one term in the continuity equation

$$\frac{\partial}{\partial z} (\rho v_z) = 0. \quad (3-5)$$

At this point, the derivatives can be expanded using the chain rule to give:

$$\rho \frac{\partial v_z}{\partial z} + v_z \frac{\partial \rho}{\partial z} = 0. \quad (3-6)$$

Equation (3-6) is most commonly used in gas phase systems and the effects of density are considered negligible in liquid phase systems. Since the primary focus of this work is on the liquid phase MTBE synthesis reaction, the gas phase velocity and density variations of Equation (3-6) were approximated using the standard algebraic expressions described below. Use of the algebraic approximation will help to reduce any solution stiffness and decrease computation time for gas phase systems. Both density and velocity component variations are calculated using (Fogler 1986; Levenspiel 1989)

Equation (3-7) was greatly simplified by the above mentioned assumptions

$$v_z = v_{zo}(1 + \epsilon X_a), \quad (3-7)$$

Assumption 7 allows for the elimination of all velocities in the r and θ for average velocity,

$$v_z(r) = \left(1 - \frac{r}{R}\right)^n \left(\frac{v_z(n+1)(2n+1)}{2n^2}\right) + \rho g \quad (3-8)$$

$$\rho = \sum_{i=1}^{N_{cs}} MW_i * C_i, \quad (3-9)$$

for the total density components of Equation (3-9).

It should be noted that application of the algebraic approximations above allows significantly greater flexibility to the model. Recent correlation by Vortmeyer and Gunn (Vortmeyer and Schuster 1983) can be applied to account for radial variations in velocity, density and porosity.

3.2.3 Momentum balances.

Only the axial (z) component of the momentum equations is considered in this work. Assumption 7 and the magnitude of axial flow in industrial reactors allows the elimination of radial (r) and theta (θ) momentum components. It should be noted that algebraic expression for velocity are used Equation (3-7).

High flow rates cause the magnitude of flow in the z direction to dominate that in the r and θ directions. Only the z component of the momentum equations is further considered. (Bird et al. 1960) This leaves only the generalized z component of the momentum balances:

Momentum balance (z component):

$$v_r \frac{\partial}{\partial r}(\rho v_z) + \frac{v_\theta}{r} \frac{\partial}{\partial \theta}(\rho v_z) + v_z \frac{\partial}{\partial z}(\rho v_z) = -\frac{\partial P}{\partial z} + \mu \left(\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial v_z}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 v_z}{\partial \theta^2} + \frac{\partial^2 v_z}{\partial z^2} \right) + \rho g_z \quad (3-10)$$

Where Equation (3-10) was greatly simplified with the above mentioned assumptions. Application of assumption 7 again, allows the elimination of all velocities in the r and θ directions, reducing the equation to

$$v_z \frac{\partial}{\partial z} (\rho v_z) = -\frac{\partial P}{\partial z} + \bar{\mu}_e \left(\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial v_z}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 v_z}{\partial \theta^2} + \frac{\partial^2 v_z}{\partial z^2} \right) + \rho g_z. \quad (3-11)$$

Note that the viscosity term μ has been replaced with an effective viscosity $\bar{\mu}_e$ to account for non-ideal effects cause by uneven flow and other turbulent issues. Application of effective transport properties, gravity and symmetry reduce Equation (3-11) to:

$$v_z \frac{\partial}{\partial z} (\rho v_z) = -\frac{\partial P}{\partial z} + \bar{\mu}_{er} \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial v_z}{\partial r} \right) + \bar{\mu}_{ez} \frac{\partial^2 v_z}{\partial z^2}. \quad (3-12)$$

Equations (3-3) and (3-12) are complex and time consuming to solve. Standard algebraic approximations are used instead. The effects of Equation (3-12) on solution complexity and computation time has been greatly studied and reviewed (Himmelblau and Bischoff 1968; Froment and Bischoff 1979; Nauman 1987). These researchers commonly applied parabolic or high order algebraic approximations for the velocity terms. As was stated with the overall continuity equation, this model will approximate the velocity using Equation (3-7) for gas phase systems and will assume a constant velocity for liquid phase systems.

3.2.4 Thermal energy balance

Non-isothermal systems require the addition of the thermal energy balance equation: (neglecting viscous dissipation terms)

$$c_p \left(v_r \frac{\partial}{\partial r} (T\rho) + \frac{v_\theta}{r} \frac{\partial}{\partial \theta} (T\rho) + v_z \frac{\partial}{\partial z} (T\rho) \right) = k_e \left(\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial T}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 T}{\partial \theta^2} + \frac{\partial^2 T}{\partial z^2} \right) + S_r. \quad (3-13)$$

Where S_r is the generation term defined as:

$$S_r = \Delta H_{rxn} R_i \quad (3-14)$$

Equation (3-13), much like the others, is simplified using the stated assumptions.

Symmetry considerations and assumption 7 allow the elimination of r and θ velocity components as well as θ derivatives:

$$c_p v_z \frac{\partial}{\partial z}(T\rho) = \tilde{k}_{er} \left(\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial T}{\partial r} \right) \right) + \tilde{k}_{ez} \frac{\partial^2 T}{\partial z^2} + S_r \quad (3-15)$$

Application of assumption 1 (constant physical properties) and expansion of the r derivatives result in the final energy equation implemented:

$$c_p v_z \frac{\partial}{\partial z}(T\rho) = \tilde{k}_{er} \left(\frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} \right) + \tilde{k}_{ez} \frac{\partial^2 T}{\partial z^2} + S_r \quad (3-16)$$

Table 3-1 summarizes the resulting set of model equations.

Table 3-1 Generalized Fixed Bed Reactor Equations

Overall mass balance

$$\rho \frac{\partial v_z}{\partial z} + v_z \frac{\partial \rho}{\partial z} = 0$$

Component mass balance

$$v_z \frac{\partial C_a}{\partial z} = \tilde{D}_{ar} \left(\frac{\partial^2 C_a}{\partial r^2} + \frac{1}{r} \frac{\partial C_a}{\partial r} \right) + \tilde{D}_{az} \left(\frac{\partial^2 C_a}{\partial z^2} \right) + R_a$$

Z-velocity component

$$v_z \frac{\partial}{\partial z}(\rho v_z) = -\frac{\partial P}{\partial z} + \tilde{\mu}_{er} \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial v_z}{\partial r} \right) + \tilde{\mu}_{ez} \frac{\partial^2 v_z}{\partial z^2}$$

Energy balance

$$c_p v_z \frac{\partial}{\partial z}(T\rho) = \tilde{k}_{er} \left(\frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} \right) + \tilde{k}_{ez} \frac{\partial^2 T}{\partial z^2} + S_r$$

3.2.5 Non-dimensionalization

Now that the basic equations for the model are determined, non-dimensionalization makes them more applicable to a wider range of applications and greatly reduces code complexity. Non-dimensionalization on the equations in Table 3-2 is done following the commonly used methods of Froment and Bischoff (Froment and Bischoff 1979). Table 3-2 defines all dimensionless groups used within the model.

The dimensionless groups when combined with the model equations in Table 3-2 provide the final form of the model equations.

Table 3-2 Dimensionless Groups Used in Non-Dimensionalization

$\bar{C} = \frac{C_i}{C_{a0}}$	$\bar{v}_z = \frac{v_z}{v_{z0}}$	$\bar{r} = \frac{r}{R}$	$\bar{z} = \frac{z}{l}$	$P = \frac{(P - P_0)}{V_{z0}^2 \rho}$
$\eta = \frac{l}{d_p}$	$m = \frac{R}{d_p}$	$\sigma = \frac{l}{R}$	$N_{fr} = \frac{v_z}{g_z R}$	$N_{re} = \frac{d_p v_z}{\mu}$
$P_{ez} = \frac{d_p v_z}{\bar{D}_{az}}$	$P_{er} = \frac{d_p v_z}{\bar{D}_{ar}}$	$\bar{T} = \frac{T}{T_0}$		

The non-dimensional mass balance equation becomes

$$\frac{\partial \bar{C}}{\partial \bar{z}} = \frac{\sigma}{P_{er} m} \left(\frac{\partial^2 \bar{C}}{\partial \bar{r}^2} + \frac{1}{\bar{r}} \frac{\partial \bar{C}}{\partial \bar{r}} \right) + \frac{1}{P_{ez} \eta} \left(\frac{\partial^2 \bar{C}}{\partial \bar{z}^2} \right) + \frac{R_a l}{v_{z0} C_0} \quad (3-17)$$

The non-dimensional overall mass balance equation becomes

$$\frac{\partial}{\partial \bar{z}} (\bar{\rho} \bar{v}_z) = 0. \quad (3-18)$$

The non-dimensional z momentum balance equation is

$$\frac{\partial \bar{v}_z}{\partial \bar{z}} = -\frac{\partial \bar{P}}{\partial \bar{z}} + \frac{\sigma}{N_{re} m} \left(\frac{\partial^2 \bar{v}_z}{\partial \bar{r}^2} + \frac{1}{\bar{r}} \frac{\partial \bar{v}_z}{\partial \bar{r}} \right) + \frac{1}{N_{re} \eta} \left(\frac{\partial^2 \bar{v}_z}{\partial \bar{z}^2} \right) + \frac{\sigma}{N_{fr}}. \quad (3-19)$$

The non-dimensional energy balance equation is

$$\frac{\partial \bar{T}}{\partial \bar{z}} = \frac{1}{P_{erh} m} \left(\frac{\partial^2 \bar{T}}{\partial \bar{r}^2} + \frac{1}{\bar{r}} \frac{\partial \bar{T}}{\partial \bar{r}} \right) + \frac{1}{P_{ezh} \eta} \left(\frac{\partial^2 \bar{T}}{\partial \bar{z}^2} \right) + \frac{S_r l}{T_o v_z C_p \rho}. \quad (3-20)$$

3.3 Effect of dimensionless groups

The dimensionless Equations (3-17) through (3-20) are the basis for the fixed bed model used within this work. Note that the effective diffusion coefficients are now contained in the axial and radial Peclet numbers. The use of effective transport properties was done strictly for comparison to literature and textbook analysis of fixed bed reactor systems (Froment 1967; Froment and Bischoff 1979; Nauman 1987). All report that fixed bed reactors with turbulent flow have axial and radial Peclet numbers that are typically much greater than 10. Highly turbulent flows have Peclet numbers significantly greater than 100. These high Peclet numbers cause the governing equations to asymptotically approach the ideal plug flow reactor equations.

3.4 Boundary conditions

Classical boundary conditions are applied to Equations (3-17) through (3-20) as outlined by (Himmelblau and Bischoff 1968; Froment and Bischoff 1979; Levenspiel 1989). Boundary conditions are applied to four primary regions of the reactor: the reactor entrance, centerline, wall and outlet. We begin by discussing the centerline and wall conditions, followed by the entrance and exit conditions.

Wall conditions are divided into two major types of conditions:

- Perfect slip at the wall (v_z at the wall = v_z of the bulk fluid) and

- No slip at the wall ($v_z=0$ at the wall).

Both types of boundary conditions come about from assumptions placed on component velocities at the wall. Perfect slip conditions assume that the velocity of all species at the wall is non-zero and that the first derivatives of the dependent variables with respect to radius are zero at all locations in the domain. No radial variations of the dependent variable occur in the radial direction. Perfect slip conditions are applicable to plug flow reactors or reactors that are very nearly plug flow. This assumption does not reflect reality, but has historically proven effective for highly turbulent reactors. Figure 3-3 graphically depicts perfect slip conditions and dependent variable profiles.

No-slip conditions (see Figure 3-4) assume that species velocities at the wall are zero and that the radial derivatives of the dependent variables are zero only at the centerline. Non-flat dependent variable profiles are the most significant visible deviation from perfect slip conditions. Results and profiles that are more realistic result from no-slip boundary conditions. This type of boundary condition is most commonly applied to reactors with moderate to large gradients in radial profiles.

Inlet and outlet conditions for slip or no-slip conditions are identical. Classical Danckwerts (Froment 1967; Himmelblau and Bischoff 1968; Froment and Bischoff 1979) boundary conditions are applied to the inlet. Danckwerts conditions are based on the assumption that no diffusion or reaction occurs in the bulk fluid prior to the entrance of the reactor. Outlet boundary conditions assume that the first derivatives in the axial direction are zero at infinity.

Table 3-1 and Table 3-3 form the foundation of the modeled fixed bed reactor system. All that remains to be provided are physical property equations and other supporting equations. The remainder of this chapter covers these equations.

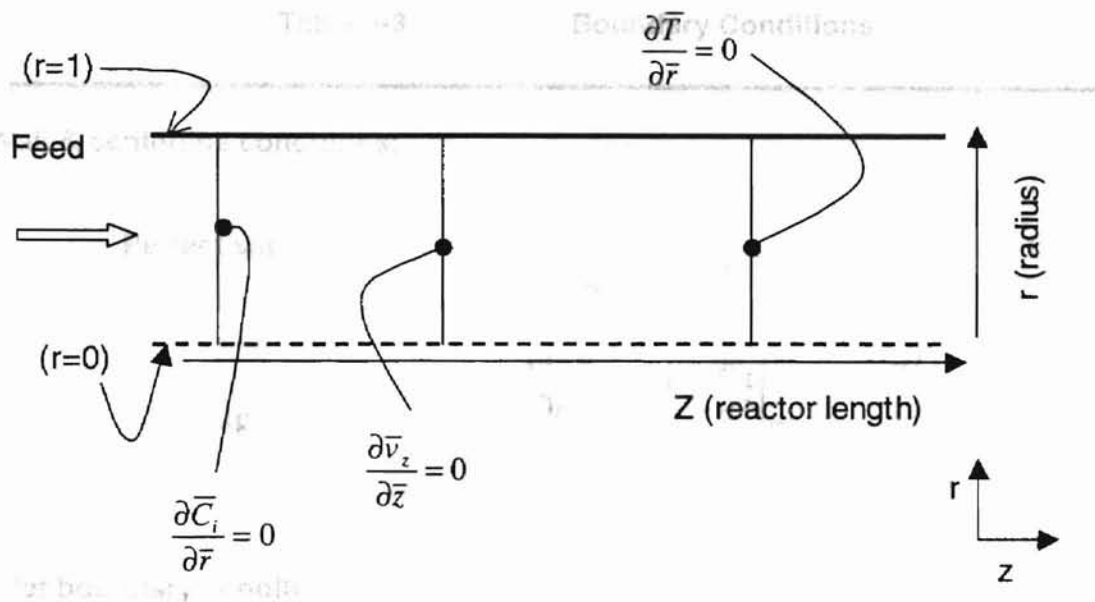


Figure 3-3 Boundary conditions & profiles (slip conditions).

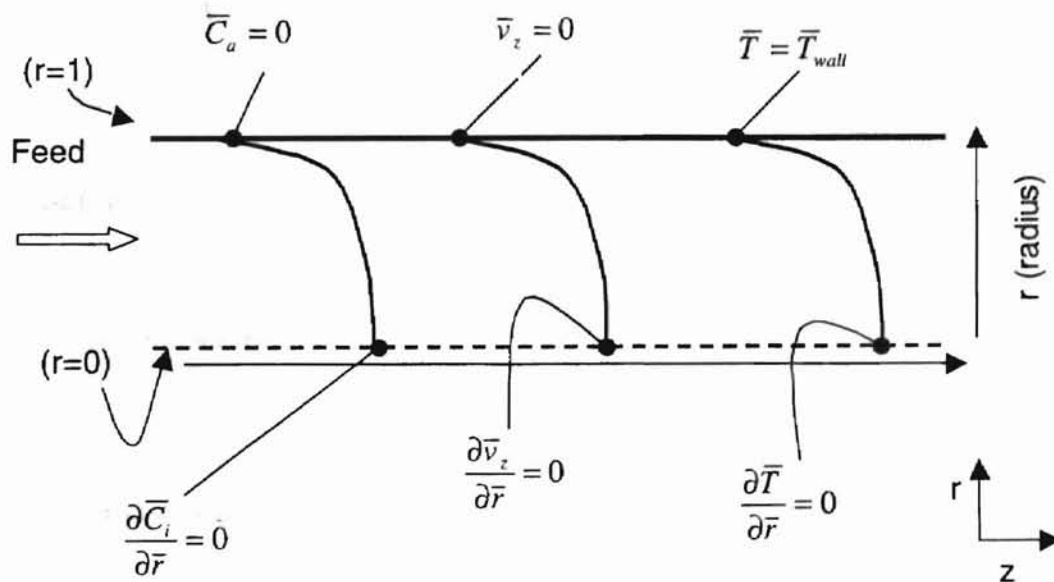


Figure 3-4 Boundary conditions & profiles (no slip conditions).

Table 3-3 Boundary Conditions

Wall & centerline conditions:

Perfect slip

$$\left. \frac{\partial \bar{C}_i}{\partial \bar{r}} \right|_{\bar{r}=\text{all}} = 0, \quad \left. \frac{\partial \bar{v}_z}{\partial \bar{r}} \right|_{\bar{r}=\text{all}} = 0, \quad \left. \frac{\partial \bar{T}}{\partial \bar{r}} \right|_{\bar{r}=\text{all}} = 0$$

No slip

$$\left. \frac{\partial \bar{C}_i}{\partial \bar{r}} \right|_{\bar{r}=0} = 0, \quad \left. \frac{\partial \bar{v}_z}{\partial \bar{r}} \right|_{\bar{r}=0} = 0, \quad \left. \frac{\partial \bar{T}}{\partial \bar{r}} \right|_{\bar{r}=0} = 0$$

$$v_z|_{\bar{r}=\text{wall}} = 0, \quad \bar{T}|_{\bar{r}=\text{wall}} = \bar{T}_{\text{wall}}$$

Inlet boundary conditions:

Concentration

$$\bar{D}_{ez} \left. \frac{\partial \bar{C}_i}{\partial \bar{r}} \right|_{\bar{r}=0, \bar{z}=0} = \bar{v}_z \bar{C}_i|_{\bar{r}=\text{all}, \bar{z}=0} - \bar{v}_z(\bar{r}) \bar{C}_{io}$$

Velocity

$$\bar{C}_i|_{\bar{r}=\text{all}, \bar{z}=0} = C_{io}$$

$$\bar{v}_z|_{\bar{r}=\text{all}, \bar{z}=0} = \bar{v}_{zo}$$

Temperature

$$\bar{T}|_{\bar{r}=\text{all}, \bar{z}=0} = \bar{T}_o$$

Outlet boundary conditions:

Concentration

$$\left. \frac{\partial \bar{C}_i}{\partial \bar{z}} \right|_{\bar{r}=\text{all}, \bar{z}=\infty} = 0$$

Velocity

$$\left. \frac{\partial \bar{v}_z}{\partial \bar{z}} \right|_{\bar{r}=\text{all}, \bar{z}=\infty} = 0$$

Temperature

$$\left. \frac{\partial \bar{T}}{\partial \bar{z}} \right|_{\bar{r}=\text{all}, \bar{z}=\infty} = 0$$

3.5 Support equations

Accurate modeling of any system requires physical property calculations.

Equations for pressure drop, effective transport properties, ideal heat capacities, and enthalpies of reaction are established in the remainder of this chapter, beginning with pressure drop calculations.

3.5.1 Pressure drop

The most commonly used equation for pressure drop calculations is the Ergun equation (Ergun and Orning 1949; Froment and Bischoff 1979):

$$\frac{\partial P}{\partial z} = \frac{v_z^2 \rho_f}{d_p} \left(\frac{1-\epsilon}{\epsilon^3} \right) \left(1.75 + \frac{150(1-\epsilon)}{N_{re}} \right). \quad (3-21)$$

The Ergun equation treats the fixed bed as a system of bundled non-connecting fluid channels between the catalyst particles. This equation has shown wide flexibility in the prediction of pressure drop but tends to fail in regions of high turbulence (Hicks and Mandersloot 1968; Hicks 1970; Froment and Bischoff 1979). The Ergun equation is limited to Reynolds numbers less than 500. In an effort to provide an equation that would accurately predict pressure drop at both low and high Reynolds numbers, Hicks developed a new equation (in non-dimensional form)

$$\frac{\partial \bar{P}}{\partial \bar{z}} = \frac{lv_z^2 \rho_f}{P_a d_p} \left(6.8 \frac{(1-\epsilon)^{1.2}}{\epsilon^3} \right) N_{re}^{-0.2}. \quad (3-22)$$

Since the developed model must be able to operate over a very wide range of operating conditions, the Hick's equation is used. The Hick's modification of the Ergun equation provides predictions over a much greater range of Reynolds numbers. Hicks performed a detailed study of many pressure drop equations. Hick's equation provided

excellent predictions when tested against the Ergun and the Handley and Hegg's equation for very high Reynolds numbers.

3.5.2 Effective transport properties

The model equations developed contain effective transport coefficients that are predicted using semi-empirical methods. These effective transport properties can be determined through tracer tests or predicted theoretically (Gunn 1987). Using second order partial differential equations and thermodynamically predicted physical constants, one can approximate actual reactor products. The method of Gunn allows 'empirical' fine tuning provided that RTD tracer curves are available. The RTD approach is widely used in non-ideal reactor modeling and can be easily found in advanced modeling or reactor design books (Himmelblau and Bischoff 1968; Froment and Bischoff 1979; Nauman 1987; Levenspiel 1989).

The radial dispersion terms, though typically neglected in highly turbulent flow (Nauman 1987), are considered here in order to account for highly exothermic / endothermic conditions that can cause large temperature and concentration gradients in the radial direction. In addition, catalyst flow pathways are highly convoluted (Nauman 1987) making it exceedingly improbable that perfect radial mixing occurs. Axial dispersion terms are also included to help account for axial variations and improve reactor modeling at both the bench scale and industrial scale.

Once reactor performance and tracer curves have been determined, least squares or other error minimization techniques determine the effective diffusivities as dictated by the governing partial differential equations. The resulting equations should predict pulse curves very similar to test results.

Resulting diffusivities are considered identical (Froment and Bischoff 1979) for all chemical species present as the physical effects of real turbulent flow greatly dominate

any molecular dispersion effects. These effective properties account for flow non-idealities such as uneven void fractions, dead spaces and other real-life phenomena encountered. Due to their empirical nature, the resulting diffusivities are applicable primarily to the conditions at which they are determined.

The most recent method for estimation of effective axial and radial coefficients (Gunn 1987) is based on catalyst, fluid and reactor properties. Equation (3-23) shows Gunn's equations in terms of axial Peclet number for fixed bed systems

$$\frac{1}{P_{ez}} = \frac{N_{re} N_{sc}}{4(1-\varepsilon)} (1-p)^2 + \frac{N_{re}^2 N_{sc}^2}{16(1-\varepsilon)^2} p(1-p)^3 \left[\exp\left(\frac{-4(1-\varepsilon)}{p(1-p)N_{re} N_{sc}} - 1\right) \right] + \frac{\varepsilon}{\tau N_{re} N_{sc}} \quad (3-23)$$

The value (p) has been shown (Gunn 1987) to be strictly a function of Reynolds number and catalyst pore structure. Equations are available for common catalyst geometry such as spheres ($\tau=1.4$)

$$p = 0.17 + 0.33 \exp\left(\frac{-24}{N_{re}}\right), \quad (3-24)$$

and cylinders ($\tau=1.93$)

$$p = 0.17 + 0.29 \exp\left(\frac{-24}{N_{re}}\right). \quad (3-25)$$

These equations have been shown to be valid for a wide range of materials and Reynolds numbers (Gunn 1987).

Radial coefficients are determined by Gunn to fit the following equation in terms of the radial Peclet number

$$\frac{1}{P_{er}} = \frac{1}{P_{ef}} + \frac{\varepsilon}{\tau N_{re} N_{sc}}. \quad (3-26)$$

Like his equation for axial terms, Gunn provides a correlation for P_{er} very similar to that for p values. The equations for spheres ($\tau=1.4$) is:

$$P_{ef} = 40 - 29 \exp\left(\frac{-7}{N_{re}}\right), \quad (3-27)$$

and for cylinders ($\tau=1.93$) is:

$$P_{ef} = 11 - 4 \exp\left(\frac{-7}{N_{re}}\right). \quad (3-28)$$

Though the semi-empirical methods outlined by Gunn are useful in estimating axial and radial effects, they are not as accurate as those obtained through RTD tracer curves.

3.5.3 Heat capacity calculations

Average heat capacities are determined using the cubic form of the heat capacity equation:

$$C_{pi} = \alpha_i + \beta_i T + \gamma_i T^2 + \delta_i T^3. \quad (3-29)$$

Coefficient values for Equation (3-29) are widely published in the literature and available from sources like Perry's Chemical Engineer's Handbook, (Bird et al. 1960; Perry et al. 1984) or commercial simulation engines like HYSYS™ or ASPEN™.

3.5.4 Enthalpy of reaction calculations

In order to accurately account for non-isothermal variations in reaction rate and system energy, the heat capacity equations must be used to calculate the heat of reaction. Heat capacity coefficient terms for all components are used to solve Equations (3-30) through (3-33) (Fogler 1986).

$$\Delta\alpha = \frac{d}{a}\alpha_d + \frac{c}{a}\alpha_c + \frac{b}{a}\alpha_b + \alpha_a \quad (3-30)$$

$$\Delta\beta = \frac{d}{a}\beta_d + \frac{c}{a}\beta_c + \frac{b}{a}\beta_b + \beta_a \quad (3-31)$$

$$\Delta\gamma = \frac{d}{a}\gamma_d + \frac{c}{a}\gamma_c + \frac{b}{a}\gamma_b + \gamma_a \quad (3-32)$$

$$\Delta\delta = \frac{d}{a}\delta_d + \frac{c}{a}\delta_c + \frac{b}{a}\delta_b + \delta_a \quad (3-33)$$

Equations (3-30) through (3-34) are of the form (reactants – products) and care should be taken to insure that stoichiometric coefficients are properly entered. Values calculated from Equations (3-30) through (3-33) were used to solve the overall heat of reaction equation:

$$\begin{aligned} \Delta H_R(T) = & \Delta H_R^c(T_R) + \Delta\alpha(T - T_R) + \frac{\Delta\beta}{2}(T^2 - T_R^2) \\ & + \frac{\Delta\gamma}{3}(T^3 - T_R^3) + \frac{\Delta\delta}{4}(T^4 - T_R^4) \end{aligned} \quad (3-34)$$

Equation (3-34) is the numerical integration for the heat of reaction from the reference temperature to the current system temperature. For non-isothermal systems, heat capacities and heats of reaction are updated at each loop of the integration engine.

3.6 Complete definition of system

To insure complete definition of the generic fixed bed reactor, an analysis is done to determine the number of equations required. For the generic case, the above mentioned phenomena and the following items must be accounted for to insure accurate model solutions:

- species material balances for all components,
- reaction expressions for each reaction present,
- overall momentum equations and
- overall energy balances.

In order for the developed model equations to be properly defined and solvable, the number of unknowns is determined by Equation (3-35):

$$N_{cc} - N_{rxn} + N_{vel} + N_{heat} = N_{eq} \quad (3-35)$$

For the case of one reaction, three chemical species and z momentum only, six equations are required for complete definition and solution. Test systems consisting of one reaction and three chemical species, provide the reaction equation and stoichiometric relations required begin solution. Determination of the remaining equations requires assumptions on the basic mass, velocity and energy equations. Solution of the developed model and description of the model test reactions are detailed in chapter 4.

Chapter 4

NUMERICAL METHODS

In this Chapter the generalized partial differential equations presented in Table 3-1 of Chapter 3 were adapted for numerical solution using the PDETWO solver. The developed computer code is described and algorithm description is divided into four primary sections:

- PDETWO description,
- Code flow and main routine,
- boundary conditions,
- equation definition & kinetics and
- physical property calculations.

We begin with an overview of the PDETWO code and presentation of the overall model flow chart.

4.1 PDETWO code

PDETWO (Melgaard and Sincovec 1981) is designed to solve systems of partial differential equations defined over a region using an ordinary differential equation integration package. Once the rectangular domain equations are defined, PDETWO forms and evaluates a discrete approximation of the partial differential equations using the method of lines (see Chapter 2). PDETWO is capable of solving partial differential equations like those given in Equation (4-1),

$$\frac{\partial u_i}{\partial t} = f_i(t, x, y, u_1, u_2, u_3, \dots, u_{npde}, \frac{\partial u_i}{\partial x}, \dots, \frac{\partial u_{npde}}{\partial x}, \frac{\partial u_i}{\partial y}, \dots, \frac{\partial u_{npde}}{\partial y}),$$

$$\frac{\partial}{\partial x} \left(DH_{i,1} \frac{\partial u_1}{\partial x} \right) \dots \frac{\partial}{\partial x} \left(DH_{i,1} \frac{\partial u_{npde}}{\partial x} \right) \quad \frac{\partial}{\partial y} \left(DV_{i,1} \frac{\partial u_1}{\partial y} \right) \dots \frac{\partial}{\partial y} \left(DV_{i,1} \frac{\partial u_{npde}}{\partial y} \right) \quad (4-1)$$

subject to constraints as outlined in Equation (4-2):

$$a_1 < x < b_1, \quad a_2 < y < b_2, \quad t > t_0, \quad i = 1, 2, \dots, NPDE \quad (4-2)$$

PDETWO consists of eight internal routines and five user provided routines.

Subroutines DRIVEP and PSETM are the primary routines used for generating solutions.

DRIVEP is a modified version of GEARB designed to handle and generate banded Jacobean matrices. The Jacobian can either be determined by the routine, or user supplied.

There are two integration options for the user: Adams-Mouton and Gears method. The routine can also be set to determine the Jacobean internally or the user can supply. PSETM is used in conjunction with DRIVEP to decrease computation time when the Jacobean is internally generated. STRSET, STIFFP, INTERP, COSET, DECBR and SOLBR are internal subroutines that allocate array sizes, integrate the system, and solve the generated Jacobean.

These routines are well defined in publications by Melgaard & Sincovec (Melgaard and Sincovec 1981; Melgaard and Sincovec 1981), and will not be discussed here. The user is only required to provide subroutines to define the spatial mesh, differential equations, diffusion coefficients, and boundary conditions. The main routine serves to call PDETWO, initialize variables, and define spatial integration mesh.

4.2 Algorithm flow

One of Conoco's requirements was that the developed model code be designed to allow easy integration into other programs such as HYSYS™, ASPEN PLUS™ or

Excel™. To meet this objective the program was compiled as a FORTRAN dynamic link library (DLL), allowing it to be called by any Windows™ program with little modification or no modification. See Appendix B for a detailed description of how to access the DLL from outside programs such as HYSYS™ and Excel™.

Programs such as HYSYS™ ASPEN PLUS™ or Excel™ call the model using a rather simple visual basic interface. Critical system data and variables are passed to the main routine for processing and equation solution. Figure 4-1 graphically demonstrates the overall solution pathway implemented for fixed bed reactor solutions using PDETWO. The main routine initializes all system variables, work arrays, assigns initial data provided from the calling program, and calls the PDETWO to complete equation solution. PDETWO solves the partial differential equations and calls the routines for calculating boundary conditions, diffusion parameters, and the partial differential equation function. At each iteration of PDETWO, the main routine updates physical properties and continues solution.

4.3 Main routine

The main routine initializes all variables and work arrays. System characteristics and physical constants from the calling program are assigned and the solution domain is established. PDETWO requires the user to supply the required number of solution points in both the z and r directions as well as the initial step size for the integrator.

Using user supplied data and solution requirements the code divides the domain into uniform sections and allocates memory to generate solutions at all nodal intersections (see Figure 4-2) as is required by PDETWO.

Chemical Reactor Simulator for non-ideal fixed bed reactor

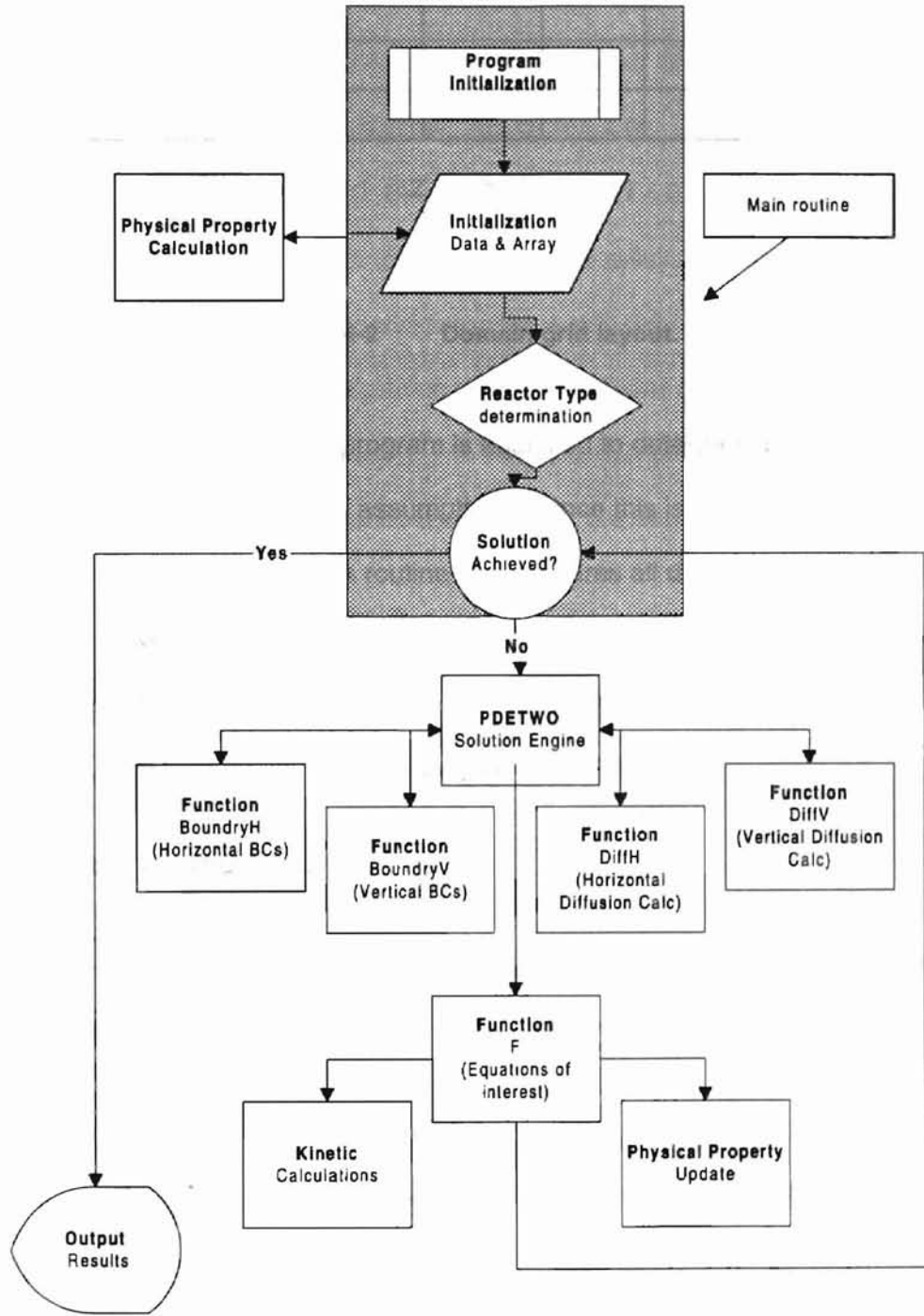


Figure 4-1 PDETWO fixed bed solution flow chart.

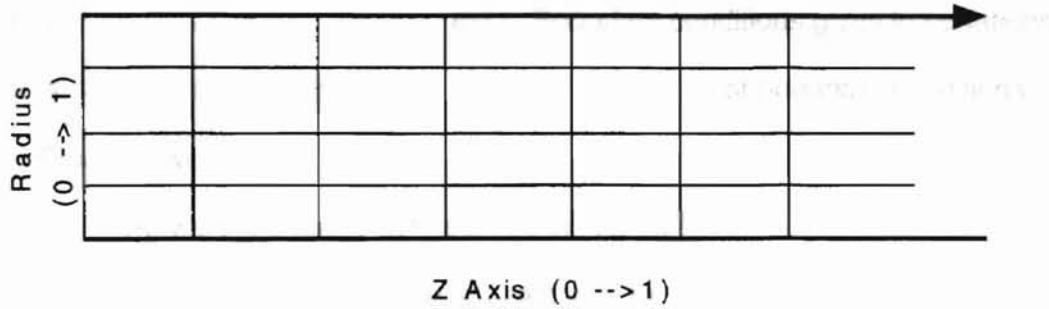


Figure 4-2 Domain grid layout.

Input data from the calling program is analyzed to determine boundary conditions, reactor type and critical assumptions. Once this is completed the main routine calls the physical properties routine and prepares all data for solution. The main routine then checks if the system has been solved. If no solution has been achieved, PDETWO is called and solution progresses. PDETWO begins solution using the boundary, diffusion, kinetic and function evaluation routines.

4.4 Boundary conditions

The coupled sets of partial differential Equations (4-1) are subject to horizontal (4-3) and vertical (4-4) boundary condition of the form:

$$AH_i u_i + BH_i \frac{\partial u_i}{\partial y} = CH_i \quad (4-3)$$

$$a_1 \leq x \leq b_1 \quad y = a_2 \text{ or } y = b_2 \quad i = 1, 2, \dots, NPDE, \quad t > t_o$$

$$AV_i u_i + BV_i \frac{\partial u_i}{\partial x} = CV_i \quad (4-4)$$

$$a_2 \leq y \leq b_2 \quad x = a_1 \text{ or } x = b_1 \quad i = 1, 2, \dots, NPDE, \quad t > t_o$$

BNDRYH, BNDRYZ subroutines are used to define the horizontal (x) and vertical (y) boundary conditions. The coefficients AH, AV, BH, BV, CH and CV are user definable and can be functions of x, y, u and t. Boundary conditions given in equations (4-3) and (4-4) allow the user to specify the three major types of boundary conditions.

1) Dirichlet, where the dependent variable is defined: BH_i or $BV_i = 0$

2) Neuman, where the flux is defined: AH_i or $AV_i = 0$

3) Mixed, defined as: $AH_i \neq 0$, $BH_i \neq 0$, $AV_i \neq 0$, $BV_i \neq 0$

It should be noted that the functions need to be at least piecewise continuous. In addition, the number of boundary conditions must match the number of equations, defined over the entire range of integration. Initial conditions do not need to be consistent with the boundary conditions. For example flow prior to the reaction with no slip at the wall, and slip at the wall once the fluid has reached the reactor inlet would be allowed.

PDETWO first evaluates the boundary conditions over the solution domain. Boundary conditions used for fixed bed reactor simulations are passed to the main routine and evaluated at each iteration of PDETWO. The user supplied horizontal and vertical boundary condition code implemented uses classical Danckwerts boundary conditions applied to the inlet of the reactor ($z=0.0$). Boundary conditions (Table 3-3) are then applied at the centerline ($r=0$) and at the wall ($r=1$).

4.4.1 Diffusion coefficients

DIFFH, DIFFV define and update the effective horizontal and vertical diffusivities for the diffusion terms of the partial differential equations. These coefficients can be constants or functions of z, r, u and t. As with the boundary conditions, these functions must be at least piecewise continuous over the entire integration domain.

4.5 Equation definition, kinetics and calculations

Subroutine F calls the physical property and kinetic routines prior to evaluation of the reactor equations. Physical property calculations and kinetic calculations are performed prior to each iteration loop. Since it is assumed that physical properties are constant over each integration slice, they must be updated between slices to minimize error. Heat capacity, density, viscosity, heat of reaction, and total mass is determined with subroutine "props90". Prior to reactor evaluation, reaction rates are updated via a user-supplied subroutine. All steady state equations with second order z terms are transformed into a set of coupled first order differential equations, allowing second order equations in z to be solved at steady state with PDETWO.

Table 4-1 shows the transformed equations used. The equations were entered in the subroutine in much the same manner as they are written (see appendix subroutine F). Boundary conditions are entered as show in Table 3-1 of the model development section. Resulting PDE equations from Table 4-1 are implemented into the function routine of PDETWO and routines are added for the calculation of kinetic parameters and physical properties.

The resulting program provides the means for numerical solution of all generated PDEs, allowing simulation and analysis of model assumptions, model feasibility and reactor performance under a wide range of conditions. Chapter 5 presents and reviews this analysis.

Table 4-1

Equation Implementation

PFR Equation	Component	$\frac{\partial \bar{C}_i}{\partial \bar{z}} = \frac{R_i l}{v_{zo} C_{zo}}$
	Energy	$\frac{\partial \bar{T}}{\partial \bar{z}} = \frac{R_i (-\Delta H_{rxm}) l}{T_o v_z C_p \rho}$
Axial only	Component	$\frac{\partial \bar{C}_i}{\partial \bar{z}} = w$
		$\frac{\partial w_i}{\partial \bar{z}} = (P_{ez} \eta) \left(w - \frac{R_i l}{v_{zo} C_{zo}} \right)$
	Energy	$\frac{\partial \bar{T}}{\partial \bar{z}} = \zeta$
		$\frac{\partial \zeta}{\partial \bar{z}} = (P_{ezh} \eta) \left(\zeta - \frac{R_i (-\Delta H_{rxm}) l}{T_o v_z C_{pav} \rho} \right)$
Radial	Component	$\frac{\partial \bar{C}_i}{\partial \bar{z}} = \frac{\sigma}{P_{er} m} \left(\frac{\partial^2 \bar{C}_i}{\partial \bar{r}^2} + \frac{1}{\bar{r}} \frac{\partial \bar{C}_i}{\partial \bar{r}} \right) + \frac{R_i l}{v_{zo} C_{zo}}$
	Energy	$\frac{\partial \bar{T}}{\partial \bar{z}} = -\frac{1}{P_{erh} m} \left(\frac{\partial^2 \bar{T}}{\partial \bar{r}^2} + \frac{1}{\bar{r}} \frac{\partial \bar{T}}{\partial \bar{r}} \right) + \frac{R_i (-\Delta H_{rxm}) l}{T_o v_z C_{pav} \rho}$
Axial & Radial Component		$\frac{\partial \bar{C}_i}{\partial \bar{z}} = w$
		$\frac{\partial w_i}{\partial \bar{z}} = P_{ez} \eta \left(w - \frac{\sigma}{P_{er} m} \left(\frac{\partial^2 \bar{C}_i}{\partial \bar{r}^2} + \frac{1}{\bar{r}} \frac{\partial \bar{C}_i}{\partial \bar{r}} \right) - \frac{R_i l}{v_{zo} C_{zo}} \right)$
	Energy	$\frac{\partial \bar{T}}{\partial \bar{z}} = w$
		$\frac{\partial \zeta}{\partial \bar{z}} = P_{ezh} \eta \left(\zeta - \frac{1}{P_{erh} m} \left(\frac{\partial^2 \bar{T}}{\partial \bar{r}^2} + \frac{1}{\bar{r}} \frac{\partial \bar{T}}{\partial \bar{r}} \right) + \frac{R_i (-\Delta H_{rxm}) l}{T_o v_z C_{pav} \rho} \right)$

Chapter 5

Results & Discussion

This Chapter evaluates the performance and accuracy of the mathematical models developed in Chapter 3 for fixed bed reactors. The analysis performed included an evaluation of integration step size effect, accuracy of mass balance, and sensitivity to input variables. Model equations were then implemented on the gas phase ethane cracking and liquid phase MTBE synthesis reactions under isothermal and adiabatic conditions to evaluate limitations and feasibility. Results were compared to HYSYS™ simulations or experimental data when available. Finally, the model code is used to evaluate the concentration based MTBE kinetics of Al-Jarallah et al. (1988) under fixed bed conditions.

5.1 PDETWO solution test (analytic comparison)

Prior to implementation of the model equations, PDETWO results were compared to known analytical results for a given test equation. Equation (5-1) represents a second order differential equation as given in example problem 9-3 of Hornbeck (1975):

$$\begin{aligned} \frac{d^2 y}{dt^2} + 2 \frac{dy}{dt} + 4y &= 0, \\ y(0) = 2, \quad \frac{dy}{dt}(0) &= 0. \end{aligned} \tag{5-1}$$

Solution of Equation (5-1) using PDETWO required transformation of the second order equation into two-coupled first order equations. Hornbeck outlines the transformed equations and boundary conditions as shown in Equation (5-2)

$$\frac{dz}{dt} = -2z - 4y, \quad z(0) = 0. \quad (5-2)$$

$$\frac{dy}{dt} = z, \quad y(0) = 2$$

Table 5-1 and Figure 5-1 compares analytical and PDETWO solutions. PDETWO solutions were generated using the equations and boundary conditions as outlined by Hornbeck, with an initial step size of 0.001 and the variable order Adams-Moulton method integration method. Note that PDETWO was able to match the analytical solution to within 0.004%, clearly demonstrating excellent accuracy. Additional accuracy comparisons for multi dimensional unsteady state equations can be found in published literature on PDETWO (Melgaard and Sincovec 1981).

Table 5-1 Analytical Comparison to PDETWO Results

Time	Analytic	PDE2	% Error
0.0	2.000	2.0000	0.000
0.5	1.319	1.3194	0.000
1.0	0.3011	0.3011	-0.005
1.5	-0.2487	-0.2497	-0.099
2.0	-0.3062	-0.3062	0.004
2.5	-0.1491	-0.1492	-0.002
3.0	-0.0045	-0.0046	-0.002
3.5	0.0512	0.0513	0.002
4.0	0.0419	0.0420	0.001
4.5	0.0141	0.0141	-0.003
5.0	-0.0043	-0.0043	0.004

5.2 Reactor model testing

The liquid phase MTBE synthesis reaction and ethane cracking gas phase reactions were used to test the reactor model. The ethane cracking reaction served as a basis for determining model performance and range of operations under rather extreme

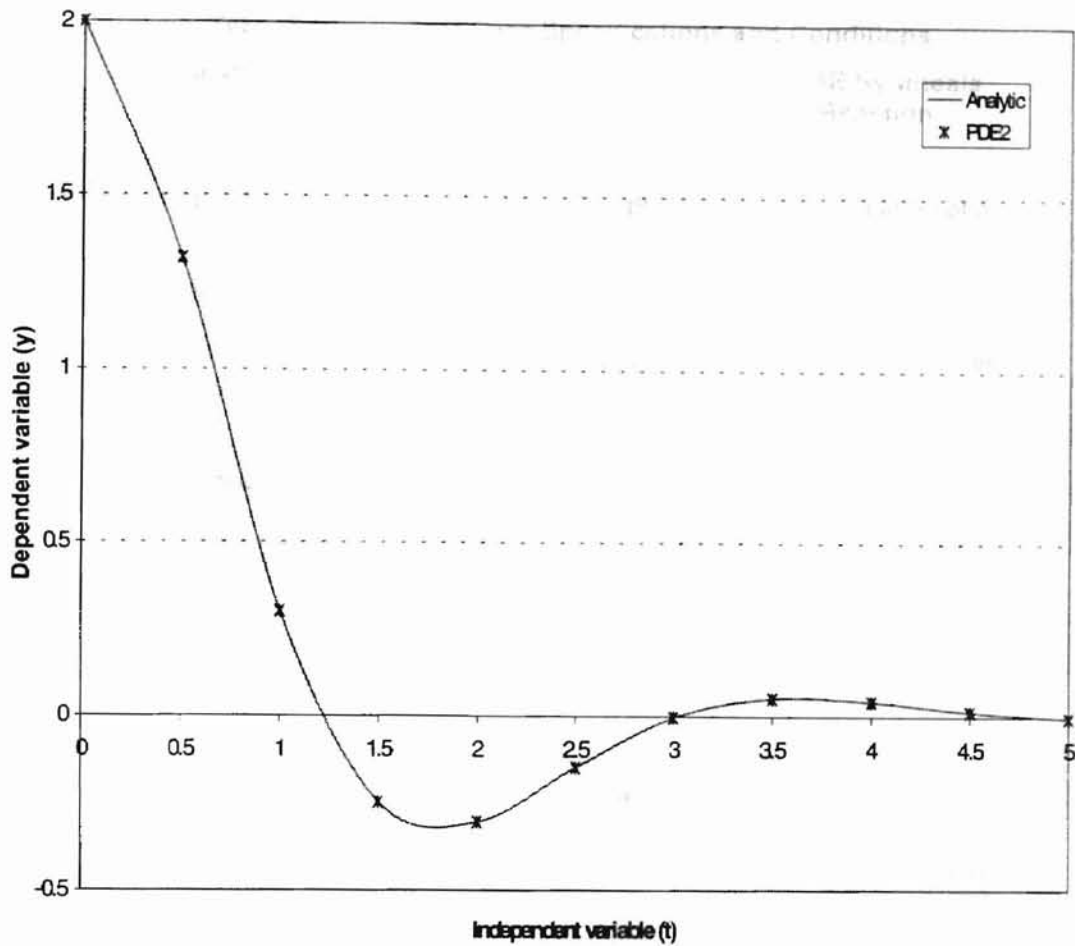


Figure 5-1 Comparison between PDETWO and analytic solution.

gas phase conditions. The MTBE synthesis reaction provided insight into model flexibility and allowed evaluation of Rideal-Eley kinetics in fixed bed reactors. Test reactions were also used to test model performance at asymptotic limits of length, flow rate and Peclet numbers. Both test reactions were compared with HYSYS™ (using the isothermal PFR model) simulations or experimental data if available. Input data and reactor specifications for the test cases are given in Table 5-2.

Table 5-2 Test Case Specifications and Conditions		
Specification	Ethane Cracking Reaction	MTBE Synthesis Reaction
Inlet conditions		
Temp (T_o)	Gas phase 1100 K	Liquid phase 333 - 353 K
Flow (F_o)	0.1927 kmol/s	Variable
Pressure (P_o)	6 atm	18 atm
Conc. (C_a)	0.06649 kmol/m ³	0.1 mol/m ³
Conc. (C_b)	0.0 kmol/m ³	0.1 mol/m ³
Conc. (C_c)	0.0 kmol/m ³	0.0 mol/m ³
Reactor		
Diameter (d_i)	0.4877 m	0.00953 m
Length (l)	12.192 m	0.333 m
Tube # (N_{tube})	1.0	1.0
Wall (T_w)	1100 K	313 - 353 K
Void % (ϵ)	100 %	32.8 %
Mass catalyst (m_c)	NA	1 to 4 grams
Particle size (d_p)	NA	0.74 mm
Reaction		
Type	Irreversible	Reversible
Kinetics	Power law	Rideal-Eley
Details	See Section 2.3.1	See Section 2.3.2

5.2.1 Plug flow reactor benchmark

Plug flow model results for the ethane cracking reaction were similar to results obtained from HYSYS™ simulations (see Table 5-3 and Table 5-4). HYSYS™ results predict an 83.1% ethane conversion compared to the 82.4% predicted with our PFR model. HYSYS™ and PFR model results predict almost identical concentration profiles with only a slight offset in profiles (Figure 5.2).

Table 5-3 HYSYS™ Simulation Results, Ethane Cracking PFR Reaction

Dimensionless Length	Ethane	Ethylene	Hydrogen
0.000	100.00%	0.00%	0.00%
0.025	81.07%	9.46%	9.46%
0.125	56.53%	21.73%	21.73%
0.225	41.45%	29.28%	29.28%
0.325	31.47%	34.26%	34.26%
0.425	24.60%	37.70%	37.70%
0.525	19.71%	40.15%	40.15%
0.625	16.15%	41.93%	41.93%
0.725	13.49%	43.25%	43.25%
0.825	11.46%	44.27%	44.27%
0.925	9.87%	45.06%	45.06%
1.000	9.21%	45.40%	45.40%

Table 5-4 Model Simulation Results, Ethane Cracking PFR Reaction

Dimensionless Length	Ethane	Ethylene	Hydrogen
0.0	100.00%	0.00%	0.00%
0.1	66.08%	16.96%	16.96%
0.2	47.79%	26.11%	26.11%
0.3	36.41%	31.79%	31.79%
0.4	28.70%	35.65%	35.65%
0.5	23.17%	38.41%	38.41%
0.6	19.03%	40.48%	40.48%
0.7	15.84%	42.08%	42.08%
0.8	13.32%	43.34%	43.34%
0.9	11.29%	44.36%	44.36%
1.0	9.63%	45.18%	45.18%

It is unlikely that the selected thermodynamics (in both cases ideal gas law) would be the primary difference in concentration profiles for the ethane cracking reaction. Both models were set to use the Ideal Gas Law for thermodynamic calculations. Additionally, if the problem was in thermodynamic calculation, the measure offset should remain the same and not correct itself down the length of the reactor.

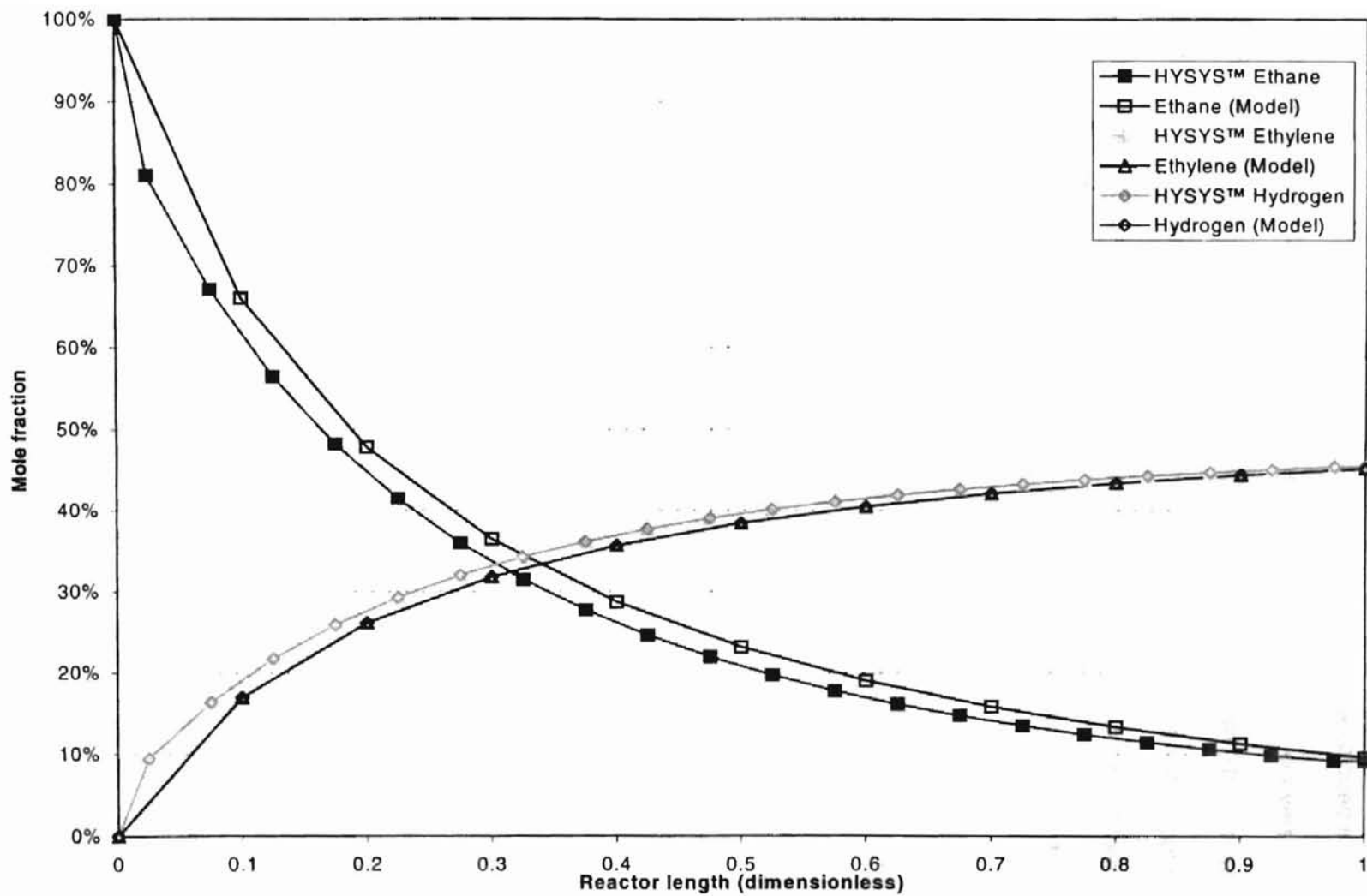


Figure 5-2 Mole fraction comparisons of HYSYS™ simulation model for the ethane cracking reaction.

Calculation methods for velocity and density profiles could play a partial role in the concentration profile miss-match. The mathematical model developed in Chapter 3 assumed velocity and density properties are uncoupled functions of conversion. If HYSYS™ treats these properties as coupled functions, solutions will differ most at the front of the reactor where conversions are the highest. It should be noted that no documentation on how HYSYS™ calculates these properties was found. More accurate results in the gas phase could be obtained if these values were coupled into the set of differential equations.

It is thought that the primary discrepancy in concentration profiles could be attributed to differences at the boundary conditions. Models developed within this work are based on the assumption that the inlet boundary is static (not effected by diffusion) and all first derivatives of the dependent variables go to zero at infinite length. Notice that predicted concentration profiles match closely at the outlet of the reactor. If the reactor is run to infinite length, both sets of solutions approach the same values. At the inlet of the reactor, there is significant variance in predicted profiles. Attempts to determine what HYSYS™ uses for inlet boundary conditions were unsuccessful.

The PFR results presented show that the model is able to effectively handle the complex gas phase ethane crack reaction kinetics with results almost identical to those obtained by the commercial simulation engines. Now that ideal prediction can be obtained, the established model must be tested under extreme non-ideal conditions to determine how effective it will be for scale-up and real reactor predictions

5.2.2 Asymptotic comparison

Asymptotic analysis allowed evaluation of model limitations and feasibility of predicted results. The reactor model equations asymptotically approach the ideal PFR equation as the axial and radial Peclet numbers increase. Using the ethane cracking

reaction, the models were run at high Peclet numbers ($P_{ez} = P_{er} = 1000$) and then at very low Peclet numbers. The upper value for Peclet numbers was determined by increasing the value until there was no perceivable change in model solutions. Low Peclet numbers were set at 1 as this value is outside of the lowest reasonable Peclet value (typically 5-10). Table 5-5 and Figure 5-3 show that at the asymptotic limit all the model types provide results identical to the PFR model.

Table 5-5 Asymptotic Comparison of Mixing Cup Concentration Profiles for the Ethane Cracking Reaction at High Peclet Numbers.

Z length	Isothermal PFR	Isothermal Axial	Isothermal Radial	Isothermal Axial & Radial
0.0	1.0000	1.0000	1.0000	1.0000
0.1	0.7957	0.7957	0.7957	0.7957
0.2	0.6467	0.6467	0.6467	0.6467
0.3	0.5339	0.5339	0.5339	0.5339
0.4	0.4461	0.4461	0.4460	0.4461
0.5	0.3762	0.3762	0.3762	0.3762
0.6	0.3198	0.3198	0.3197	0.3197
0.7	0.2734	0.2734	0.2734	0.2734
0.8	0.2350	0.2350	0.2350	0.2350
0.9	0.2029	0.2029	0.2029	0.2029
1.0	0.1757	0.1757	0.1757	0.1757
P_{ez}	1000	1000	NA	1000
P_{er}	1000	NA	1000	1000

Model equations show marked deviation from PFR behavior as Peclet numbers approach zero. Table 5-6 and Figure 5-4 show mixing cup results for all model equations at Peclet numbers of 1.0 in the axial and radial directions.

Table 5-6 Asymptotic Comparison of Mixing Cup Concentration Profiles for the Ethane Cracking Reaction at Low Peclet Numbers

Z length	Isothermal PFR	Isothermal axial	Isothermal radial	Isothermal axial & radial
0.0	1.0000	1.0000	1.0000	1.0000
0.1	0.7957	0.7961	0.7328	0.7333
0.2	0.6467	0.6480	0.5519	0.5534
0.3	0.5339	0.5363	0.4239	0.4269
0.4	0.4461	0.4498	0.3312	0.3356
0.5	0.3762	0.3813	0.2623	0.2681
0.6	0.3198	0.3263	0.2098	0.2170
0.7	0.2734	0.2813	0.1691	0.1777
0.8	0.2350	0.2443	0.1372	0.1471
0.9	0.2029	0.2135	0.1119	0.1230
1.0	0.1757	0.1877	0.0916	0.1038
P_{ez}	1000	1	1000	1
P_{er}	1000	1000	1	1

The axial diffusion model clearly indicates a decrease in overall conversion when compared to the ideal PFR reactor. This effect is expected as axial diffusion works in a given reactor element to decrease high concentration components and increase low concentration components. This reduces the reaction driving force, decreasing cell conversions.

Equations with radial diffusion terms show a strong increase in conversion over the PFR reactor (Figure 5-4). Increased conversions were attributed to problems with the portions on the computer code that calculate the radial derivatives. Detailed review of the computer suggests that radial solution problems could be attributed to how PDETWO is handling boundary conditions between the velocity calculations and the assumption that the used radial profiles are applicable at the flow rates tested.

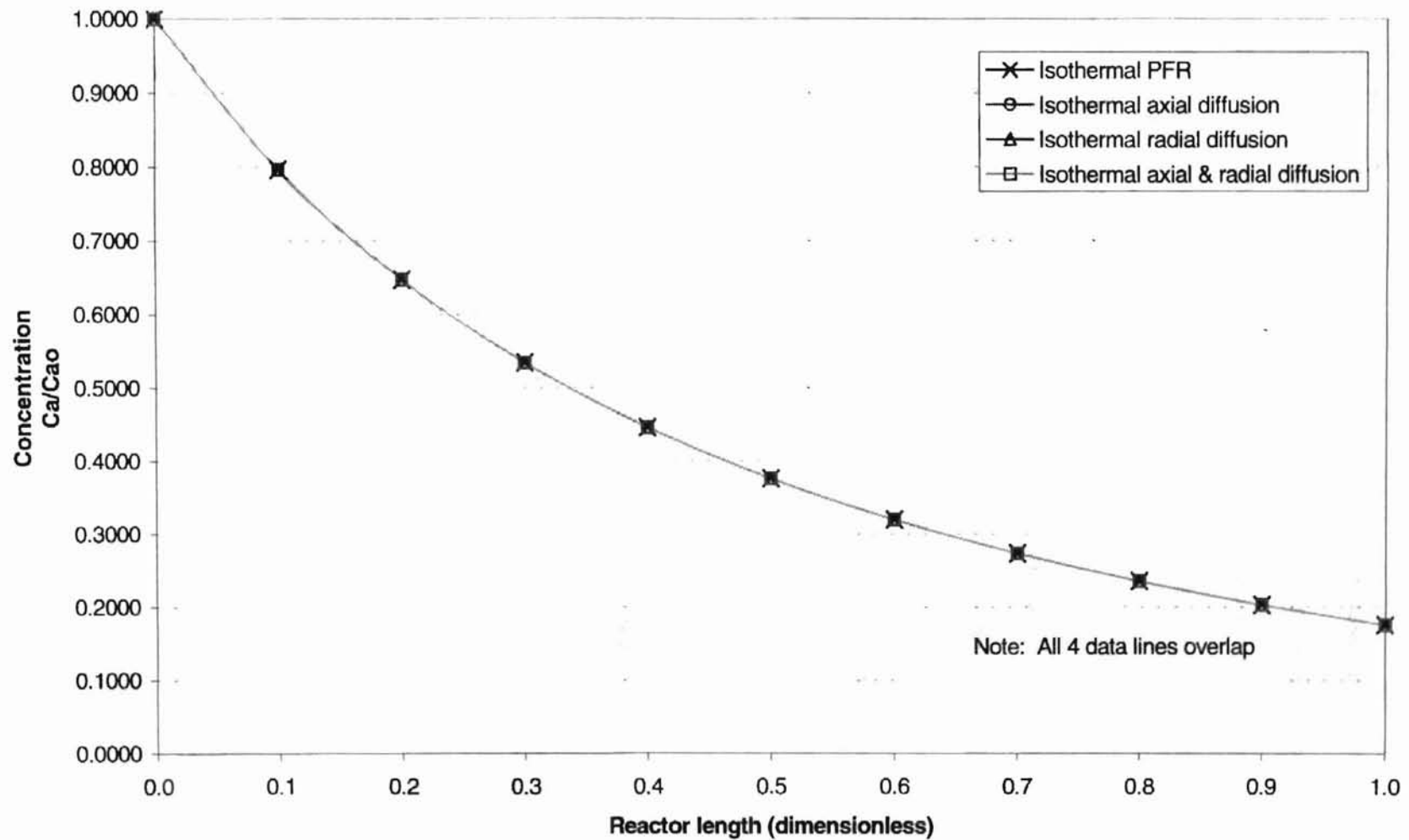


Figure 5-3 Asymptotic comparison of reactor types at high Peclet numbers for the ethane cracking reaction.

Velocity profiles (1/7 power law) are reasonable assumptions for the reactions and conditions tested. In both reaction flow rates and Reynolds numbers are well into the turbulent region. It is recommended that future simulations look very closely at how PDETWO handles radial boundary conditions, and determine where the problem is.

5.2.3 Effect of domain step size

Initial integration step size was varied to insure that solutions were independent of the initial step size. Initial integration step sizes (z direction) of 1e-10 and 1e-8 were used for the ethane cracking and MTBE synthesis reactions, respectively. Both test cases were analyzed using adiabatic and low Peclet numbers with the axial and radial diffusion model. Table 5-7 and Table 5-8 show results based on initial step size for each reaction.

Table 5-7 Effect of Step Size on the Numerical Results in the Ethane Cracking Reaction

Length	Step size				
	0.1	0.001	0.00001	0.000001	0.0000001
0	ERROR	ERROR	1.00000	1.00000	1.00000
0.1	ERROR	ERROR	0.87116	0.87091	0.87091
0.2	ERROR	ERROR	0.77857	0.77849	0.77849
0.3	ERROR	ERROR	0.69864	0.69836	0.69836
0.4	ERROR	ERROR	0.62914	0.62929	0.62929
0.5	ERROR	ERROR	0.57066	0.56984	0.56984
0.6	ERROR	ERROR	0.52022	0.51856	0.51856
0.7	ERROR	ERROR	0.47555	0.47390	0.47390
0.8	ERROR	ERROR	0.43592	0.43448	0.43448
0.9	ERROR	ERROR	0.40050	0.39990	0.39990
1	ERROR	ERROR	0.36926	0.36913	0.36913

Note: Errors shown were convergence errors reported by PDETWO

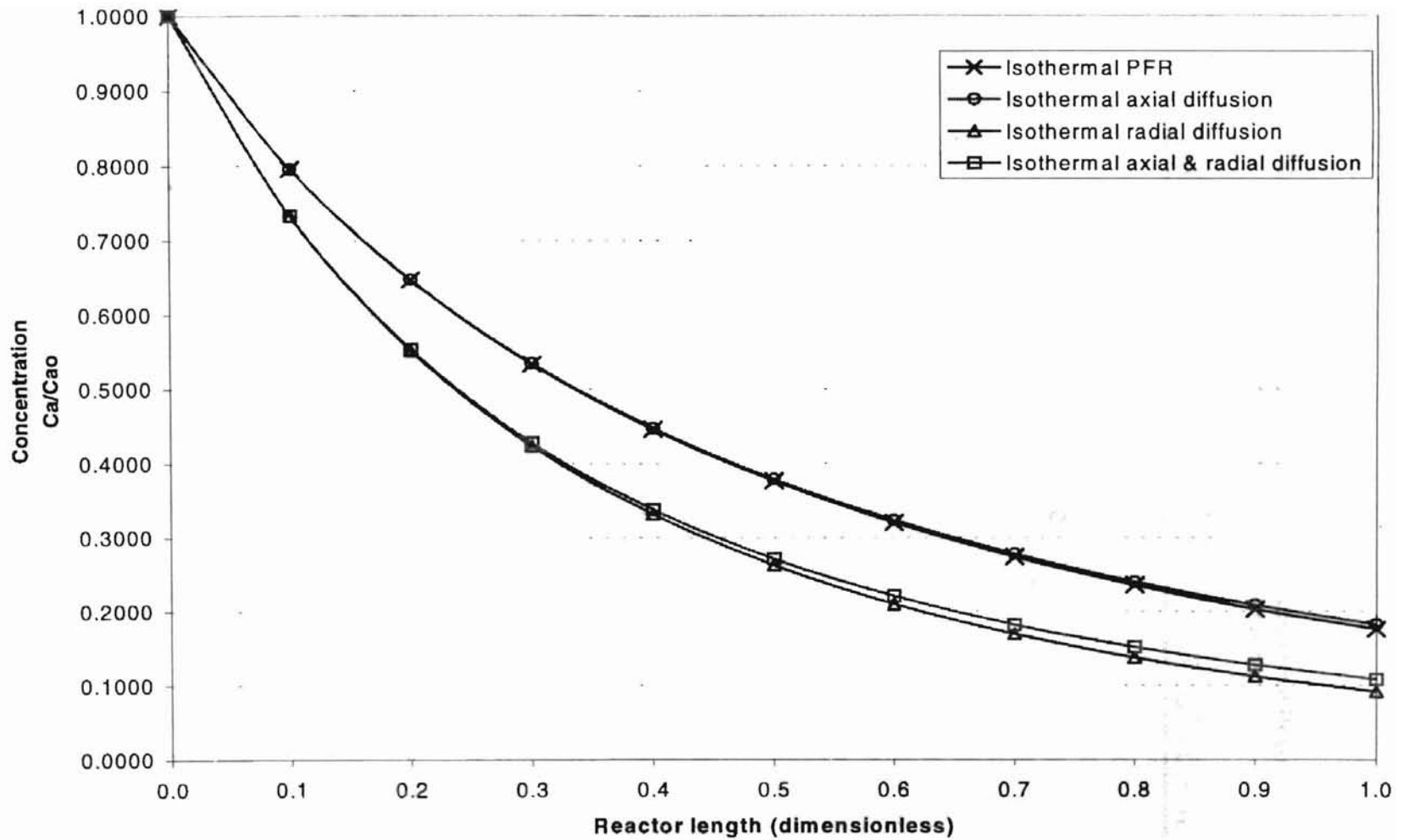


Figure 5-4 Asymptotic comparison of reactor types at low Peclet numbers for the ethane cracking reaction.

Table 5-8 Effect of Step Size on the Numerical Results in the MTBE Synthesis Reaction

Length	Step size				
	0.1	0.001	1.00E-06	1.00E-08	1.00E-10
0	ERROR	ERROR	1.0000	1.0000	1.0000
0.1	ERROR	ERROR	0.9295	0.9295	0.9295
0.2	ERROR	ERROR	0.8554	0.8554	0.8554
0.3	ERROR	ERROR	0.7774	0.7774	0.7774
0.4	ERROR	ERROR	0.6986	0.6986	0.6986
0.5	ERROR	ERROR	0.6211	0.6211	0.6211
0.6	ERROR	ERROR	0.5468	0.5468	0.5468
0.7	ERROR	ERROR	0.4771	0.4771	0.4771
0.8	ERROR	ERROR	0.4132	0.4132	0.4132
0.9	ERROR	ERROR	0.3562	0.3562	0.3562
1	ERROR	ERROR	0.3068	0.3068	0.3068

Note: Errors shown were convergence errors reported by PDETWO

It was found that the ethane cracking reaction solutions were not a function of initial step size (to 4 decimal places) provided that the step was less than 10^{-10} for temperatures lower than 1200 K. Above 1200 K severe conversion problems were observed when Peclet numbers were lower than 10.

Convergence problems in the ethane cracking reaction above 1200 K were attributed to heat capacity estimations. Heat capacity coefficients from HYSYS™ or Yaw et al. (1992) were used in all non-isothermal simulations. Values for the components in the ethane cracking reaction were experimentally curve fit, and are only valid up to 1100 K. Figure 5-5 shows correlated ideal gas component heat capacities for the ethane cracking reaction.

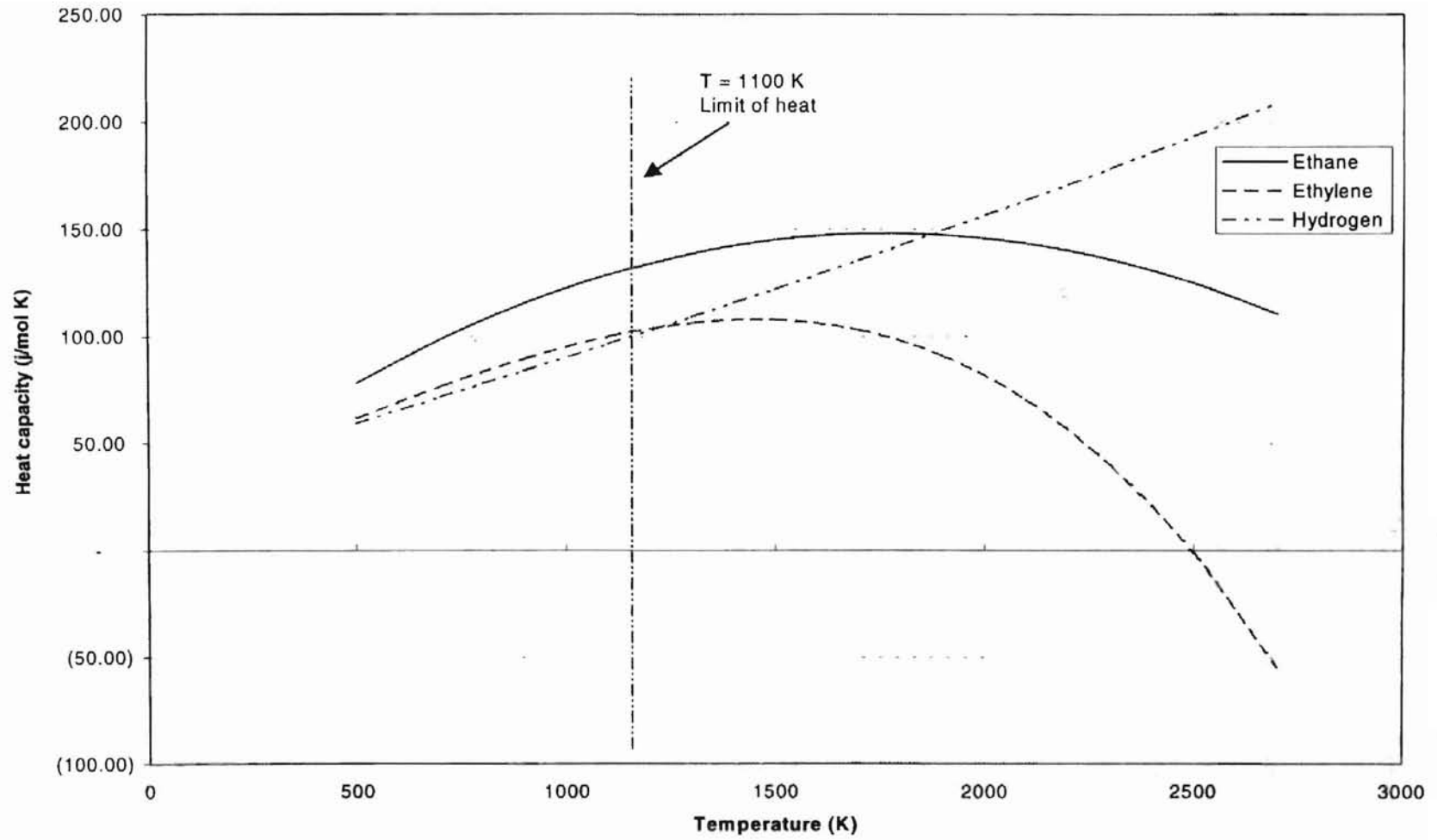


Figure 5-5 Gas phase heat capacity versus temperature profiles for the ethane cracking reaction.

Hydrogen and ethane component heat capacities begin to show marked deviations at temperatures greater than 1150 K. Above 1150 K, small changes in temperature result in large heat capacity variation.

Model solutions for the MTBE synthesis reaction were independent of integration step size provided the step size was less than 10^{-6} (Table 5-8). No convergence errors were observed on the MTBE synthesis reaction over the range of conditions studied.

5.2.4 Mass conservation test

The developed model obeyed mass conservation for the tested reactions. In all cases, the average mass was maintained at the integration points. Table 5-10 and Table 5-11 show cell total mass for each test case. Since system mass is conserved throughout the reactor, this further validated simulation results. The ethane cracking and MTBE synthesis reactions maintained an average mass of 30.7 kg and 88.15 throughout the reactor. Mass conservation is preserved.

5.3 Parametric analysis

The effect of critical feed, flow and reactor variables were analyzed to determine the flexibility and model limitations. Parameters analyzed were inlet temperature, reactor length Peclet number and feed flow rates. In order to allow for clear analysis only one parameter was analyzed at a time. Table 5-9 shows the range of analysis done for the ethane cracking reaction.

Table 5-9 Parametric Analyses - Variable Range

	Temperature (Kelvin)	Flow rate (kmol/s)	Peclet Number	Reactor Length (Meters)
High value	1600	0.1335	1000	50
Low value	880	0.01	1.	5.0

Table 5-10 Mass Balance Data for the Ethane Cracking Reaction

EthRxn	r=0.0	r=0.1	r=0.2	r=0.3	r=0.4	r=0.5	r=0.6	r=0.7	r=0.8	r=0.9	r=1.0	MixCup
0	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07
0.1	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07
0.2	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07
0.3	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07
0.4	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07
0.5	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07
0.6	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07
0.7	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07
0.8	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07
0.9	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07
1	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07	30.07

Table 5-11 Mass Balance Data for the MTBE Synthesis Reaction

MTBE	r=0.0	r=0.1	r=0.2	r=0.3	r=0.4	r=0.5	r=0.6	r=0.7	r=0.8	r=0.9	r=1.0	MixCup
0	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15
0.1	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15
0.2	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15
0.3	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15
0.4	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15
0.5	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15
0.6	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15
0.7	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15
0.8	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15
0.9	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15
1	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15	88.15

5.3.1 Effect of inlet temperature

Parametric analysis on inlet feed temperature demonstrated typical conversion verses temperature profiles for the ethane cracking reaction. Figure 5-6 shows model outputs for both the adiabatic and isothermal ethane cracking reaction. The model was simulated at temperatures from 880 K to 1200 K. Note that the adiabatic reaction results projected out to higher temperatures may be in error due to limitations in the heat capacity data.

The irreversible ethane cracking reaction was very endothermic ($\Delta H_{rxn}=82,000$ cal/mol at 298 K). Figure 5-6 clearly showed that in all high Peclet simulations the adiabatic cases have a lower conversion at any given temperature. The endothermic effect can also be seen in the temperature required to reach 100% conversion. Complete conversion occurred at 1150 K (isothermal) and was estimated at temperatures above 1300 K for the adiabatic simulation.

Simulations with low radial Peclet numbers resulted in unrealistic conversion values for the conditions given in

Table 5-2. HYSYS™ and ASPEN PLUS™ simulations indicate that the ethane cracking reaction was essentially quenched at temperatures lower than 900 K. PFR simulations were in agreement with these results.

Non-ideal simulations with radial dispersion (low Peclet values) predict a lowest possible conversion of 40%. These appear to be unrealistic results. The radial equations assume that there was no slip of material at the reactor wall. This assumption (as coded) allowed for almost infinite residence time and could explain why conversion was obtained at low temperatures. In addition, the very high flow rates involved with this reaction suggest almost plug flow condition. Strong variations from plug flow result in unrealistic solutions.

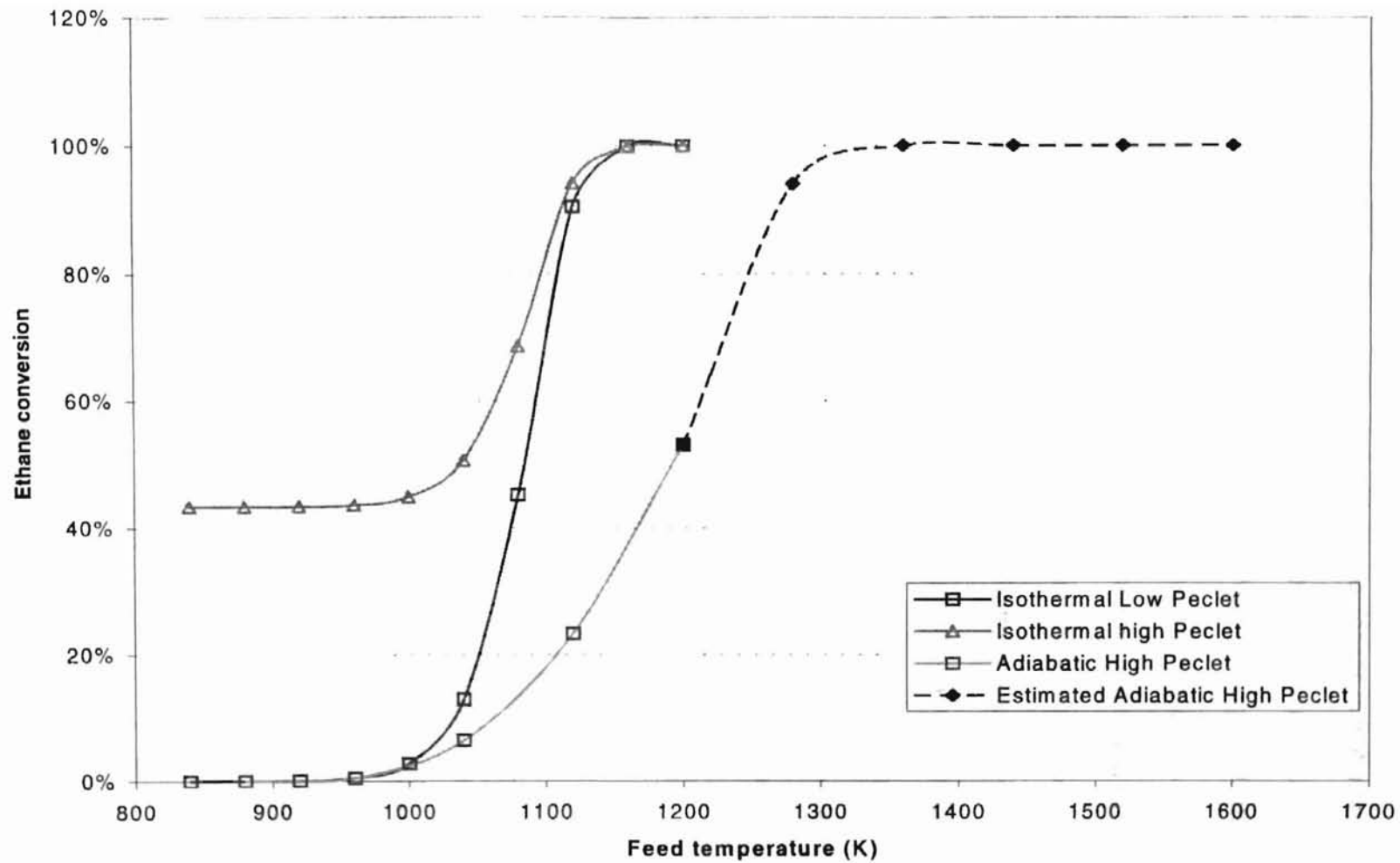


Figure 5-6 Effect of inlet temperature on conversion in the ethane cracking reaction under isothermal and adiabatic conditions.

The reactor model predicts that the MTBE synthesis reaction system was equilibrium limited. Figure 5-7 showed significant decreases in isobutene conversion for inlet feed temperatures greater than ~400 K. This phenomenon occurred in all 4 variations of the simulations. Experimental results from (Zhang and Datta 1995) and others indicate that equilibrium favors reactants at high temperatures. This is clearly predicted by the model. In addition, comparison of the adiabatic and isothermal curves at 350K indicated where the equilibrium transition from products to reactants occurred.

Reactor conditions as given by (Zhang and Datta 1995) state that the reactor used was essentially isothermal with only slight temperature variations down the reactor. The close proximity of the adiabatic and isothermal curves indicates almost isothermal conditions. The radial diffusion equations failed to predict realistic conversion at temperatures lower than 375 K. As was discovered in the ethane cracking analysis, radial variations imposed higher conversions at lower temperatures. This can again be attributed to the assumed radial velocity profile.

5.3.2 Reactor length analysis

Model predictions approach equilibrium conversion as the reactor length increases. Figure 5-8 shows that ethane conversion approach 100% (as is expected in irreversible reactions) at reactor lengths greater than 30 meters. ASPEN™ simulations also predict 100% conversion for reactors larger than 30 meters. The adiabatic reaction was unable to achieve conversion greater than 19%. This was due to reaction quenching as temperatures dropped below 1000 K.

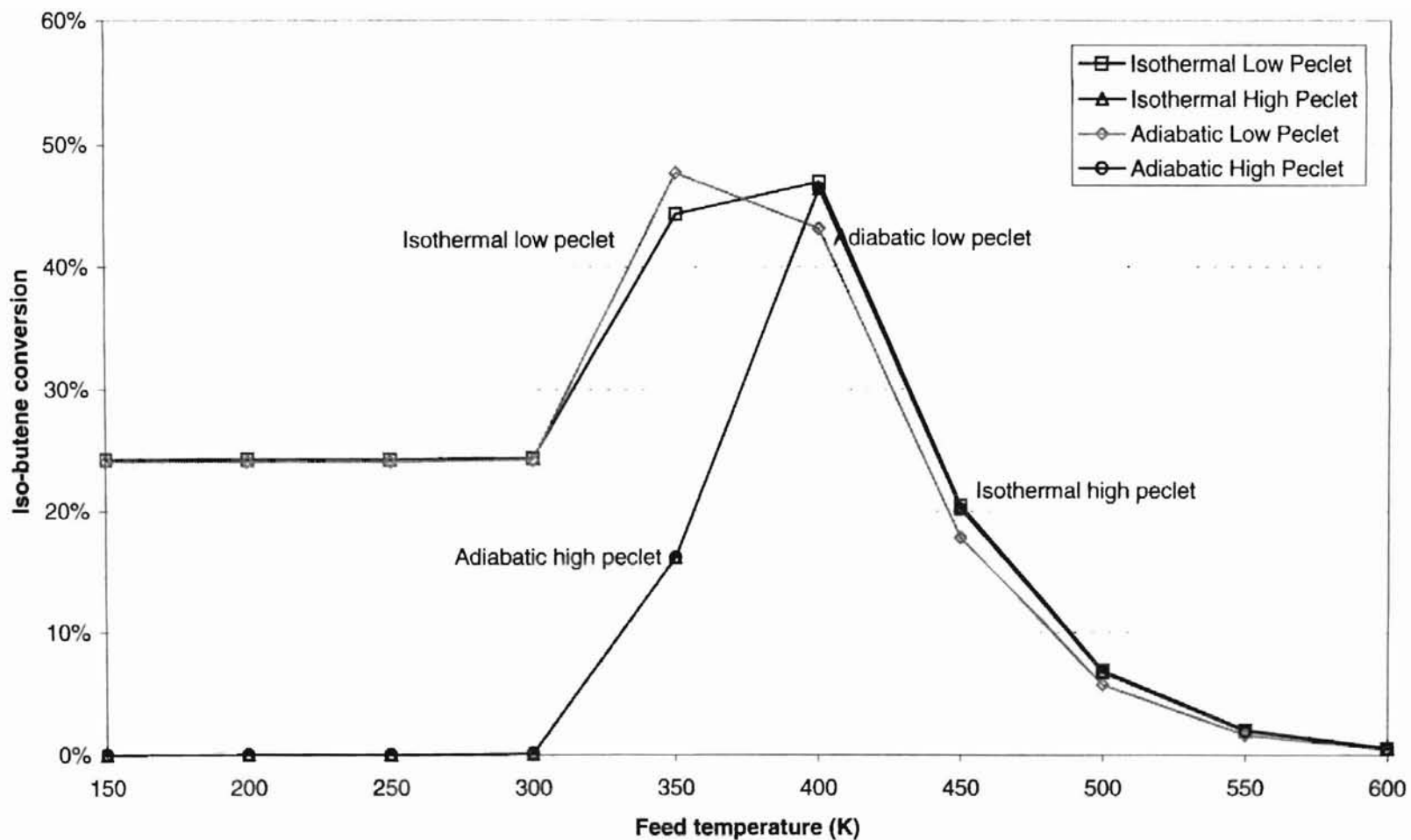


Figure 5-7 Effect of inlet temperature on conversion in the MTBE synthesis reaction under isothermal and adiabatic conditions.

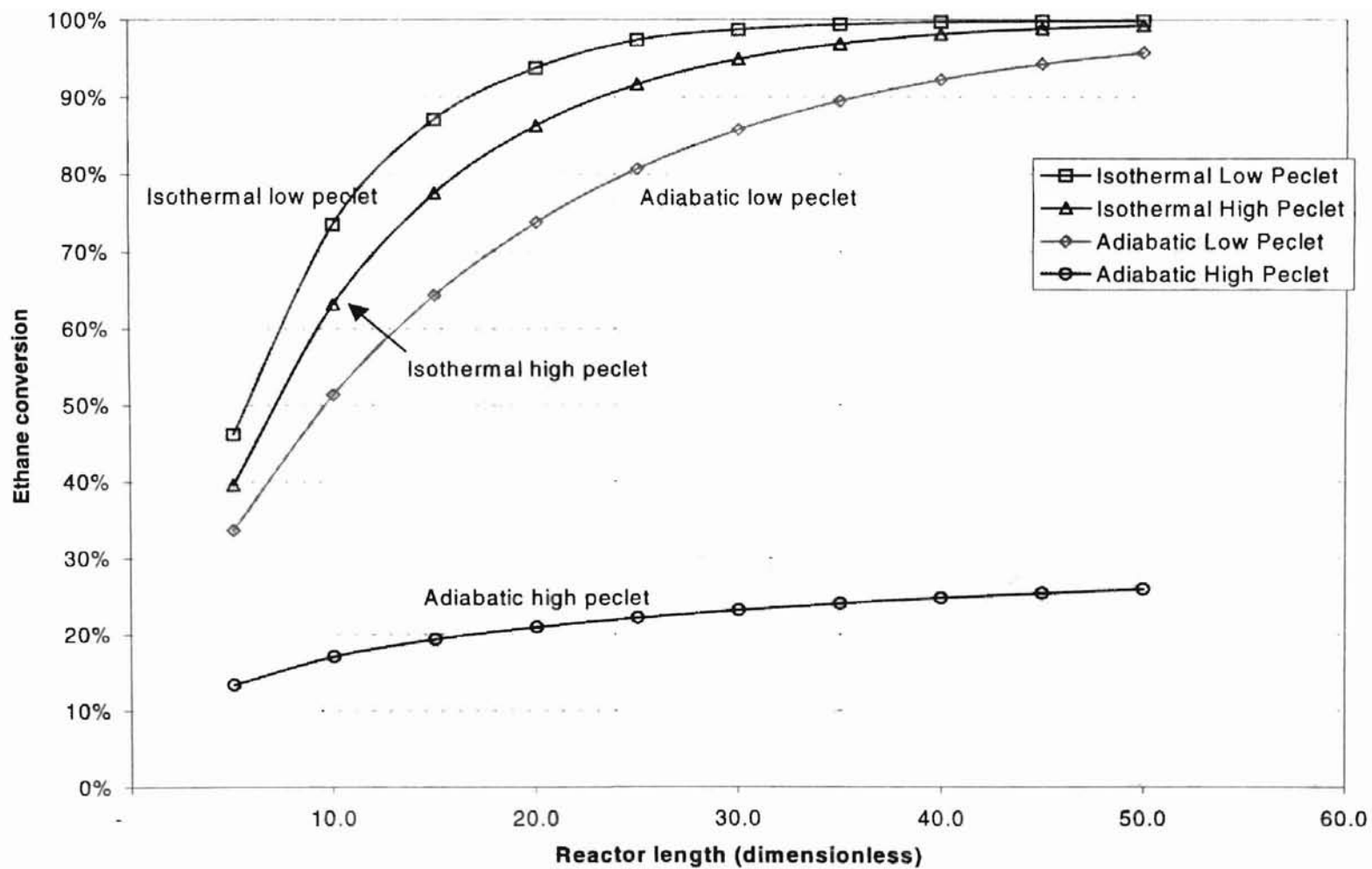


Figure 5-8 Effect of reactor length on conversion in the ethane cracking reaction under isothermal and adiabatic conditions.

Low Peclet simulations again failed to accurately predict reactor conditions and conversion. The adiabatic low Peclet simulation gave conversions almost as good as those obtained in the isothermal case. Again, this could in part be attributed to the radial velocity profiles assumed, as flow rates are in the turbulent regime.

High Peclet MTBE synthesis simulation results accurately predicts equilibrium conversions. Figure 5-9 shows equilibrium conversion of 90% for both the adiabatic and isothermal runs. This was expected as the reaction conditions are nearly isothermal (Zhang and Datta 1995). Zhang and Datta indicate that at 343K equilibrium conversion isobutene was ~90%. This simulation predicts 91%.

Low Peclet simulations are inappropriate for this reactor. Low Peclet simulations were unable to accurately predict equilibrium conversion for reactor conditions. In addition, previous simulations indicate poor predictions using the radial assumptions.

5.3.3 Flow rate analysis

Flow rate analysis showed that the model accurately predicted experimental MTBE synthesis data. Data from (Zhang and Datta 1995) was used to test the flow sensitivity of the model. (Zhang and Datta 1995) extensively tested the MTBE synthesis reaction over a wide range of space velocities and temperatures (Zhang and Datta 1995). Our model was used to simulate experimental runs at 333 K, 343 K and 353 K. Figure 5-10 plotted model results and experimental values of Zhang and Datta. In all three test cases, a high degree of accuracy was achieved.

Reactor conversion was consistently lower than experimental for the low temperature runs (343 K). This could be attributed to the kinetics used in the model. Analysis of the temperature dependent terms of the MTBE synthesis kinetics showed greatest deviation from experimental at low temperatures (Al-Jarallah et al. 1988) (see Table 5-12).

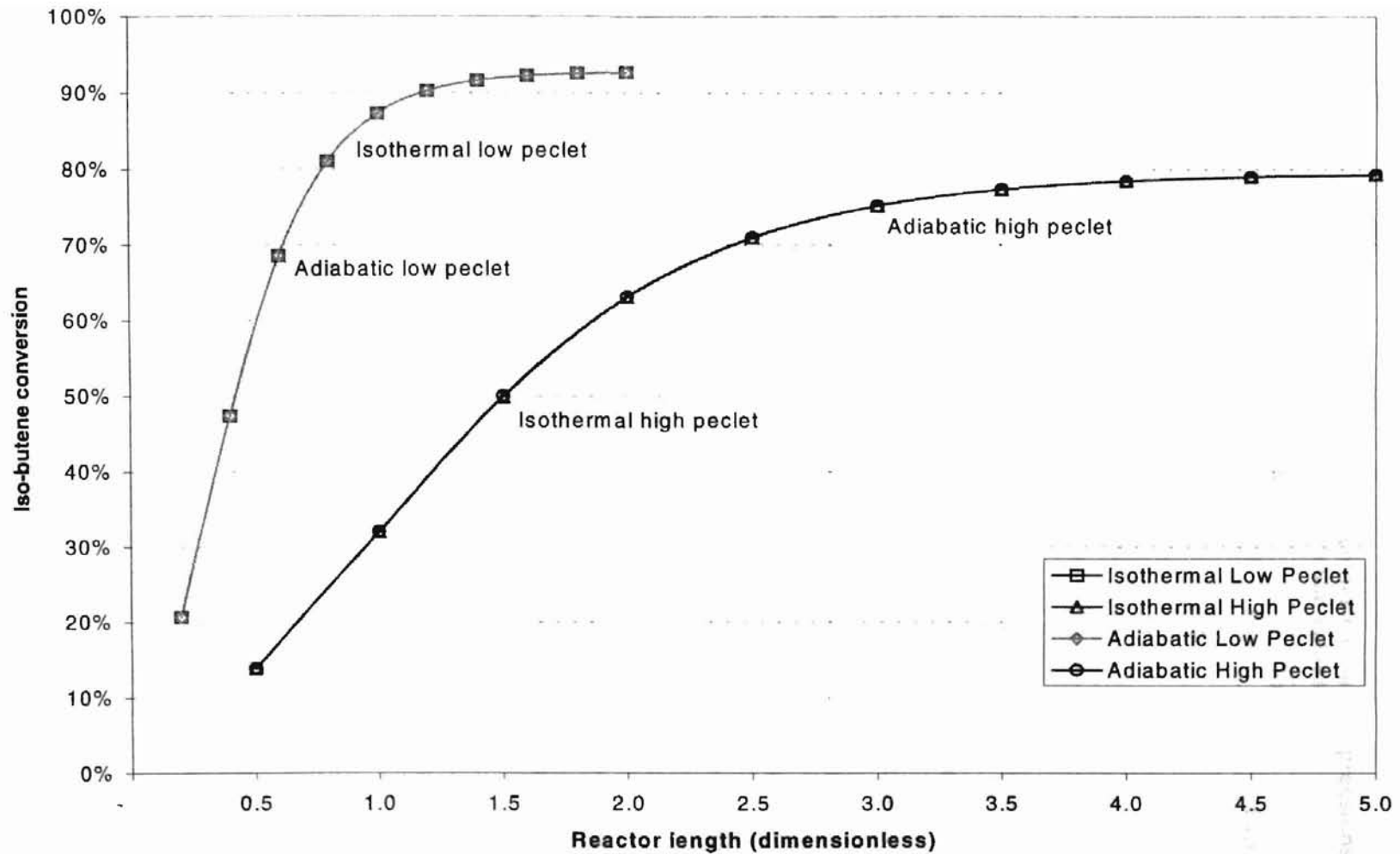


Figure 5-9 Effect of reactor length on conversion in the MTBE synthesis reaction under isothermal and adiabatic conditions.

Table 5-12 Comparisons of MTBE Synthesis Kinetic Expressions to Experimental Values

Temp	K _{eq}		K _s		K _a		K _c	
	Exp.	Model	Exp.	Model	Exp.	Model	Exp.	Model
343 (K)	38.0	34.17	0.512	0.499	359.8	365.53	202.1	208.49
353 (K)	15.8	22.89	1.065	1.195	159.8	134.70	73.3	63.991
363 (K)	13.0	20.96	2.537	2.725	47.6	53.982	18.5	20.961
373 (K)	6.9	10.80	6.080	5.946	25.5	22.719	7.64	7.289

The model failed to predict incursions in the higher temperature profiles at space velocities between 50 and 100 (Note the experimental bumps at this range in Figure 5-10). Explanation of the observed incursions is most likely not caused by flow issues, given the fact that the incursion corrects itself and does not continue to effect the initial curve. If the bump were caused by a shift from laminar to turbulent flow, the conversion curves would not recover and would instead continue at the lower slope. Zhang and Datta attributed the variations in conversion to the formation of di-isobutene. Other researchers also indicate that this was a relevant side reaction. Our model does not account for this reaction, and therefore does not predict its effects. However, the effect of this reaction was minimal and was not accounted for.

5.4 Summary of Results

Extensive testing on the MTBE synthesis and ethane cracking reactions yielded results that were consistent with published literature and experiments over a rather wide range of conditions. The model predicted realistic equilibrium conversion and performance under plug flow and axial dispersion conditions. Unrealistic results were observed for all test reactions when radial dispersion was included. Analysis of the computer code suggests a problem with all models using the radial profiles, and must be investigated further.

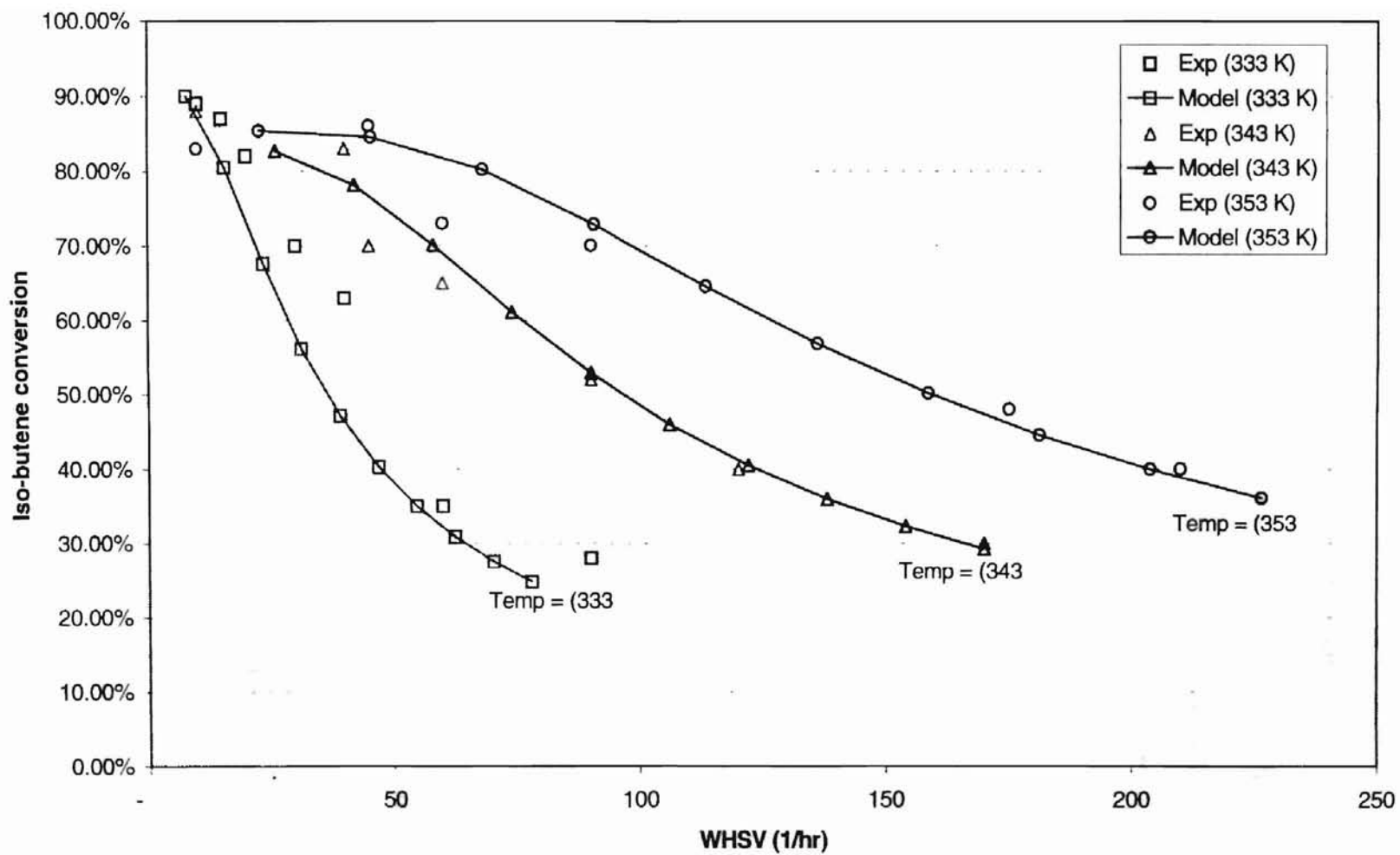


Figure 5-10 Comparison of model performance to Zhang and Datta (1995) experimental data for the MTBE synthesis reaction.

Table 5-13 summarizes all results presented in this section.

Table 5-13 Summary of Results

Test	Reaction	Results
PDETWO accuracy	Hornbeck sample problem	Accurate to within 0.005%.
HYSYS™ comparison	ethane cracking reaction	Only slight difference in profiles. Attributable to model assumptions.
Asymptotic limits:		
High Peclet numbers	ethane cracking reaction	Model solutions approach PFR results as Peclet numbers increase.
Low Peclet numbers	ethane cracking reaction	Model solutions show an increase in conversion for decreasing radial Peclet numbers, while decreasing axial Peclet numbers decrease conversion and is in error.
Effect of domain size		
	ethane cracking reaction	Solution is independent of step size if steps are sufficiently small.
	MTBE synthesis reaction	Solution is independent of step sized if steps are sufficiently small.
Conservation Check:		
Mass conservation	Both reactions	Model obeyed mass conservation laws at all considered integration points within the reactor.
Model stability	Ethane cracking reaction	Stable for isothermal and adiabatic conditions within the limits of the provided physical data.

Parametric Analysis:

NOV 11 1998 10:58:10 AM

Inlet temperature	ethane cracking reaction	Model provides realistic results and profiles for an endothermic isothermal or adiabatic reaction under PFR and axial diffusion simulations. Radial diffusion simulations provide unrealistic results. Adiabatic solutions over 1200 K were unstable due to limitations in the heat capacity data, and are in error.
	MTBE synthesis reaction	Model and kinetics provide reasonable results for PFR and axial dispersed reactors. Radial dispersion was unrealistic. Error is again attributed to problems within the radial portion of the code. Equilibrium characteristics were comparable to analytical results.
Reactor length	Both reactions	Model predicted increasing conversions as reactor length was increased. Conversions obtained were consistent with ideal and experimental results. Both reactions showed unrealistic results under radial dispersion conditions.
Flow analysis	MTBE synthesis reaction	Comparison of model results to experimental data show excellent agreement over a wide range of flow rates.

**MTBE Synthesis
Kinetics Test**

Provided results consistent with the experimental data of Zhang and Datta at a high level of accuracy. Greatest error occurred at low temperatures, well outside of the range under which they were derived.

The generic model was effective in predicting performance and conversion for the reactions tested. Simulation results were consistent with both experimental and proven HYSYS™ models. The code also provided the first documented evidence that the concentration based MTBE synthesis kinetics of Al-Jarallah et al. can be used to accurately predict MTBE synthesis fixed bed reactor performance.

Chapter 6

Conclusions and Recommendations

6.1 *Conclusions*

In this study, a computer model was developed to describe and predict the performance of a generic fixed bed reactor system. The model was subjected to parametric testing using the gas phase ethane cracking reaction and liquid phase MTBE synthesis reaction. The developed model is the first to be used for the evaluation of MTBE synthesis kinetics (using the concentration kinetics of Al-Jarallah et al.) under fixed bed industrial conditions. The following conclusions were drawn from simulation results and model evaluations.

Model simulations using the ethane cracking reaction resulted in solutions that closely matched experimental results and the isothermal PFR model of the HYSYS™ simulation package. Slight profile differences at the front of the reactor are thought to be a difference on inlet boundary conditions. Model output conversions were within 5% of those predicted by HYSYS™.

The model showed convergence problems at high adiabatic temperatures for the ethane cracking reaction. Convergence problems were attributed to limitations in the heat capacity equations used. The experimentally determined heat capacity equations had an upper limit of 1100 K and problems occurred above this temperature with the adiabatic simulations.

Asymptotic analyses provided results are consistent compared to theoretical for all reactions considered except those containing radial diffusion components. Model results approached plug flow reactor solutions as axial and radial Peclet numbers approached infinity, consistent with established reactor theory. Evaluation of low axial

Peclet numbers (axial transport) showed a decrease in conversion as is predicted in reactor theory. Evaluation of the model with low radial Peclet numbers (radial transport) resulted in unrealistic solutions for the conditions and reactions tested as reactor conversions increased. Analysis of the computer models and code indicates that the problems are in how the computer handles radial boundary conditions.

Simulations for the MTBE synthesis reaction demonstrated that the Rideal-Eley kinetic expression proposed by Al-Jarallah accurately predicted conversions in fixed bed reactor conditions. Simulations correctly predicted equilibrium conversions and closely matched experimental fixed bed data from Zhang and Datta.

With exception of radial velocity profile, model assumptions appear to be valid for the conditions tested. Radial unfeasibility was attributed to the assumed velocity profile; this should be updated in future work.

6.2 Recommendations

Overall, the model proved to be an effective tool in predicting reactor performance over a wide range of operating conditions for the reactions tested. Test results provided strong agreement with simulations engines and published data. However, further testing should be done to determine its usability as a generic fixed bed modeling routine and correct all problems associated with the radial profile problem.

Initial focus should work towards correcting the problems associated with the unrealistic radial conversion profiles. This work was able to correct problems associated with correctly calculating the power law radial velocity profiles in an attempt to correct the observed problem. Corrections implemented served to increase the reliability of the code, but were unable to effect the erroneous conversion profiles being generated. Future work should look at how past researchers have overcome these types of

problems and how PDE TWO implements radial boundary conditions over a given radius. It is thought current radial solution problems are associated with these boundary conditions.

It is recommended that future tests look at reaction currently thought to be most suitable for Nafion™ catalytic studies. Reactions include those found in alkylation (Albright and Kranz 1992; Kranz 1992; Albright et al. 1993; Kranz 1993), catalytic polymerization (Ipatieff and Pines 1936; Ipatieff and Schaad 1936; Deeter 1950; Haag 1983) and butene isomerization (Sun et al. 1995; Kagawa 1974; Sun 1995). Conoco has indicated that these reactions are ones they feel will most benefit from the Nafion™ catalyst. Additionally, the wide spread publication of these reactions makes them ideal candidates for testing of the Nafion™ reactor.

In order for Conoco to confidently use the model most effectively, it should be tested under actual plant conditions. Testing and comparison of the model under these conditions will determine the model accuracy and limitations in predicting pilot and plant scale reactors. Conoco has offered an extensive amount of adiabatic MTBE synthesis data from their Ponca City plant. The Ponca City data would provide a strong base with which the model can be tested. Conoco has indicated that this data includes deactivation and significant history, providing an excellent benchmark for current and future work.

The current work has focussed on steady state conditions; future work should expand this to non-steady state conditions. Extension to non-steady state would allow analysis of reactor performance over time, and could account for catalyst aging. A transient model of this type would prove a valuable tool in pilot and plant scale reactor analysis. Provided the model is accurate, there is a strong potential for Conoco to save time and money at all levels of reactor operation.

In this work, a reactor model was developed in an attempt to create a generalized code capable of accurately predicting fixed bed reactor performance. The developed model was able to simulate two very different reactions under a wide range of conditions. Additionally, the Rideal-Eley kinetic model (Al-Jarallah et al. 1988) for the MTBE synthesis reaction was shown to be effective in predicting fixed bed reactor performance. It is proposed that the developed model will be feasible for the accurate prediction of reaction similar to MTBE such as the above mentioned Nafion™ reactions.

In summary, this work was provided the following conclusions:

- Model solutions closely matched commercially proven simulators.
- The model showed difficulties solving the gas phase ethane cracking reaction at temperatures outside the range of the physical data.
- Asymptotic analysis proved the models behave as expected under extremes of operation for:
 - Inlet temperature variations,
 - reactor length and
 - flow rate analyses (MTBE synthesis reaction).
- MTBE synthesis simulations showed that the Rideal-Eley kinetics of Al-Jarallah et al. (1988) are indeed applicable to the realistic fixed bed experiments of Zhang and Datta (1995).

BIBLIOGRAPHY

- Albright, L. F. and K. E. Kranz (1992). "Alkylation of Isobutane with Pentenes Using Sulfuric Acid as a Catalyst: Chemistry and Reaction Mechanisms." Industrial and Engineering Chemical Research **31**: 475-481.
- Albright, L. F. et al. (1993). "Alkylation of Isobutane with Light Olefins: Yields of Alkylates for Different Olefins." Industrial and Engineering Chemistry Research **32**: 2991-2996.
- Al-Jarallah, A. et al. (1988). "Kinetics of Methyl Tert-Butyl Ether Synthesis Catalyzed by Ion Exchange resin." The Canadian Journal of Chemical Engineering **66**: 802-806.
- Bird et al. (1960). Transport Phenomena. New York, John Wiley & Sons.
- Carnahan, B., H. A. Luther, et al. Applied Numerical Methods. New York, John Wiley & Sons.
- Cussler, E. L. (1997). Diffusion Mass Transfer in Fluid Systems, Cambridge University Press.
- Deeter, W. F. (1950). "Propylene Polymerization for Motor-Gasoline Production." The Oil and Gas Journal **March 23, 1950**: 252-258.
- Ergun, S. and A. A. Orning (1949). "Fluid Flow through Randomly Packed Columns and Fluidized Beds." Industrial and Engineering Chemistry **41(6)**: 1179-1184.
- Fogler, H. S. (1986). Elements of Chemical Reaction Engineering, Prentice-Hall.
- Froment, G. F. (1967). "Fixed Bed Catalytic Reactors." Industrial and Engineering Chemistry **59(2)**: 18-27.
- Froment, G. F. and K. B. Bischoff (1979). Chemical Reactor Analysis and Design, John Wiley & Sons.
- Gomez, C. et al. (1997). "Experimental Study of the Simultaneous Synthesis of Methyl Tert-Butyl Ether and Ethyl Tert-Butyl Ether in Liquid Phase." Industrial and Engineering Chemical Research **36**: 4756.
- Gunn, D. J. (1987). "Axial and Radial Dispersion in Fixed Beds." Chemical Engineering Science **42(2)**: 363-373.
- Gunn, D. J., M. M. Ahmad, et al. (1987). "Radial Heat Transfer to Fixed Beds of Particles." Chemical Engineering Science **42(9)**: 2163-2171.
- Haag, W. O. (1983). "Oligomerization of Isobutylene on Cation Exchange Resins." Chemical Engineering Progress Symposium Series **73(63)**: 140-147.
- Hicks, R. E. (1970). "Pressure Drop in Packed Beds of Spheres." Industrial and Engineering Chemical Fundamentals **9(3)**: 500 - 502.
- Hicks, R. E. and W. G. B. Mandersloot (1968). "The Effect of Viscous Forces on Heat and Mass Transfer in Systems with Turbulence Promoters and in Packed Beds." Chemical Engineering Science **23**: 1201-1210.

- Himmelblau, D. M. (1996). Basic Principles and Calculations in Chemical Engineering. Upper Saddle River, NJ, Prentice Hall International Series.
- Himmelblau, D. M. and K. B. Bischoff (1968). Process Analysis and Simulation: Deterministic Systems, John Wiley & Sons.
- Hornbeck, R. W. (1975). Numerical Methods. Englewood Cliffs, New Jersey, Prentice-Hall Inc.
- Ipatieff, V. N. and H. Pines (1936). "Propylene Polymerization Under High Pressure and Temperature with and without Phosphoric Acid." Industrial and Engineering Chemistry **28**(6).
- Ipatieff, V. N. and R. E. Schaad (1936). "Mixed Polymerization of Butenes by Solid Phosphoric Acid Catalyst." Industrial and Engineering Chemistry **30**(5): 596-600.
- Kagawa, S. (1974). "Analysis of Reaction Path in Triangular Network Consisting of Langmuir-Hinshelwood Type Reactions." Journal of Catalysis **37**: 182-185.
- Kleir, K., et al. (1982). "Catalytic Synthesis of Methanol from CO/H₂." Journal of Catalysis **74**: 343-360.
- Kranz, L. A. a. K. (1992). "Alkylation of Isobutane with Pentenes Using Sulfuric Acid as a Catalyst: Chemistry and Reaction Mechanisms." Industrial and Engineering Chemical Research **31**: 475-481.
- Kranz, L. A. a. K. (1993). "Alkylation of Isobutane with Light Olefins: Yields of Alkylates for Different Olefins." Industrial and Engineering Chemical Research **32**: 2991-2996.
- Levenspiel, O. (1972). Chemical Reaction Engineering, John Wiley & Sons.
- Levenspiel, O. (1972). "Experimental Search for a Simple Rate Equation to Describe Deactivating Porous Catalyst Particles." Journal of Catalysis **25**: 265-272.
- Levenspiel, O. (1989). The Chemical Reactor Omnibook. Corvallis, Oregon, OSU Book Stores.
- Melgaard, D. and R. Sincovec (1981). "Algorithm 565 PDETWO/PSETM/GEARB: Solution of Systems of Two-Dimensional Nonlinear Partial Differential Equations." ACM Transactions on Mathematical Software **7**(1): 126-135.
- Melgaard, D. and R. Sincovec (1981). "General Software for Two-Dimensional Nonlinear Partial Differential Equations." ACM Transactions on Mathematical Software **7**(1): 106-125.
- Nauman, E. B. (1987). Chemical Reactor Design. Troy, New York, John Wiley & Sons.
- Perry, R. H. et al., Eds. (1984). Perry's Chemical Engineers' Handbook 6th Edition. New York, McGraw-Hill Book Company.
- Rehfinger, A. and U. Hoffman (1990). "Kinetics of Methyl Tert-Butyl Ether Liquid Phase Synthesis Catalyzed by Ion Exchange Resin. Macropore Diffusion of Methanol as Rate-Controlled Step." Chemical Engineering Science **45**(6): 1619-1626.
- Rehfinger, A. and U. Hoffmann (1990). "Formation of di-Isobutene, Main By-Product of Methyl Tert-Butyl Ether Synthesis Catalyzed by Ion Exchange Resin." Chemical Engineering Technology **13**: 150-161.

Rice, R. G. and D. D. Do (1995). Applied Mathematics and Modeling for Chemical Engineers, John Wiley & Sons, Inc.

Riggs, J. B. (1994). An Introduction to Numerical Methods for Chemical Engineers, Texas Tech University Press.

Sattersfield, C. N. (1980). Heterogeneous Catalysis in Practice. New York, NY, McGraw-Hill.

Shikata, S., S.-i. Natakata, et al. (1997). "Catalysis by Heteropoly Compounds. 32. Synthesis of Methyl Tert-Butyl Ether Catalyzed by Heteropolyacids Supported on Silica." Journal of Catalysis **166**: 263-271.

Sola, L. and M. Pericas (1997). "A Comparative Thermodynamic and Kinetic Study of the Reaction Between Olefins and Light Alcohols Leading to Branched Ethers. Reaction Calorimetry Study of the Formation of Tert-amyl-methyl-ether and Tert-butyl isopropyl Ether." Industrial and Engineering Chemical Research **36**: 2012-2018.

Sun, Q. (1995). Summary of Nov 20, 1995 Meeting at Conoco, Ponca City.

Sun, Q. et al. (1995). 1-Butene Isomerization over Nafion/Silica Composite Catalyst. Wilmington, DE 19880, The DuPont Company, Central Research and Development.

Szukiewicz, M., K. Kaczmarski, et al. (1998). "Modeling of Fixed-Bed Reactors: Two Models of Industrial Reactor for Selective Hydrogenation of Acetylene." Chemical Engineering Science **53**(1): 149-155.

Vortmeyer, D. and J. Schuster (1983). "Evaluation of Steady Flow Profiles in Rectangular and Circular Packed Beds by a Variational Method." Chemical Engineering Science **38**(10): 1691-1699.

Zhang, T. and R. Datta (1995). "Integral Analysis of Methyl Tert-Butyl Ether Synthesis Kinetics." Industrial and Engineering Chemical Research **34**: 730-740.

APPENDICIES

APPENDIX A

Determination of RTD and Effective Diffusion Parameters

The residence time distribution method (RTD) is one the easiest and most common method used to model non-ideal behavior in industrial reactors. The RTD method is also the basis for determination of the parameters utilized in the effective transport method. Initial determination of the effective transport properties requires initial RTD testing to determine effective transport properties. This appendix provides a detailed description of the RTD method and presents a method to determine effective transport properties using RTD data. We begin with the description of the RTD method.

The residence time distribution (RTD) uses empirical testing to determine reactor performance. The reactor to be modeled must undergo a series of concentration step or inert tracer test. An initial concentration or trace step is introduced at the inlet of the reactor and outlet concentration monitoring begins. Analysis of reactor tracer outflow allows the analysis of reactor performance. Reactor concentrations are recorded and plotted against residence time, resulting in a plot having the mathematical form:

$$\int_0^{\infty} E(\theta) d\theta = 1 \quad \text{A.1}$$

The function within the integral is defined as the fraction of fluid leaving the vessel that has residence time of $\theta+d\theta$. From Equation 1 mean residence times and reactor flow profiles can be determined. Figure 1 and Figure 2 show typical distribution curves for the two extremes of flow patterns (PFR & CSTR).

At $t=0$, a known concentration of tracer compound is introduced into a fixed bed reactor operating under steady state conditions. Reactor products are continuously sampled after introduction of the tracer compound. Concentration of the tracer in the outlet stream is measured and a concentration verses residence time curve is generated.

For plug flow reactors, concentration verses resident time distributions (Figure A-1) assume the shape of the general Dirac function (Froment and Bischoff 1979). To understand this, consider the flow assumptions used to model plug flow reactors. Plug flow reactors assume that there are no radial variations in temperature concentrations and flow rates. In other words, all components in a given cross section of the reactor have identical residence times for all points along the radius. Additionally it assumes no diffusion of chemical species in the axial direction. Compounds in any given cross section do not influence or communicate with any other cross section. Based on these assumptions, all tracer input to the reactor will exit the reactor at exactly the same moment. This results in the characteristic profile shown in Figure A-1.



Figure A-1 RTD plug flow exit age distribution.

Continuous Stirred Tank Reactors (CSTRs) provide a very different type of distribution curve (Figure A-2).

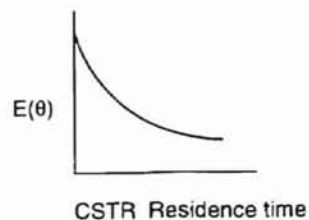


Figure A-2 RTD, CSTR exit age distribution.

Again, a trace pulse is introduced to the reactor instantaneously and it is detected immediately at the reactor outlet. Immediate detection of the tracer in the outlet is a direct result of the perfect mixing assumptions essential to the CSTR reactor models. Once tracer is detected in the outlet, concentrations gradually taper off until undetectable.

Upon completion of tracer testing, the data is then applied based on the extent of reaction and summing over all elements at the reactor outlet as shown is Equation set A-2 and plotted.

$$\begin{aligned}\bar{c}_a &= \int_0^{\infty} c_a(\theta) E(\theta) d\theta, \\ -\frac{dc_a}{d\theta} &= ra(c_a(\theta))\end{aligned}\tag{A-2}$$

The resulting residence time distribution plots allow quick and relatively simple determination of reactor performance, mean residence times and contacting patterns. It should be noted that this method is strictly empirical and is not based on any physical model (Froment and Bischoff 1979). To overcome this problem the effective diffusion coefficient method, which combines both physical and empirical methods, is used.

The effective diffusion method utilizes material and energy balances of classical reactor modeling including effective transport properties. The effective properties can be predicted using theoretical and empirical correlations. Typically, effective properties are established using RTD results or error minimization techniques.

Effective properties from RTD data involve calculation of the tracer mean residence time and its standard deviation. Mean residence time is calculated using Equation A-3. Where t_m is the mean residence time for a given element of tracer.

$$t_m = \int_0^{\infty} tE(t)dt\tag{A-3}$$

Solution of Equation A-3 is typically done on RTD data using an integration method such as the Trapezoid rule or Simpson's rule. Once the t_m is calculated, determination of the standard deviation is completed using Equation A-4.

$$\sigma^2 = \int_0^{\infty} (t - t_m)^2 E(t) dt \quad \text{A-4}$$

Solution of Equation A-4 much like that of t_m requires an integration technique such as the Trapezoid or Simpson's rule. A set of simple examples for these calculations can be found in Fogler (1986).

For axial dispersion systems, Equation A-5 and iterative solution techniques provide effective Peclet values.

$$\frac{\sigma^2}{t_m^2} = \frac{2}{Pe} - \frac{2}{Pe^2} (1 - e^{-Pe}) \quad \text{A-5}$$

For methods involving both axial and radial components the concentration curves generated using the RTD method are used in conjunction with an optimization program to determine the effective properties. In this approach, Peclet numbers in both the axial and radial direction are adjusted by an optimization routine to minimize the error between the model predictions and the generated reactor data.

APPENDIX B

***Implementing the Software into Simulation Engines
using Dynamic Link Libraries***

All model code was developed to allow easy access from other programs. Simulation models were implemented as Fortran dynamic link libraries (DLL). Programs compiled as DLLs offer a much wider range of usability than stand along executables. DLL programs are essentially a system of subroutines that can be easily linked to Windows™ based programs such as HYSYS™, ASPEN PLUS™ and EXCEL™, to name only a few. Properly coded DLLs can be used in programs with little or no code modification, allowing great flexibility and ease of use.

The developed DLLs require only an interface such as Visual Basic or Visual Basic for Applications to collect required data and call the routines. The remainder of this section outlines the parameters required for solution, how to call the DLL from a Visual Basic interface, how the DLL is coded in Fortran and a brief overview of how to implement them into HYSYS™. It is assumed that the reader has a rudimentary knowledge of Visual Basic, Fortran and HYSYS™ programming.

To obtain an understanding of how the Visual Basic application interfaces with the DLL, we must first define and determine the data and variables critical to the model. Variables include reactor characteristics, flow assumptions, component data and relevant critical properties. The model DLL presently accepts the arrays and variables listed in the Table B-1.

Table B-1 Required Model Variables

Array Name	Variable	Definition
Feed data		
Feed(1)	Cao	Initial concentration of component A
Feed(2)	Cbo	Initial concentration of component B
Feed(3)	Cco	Initial concentration of component C
Feed(4)	Temp	Feed temperature
Feed(5)	Pres	Feed pressure
Feed(6)	Vzo	Inlet velocity
Feed(7)	Twall	Reactor wall temperature
Feed(8)	GasR	Gas constant for given units

Reactor Values

Reactr(1)	Zlen	Reactor length
Reactr(2)	Diam	Reactor diameter
Reactr(3)	Ntubes	Number of tubes

Peclet data

Pec(1)	Pez	Axial Peclet number
Pec(2)	Per	Radial Peclet number

Reaction data

Kin(1)	Ko	Kinetic constant
Kin(2)	Ea	Activation energy
Kin(3)	GasR	Gas constant for given units
Kin(4)	Hrxn	Heat of reaction
Kin(5)	Reft	Reference temperature for heat of reaction

Integer Array

Dimn(1)	8	Number of values in the feed array
Dimn(2)	Npde	Number of partial differential equations
Dimn(3)	Ncc	Number of components
Dimn(4)	Nzstp	Number of output steps in axial direction
Dimn(5)	Nrstp	Number of output steps in radial direction
Dimn(6)	Nrtype	Type of reactor 1= PFR 2= Axial only 3= Radial only 4= Axial and Radial
Dimn(7)	Ntype	1= Isothermal reactor 2= Adiabatic reactor

Component IN (a 2d array where the first number represents the component)

Cmp(*,1)	Mw	Component molecular weight
Cmp(*,2)	St1	Component stoichiometric coefficient
Cmp(*,3)		Unused
Cmp(*,4)	Tc	Critical temperature
Cmp(*,5)	Pc	Critical pressure
Cmp(*,6)	Vc	Critical volume
Cmp(*,7)	Cpa	Heat capacity coefficient A (see Eq. 3-27)
Cmp(*,8)	Cpb	Heat capacity coefficient B
Cmp(*,9)	Cpc	Heat capacity coefficient C
Cmp(*,10)	Cpd	Heat capacity coefficient D
Cmp(*,11)	Avis	Viscosity coefficient A (see Eq. 3-27)
Cmp(*,12)	Bvis	Viscosity coefficient B
Cmp(*,13)	Cvis	Viscosity coefficient C
Cmp(*,14)	Dvis	Viscosity coefficient D

Note that * indicated the component number, and cmp is dimensioned as cmp(ncc,15)

Other data

Eps	Allowable integration error tolerance
Zstp	Initial integration step size

mfxrn	Reaction routine switch 1= power law 2=user defined
Tmp(100,100)	Result array returning temperature profiles in the z and r direction
Cc(ncc,100,100)	Concentration results array for each component in the z and r direction

The above variables must be passed to the Fortran DLL as real values, with dimn as the only integer array. It is important to note that the kinetics array described above is adaptable for different types of user supplied kinetics and parameters are determined as dictated by the user developed kinetic routine.

Program results return to the interface through the CC, TMP arrays and all calculated data output to backup text files for more detailed analysis and review. Radial and axial profiles for temperature and concentration are the only values currently returned to the interface. Profiles for all other calculations are returned as text files (see below for a listing).

Summary of Text File Outputs

CompX.txt	Dimensionless concentration profiles for component X ranging from 1→ number of components
AMW.txt	Average molecular mass at each node (kg)
Cpinfo.txt	Average heat capacity at each node (kj/kmol *K)
Gvis.txt	Average viscosity at each node (cP)
Dhrxn.txt	Heat of reaction at each point (kj/kmol)
Faflw.txt	Cell flow rate for component 1 (kmol/s)
Massbal.txt	Average system mass at each node (kg)
Massdens.txt	Average system mass density at each node
Pressure.txt	System pressure profiles (atm)
Temp.txt	System temperature profiles (K)
Velocity.txt	Cell velocity (m/s)
Error.txt	Diagnostic file for program errors

All Visual Basic type interfaces, such as those found in HYSYS™, ASPEN PLUS™ or Excel™ require that the external DLL is properly declared and called. The following is a simple but typical DLL declaration in Visual Basic (See Appendix D for full Visual Basic code).

```
Declare Sub DLLfort Lib "fort\Debug\fort.dll" (feed As double, dimn As Integer, zstpin As Single, eps1 As Single)
```

The DLL declaration must go in as the first set of lines in the Visual Basic interface. The first 4 words declare a subroutine called DLLFORT (*Declare Sub DLLFORT*) as a library (*LIB*) that is located in the current directory under the folder named fort/debug (*fort/debug/fort.dll*). All information after the location indicates the variables being passed. It should be noted that only the name of the variable is used, and there is no need to indicate the dimensioning if arrays are present. For example, the double precision variable feed is an array and the variable zstpin is single precision variable. Determination of arrays and variables is done later in the Visual Basic program.

Now that the DLL has been properly declared, variables must be assigned and passed to the DLL. Assigning values to variables is simple, so we therefore focus our attention on calling the DLL. The DLL routine call is identical to normal subroutine calls in Visual Basic:

```
Call DLLfort(feed(1), dimn(1), zstpin, eps1)
```

The major point to notice here is that arrays must be passed by referencing the first element. Failure to do so will result in fatal program errors. Provided the user is familiar with Visual Basic programming the code in Appendix D (for Visual Basic in Excel™) can be easily modified to initialize and prepare the data required by the DLL.

Compilation of the DLL FORTRAN subroutines is only slightly different than that used to generate executable files. In compilers such as Digital Visual Fortran, you must create a DLL project instead of an executable project for proper compiling. In addition, the following series of comment lines must flow immediately after the main subroutine declaration (See Appendix C for more detailed source code):


```
subroutine dllfort(feed,dmn1,zstpin,eps1)
!DEC$ ATTRIBUTES DLLEXPORT::DLLfort
!DEC$ ATTRIBUTES ALIAS :'DLLfort'::DLLfort
```

Note that the first line is standard Fortran programming, and that the name of the main routine (DLLFORT) matches the name used to call the routine in Visual Basic. The next two lines are required, and are used by the operating system to name the routine for outside programs and allow access to the program. Other than these two additions, the remaining coding is standard Fortran. Now that the basics of the interface and DLL have been established, a brief overview of implementation into HYSYS™ is presented. Modifications to the main DLL are required only if the user needs implement kinetics other than power law or Rideal-Eley or wants additional data redirected from the text output to the interface. This is facilitated using standard Fortran 77 or Fortran 90 coding conventions.

Application of these DLL routines to HYSYS™ requires a visual basic interface as well as the HYSYS™ extension interface provided with HYSYS™. In order to use external routines such as the one developed it is suggested that the user modify the generic unit operations module provided with HYSYS™. The generic module provided by HYSYS™ provides an excellent interface, and a moderate amount of changes to work correctly. The changes are required to get physical properties and user information from HYSYS™ to the routine. Also note that all kinetic calculations must be entered into the kinetics subroutine of the DLL and not in HYSYS™.

APPENDIX C
FORTRAN CODE

C.1 MAIN SUBROUTINE

```

subroutine dlfort(feed,dmn1,reactr,kin,cmp1,res,cc,pec,tmp1,mfxrn,zstpin,eps1)
!DEC$ ATTRIBUTES DLLEXPORT::DLLfort
!DEC$ ATTRIBUTES ALIAS :'DLLfort'::DLLfort
! ALGORITHM 565
! PDETWO/PSTEM/GEARB: SOLUTION OF SYSTEMS OF TWO DIMENSIONAL
! NONLINEAR PARTIAL DIFFERENTIAL EQUATIONS
! BY D.K. MELGAARD AND R.F. SINCOVEC
! ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE 7,1 (MARCH 1981)
!
! Code modified by Rob Wade (OSU 1998) for the simulation
! of a pseudo homogenous fixed bed reactor systems with effective transport
!
=====
! variable, module and common declarations
  use data1
  implicit real (A-H,O-Z)
  include 'cnst1.inc'
  REAL H,S,XDX,TOUT,DY,HUSED,EXP,R,T0,EPS,ABS,DL,DLI,DEV
  INTEGER IX,NPDE,NSTEP,NX,NFE,NODE,MF,NY,NJE,NQUSED,IY,INDEX,I,IK
  REAL, ALLOCATABLE :: U3(:,:,:),U3OLD(:,:,:)
  REAL, ALLOCATABLE :: X(:),Y(:),uu(:)
  dimension iwork(65000),work(850000)
  real feed(8),reactr(5),kin(11),eps1
  real cmp1(4,25),cc(8,100,100)
  real dmn1(10),pec(2),tmp1(100,100)
  integer nrxn

  TYPE (PROPS)::COMP
  TYPE (kinet)::rxn

! Required by PDETWO for integration and jacobian determination
  COMMON /GEAR3/ HUSED,NQUSED,NSTEP,NFE,NJE
  COMMON /PROB/ DL,DLI

! =====
! ===== Open diagnostic results files =====
! =====
  OPEN (33, FILE = 'comp1.TXT')           ! for comp1
  OPEN (34, FILE = 'comp2.TXT')           ! for comp2
  OPEN (35, FILE = 'comp3.TXT')           ! for comp3
  OPEN (36, FILE = 'temp.TXT')            ! for temperature
  OPEN (37, FILE = 'amw.txt')              ! for pressure
  OPEN (38, FILE = 'velocity.TXT')         ! for molefractions
  OPEN (39, FILE = 'cpinfo.TXT')           ! for cp information
  OPEN (40, FILE = 'massbal.txt')          ! for # of comps present
  OPEN (41, FILE = 'gvis.txt')             ! for gas viscosity
  OPEN (42, FILE = 'molefrac.txt')         ! just what is says
  open (43, file = 'totalflow.txt')        ! total molar flow rate
  open (44, file = 'faflw.txt')            ! component 1 flow rate
  open (45, file = 'Reynolds.txt')         ! Reynolds number
  open (46, file = 'massdens.txt')         ! average mass density
  open (47, file = 'pressure.txt')         ! average pressure
  open (48, file = 'dhrxn.txt')            ! heat of reaction
  open (99, FILE = 'errorlog.txt',STATUS ='REPLACE') ! error log

```

```

=====
===== Initialize Constants =====
=====
nrxn=kin(9)          ! Reaction number
tiny=1.0e-6         ! Smallest real number used in code
NPDE=dmn1(2)        ! # OF PDES
NCC=dmn1(3)         ! # OF COMPONENTS
NX=dmn1(4)          ! # OF STEPS IN X DIRECTION
NY=dmn1(5)          ! # OF STEPS IN Y DIRECTION
! NRXN=1            ! # OF REACTION 1=ETHANE, 2=MTBE
NRL=7               ! # of points in r11 vector
slip=dmn1(8)        ! Set slip at wall condition (0=no, 1=yes)
refT=kin(8)         ! Reference temperature for heat of reaction
nrun=0              ! Equals 0 if first time routine is running
ntemp=dmn1(7)       ! If ntemp=1 then nonisothermal if =0 then isothermal
KODE=dmn1(6)        ! Sets reactor type: 1=pfr,2=axial,3=radial,4=ax/rad

! set switch and determine number of PDEs required for calculation
  if ((kode.eq.2)) npde=2*2
  if ((kode.eq.4)) npde=2*2

! determine model type and print header to output files
  do 911 ixi=33,44
    if (kode.eq.1) write(ixi,*)"Plug flow"
    if (kode.eq.2) write(ixi,*)"Axial only"
    if (kode.eq.3) write(ixi,*)"Radial only"
    if (kode.eq.4) write(ixi,*)"Axial & Radial only"
    if (ntemp.eq.0) write(ixi,*)"Isothermal"
    if (ntemp.eq.1) write(ixi,*)"Adiabatic "

! check reaction and phase while printing to output files
    if (nrxn.eq.1) then
      write(ixi,*)"Ethane Reaction - Vapor Phase"
      nphase=1
    endif
    if (nrxn.eq.2) then
      write(ixi,*)"MTBE Reaction - Liquid Phase"
      nphase=2
    endif
  endif
911 continue

! ===== Required by PDETWO for automatic array dimensioning
  niw=nx*ny*npde*4
  nw=850000
  NODE=NPDE*NX*NY

! ===== Sets integration method (22 = Adams with jacobian, 12 = gears with jacobian)
  MF=22
! ===== Required by PDETWO for initialization
  INDEX=1
! ===== Set initial value for length calculations (time if transient)
  T0=0.0
! ===== Supplied by user in interfacing program =====
  H=zstp            ! initial step size
  EPS=eps1         ! per step max error tolerance

```

```

! ===== Divides the domain outputs into 11 equal steps in x and y and sets final time/length
nystp=ny/10
nxstp=nx/10
ntout=nx-1          ! Sets the number of outer integration loops

```

```

! =====
! ===== Allocate Arrays =====
! =====

```

```

ALLOCATE (U3(NPDE,NX,NY),uu(ncc+1))
ALLOCATE (X(NX+1),Y(NY+1))

```

```

! Calls system clock to assist in determining computation time at end of calculations
CALL CPU_TIME ( timebegin )

```

```

! =====
! ===== Set array values and constants =====
! =====

```

```

RL1(1)=kin(1)          ! Ko KINETCS
RL1(2)=kin(2)          ! ACTIVATION ENERGY
RL1(3)=feed(8)         ! GAS CONSTANT
RL1(4)=reactr(1)       ! TUBE LENGTH
RL1(5)=reactr(2)       ! TUBE DIAMETER      7
RL1(6)=reactr(3)       ! NUMBER OF TUBES
vzo=feed(6)            ! INLET VELOCITY IN Z (M/S)
velr=feed(6)           ! Initial radial velocity (first run same as vzo)
zlen=r1(4)              ! Set reactor length
diam=r1(5)              ! Set reactor diameter
cto=sum(feed(1:3),1)   ! Set total inlet concentration
Twall=feed(7)          ! Set reactor wall temperature
CAO=feed(1)            ! INITIAL CA CONCENTRATION
CbO=feed(2)            ! INITIAL Ca CONCENTRATION
CcO=feed(3)            ! INITIAL Cc CONCENTRATION
radi=r1(6)*diam/2.0    ! Calculates radius of reactor
area=3.14*(radi**2)    ! Calculates area of reactor

Tin=sin1(ncc+2)        ! Dummy variable for inlet temperature
temp=Tin                ! Current temperature
xpin=feed(5)            ! Dummy variable for inlet pressure
pres=6 !xpin           ! Current pressure
xp=1.0 !xpin/xpin      ! Non-dimensionalized pressure
xKtgrm=kin(11)         ! Amount of catalyst in reactor

```

```

! ===== Calculation of dimensionless coefficients
dimN=zlen/(2*radi)     ! Dimensionless N see table 3-2
dimM=(2*radi)/(2*radi) ! Dimensionless M see table 3-2
dmsig=dimN             ! Dimensionless sigma see table 3-2
Pez=pec(1)             ! Axial Peclet number see table 3-2
Per=pec(2)             ! Radial Peclet number see table 3-2
DFH=1/(Pez*dimN)      ! Set final axial dimensionless group for calcs (eq 3-15)
DFV=1/(Per*dimM)      ! Set final radial dimensionless group for calcs (eq 3-15)

```

```

! ===== Set up reaction specific Kinetic information =====
if (nrxn.eq.1) then

```

```

        rxn(1)%name='EthRxn'           ! Reaction name
        rxn(1)%ko=r1(1)               ! PreExponential for k
        rxn(1)%EoverR=r1(2)/r1(3)    ! E/RT (activation energy)
        rxn(1)%hrxn=kin(4)*ntemp     ! Set heat of reaction
        dens=1.994345503611          ! Density of ethane @ 1100K
        refdHrxn=rxn(1)%hrxn         ! Heat of reaction at reference T
    endif
    if (nrxn.eq.2) then
    ! ===== Ridel-Eley parameters from Al-Jarallah kinetics MTBE ===
        rxn(1)%name='MTBE_Rxn'       ! Reaction name
        rxn(1)%ks=kin(1)              ! Kinetic parameter S
        rxn(1)%ka=kin(2)              ! Kinetic parameter A
        rxn(1)%kc=kin(3)              ! Kinetic parameter c
        rxn(1)%Keq=kin(4)             ! Equilibrium parameter
        rxn(1)%Eaks=kin(5)            ! Activation energy for S
        rxn(1)%Eaka=kin(6)            ! Activation energy for A
        rxn(1)%Eakc=kin(7)            ! Activation energy for C
        rxn(1)%hrxn=kin(10)*ntemp     ! Heat of reaction
        refdHrxn=rxn(1)%hrxn         ! Heat of reaction at ref T
        dens=654.89546                ! Average feed density at inlet
        epsilon=.32                    ! Reactor bed porosity
        dpart=0.00074                 ! Average catalyst particle size
    endif

! ===== Transfer physical property coefficients and data from interface
    comp(1)%name='comp1'              ! Component name
    comp(1)%xmw=cmp1(1,1)             ! Component molecular mass
    comp(1)%avis=cmp1(1,12)           ! Viscosity coefficient (a+bT+cT^2+dT^3)
    comp(1)%bvis=cmp1(1,13)           ! Viscosity coefficient (a+bT+cT^2+dT^3)
    comp(1)%cvis=cmp1(1,14)           ! Viscosity coefficient (a+bT+cT^2+dT^3)
    comp(1)%dvis=cmp1(1,15)           ! Viscosity coefficient (a+bT+cT^2+dT^3)
    comp(1)%acp=cmp1(1,8)              ! Heat capacity coeff. (Equation 3-27)
    comp(1)%bcp=cmp1(1,9)              ! Heat capacity coeff. (Equation 3-27)
    comp(1)%ccp=cmp1(1,10)            ! Heat capacity coeff. (Equation 3-27)
    comp(1)%dcp=cmp1(1,11)            ! Heat capacity coeff. (Equation 3-27)
    comp(1)%TC=cmp1(1,4)               ! Critical temperature
    comp(1)%PC=cmp1(1,5)               ! Critical pressure
    comp(1)%VC=cmp1(1,6)               ! Critical volume
    comp(1)%cLJ=0.809*comp(1)%vc**(1.0/3.0) !
    comp(1)%STOIK=cmp1(1,2)           ! Reaction stoichometric coefficient
!
! ===== See Comp(1) for comments
    comp(2)%name='comp2'
    comp(2)%xmw=cmp1(2,1)
    comp(2)%avis=cmp1(2,12)
    comp(2)%bvis=cmp1(2,13)
    comp(2)%cvis=cmp1(2,14)
    comp(2)%dvis=cmp1(2,15)
    comp(2)%acp=cmp1(2,8)
    comp(2)%bcp=cmp1(2,9)
    comp(2)%ccp=cmp1(2,10)
    comp(2)%dcp=cmp1(2,11)
    comp(2)%TC=cmp1(2,4)
    comp(2)%PC=cmp1(2,5)
    comp(2)%VC=cmp1(2,6)
    comp(2)%cLJ=0.809*comp(2)%vc**(1.0/3.0)

```

```

    comp(2)%STOIK= cmp1(2,2)
!
! ===== See Comp(1) for comments
    comp(3)%name='comp3'
    comp(3)%xmw=cmp1(3,1)
    comp(3)%avis=cmp1(3,12)
    comp(3)%bvis=cmp1(3,13)
    comp(3)%cvis=cmp1(3,14)
    comp(3)%dvis=cmp1(3,15)
    comp(3)%acp=cmp1(3,8)
    comp(3)%bcp=cmp1(3,9)
    comp(3)%ccp=cmp1(3,10)
    comp(3)%dcp=cmp1(3,11)
    comp(3)%TC=cmp1(3,4)
    comp(3)%PC=cmp1(3,5)
    comp(3)%VC=cmp1(3,6)
    comp(3)%cLJ=0.809*comp(3)%vc**(1.0/3.0)
    comp(3)%STOIK= cmp1(3,2)

! ===== Determine sum of stoichiometric coefficients for expansion calculations
    SUMSTK=0.0
    do 111 k=1,ncc
        sumstk=sumstk+comp(k)%stoik
    111 continue

! ===== Establish output grid and prep for solution
    DX = 1.0/(FLOAT(NX)-1.0)
    DY = 1.0/(FLOAT(NY)-1.0)
    do 120 IY=1,NY
        Y(IY)=(FLOAT(IY)*DY-DY)
    120 continue
    DO 121 IX=1,NX
        X(IX)=(FLOAT(IX)*DX-DX)
    121 CONTINUE

! ===== PDETWO work array for reference and dimensioning
    IWORK(1) = NPDE
    IWORK(2) = NX
    IWORK(3) = NY
    IWORK(4) = 12
    IWORK(5) = nw
    IWORK(6) = niw

! ===== Print and initialize solution arrays
write(33,*)"tin",tin
    if ((kode.eq.1).or.(kode.eq.3)) then
        DO 160 IY = 1,NY
            DO 160 IX=1,NX
                do 160 ii=1,npde
                    U3(1,ix,IY)=sin1(1)           ! Dimensionless concentration
                    U3(2,ix,IY)=Tin/Tin           ! Dimensionless temperature
                160 CONTINUE
            endif
            if ((kode.eq.2).or.(kode.eq.4)) then
                DO 560 IY = 1,NY
                    DO 560 IX=1,NX

```

```

do 560 ii=1,npde
    U3(1,ix,IY)=1.0
    uu(ii)=sin1(ii)
    u3(2,ix,iy)=0.0
    u3(3,ix,iy)=tin/tin      !temp
    u3(4,ix,iy)=0.0
560 CONTINUE
endif

! ===== Set dummy and variable transfer arrays
do 161 iy=1,ny
do 161 ix=1,nx
do 161 ii=1,ncc
    c(ii,iy)=sin1(ii)
    cc(ii,1,iy)=sin1(ii)
161 continue
    ctotal=cto                ! Total concentration
    xmbal=comp(1)%xmw         ! Mass balance
    tmp=temp/Tin              ! Dummy temp
    tmp1=Tin                  ! Dummy temp
    ttmp=temp/Tin            ! Dummy temp
    fa=vzo*area*cao          ! Component a flow rate
    fb=vzo*area*cbo          ! Component b flow rate
    fc=vzo*area*cco          ! Component c flow rate
    nlpr=0                    ! Dummy index variable
    vzmt=vzo                  ! Dummy velocity

! ===== Determine physical properties and component fractions

do 165 IY=1,ny
do 165 ix=1,nx
    call props1(uu,npde,ix,iy)
165 continue

! ===== Print out remaining header information for output files
!
write(44,*)velr,area,c(1,1),cao
write(33,*)"This is the matrix (z down, r righth)"
write(33,263)zout,cc(1,1,1:ny:nystp)
write(34,*)"This is the matrix (z down, r righth)"
write(34,263)zout,cc(2,1,1:ny:nystp)
write(35,*)"This is the matrix (z down, r righth)"
write(35,263)zout,cc(3,1,1:ny:nystp)
write(36,*)"System cell temps (k) (z down, r righth)"
write(36,264)zout,ttmp(1,1:ny:nystp)*temp
write(37,*)"Cell average molecular mass (kgmol/kg)"
write(37,264)zout,amw(1,1:ny:nystp)
write(38,*)"Cell average molar velocity (m/s)"
write(38,264)zout,vzmt(1,1:ny:nystp)
write(39,*)"Average system heat capacity (Cp) (kj/kgmol*K)"
write(39,264)zout,cp(4,1,1:ny:nystp)
write(40,*)"Cell total mass (kg)"
write(40,264)zout,xmbal(1,1:ny:nystp)
write(41,*)"Average system gas viscosity (gvis) 1(kg/m/s)=.1cp"
write(41,263)zout,gvis(4,1,1:ny:nystp)
write(42,*)"This is the matrix overall mole fraction"

```



```

write(42,263)zout,xfr(4,1,1:ny:nystp)
write(43,*)"This is Cell total molar concentration (kmol/m3)"
write(43,263)zout,ctotal(1,1:ny:nystp)
write(44,*)"This is Cell Fa flow rate (kmol/s)"
write(44,263)zout,fa(1,1,1:ny:nystp)
write(45,*)"This is cell Reynolds #"
write(45,264)zout,reyn(1,1:ny:nystp)
write(46,*)"Cell mass density (kg/m3)"
write(46,263)zout,xdens(1,1:ny:nystp)
write(47,*)"System pressure (atmospheres)"
write(47,264)zout,xp(1,1:ny:nystp)*xpin
write(48,*)"System heat of reaction kj/kmole"
write(48,265)zout,xdrxn(1,1:ny:nystp)

```

```

! =====
! ===== SET UP INTEGRATION & LOOP =====
! =====
!
ZZSTP=1.0/real(nx-1)      ! Set PDETWO output step
ZOUT=ZZSTP

DO 260 I=1,ntout
    ii=i+1
    nrun=1000
!    write(99,*)"Looping Integrator...ZOUT=",zout
CALL DRIVEP (NODE,T0,H,U3,zout,EPS,MF,INDEX,WORK,IWORK,X,Y)
!
! =====
! ===== After return from integration loop, update storage arrays and physical properties
do 261 iy=1,ny
do 261 ix=1,nx

! ===== Update concentrations and conversions ethane reaction
if (nrxn.eq.1) then
    uu(1)=u3(1,ix,iy)
    call conup(UU,NPDE,ix,iy)
endif

! ===== Update concentrations and conversions MTBE reaction

if (nrxn.eq.2) then
    uu(1)=u3(1,ix,iy)
    call conup(UU,NPDE,ix,iy)
endif
261 continue
do 266 iy=1,ny
do 266 ic=1,ncc
cc(ic,i+1,iy)=c(ic,iy)
ttmp(i+1,iy)=tmp(iy)
tmp1(i+1,iy)=ttmp(i+1,iy)*tin
266 continue
do 159 IY=1,ny
do 159 ix=1,nx
do 158 ji=1,ncc
uu(ji)=c(ji,iy)

```

```

158   continue
           call props1(uu,npde,ix,iy)
159   continue
           write(33,263)zout,cc(1,ii,1:ny:nystp)
263   format (F8.4,4x,11F8.4)
264   format (f8.4,4x,11f10.2)
265   FORMAT (f8.4,4x,11E10.4)
! ===== Output loop integration results
           write(34,263)zout,cc(2,ii,1:ny:nystp)
           write(35,263)zout,cc(3,ii,1:ny:nystp)
           write(36,264)zout,tmp(1:ny:nystp)*tin
           write(37,264)zout,amw(ii,1:ny:nystp)
           write(38,264)zout,vzmt(ii,1:ny:nystp)
           write(39,264)zout,cp(4,ii,1:ny:nystp)
           write(41,263)zout,gvis(4,ii,1:ny:nystp)
           write(40,264)zout,xmbal(ii,1:ny:nystp)
           write(42,263)zout,xfr(4,ii,1:ny:nystp)
           write(43,263)zout,ctotal(ii,1:ny:nystp)
           write(44,263)zout,fa(1,ii,1:ny:nystp)
           write(45,264)zout,reyn(1,1:ny:nystp)
           write(46,263)zout,xdens(1,1:ny:nystp)
           write(47,264)zout,xp(ii,1:ny:nystp)*xpin
           write(48,265)zout,xdhrxn(ii,1:ny:nystp)
! ===== increment zout and return to integration loop
           zout=zout+ZZSTP
260   CONTINUE
! =====
! ===== CLEAN UP & TURNOUT THE LIGHTS =====
! =====
! ===== Determine current time and calculate computation time
           CALL CPU_TIME ( timeend )
           cputime=(timeend-timebegin)
           cputm=cputime
!   do 888 k=33,33+ncc-1
           write(33,*) '
           write(33,*) '
           write(33,'(a9,4x,f6.3)')'time ',cputime
           write(33,'(a6,4x,f9.1)')'Pez=',pez
           write(33,'(a6,4x,f9.1)')'Per=',per
           write(99,*)"No futher errors....."
           write(99,'(a9,4x,f6.3)')'time ',cputime
888   continue
!           write(*,*) 'Time of operation was', (timeend-timebegin), ' seconds'

! ===== Free up memory and end =====
           DEALLOCATE (U3,X,Y,UU)
           do 889 i=1,49
               CLOSE (i)
889   continue
           close (99)           ! error log
! 1000 STOP
           RETURN
           END

```

C.2 Horizontal boundary condition routine

```
SUBROUTINE BNDRYH (T,X,Y,U,AH,BH,CH,NPDE)
!
! DEFINE THE HORIZONTAL BOUNDARY CONDITIONS
!
  implicit real (A-H,O-Z)
  REAL T,U,X,Y,BH,AH,CH
  INTEGER NPDE

  COMMON /PROB/ DL,DLI
    include 'cnst1.inc'

  DIMENSION U(NPDE),AH(NPDE),BH(NPDE),CH(NPDE)

! ===== Determine reactor type
  IF(kode.EQ.1) GO TO 100      ! PFR
  IF(kode.EQ.2) GO TO 2000    ! Axial only
  IF(kode.EQ.3) GO TO 3000    ! Radial only
  IF(kode.EQ.4) GO TO 4000    ! Axial and Radial

! =====
! 2. PFR Reactor      !! AH*U(l) + BH du/dy = CH
! =====

100  do 110 i=1,npde
      AH(i) = 0.0      ! A FOR HORIZONTAL BC
      BH(i) = 1.0      ! B FOR HORIZONTAL BC
      CH(i) = 0.0      !! FOR HORIZONTAL BC
110  continue
      goto 9999

! =====
! 2. Axial Profile  !! AH*U(l) + BH du/dy = CH
! =====
2000 continue
      DO 2210 l=1,NPDE
          ah(l)=0.0
          bh(l)=1.0
          ch(l)=0.0
2210  CONTINUE
      GOTO 9999

! =====
! 2. Radial Profile      !! AH*U(l) + BH du/dy = CH
! =====

3000  do 3100 i=1,npde
      if (y.EQ.0)then
          AH(i)=0.0
          BH(i)=1.0
          CH(i)=0.0
      end if
      IF (Y.EQ.1) THEN
          AH(l)=1.0
```

```

        BH(l)=slip
        CH(l)=0.0

        ah(1)=1.0
        bh(1)=slip
        ch(1)=0.0

        ah(2)=1.0
        bh(2)=1.0
        ch(2)=twall/tin
    END IF
3100 continue
    GO TO 9999

! =====
! 1. Ax/Radial diff w/v Profile
! =====

4000 do 4100 i=1,npde
    if (y.EQ.0)then
        AH(i)=0.0
        BH(i)=1.0
        CH(i)=0.0
    end if
    IF (Y.EQ.1) THEN
        AH(l)=1.0
        BH(l)=slip
        CH(l)=0.0

        ah(1)=1.0
        bh(1)=slip
        ch(1)=0.0

        ah(3)=1.0
        bh(3)=1.0
        ch(3)=twall/tin
    END IF
4100 continue
    goto 9999

! ***** TERMINATE SUBROUTINE AND RETURN *****
9999 CONTINUE
    RETURN
    END

```

C.3 Vertical boundary condition routine

```
SUBROUTINE BNDRYV (T,X,Y,U,AV,BV,CV,NPDE)
!
! DEFINE THE VERTICAL BOUNDARY CONDITIONS
!
  REAL T,U,X,Y,BV,AV,CV
  INTEGER NPDE

  COMMON /PROB/ DL,DLI
    include 'cnst1.inc'

  DIMENSION U(NPDE),AV(NPDE),BV(NPDE),CV(NPDE)

  if (nlooper.ne.100) nlooper=100

! ===== Determine reactor type
  IF(kode.EQ.1) GO TO 100          ! PFR
  IF(kode.EQ.2) GO TO 2000       ! Axial only
  IF(kode.EQ.3) GO TO 3000       ! Radial only
  IF(kode.EQ.4) GO TO 4000       ! Axial & Radial

! =====
! 2. PFR Reactor
! =====
! == Sets derivative equal to zero over entire radial slice
100  continue
    do 110 i=1,npde
      AV(i)=0.0
      BV(i)=1.0
      CV(i)=0.0
110  continue
    goto 9999

! =====
! 2. Axial Profile      !! AV*U(l) + BV du/dx = CV
! =====

2000  if (x.eq.0) then
      DO 2210 l=1,NPDE
        aV(l)=1.0
        bV(l)=0.0
        cV(l)=U(l)
2210  CONTINUE
      end if
      if (x.gt.0) then
        DO 2220 l=1,NPDE
          AV(i) = 0.0
          BV(i) = 1.0
          CV(i) = 0.0
2220  CONTINUE
        end if
        goto 9999

! =====
```

```

! 2. Radial Profile                                !! AV*U(l) + BV du/dx = CV
! =====
3000 if (x.EQ.0) then
      DO 3210 l=1,NPDE
          aV(l)=1.0
          bV(l)=0.0
          cV(l)=u(i)

3210  CONTINUE
      end if

      if (x.gt.0) then
          do 3300 i=1,npde
              AV(i) = 0.0
              BV(i) = 1.0
              CV(i) = 0.0
3300  continue
          end if
          GOTO 9999
! =====
! 1. Ax/Radial diff w/v Profile
! =====

4000 if (x.EQ.0) then
      DO 4210 l=1,NPDE
          aV(l)=1.0
          bV(l)=0.0
          cV(l)=U(l)
4210  CONTINUE
      end if
      if (x.gt.0) then
          do 4300 i=1,npde
              AV(i) = 0.0
              BV(i) = 1.0
              CV(i) = 0.0
4300  continue
          end if
          GOTO 9999

! ***** TERMINATE SUBROUTINE AND RETURN *****
9999 CONTINUE
      RETURN
      END

```

C.4 SUBROUTINE DIFFH (T,X,Y,U,DH,NPDE)

```
! ===== Required by PDETWO serves only as a dummy routine for this work
!
! DEFINE THE HORIZONTAL DIFFUSION COEFFICIENTS
!
  implicit real (A-H,O-Z)
  REAL T,U,X,Y,DH,DFH,DFV
  INTEGER NPDE

  COMMON /PROB/ DL,DLI
    include 'cnst1.inc'

  DIMENSION U(NPDE),DH(NPDE,NPDE)
  IF(kode.EQ.1) GO TO 100
  IF(kode.EQ.2) GO TO 2000
  IF(kode.EQ.3) GO TO 3000
    IF(kode.EQ.4) GO TO 4000

! =====
! 2. PFR Reactor
! =====
100 do 110 i=1,npde
    do 110 l=1,npde
        DH(I,L) =dfh
110 continue
    GO TO 9999

! =====
! 2. Axial Profile
! =====
2000 DO 2210 l=1,NPDE
        DO 2210 L=1,NPDE
            DH(I,L) =dfh
2210 CONTINUE
    GO TO 9999

! =====
! 2. Radial Profile
! =====
3000 do 3100 i=1,npde
        do 3100 l=1,npde
            DH(I,L)=dfh
3100 continue
    GO TO 9999

! =====
! 1. Ax/Radial diff w/v Profile
! =====
4000 DO 4210 l=1,NPDE
        DO 4210 L=1,NPDE
            DH(I,L) =dfh
4210 CONTINUE
    GOTO 9999
! ***** TERMINATE SUBROUTINE AND RETURN *****
9999 CONTINUE
    RETURN
    END
```

C.5 Vertical diffusion routine

! ===== Required by PDETWO serves only as a dummy routine for this work

```
SUBROUTINE DIFFV (T,X,Y,U,DV,NPDE)
!
! DEFINE THE VERTICAL DIFFUSION COEFFICIENTS
!
  implicit real (A-H,O-Z)
  REAL T,U,X,Y,DV,DFH,DFV
  INTEGER NPDE
  COMMON /PROB/ DL,DLI
  include 'cnst1.inc'
  DIMENSION U(NPDE),DV(NPDE,NPDE)
  IF(kode.EQ.1) GO TO 100
  IF(kode.EQ.2) GO TO 2000
  IF(kode.EQ.3) GO TO 3000
  IF(kode.EQ.4) GO TO 4000
! =====
! 2. PFR Reactor
! =====
100 do 110 i=1,npde
  do 110 l=1,npde
    DV(i,l) = dfv
110 continue
  GO TO 9999
! =====
! 2. Axial Profile
! =====
2000 DO 2210 l=1,NPDE
  DO 2210 L=1,NPDE
    DV(l,L) =dfv
2210 CONTINUE
  GO TO 9999
! =====
! 2. Radial Profile
! =====
3000 do 3100 i=1,npde
  do 3100 l=1,npde
    Dv(l,L)=dfv
3100 continue
  goto 9999
! =====
! 1. Ax/Radial diff w/v Profile
! =====
4000 DO 4210 l=1,NPDE
  DO 4210 L=1,NPDE
    DV(l,L) =dfv
4210 CONTINUE
  GOTO 9999
! ***** TERMINATE SUBROUTINE AND RETURN *****
9999 CONTINUE
  RETURN
  END
```


C.6 Function evaluation routine

```

SUBROUTINE F(T,X,Y,U,UX,UY,DUXX,DUYY,DU DT, NPDE)
!
! DEFINE THE PDE
!
      use data1
      implicit real (a-h,o-z)

REAL T,U,X,Y,UX,UY,DUXX,DUYY,DU DT,EXP
INTEGER NPDE

COMMON /PROB/ DL,DLI
DIMENSION U(NPDE),UX(NPDE),UY(NPDE),DUXX(NPDE,NPDE),&
DUYY(NPDE,NPDE),DU DT(NPDE),alph(npde)

      include 'cnst1.inc'
      type (kinet)::rxn
      type (props) ::comp
! =====
! ===== Calculate current x and y grid point
! =====
      njx=nint(x/dx)+1           ! Determine current x grid location
      njy=nint(y/dy)+1           ! Determine current y grid location
      !       write(33,*)"func"
      vnpr=7.0                   ! Set n for power law velocity profiles
! ===== Call reaction specific kinetics routines

      cc1(1,njx,njy)=u(1)         ! Fill temp. concentration array
      cav=0.0                     ! Initialize current average concentration
      do 5 inj=2,nj
          cav=cav+cc1(1,njx,inj)/10 ! Average the concentration over the current
radius
!       write(33,*)"cav=",cav
      5 continue

! ===== Determine Reaction and call kinetics routines =====
      if (nrxn.eq.1) then
          call KN1(RATEf,U,NPDE,njx,njy) !name of ethane routine
      endif
      if (nrxn.eq.2) then
          call mtbKN1(RATEf,U,NPDE,njx,njy) !name of MTBE routine
      endif

! ===== Set dummy temperature and update physical properties
      tmp(njy)=u(2)                ! temp array for temperature
      if ((kode.eq.2).or.(kode.eq.4)) tmp(njy)=u(3) ! if axial profiles present store array for
temp
      call props1(u,npde,njx,njy) ! call and update physical
properties

! ===== Check reaction and update average conversion Xconv and velocities
      if (nrxn.eq.1) then
          xcon=(1-cav)/(1+cav) ! calculate radial average conversion
          velr=vzo*(1+sumstk*xcon)/tmp(njy) ! calculate average radial velocity

```

```

          vav=velr
later use                                     ! temp store average velocity for

! ===== If radial profiles are present then implement 1/n velocity power law
          if (kode.ge.3) then

              vmax=(area/6.28)*((vnpr+1.0)*(2.0*vnpr+1.0)/(vnpr*vnpr))*vzo*(1.0+sumstk*xcon)
              ! calculate vmax using 1/7 powerlaw
              velr=(vmax/tmp(njy))*(1-y)**((1.0)/vnpr) !(0.14286)
              ! determine v(r) using 1/7 powerlaw
              endif
          endif
!      WRITE(33,*)"VELR=",VELR

! === If gas phase then current conversion is determined and velocity updated
          if (nrxn.eq.2) velr=vzo
          if (velr.eq.0) velr=tiny                ! if velocity is zero set to smallest
possible value
          vzmt(njx,njy)=velr                    ! fill storage array for velocities

          if (xPin.eq.0) write(99,*)"xpin is zero"      ! test and output error set if xpin=0
          if (diam.eq.0) write(99,*)"diam=0 in function" ! test and output error set if diam=0

! ===== hicks pressure drop equation =====
          fr=6.8*(((1-epsilon)**(1.3))/(epsilon**3.0))*(reyn(njx,njy)**(-0.2))
          !xp(njx,njy)=1.0-(fr*654*velr*velr*zlen/xPin)*x
          xp(njx,njy)=1-(fr*area*zlen/xPin/area/100)*x

! =====
! ===== Update current component flow rates
! =====
          fa(1,njx,njy)=velr*area*c(1,1)*cao
          fa(2,njx,njy)=velr*area*c(2,1)*cao
          fa(3,njx,njy)=velr*area*c(3,1)*cao
          ftotal=sum(fa(1:ncc,njx,njy),1)

! =====
! ===== Update velocity profiles to account for radial conditions
! =====

! ===== Set 1/r terms so division by zero does not occur at r=0 via L'HoPitols rule
          DO 10 l=1,NPDE
              if (y.eq.0) alph(l)=duyy(l,l)
              if (y.ne.0) alph(l)=(1/y)*Uy(l)
10      CONTINUE
!      ENDIF

```

```

=====
! ===== Determine Routine for reactors =====
=====
IF(kode.EQ.1) GO TO 100          ! PFR
IF(kode.EQ.2) GO TO 2000       ! Axial only
IF(kode.EQ.3) GO TO 3000       ! Radial only
IF(kode.EQ.4) GO TO 4000       ! Axial and radial

! ===== PFR (NO Axial/radial diff) =====
100  continue
!    do 110 i=1,npde
!      write(77,*)ratef(1),zlen,velr
! =====Concentration
      DUDT(1)=(ratef(1)*zlen/(velr))
      if(nrxn.eq.1) dudt(1)=dudt(1)/tmp(njy)  ! correction if gas phase
! =====Energy
      dudt(2)=(ratef(1)*dHrxn*zlen/(VELR*xdens(njx,njy)*cp(4,njx,njy)*tin))*ntemp
      !dudt(2)=dudt(2)-4*UH*(tmp(njy)-Twall/Tin)*zlen/diam
! 110 CONTINUE
      GO TO 9999

! ===== Axial only diffusion =====
2000 continue

! =====Concentration
      dudt(1)=(u(2)+ratef(1)*zlen/(velr))

      if(nrxn.eq.1) dudt(1)=dudt(1)/tmp(njy)  ! correction if gas phase
      dudt(2)=u(1)*dfh
      ztemp=dfh*10/(velr*dens*cp(4,njx,njy)*tin)
! =====Energy
      dudt(3)=(u(4)+ratef(1)*dHrxn*zlen/(tin*velr*xdens(njx,njy)*cp(4,njx,njy)))+ztemp
      dudt(4)=u(3)*dfh*ntemp

      GO TO 9999

! ===== Radial only diffusion =====

3000 continue
      dddy=dfv*duyy(1,1)+dfv*alph(1)          ! Sets (1/r)(dC/dr) term
      dddy2=(dfv*duyy(2,2)+dfv*alph(2))*ntemp ! Sets (1/r)(dT/dr) term

! =====Concentration
      DUDT(1)=(ratef(1)*zlen/(velr))+dddy*zlen/velr
      if(nrxn.eq.1) dudt(1)=dudt(1)/tmp(njy)  ! correction if gas phase
! =====Energy
      dudt(2)=(ratef(1)*dHrxn*zlen/(velr*xdens(njx,njy)*cp(4,njx,njy)*tin))*ntemp+dddy2*zlen/velr
r
      GOTO 9999

! ===== Axial and radial diff =====
4000  continue
      dddy=dfv*duyy(1,1)+dfv*alph(1)
      dddy2=dfv*duyy(2,2)+dfv*alph(2)
      ztemp=dfh/(velr*xdens(njx,njy)*cp(4,njx,njy)*tin)+dddy2*zlen/velr

```

```
dudt(1)=(u(2)+ratef(1)*zlen/(velr)+ddy*zlen/velr)
if(nrxn.eq.1) dudt(1)=dudt(1)/tmp(njy)
dudt(2)=(u(1))*dfh
dudt(3)=u(4)+ratef(1)*dHrxn*zlen/(VELR*xdens(njx,njy)*cp(4,njx,njy)*tin)+ztemp
dudt(4)=(u(3))*dfh*ntemp
GOTO 9999
```

```
! ***** TERMINATE SUBROUTINE AND RETURN *****
9999 CONTINUE
```

```
RETURN
END
```

C.7 Ethane kinetics routine

```
! ETHANE CRACK REACTION POWER LAW KINETICS
! =====
! ===== THIS ROUTINE CALCULATES THE KINETICS =====
! =====

SUBROUTINE KN1(RATE,U,NPDE,njx,njy)

    use data1
    implicit real (A-H,O-Z)

    include 'cnst1.inc'

    real U(NPDE),rate(nncc)
    type (kinet)::rxn
    type (props) ::comp

! ===== Update current conversion & calculate Cb and Cc from Ca
    call conup(U,NPDE,njx,njy)

! ===== Set and calculate rate constant
    ako=rxn(1)%ko
    ak1=ako*exp(-(rxn(1)%EoverR/(tin*tmp(njy))))

! ===== Calculate and update reaction rates (rate(1) in only one currently used
    RATE(1)=ak1*comp(1)%stoik*u(1)
    RATE(2)=ak1*comp(2)%stoik*u(1)
    RATE(3)=ak1*comp(3)%stoik*u(1)

RETURN
END
```

C.8 MTBE kinetics routine

```
! Kinetics expression for MTBE synthesis (Al-Jarallah et al. 1988)
! See Table 2-1 for detailed view of kinetics
! =====
! ===== THIS ROUTINE CALCULATES THE KINETICS =====
! =====
SUBROUTINE mtbKN1(RATE,U,NPDE,njx,njy)
  use data1
  implicit real (A-H,O-Z)
  include 'cnst1.inc'
  real U(NPDE),rate(nncc),rxup,rxdn
  type (kinet)::rxn
  type (props) ::comp

! ===== Update current conversion & calculate Cb and Cc from Ca
  call conup(U,NPDE,njx,njy)
  tm=tmp(njy)*tin          ! update dummy temperature
  Alf=1.0                  ! Al-Jarallah et al. 1988 exponential on A
  Bet=0.5                  ! Al-Jarallah et al. 1988 exponential on B
  Cap=1.5                  ! Al-Jarallah et al. 1988 exponential on C

! ===== Assign activation energies for kinetic constant calculations
  Eaks=rxn(1)%Eaks
  EaKa=rxn(1)%EaKa
  EaKc=rxn(1)%EaKc

  xks=(1.2E+13)*exp(Eaks/(8.314*tm))
  xKa=(5.1E-13)*exp(EaKa/(8.314*tm))
  xKc=(1.6E-16)*exp(EaKc/(8.314*tm))
  xkeq=-13.482+4388.7/tm+1.2353*log(tm)-0.013849*tm
  xkeq=exp(xkeq+(2.5923e-5)*tm**2-(3.1881e-8)*tm**3)

! ===== Calculate numerator, denominator and result (see kinetic expression in table 2-1)
  rxnum1=(c(1,njy)**alf)*(c(2,njy)**bet)-(c(3,njy)*cap)/xkeq
  rxden=(1+xKa*c(1,njy)+xKc*c(3,njy)**alf)
  rxnum1=rxnum1*xks*xKa**alf

! ===== Complete reaction rate calculations
  rate(1)=-(rxnum1/rxden)*xktgrm
  rate(2)=-(rxnum1/rxden)*xktgrm
  rate(3)=-(rate(1)+rate(2))

RETURN
END
```

C.9 Physical property calculation routine

```

subroutine props1(u,npde,njx,njy)
  use data1
  implicit real (A-H,O-Z)
  include 'cnst1.inc'
  real U(NPDE)
  type (kinet)::rxn
  type (props)::comp

! ===== Check reactor type and variable transfer
  if ((kode.eq.2).or.(kode.eq.4)) tmp(njx)=u(3)
  !IF ((KODE.EQ.3)) TMP(NJX)=U(2)

! ===== Transfer of dummy variables and determine if this is the first function call
  tm=tmp(njy)*Tin
  IF (NRUN.EQ.0) TMP(NJY)=TEMP/Tin
  IF (TM.EQ.0) TM=TEMP

!
! =====
!
! ===== Count # of components present =====
!
  ncp=0          ! Set value to 0
  do 10 k=1,ncc
    if (c(k,njy).gt.(0.000001)) ncp=ncp+1
  10 continue

!
! =====
!
! ===== Determine cell molefraction =====
!
  cttotal=0      ! Set value to 0
  do 100 k=1,ncc
    cttotal=cttotal+c(k,njy)
  100 continue
  do 110 k=1,ncc
    xfrk(k,njx,njy)=c(k,njy)/cttotal          ! Simple mole% calc
    if (cttotal.lt.(0.000001))xfrk(k,njx,njy)=0.0    ! Set to 0 if too small
  110 continue

  !===== Total mole fraction for calculation checks
  xfrk(nc+1,njx,njy)=xfrk(1,njx,njy)+xfrk(2,njx,njy)+xfrk(3,njx,njy)
  !===== Update average molecular weight using mole% and MW
  amw(njx,njy)=xfrk(1,njx,njy)*comp(1)%xmw+xfrk(2,njx,njy)*comp(2)%xmw
  amw(njx,njy)=(amw(njx,njy)+xfrk(3,njx,njy)*comp(3)%xmw)

!
! =====
!
! ===== determine cell total mass =====
!
  xmbal(njx,njy)=0.0
  do 200 i=1,ncc
    ! ===== Perform mass balance at current point
    xmbal(njx,njy)=comp(1)%xmw*c(1,njy)+comp(2)%xmw*c(2,njy)+comp(3)%xmw*c(3,njy)
    !
    write(33,*)comp(i)%xmw,c(i,njy),xmbal(njx,njy)
  200 continue
  7  xdens(njx,njy)=xmbal(njx,njy)*cao          ! approximate density

!
! =====

```

```

! ===== Determine cell total flow rate =====
! =====
ct=0.0          ! Set total dummy flow to 0
do 300 i=1,ncc
    ct=ct+c(i,njy) ! Add all dummy flows
300 continue
ctotal(njx,njy)=ct

!! =====
!! ===== Determine cell heat capacity =====
!! =====
! ===== See equation 3-27 =====
! From Felder in kJ/kgmol*K
cptotal=0.0
cp(ncc+1,njx,njy)=0.0
do 400 k=1,ncc
    cpval=comp(k)%acp+comp(k)%bcp*tm+comp(k)%ccp*tm**2+comp(k)%dcp*tm**3
    cp(k,njx,njy)=cpval*xfrk(k,njx,njy)
    cptotal=cptotal+cp(k,njx,njy)
    cp(ncc+1,njx,njy)=cptotal
400 continue
!! =====
!! ===== Determine current DHrxn =====
!! =====
! ===== See equations 3-28 through 3-31 =====
! From Felder in kJ/kgmol*K
covera=comp(3)%stoik/comp(1)%stoik
bovera=comp(2)%stoik/comp(1)%stoik
dalpha=(covera*comp(3)%acp-covera*comp(2)%acp-comp(1)%acp)
dbeta=(covera*comp(3)%bcp-covera*comp(2)%bcp-comp(1)%bcp)
dgamma=(covera*comp(3)%ccp-covera*comp(2)%ccp-comp(1)%ccp)
ddelta=(covera*comp(3)%dcp-covera*comp(2)%dcp-comp(1)%dcp)
! ===== Equation 3-32 =====
dHrxn=refdHrxn+dalpha*(tm-refT)+(dbeta/2)*(tm**2-refT**2)
dHrxn=dHrxn+(dgamma/3)*(tm**3-refT**3)+(ddelta/4)*(tm**4-refT**4)
xdhrxn(njx,njy)=dhrxn          ! Store dHrxn history
dHrxn=dHrxn*ntemp             ! Set to 0 of isothermal
! =====
! ===== determine cell gas or liquid viscosity =====
! =====
gvttl=0.0
gvis(ncc+1,njx,njy)=0.0
do 500 k=1,ncc
    gvis(k,njx,njy)=comp(k)%avis+comp(k)%bvis/tm+comp(k)%cvis*log(tm)+comp(k)%dvis*(t
m**0.0)
    gvis(k,njx,njy)=gvis(k,njx,njy)*xfrk(k,njx,njy)
    gvttl=gvttl+gvis(k,njx,njy)
500 continue
! convert cp to kg/m/s
gvttl=gvttl*10
gvis(4,njx,njy)=gvttl
! stop

```


C.10 Data module

```
module data1
! ===== Reactor physical values (length, diameter, area, # tubes and type)
  TYPE RCTR
    REAL ::      ZLEN,DIAM,AREA,E
    INTEGER  :: NTUBES,NTYPE
  END TYPE RCTR

! ===== Stream data (concentration, flow, temp, pressure, molecular mass, density)
  TYPE STREAMS
    REAL ::      C(3),FLOW,TMP,PRS,AMW,DENS
  END TYPE STREAMS

! ===== Component physical properties and coefficients
  TYPE PROPS
    REAL ::      TC,PC,VC,XMW,STOIK,CP,AVIS,BVIS,CVIS,DVIS,CLJ,w
    REAL ::      ACP,BCP,CCP,DCP,gdens,ldens
    CHARACTER(LEN=10) :: NAME
  END TYPE PROPS

! ===== Kinetic reaction information
  TYPE KINET
    REAL ::      ko,EoverR,ratef,hrxn,ks,ka,kc,Keq,Eaks,Eaka,Eakc
    character(len=10)::name
  end type kinet

! contains
end module data1
```

C.11 Common blocks

```
! ===== Declaration of global variables =====  
COMMON /prps/ comp(3),rxn(1)  
COMMON /CNST1/ zlen,diam,cao,cbo,cco,cto,temp,pres,vzo,velz,zout,velr,area,xPin,Tin  
common /cnst2/ sumstk,xcon,dens,tiny,reft,refdHrxn,twall,uh,dpart,epsilon  
COMMON /DIFFS/ DFH,DFV,PEZ,PER,dx,dy  
COMMON /rxns1/ ratef(3),sin1(12),c(3,61),xP(61,61),dHrxn,xktgrm  
common /bals/ xmbal(61,61),Hbal(61,61),vzmt(61,61),ctotal(61,61),reyn(61,61)  
common /phys1/ cp(4,61,61),xfrc(4,61,61),gvis(4,61,61),ttmp(61,61),xdens(61,61)  
common /phys2/ amw(61,61),den(61,61),tmp(61),fa(4,61,61),rl1(61),xdhrxn(61,61)  
common /ints/ nnpde,nccc,nx,ny,kode,ncc,nxstp,nystp,slip,ntemp,kkk  
COMMON /LOGI/ NRUN,nrxn,nphase
```

APPENDIX D

Excel™ Visual Basic Interface Routines

' Declaration of Fortran DLL. Required for calculations. Tells VB how to pass the variables, 'and the location of the DLL in reference to the current excel file.

Declare Sub DLLfort Lib "fort\Debug\fort.dll" (feed As Single, dimn As Single, rctr As Single, kin As Single, cmp As Single, res As Single, cc As Single, pec As Single, tmp1 As Single, mfxrn As Integer, zstpin As Single, eps1 As Single)

' Declaration of public variables and arrays

```
Public kin(1 To 11) As Single      ' Array for kinetics values
Public eps1 As Single             ' User defined error tolerance
Public cmp(1 To 4, 1 To 25) As Single ' Component properties array
Public dimn(1 To 10) As Single    ' Array of integers for dimensioning
Public feed(1 To 8) As Single     ' Array for feed data and components
Public rctr(1 To 6) As Single     ' Array of reactor properties
Public pec(1 To 5) As Single      ' Array of Peclet numbers
Public c(1 To 8, 1 To 100, 1 To 100) As Single ' Storage array for concentrations
Public ste(1 To 20) As Single     ' Dummy array
Public ntubes As Single          ' Number of reactor tubes
Public ntype As Single           ' Reactor type
Public cao As Single             ' Component A concentration
Public cbo As Single             ' Component B concentration
Public cco As Single             ' Component C concentration
Public thermtype As Single       ' Determines if Isothermal or not
Public rtype As Single           ' Reactor type storage variable
Public nrtype As Single          ' Reactor type storage variable
Public nthermtype As Single      ' Thermal type storage variable
Public mfxrn As Integer          ' Selects ethane or MTBE reaction
Public nslip As Single           ' Determines flow character at reactor wall
' ===== Below are integer counters for reactor position and step number
Public nrstp As Integer, nzstp As Integer, nz As Integer, nr As Integer
Public tmp1(1 To 100, 1 To 100) As Single ' Storage array for temperatures

Public res As Single             ' Dummy variable
Public n As Single               ' Dummy variable
Public dm As Single              ' Dummy variable
```

```
' =====
' ===== This routine executes and calls the DLL =====
' ===== This is run when the update button is pressed =====
' ===== on the main screen of this excel file =====
' =====
```

```
Sub Button4_Click()
```

```
' Button4_Click Macro
' Macro recorded 12/22/98 by Rob Wade
```

```
' ===== Declare worksheet and subroutine variables
```

```
Dim wb As New Worksheet
Dim cc(1 To 8, 1 To 100, 1 To 100) As Single
' Dim aiTmp() As Integer
' Dim iTmp As Integer
' Dim iArraySize As Integer
```

```

' Dim iterations As Integer
  Dim slipval As Boolean

' We change current drive/directory so Excel can find DLLs (in case user started this
' file from File Manager or Explorer.
  ChDrive ThisWorkbook.Path
  ChDir ThisWorkbook.Path
  nslip = 0#

' Set excel screen refreshing to OFF for calculation speed reasons
  Application.ScreenUpdating = False
  Application.DisplayAlerts = False
' Put indication in status bar that program is running
  Application.StatusBar = "Running Fortran routines..."

' Put secondary indication on main page of calculation status
  Range("b16").Select
  With Selection.Font
    .Name = "Arial"
    .FontStyle = "BOLD"
    .ColorIndex = 3
  End With
  ActiveCell.Value = "Working"

'==== Initialize dummy concentration array prior to calculation
  For i = 1 To 8
    For j = 1 To 100
      For k = 1 To 100
        cc(i, j, k) = 0#
      Next k
    Next j
  Next i

'=====  

' Program selects worksheet called main and collects data for DLL call  

'=====  

  With ThisWorkbook.Sheets("Main")
    If .Range("slip").Value = True Then
      nslip = 1#
    End If

'=====  

' ===== get reactor type data from and set switches for DLL
    nrtype = .Range("rtype").Value
    nthermtype = .Range("thermtype").Value
    If nrtype = 1 Then npde = 2 'pfr
    If nrtype = 2 Then npde = 4 'axial
    If nrtype = 3 Then npde = 2 'radial
    If nrtype = 4 Then npde = 4 'axial & radial

    If nthermtype = 1 Then ntype = 0
    If nthermtype = 2 Then ntype = 1

    If nrtype = 1 And nthermtype = 1 Then name1 = 1
    If nrtype = 1 And nthermtype = 2 Then name1 = 2

```

```

If nrtype = 2 And nthermtype = 1 Then name1 = 3
If nrtype = 2 And nthermtype = 2 Then name1 = 4
If nrtype = 3 And nthermtype = 1 Then name1 = 5
If nrtype = 3 And nthermtype = 2 Then name1 = 6
If nrtype = 4 And nthermtype = 1 Then name1 = 7
If nrtype = 4 And nthermtype = 2 Then name1 = 8

! ===== Fill feed data array =====
feed(1) = .Range("cao").Value
feed(2) = .Range("cbo").Value
feed(3) = .Range("cco").Value
feed(4) = .Range("temp").Value
feed(5) = .Range("pres").Value
feed(6) = .Range("vzo").Value
feed(7) = .Range("twall").Value
feed(8) = .Range("gasR").Value

zstpinit = .Range("zstpinit").Value 'set initial step size
rctr(1) = .Range("zlen").Value 'length
rctr(2) = .Range("diam").Value 'diameter
rctr(3) = .Range("ntubes").Value '#tubes

pec(1) = .Range("pez").Value ' Axial Peclet
pec(2) = .Range("per").Value ' Radial Peclet

' ===== Set number of output steps, reaction , and mass balance check
nzstp = .Range("nzstp").Value
nrstp = .Range("nrstp").Value
mfxrn = .Range("mfxrn").Value
mbal = .Range("mbal1").Value

' ===== If ethane crack reaction, collect data and fill kinetic array
If mfxrn = 1 Then
kin(1) = .Range("ko").Value
kin(2) = .Range("Ea").Value
kin(3) = .Range("gasR").Value
kin(4) = .Range("hrxn").Value
kin(5) = .Range("refT").Value
kin(10) = 1#
End If

' ===== If MTBE crack reaction, collect data and fill kinetic array
'MTkeq = MTBEkeq(feed(4))
mtkeq = .Range("mkeq").Value
'.Range("mKeq").Value = MTkeq
If mfxrn = 2 Then
kin(1) = .Range("mA1").Value
kin(2) = .Range("mB1").Value
kin(3) = .Range("mC1").Value
kin(4) = mtkeq
kin(5) = .Range("mEa1").Value
kin(6) = .Range("mEa2").Value
kin(7) = .Range("mEa3").Value
kin(8) = .Range("refT").Value
kin(9) = 2#
kin(10) = .Range("hrxn").Value

```

```

kin(11) = .Range("catgrams").Value
End If

' ===== Check to determine if components & reaction is in mass balance
' ===== if not then abort function call
    If mbal > 0.00001 Or mbal < -0.00001 Then
        MsgBox "Initial mass balance invalid.... check component MW or stoich"
    End
End If

' ===== Determine number of axial and radial loops
    nz = nzstp / 11
    nr = nrstp / 11

' ===== Fill initial temperature array
    For z = 1 To nzstp
        For r = 1 To nrstp
            tmp1(r, z) = feed(4)
        Next r
    Next z

' ===== Fill integer array for DLL dimensioning
    dimn(1) = 6           'number of values in feed
    dimn(2) = npde       'number of pdes
    dimn(3) = 3         'number of components
    dimn(4) = nzstp      'number of x steps
    dimn(5) = nrstp      'number of y steps
    dimn(6) = nrtype     'type of reactor 1=pfr,2=axial
    dimn(7) = ntype      'isothermal/nonisothermal
    dimn(8) = nslip      ' determine if using slip at wall

    res = 0#

' ===== Set and fill data for component arrays from main worksheet
    cmp(1, 1) = .Range("mw1").Value
    cmp(1, 2) = .Range("st1").Value
    cmp(1, 3) = .Range("mw1").Value
    cmp(1, 4) = .Range("Tc1").Value
    cmp(1, 5) = .Range("Pc1").Value
    cmp(1, 6) = .Range("Vc1").Value
    cmp(1, 8) = .Range("cpa1").Value
    cmp(1, 9) = .Range("cpb1").Value
    cmp(1, 10) = .Range("cpc1").Value
    cmp(1, 11) = .Range("cpd1").Value
    cmp(1, 12) = .Range("agvis1").Value
    cmp(1, 13) = .Range("bgvis1").Value
    cmp(1, 14) = .Range("cgvis1").Value
    cmp(1, 15) = .Range("dgvis1").Value

    cmp(2, 1) = .Range("mw2").Value
    cmp(2, 2) = .Range("st2").Value
    cmp(2, 3) = .Range("mw2").Value
    cmp(2, 4) = .Range("Tc2").Value
    cmp(2, 5) = .Range("Pc2").Value
    cmp(2, 6) = .Range("Vc2").Value
    cmp(2, 8) = .Range("cpa2").Value

```

```

cmp(2, 9) = .Range("cpb2").Value
cmp(2, 10) = .Range("cpc2").Value
cmp(2, 11) = .Range("cpd2").Value
cmp(2, 12) = .Range("agvis2").Value
cmp(2, 13) = .Range("bgvis2").Value
cmp(2, 14) = .Range("cgvis2").Value
cmp(2, 15) = .Range("dgvis2").Value

cmp(3, 1) = .Range("mw3").Value
cmp(3, 2) = .Range("st3").Value
cmp(3, 3) = .Range("mw3").Value
cmp(3, 4) = .Range("Tc3").Value
cmp(3, 5) = .Range("Pc3").Value
cmp(3, 6) = .Range("Vc3").Value
cmp(3, 8) = .Range("cpa3").Value
cmp(3, 9) = .Range("cpb3").Value
cmp(3, 10) = .Range("cpc3").Value
cmp(3, 11) = .Range("cpd3").Value
cmp(3, 12) = .Range("agvis3").Value
cmp(3, 13) = .Range("bgvis3").Value
cmp(3, 14) = .Range("cgvis3").Value
cmp(3, 15) = .Range("dgvis3").Value
eps1 = .Range("eps").Value
End With

' ===== Here is the DLL call for the reactor =====
Call DLLfort(feed(1), dimn(1), rctr(1), kin(1), cmp(1, 1), res, cc(1, 1, 1), pec(1), tmp1(1, 1),
mfxrn, zstpin, eps1)
' =====

' ===== Select temporary worksheet TPFL (used for storage and plotting)
With ThisWorkbook.Sheets("Tpfl")
For i = 1 To 3
For j = 1 To nzstp
For k = 1 To nrstp
' ===== place concentrations in array and fill spreadsheet
c(i, j, k) = cc(i, j, k)
Next k
Next j
Next i

' ===== Set excel calculations and refresh parameters
Application.MaxChange = 0.001
ActiveWorkbook.PrecisionAsDisplayed = False
' ActiveSheet.Calculate

' ===== Once data is placed into tpfl copy to required worksheet (base on reactor type)
Sheets("Tpfl").Select

' ===== Fill component 1 concentrations
nnz = 3
nnr = 20
For z = 1 To nrstp Step nr
For r = 1 To nzstp Step nz

```



```

        nnr = nnr + 1
        Cells(nnr, nnz).Value = c(1, r, z)
    Next r
    nnz = nnz + 1
    nnr = 20
Next z

' === Fill component 2 concentrations
nnz = 3
nnr = 36
For z = 1 To nrstp Step nr
    For r = 1 To nzstp Step nz
        nnr = nnr + 1
        Cells(nnr, nnz).Value = c(2, r, z)
    Next r
    nnz = nnz + 1
    nnr = 36
Next z

' === Fill component 3 concentrations
nnz = 3
nnr = 52
For z = 1 To nrstp Step nr
    For r = 1 To nzstp Step nz
        nnr = nnr + 1
        Cells(nnr, nnz).Value = c(3, r, z)
    Next r
    nnz = nnz + 1
    nnr = 52
Next z

' === Fill temperatures into worksheet
nnz = 3
nnr = 67
For z = 1 To nrstp Step nr
    For r = 1 To nzstp Step nz
        nnr = nnr + 1
        Cells(nnr, nnz).Value = tmp1(r, z)
    Next r
    nnz = nnz + 1
    nnr = 67
Next z

' ===== Set worksheet headings and exit
Range("b2").Select
Select Case name1
Case 1
    ActiveCell.Value = "Isothermal PFR"
    pag = 1
Case 2
    ActiveCell.Value = "Adiabatic PFR"
    pag = 1
Case 3
    ActiveCell.Value = "Isothermal Ax"
    pag = 2

```

```

Case 4
    ActiveCell.Value = "Adiabatic Ax"
    pag = 2
Case 5
    ActiveCell.Value = "Isothermal Rad"
    pag = 3
Case 6
    ActiveCell.Value = "Adiabatic Rad"
    pag = 3
Case 7
    ActiveCell.Value = "Isothermal Ax/Rad"
    pag = 4
Case 8
    ActiveCell.Value = "Adiabatic Ax/Rad"
    pag = 4
Case Else
    ActiveCell.Value = "Invalid Section"
End Select
Range("m2").Select
    ActiveCell.Value = Format(Now, "yy/mm/dd")
Range("n2").Select
    ActiveCell.Value = Format(Now, "hh:mm:ss")
Sheets("main").Select
Range("b16").Select
    With Selection.Font
        .Name = "Arial"
        .FontStyle = "BOLD"
        .ColorIndex = 4
    End With
ActiveCell.Value = "Done"
Sheets("tpfl").Select
    Application.MaxChange = 0.001
ActiveWorkbook.PrecisionAsDisplayed = False
ActiveSheet.Calculate
Select Case pag
    Case 1
        Sheets("PFRout").Select
        Range("A1:O81").Select
        Selection.ClearContents
        Selection.Borders(xlDiagonalDown).LineStyle = xlNone
        Selection.Borders(xlDiagonalUp).LineStyle = xlNone
        Selection.Borders(xlEdgeLeft).LineStyle = xlNone
        Selection.Borders(xlEdgeTop).LineStyle = xlNone
        Selection.Borders(xlEdgeBottom).LineStyle = xlNone
        Selection.Borders(xlEdgeRight).LineStyle = xlNone
        Selection.Borders(xlInsideVertical).LineStyle = xlNone
        Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
        Selection.Interior.ColorIndex = 2
        Range("A1").Select
        Sheets("TPFL").Select
        Range("A1:N17").Select
        Selection.Copy
        Sheets("PFRout").Select
        'ActiveSheet.DropDowns.Add(617.4, 62.4, 52.8, 15).Select
        'ActiveSheet.CheckBoxes.Add(335.4, 172.2, 77.4, 14.4).Select
        ActiveSheet.Paste

```

```

Sheets("TPfl").Select
Range("A18:O81").Select
Application.CutCopyMode = False
Selection.Copy
Sheets("PFRout").Select
Range("A18").Select
ActiveSheet.Paste
Sheets("tpfl").Select
Range("c13:c14").Select
Selection.Copy
Sheets("pfrou").Select
Range("c13").Select
Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
    False, Transpose:=False
Range("a1").Select

```

Case 2

```

Sheets("AxialOut").Select
Range("A1:O81").Select
Selection.ClearContents
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
Selection.Borders(xlEdgeLeft).LineStyle = xlNone
Selection.Borders(xlEdgeTop).LineStyle = xlNone
Selection.Borders(xlEdgeBottom).LineStyle = xlNone
Selection.Borders(xlEdgeRight).LineStyle = xlNone
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
Selection.Interior.ColorIndex = 2
Range("A1").Select
Sheets("TPFL").Select
Range("A1:N17").Select
Selection.Copy
Sheets("Axialout").Select
'ActiveSheet.DropDowns.Add(617.4, 62.4, 52.8, 15).Select
'ActiveSheet.CheckBoxes.Add(335.4, 172.2, 77.4, 14.4).Select
ActiveSheet.Paste
Sheets("TPfl").Select
Range("A18:O81").Select
Application.CutCopyMode = False
Selection.Copy
Sheets("Axialout").Select
Range("A18").Select
ActiveSheet.Paste

```

```

Sheets("tpfl").Select
Range("c13:c14").Select
Selection.Copy
Sheets("Axialout").Select
Range("c13").Select
Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
    False, Transpose:=False
Range("a1").Select

```

Case 3

```

Sheets("RadOut").Select
Range("A1:O81").Select

```

```

Selection.ClearContents
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
Selection.Borders(xlEdgeLeft).LineStyle = xlNone
Selection.Borders(xlEdgeTop).LineStyle = xlNone
Selection.Borders(xlEdgeBottom).LineStyle = xlNone
Selection.Borders(xlEdgeRight).LineStyle = xlNone
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
Selection.Interior.ColorIndex = 2
Range("A1").Select
Sheets("TPFL").Select
Range("A1:N17").Select
Selection.Copy
Sheets("Radout").Select
'ActiveSheet.DropDowns.Add(617.4, 62.4, 52.8, 15).Select
'ActiveSheet.CheckBoxes.Add(335.4, 172.2, 77.4, 14.4).Select
ActiveSheet.Paste
Sheets("TPfl").Select
Range("A18:O81").Select
Application.CutCopyMode = False
Selection.Copy
Sheets("Radout").Select
Range("A18").Select
ActiveSheet.Paste

Sheets("tpfl").Select
Range("c13:c14").Select
Selection.Copy
Sheets("Radout").Select
Range("c13").Select
Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
    False, Transpose:=False
Range("a1").Select
Case 4
Sheets("AxRadOut").Select
Range("A1:O81").Select
Selection.ClearContents
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
Selection.Borders(xlEdgeLeft).LineStyle = xlNone
Selection.Borders(xlEdgeTop).LineStyle = xlNone
Selection.Borders(xlEdgeBottom).LineStyle = xlNone
Selection.Borders(xlEdgeRight).LineStyle = xlNone
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
Selection.Interior.ColorIndex = 2
Range("A1").Select
Sheets("TPfl").Select
Range("A1:N17").Select
Selection.Copy
Sheets("AxRadout").Select
'ActiveSheet.DropDowns.Add(617.4, 62.4, 52.8, 15).Select
'ActiveSheet.CheckBoxes.Add(335.4, 172.2, 77.4, 14.4).Select
ActiveSheet.Paste
Sheets("TPfl").Select

```

```
Range("A18:O81").Select
Application.CutCopyMode = False
Selection.Copy
Sheets("AxRadout").Select
Range("A18").Select
ActiveSheet.Paste
Sheets("tpfl").Select
Range("c13:c14").Select
Selection.Copy
Sheets("AxRadout").Select
Range("c13").Select
Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
    False, Transpose:=False
Range("a1").Select
End Select
End With

Application.DisplayAlerts = False
Application.ScreenUpdating = True
Application.StatusBar = False
Sheets("main").Select
End Sub
```

VITA

Robert El Wade

Candidate for the Degree of

Master of Science

Thesis: FIXED BED REACTOR MODEL AND KINETICS EVALUATION

Major Field: Chemical Engineering

Biographical:

Personal Data: Born in Great Falls, Montana on December 14, 1969, the son of Donald and Sharon Wade.

Education: Graduated from Hobson Public School, Hobson Montana in May of 1988.

Received Bachelors of Science in Chemical Engineering at Montana State University, Bozeman, Montana in 1992.

Completed the requirements for Masters of Science degree in Chemical Engineering at Oklahoma State University in July 1999.

Experience: Worked for Texaco Lubricants Company in Denver, Colorado from 1993 to 1995.

Provided sales support, engineering design and technical instruction as a representative of Loctite Corporation from 1995 to 1997.

Scheduled to start work for Celanese Research in Corpus Christi, TX in early June 1999.