

Numerical Simulation of Pressure Swing Adsorption Process

by

Lin Lin

B.Sc. XIDIAN University, Xi'an, China, 1990

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in the
School of Engineering Science

© Lin Lin 1997

SIMON FRASER UNIVERSITY

August 1997

All rights reserved. This work may not be reproduced
in whole or in part, by photocopy or other means,
without the permission of the author.



National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-24184-X

Approval Page

Title of Thesis: Numerical Simulation of Pressure Swing Adsorption Process

Name: Lin Lin

Degree: Master of Applied Science

Examining Committee: Dr. M. Parameswaran
Chair

Dr. John D. Jones
Senior Supervisor

Dr. Bowie Keefer
Supervisor

Dr. Bob Russell
Supervisor

Dr. William Gruver
External Examiner

Date Approved:

28 August 1997

Abstract

The object of this thesis project is to build a realistic model of a PSA cycle, in which velocity of flow varies due to significant adsorption; also heat effect needs to be considered due to adiabatic thermal condition. Such a realistic model is desired for design of a PSA unit and a PSA cycle. However, the mathematical model used to describe such a realistic model would be very complex, and numerical simulation needs to be handled carefully, since it will result in a self-sharpening concentration breakthrough curve. Such a mathematical model would require the numerical method to have high standard of performance on numerical diffusion (free from numerical diffusion) in order to accurately predict the productivity or purity of product. In this thesis, we examine two numerical methods ---- the numerical method of lines (finite difference formulation for space difference) and the orthogonal collocation method ---- by studying various mathematical problems and adsorption systems. We have shown that these two numerical methods with higher order formulations embedded provide the solution to numerical diffusion; and the orthogonal collocation methods is superior to the numerical method of lines in computation time. In the last two chapters, we take a step by step procedure to solve our realistic model by double collocation method, which is the extension of the orthogonal collocation method. The results have shown that the theoretical breakthrough curve of concentration is self-sharpening, and the temperature of the adsorption column as well as the velocity of flow are affected by the significant adsorption.

Dedication

I'd like to dedicate this work

to my dearest friend: YIJun

and many thanks for her generous support to me.

Acknowledgments

I'm very thankful to my senior supervisor Dr. John Jones for his guidance during the work on this thesis project and for reviewing of the thesis.

I would like also to thank Highquest Engineering Inc. for providing me this research opportunity and partial financial support to this work.

I'm also thankful to supervisor Dr. Bob Russel for reviewing this thesis.

And I also like to give thanks to my colleague Kulbir Singh for many helps.

Table of Content

Approval

Abstract

Dedication

Acknowledgments

Chapter 1 Introduction 1

1-1 Pressure Swing Adsorption Process 1

1-1.1 Adsorption Separation Process 1

1-1.2 Pressure Swing Adsorption Process 1

1-2 Modelling of PSA Cycles 2

1-3 Adsorption Column Dynamics and Breakthrough Curve 3

1-3.1 Equilibrium Theory 3

1-3.2 Bulk Separations 4

1-4 Numerical Diffusion and Breakthrough Curve 5

1-4.1 Numerical Diffusion 5

1-4.2 Method Free from Numerical Diffusion 6

Chapter 2 A Survey on Numerical Diffusion 7

2-1 Introduction 7

2-1.1 Partial Differential Equation- A Model Problem 7

2-1.2 Finite Difference Method 7

2-1.3 Control Volume Approach 9

2-1.4 Numerical Diffusion: 10

2-2 Numerical Analysis of the Existing Finite Difference Schemes: 11

2-2.1 Quantitative Evaluation of Numerical Diffusion 11

2-2.2 Transient and Steady-State Analysis On Artificial Viscosity 13

2-2.3 The Upwind Difference Schemes 15

2-2.4 A Consistency Check for Estimating Truncation Error due to Upstream Differencing 16

2-2.5 Comments on Section 2-2 19

2-3 Advanced Study on Numerical Diffusion and Improvement of Upwind Difference Methods 19

2-3.1 A Survey of Finite Differences with Upwinding for Numerical Modelling of the Incompressible Convective Diffusion Equation 19

2-3.2 Beyond First-Order Upwinding: The ULTRA-SHARP Alternative for Non-

oscillatory Steady-State Simulation of Convection	24
2-3.3 A Method for Predicting and Minimizing Numerical Diffusion	27
2-3.4 Comments on Section 2-3	29
2-4 Explicit Artificial Viscosity	29
2-4.1 An Artificial Dissipation Scheme for the Navier-Stokes Equations	29
2-5 Conclusion for Chapter 2	31
Chapter 3 Convection Equation	32
3-1 Convection Equation	32
3-2 Solving Convection Equation with the Numerical Method of Lines	33
3-2.1 A Brief Introduction to the Numerical Method of Lines	33
3-2.2 Spatial differentiation	34
3-2.3 ODE integration	35
3-2.4 The Computation Results	37
3-2.5 Comments	39
3-3 Solving Convection Equation With the Orthogonal Collocation Method	39
3-3.1 A Brief Introduction to the Orthogonal Collocation Method	39
3-3.2 Solving Convection Equation with the Orthogonal Collocation Method	42
3-3.3 The Computation Results	43
3-3.4 Comments:	43
3-4 Comparison between NUMOL and Orthogonal Collocation	44
3-4.1 observation of performance on numerical diffusion and oscillation	44
3-4.2 Comparison of Computation Time	45
3-4.3 Conclusion for Chapter 3	45
Chapter 4 Convective Diffusion Equation	46
4-1 Convective Diffusion Equation	46
4-2 The NUMOL Solution of Convective Diffusion Equation	46
4-2.1 Space Discretization	46
4-2.2 ODE Integration	47
4-2.3 The Computation Result	47
4-2.4 Comments	49
4-3 Solving Convective Diffusion Equation with the Orthogonal Collocation Method.	49
4-3.1 Collocation Form	49
4-3.2 The Computation Result	50
4-3.3 Comments	51
4-4 Comparison of the Results and Conclusion	52
Chapter 5 Simulation of a Fixed-Bed Adsorption Column	53
5-1 Description of the Fixed-Bed Adsorption column	53

5-2 Mathematical Model: 53

5-2.1 Dimensional Form: 53

5-2.2 Dimensionless Form: 54

5-2.3 Space Discretization of the Computational Domain 55

5-3 Simulation By the Numerical Method of Lines(NUMOL) 56

5-3.1 Space differentiation: 56

5-3.2 The result 58

5-4 Numerical Simulation by the Orthogonal Collocation Method 59

5-4.1 Collocation Form 59

5-4.2 Simulation Result 61

5-5 Experiments and Comparison of Results 62

5-5.1 Comparison between Numerical and Analytical results 62

5-5.2 Effects of Axial Dispersion and External Mass Transfer Resistance 63

5-5.3 Effects of Distribution Ratio 63

5-5.4 The Computation Time 64

5-6 Conclusion for Chapter 5 65

Chapter 6 PSA Simulation by Double Collocation Method 66

6-1 Introduction 66

6-2 A System of PSA Process Description 66

6-3 Mathematical Model 67

6-3.1 Step 1: High-Pressure Flow in Bed 2 and Low-Pressure Flow in Bed 1 67

6-3.2 Step 2: Blowdown of Bed 2 and Pressurization of Bed 1 71

6-3.3 Step 3 72

6-3.4 Step 4 72

6-3.5 Cyclic Steady-State: 73

6-3.6 Simulation: 73

6-4 Double Collocation method: 73

6-4.1 Preparation: 74

6-4.2 Computation for one time step by double collocation: 75

6-5 The Computation Results and Discussion 78

6-5.1 The sensitivity of the assumptions for repressurization to the number of collocation points 78

6-5.2 Observation of Mass balance 79

6-5.3 Output of the Simulation by Polynomials 80

6-5.4 Result of Steady-state PSA operation 82

6-5.5 Computation Time. 82

6-6 Conclusion: 83

Chapter 7 Simulation of A Conventional PSA Process 84

- 7-1 Theoretical Model 84**
- 7-2 Mathematical Model 84**
 - 7-2.1 Dimensional Form 84
 - 7-2.2 Dimensionless Form 86
- 7-3 Implementation of double collocation 89**
 - 7-3.1 Step 1 89
 - 7-3.2 Step 2 89
 - 7-3.3 Step 3 90
 - 7-3.4 Step 4 90
- 7-4 The Simulation Result: 91**
- 7-5 The Computation Time 95**
- 7-6 Conclusion for Chapter 7 95**
- 7-7 List of Parameters 96**

Chapter 8 General Conclusion and Future Work 98

- 8-1 The Present Study Has the Following Findings: 98**
- 8-2 Recommendation for Future Work 99**

Appendix: The Orthogonal Collocation Method and Orthogonal Polynomials 100

A-1 A Brief Introduction to The Orthogonal Collocation Method 100

- A-1.1 Trial Function & Orthogonal Polynomials 100
- A-1.2 Interior formulas based on ordinates 101

A-2 Orthogonal polynomials 104

- A-2.1 Jacobi Polynomials 104
- A-2.2 Shifted Legendre Polynomials 105

A-3 Solution in the form of polynomials 106

- A-3.1 Solution in the form of Jacobi polynomials 106
- A-3.2 Solution in the form of shifted Legendre Polynomials 106

A-4 Table of Polynomials and Their Roots 107

A-5 Program List 109

- A-5.1 The program for symbolic computation of the Jacobi polynomials in Maple V: 109
- A-5.2 The program for symbolic computation of the shifted Legendre polynomials in Maple V: 110

A-6 Reference: 110

Bibliography 111

Chapter 1 Introduction

- Explain what PSA is, and why we want to model it.
- Explain what a breakthrough curve is, why it is self-sharpening, and why this means that we need a model free from numerical diffusion.

1 - 1 Pressure Swing Adsorption Process

Pressure swing adsorption (PSA) is a widely used process for separation of gases. The major applications of PSA include the recovery and purification of hydrogen, the separation of oxygen from air, the separation of normal and isoparaffins, and a variety of drying operations. The major commercial PSA separation processes have been reviewed by Cassidy and Holmes, (1984).

1 - 1.1 Adsorption Separation Process

The essential requirement of all adsorption separation processes is an adsorbent which preferentially adsorbs one component (or one family of related components) from a mixed feed. The selectivity of the adsorbent may depend on a difference in adsorption equilibrium or on a difference in sorption rates (kinetic selectivity). (Ruthven 1994)

All adsorption separation processes involve two principal steps: *adsorption*, during which the preferentially adsorbed species are picked up from the feed; *regeneration* or *desorption*, during which the adsorbed species are removed from the adsorbent, thus “regenerating” the adsorbent for use in the next cycle.

The adsorption can be affected by changing either the pressure or the temperature, i.e., the degree of adsorption increases with pressure and decreases with temperature.

1 - 1.2 Pressure Swing Adsorption Process

Pressure swing adsorption processes are cyclic processes for separation of gaseous mixtures in which the adsorbent is regenerated by reducing the partial pressure of the adsorbed component. This partial pressure reduction can be accomplished rather rapidly by lowering the total pressure or using a purge gas. (Farooq 1990)

In the following diagram, we illustrate the two principal steps of a PSA cycle.

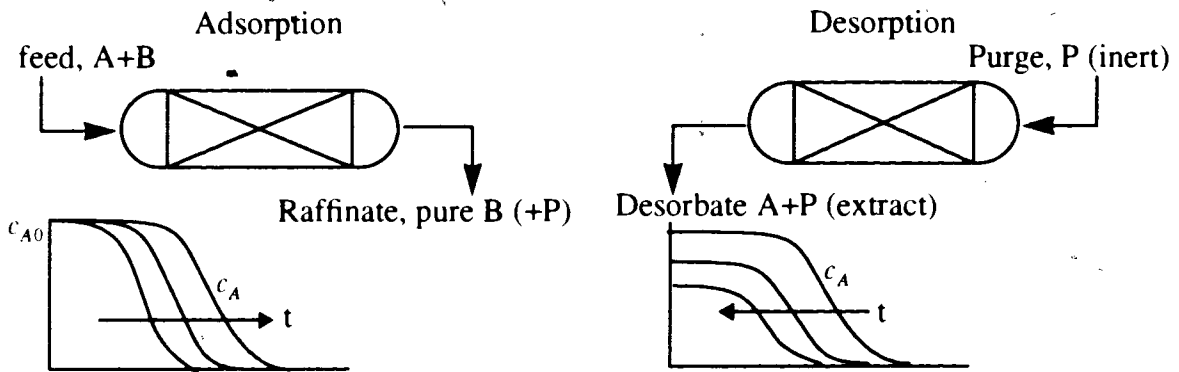


Figure 1 - 1.2.1 The Concept of a PSA process

The adsorption column (bed) is packed with adsorbent. During the adsorption (feed) step, the feed gas is entering the bed from inlet, and as the bed is pressurized with feed gas, the product gas A is collected from the outlet. During the desorption step, the bed blows down to lower pressure, with or without purge gas. The feed step is normally terminated before the more strongly adsorbed component breaks through the bed, while the regeneration step is generally terminated before the bed is fully desorbed. At cyclic steady state the profile therefore oscillates about a mean position in the bed.

Thus the pressure of adsorbent bed is changed during each separation cycle. The advantage of changing the pressure instead of temperature is that pressure can be changed much more rapidly than the temperature, thus making it possible to operate a PSA process on a much faster cycle, thereby increasing the throughput per unit of adsorbent bed volume.

The earliest development of PSA technology can be traced back to Finlayson, Sharp, Hasche, Dargan and Perley in their pioneering patents. However, Skarstrom (1960, 1972) first introduced the low-pressure purge step to clean the adsorbent bed following the blowdown step in his patent. Many modified versions of Skarstrom PSA cycles have been developed for particular uses since then.

1 - 2 Modelling of PSA Cycles

Efficient performance of a PSA unit depends on achieving the correct combination of process variables such as bed length, flow rate, cycle time, pressure ratio, and purge to feed ratio. The interaction of these process variables and kinetic / equilibrium parameters is so complicated that it is difficult to arrive at an optimal design simply by intuition and empiricism; a reliable mathematical simulation of the system is therefore required. Partial differential equations are used to describe the dynamic behavior of the PSA system at each step of a PSA cycle. Basic modelling approaches include equilibrium models, dynamic LDF models and pore diffusion models. Most research on PSA modelling and analysis has been done based on the assumption of

isothermal behavior.

The objective of my thesis project is to build realistic computer models of PSA processes, to support the design and optimization of PSA equipment by our industrial partner. Currently, our industrial partner has developed various PSA cycles and laboratory scale apparatus. However, when these benchtop prototype apparatus are scaled up to industrial-scale apparatus, the thermal boundary conditions change from near-isothermal to near-adiabatic. Because of the temperature sensitivity of the adsorption process, this can lead to performance well below the potential optimum. Therefore a reliable computer model is desired to realistically represent the critical characteristics of the PSA process, including heat transfer, varying pressure, and varying superficial velocity.

1 - 3 Adsorption Column Dynamics and Breakthrough Curve

For modelling a PSA process, we need to understand the dynamic behavior of a packed adsorbent bed, which depends on the interplay between adsorption kinetics, adsorption equilibrium, and fluid dynamics. The overall pattern of the dynamic behavior is generally determined by the form of the equilibrium relationship, and may be strongly modified by kinetic effects (finite resistance to mass transfer). It is useful to consider the analysis of the dynamics of an ideal system with infinitely rapid mass transfer (equilibrium theory). This dynamic behavior can be represented by the breakthrough curve. The emphasis is on the introduction of a breakthrough curve. A reference on how, in a real system, the ideal patterns of behavior are modified by the intrusion of finite resistance to mass transfer can be found in Ruthven (1994).

1 - 3.1 Equilibrium Theory

The formal analysis of adsorption column dynamics starts from the basic differential equation derived from a transient mass balance on an element of the column. If the flow pattern is represented by the axially dispersed plug flow model, this assumes the form:

$$-D_L \frac{\partial^2 c}{\partial z^2} + \frac{\partial}{\partial z} (vc) + \frac{\partial c}{\partial t} + \left(\frac{1-\epsilon}{\epsilon} \right) \frac{\partial \bar{q}}{\partial t} = 0 \quad (\text{Eq 1 - 3.1.1})$$

If axial dispersion and pressure drop through the column can be neglected and if the concentration of the adsorbable species is small, this expression reduces to:

$$v \frac{\partial c}{\partial z} + \frac{\partial c}{\partial t} + \left(\frac{1-\epsilon}{\epsilon} \right) \frac{\partial \bar{q}}{\partial t} = 0 \quad (\text{Eq 1 - 3.1.2})$$

In the absence of mass transfer resistance local equilibrium prevails at all points (i.e., $\bar{q} = q^*$ and if the system is isothermal, $q^* = f(c)$, where $f(c)$ represents the equilibrium isotherm. Under these conditions (Eq 1 - 3.1.2) becomes:

$$v / \left(1 + \left(\frac{1-\epsilon}{\epsilon} \right) \frac{dq^*}{dc} \right) \frac{\partial c}{\partial z} + \frac{\partial c}{\partial t} = 0 \quad (\text{Eq 1 - 3.1.3})$$

This equation has the form of a traveling wave equation with the wave velocity given by:

$$w_c = v / \left(1 + \left(\frac{1-\epsilon}{\epsilon} \right) \frac{dq^*}{dc} \right) \quad (\text{Eq 1 - 3.1.4})$$

If the equilibrium relationship is linear $q = Kc$,

$$w_c = v / \left(1 + \left(\frac{1-\epsilon}{\epsilon} \right) K \right) \quad (\text{Eq 1 - 3.1.5})$$

and it is evident that the wave velocity is independent of concentration.

For an unfavorable equilibrium relationship dq^*/dc increases with concentration so w decreases with concentration, leading to a profile that spreads as it propagates. Since the profile spreads in a direction proportion to the distance traveled, this referred to as "proportionate pattern" behavior.

The case of a favorable equilibrium isotherm is slightly more complex. dq^*/dc decreases with concentration, so according to (Eq 1 - 3.1.4), w will increase with concentration. This leads to what is commonly referred to as "self-sharpening" behavior. An initially dispersed profile will become less and less dispersed as it propagates, eventually approaching a shock transition.

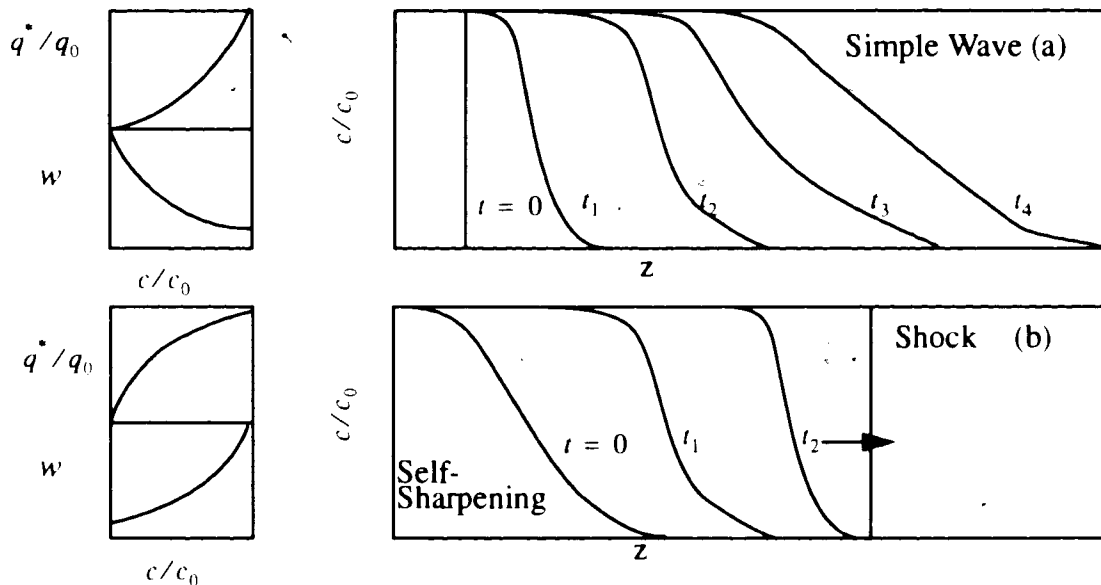


Figure 1 - 3.1.1 Development of the concentration profile in an adsorption column. (a) For an "unfavorable" equilibrium relationship the profile spreads as it propagates, approaching proportionate pattern behavior. (b) For a "favorable" equilibrium relationship an initially dispersed profile is sharpened as it propagates, approaching a shock wave.

1 - 3.2 Bulk Separations

Another situation in which a shock solution is obtained arises in bulk separations, where the

change in flow rate due to adsorption is relatively large.

For a bulk separation we have,

$$v \frac{\partial c}{\partial z} + c \frac{\partial v}{\partial z} + \frac{\partial c}{\partial t} + \left(\frac{1-\epsilon}{\epsilon} \right) \frac{\partial \bar{q}}{\partial t} = 0 \quad (\text{Eq 1 - 3.2.1})$$

where, for an isobaric and isothermal system with an adsorbable component in an inert carrier:

$$\frac{v}{v_0} = \frac{1-y_0}{1-y} \quad (\text{Eq 1 - 3.2.2})$$

Expressed in terms of y , the mole fraction of the adsorbable (or more adsorbable) component, (Eq 1 - 3.2.1) becomes:

$$\left\{ v_0 (1-y_0) / (1-y)^2 \left[1 + \left(\frac{1-\epsilon}{\epsilon} \right) \frac{dq^*}{dc} \right] \right\} \frac{\partial y}{\partial z} + \frac{\partial y}{\partial t} = 0 \quad (\text{Eq 1 - 3.2.3})$$

which evidently represents a traveling wave with the wave velocity given by:

$$\frac{w}{v_0} = \left\{ (1-y_0) / (1-y)^2 \left[1 + \left(\frac{1-\epsilon}{\epsilon} \right) \frac{dq^*}{dc} \right] \right\} \quad (\text{Eq 1 - 3.2.4})$$

For the linear equilibrium system:

$$\frac{w}{v_0} = \left\{ (1-y_0) / (1-y)^2 \left[1 + \left(\frac{1-\epsilon}{\epsilon} \right) K \right] \right\} \quad (\text{Eq 1 - 3.2.5})$$

Clearly w increases with increasing y , just as in the case of a trace system with favorable equilibrium, so that, according to equilibrium theory, there will be a shock transition.

From the above analyses, we can see that in both cases, the self-sharpening breakthrough curves are produced. For the PSA systems we are modelling, the situation is more complicated; however, fundamentally, due to adsorption in bulk separation, the change of flow rate is relatively large. Also, the systems have favorable isotherms; as a result, the concentration profile is the form of a self-sharpening breakthrough curve.

1 - 4 Numerical Diffusion and Breakthrough Curve

The breakthrough curve of our current PSA model is self-sharpening; as a result, the numerical method for modelling PSA processes is required to have quality performance on numerical diffusion.

1 - 4.1 Numerical Diffusion

Simulation of the behavior of an adsorption column bed produces a certain amount of axial diffusion, but this diffusion is entirely an artifact of the modelling process, and it is shown that its extent depends on the relative values of the fluid velocity, the size of the crank angle increment, and the length of a control volume. (Jones 1995)

Consider a square wave moving along the bed without diffusion:

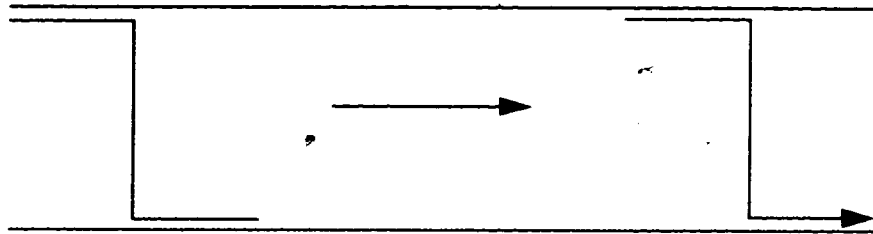


Figure 1 - 4.1.1 Square wave flow

If there is axial diffusion, the square wave will broaden out as it moves along the bed.

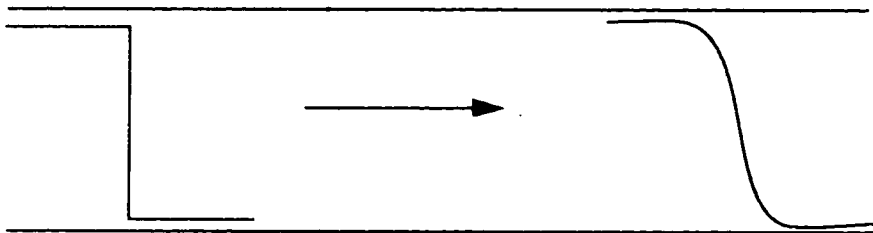


Figure 1 - 4.1.2 square wave flow plus axial dispersion

If this amount of axial diffusion is produced by the numerical method we use, it is called numerical diffusion, or false diffusion, distinct from the physical diffusion. A further study on numerical diffusion will be presented in chapter 2.

1 - 4.2 Method Free from Numerical Diffusion

Axial diffusion has an important effect on bed performance. Gas separation is most effective when the concentration profile in the bed is a square wave. Since it is shown that the concentration profile is self-sharpening and moves along the bed, an almost pure product flows out from the downstream end. Then, just as the shock wave is about to reach the downstream end, we reverse the flow. If there is axial dispersion, however, when the self-sharpening wave transfers to shock wave, it broadens out as it moves along the bed. So as the tail of the wave reaches the downstream end, we either stop collecting the product immediately, in which case we can't obtain a high percentage of the product, or we go on collecting the product for a little while, in which case we get some of the other species mixed in reducing purity. If our model includes an unrealistically high degree of axial dispersion, we will find that it always predicts low purity or low recovery. It is therefore important that our model is free from numerical diffusion.

Therefore, another object of this thesis project is to study the fundamentals of various different numerical methods and their performance on numerical diffusion; in return, we will have the basis to choose among the numerical methods to improve our modelling of PSA processes.

Chapter 2 A Survey on Numerical Diffusion

During the modelling of a PSA process, partial differential equations are used to describe the system. Numerical methods are used to approximate solutions of the partial differential equations. When solving a PDE numerically, for example, by the finite difference method, the numerical diffusion causes a great deal of concern. It is desired to reduce or even eliminate this non-physical false diffusion in the modelling. In this chapter, through studying selected papers on numerical diffusion, past research and achievements in this area will be presented, and the most fundamental questions arising will be answered.

- What is numerical diffusion?
- How is numerical diffusion generated?
- How to reduce or eliminate numerical diffusion?

2 - 1 Introduction

In this section, the preliminary knowledge of partial differential equations and numerical methods such as the finite difference method and control volume method will be introduced.

2 - 1.1 Partial Differential Equation- A Model Problem

Consider a simple transient convective-diffusion equation in one dimension (initial-boundary value problem):

$$\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} - D \frac{\partial^2 \phi}{\partial x^2} = 0, 0 < x < 1, t > 0 \quad (\text{Eq 2 - 1.1.1})$$

subject to the initial condition:

$$\phi(x, 0) = 0, 0 \leq x \leq 1 \quad (\text{Eq 2 - 1.1.2})$$

and boundary conditions:

$$\phi(0, t) = 0 \quad \text{and} \quad \phi(1, t) = 1, t > 0 \quad (\text{Eq 2 - 1.1.3})$$

The physical interpretation of this set of equations is: a fluid moving along a thin tube of length 1 at velocity u . ϕ represents the temperature, or the concentration of a trace component. The temperature is forced to be 0 at inlet and 1 at outlet. Initially the fluid is at a temperature 0. The constant D is called the thermal diffusivity of the fluid. The solution of $\phi(x, t)$ will be the temperature at each position x in the tube at time t .

2 - 1.2 Finite Difference Method

Taylor's theorem:

An elementary approach to the finite difference method is provided by Taylor's theorem, which we state as follows:

Let ϕ be a class of functions that are $n+1$ times continuously differentiable on the interval

$[a, b]$. Then there exists a number $\xi, a < \xi < b$, such that:

$$\phi(b) = \sum_{i=0}^n \frac{\phi^{(i)}(a)}{i!} (b-a)^i + R_N, \text{ where } R_N = \frac{\phi^{(n+1)}(\xi)(b-a)^{n+1}}{(n+1)!} \quad (\text{Eq 2 - 1.2.1})$$

Finite difference approximation to $\frac{d\phi}{dx}(a)$,

1. Forward difference (first-order accuracy): $(\phi(a+h) - \phi(a))/h$
2. Backward difference (first-order accuracy): $(\phi(a) - \phi(a-h))/h$
3. Centered difference (second-order accuracy): $(\phi(a+h) - \phi(a-h))/(2h)$

Finite difference approximation to $\frac{d^2\phi}{dx^2}(a)$,

4. Second centered difference (second order accuracy): $(\phi(a+h) - 2\phi(a) + \phi(a-h))/h^2$.

2 - 1.2.1 Spatial Discretization:

Assume for the moment that ϕ does not depend on time. The result is the steady-state boundary value problem¹

$$\begin{aligned} u\phi_x - D\phi_{xx} &= 0, 0 < x < 1 \\ \phi(0) &= 0, \phi(1) = 1 \end{aligned} \quad (\text{Eq 2 - 1.2.2})$$

The analytical solution to this boundary value problem is

$$\phi(x) = \frac{1 - e^{Rx}}{1 - e^R} \quad (\text{Eq 2 - 1.2.3})$$

where $R = (uL)/D$ is called the Peclet number, it can be seen that R is the ratio of the strengths of convection and diffusion.

If **centered difference** approximation is used for the first and second derivatives in (Eq 2 - 1.2.2), we obtain:

$$u \left(\frac{\phi_{j+1} - \phi_{j-1}}{2\Delta x} \right) - D \left(\frac{\phi_{j-1} - 2\phi_j + \phi_{j+1}}{\Delta x^2} \right) = 0, \quad 1 \leq j \leq N-1 \quad (\text{Eq 2 - 1.2.4})$$

The closed form solution to (Eq 2 - 1.2.4) is $\phi_j = \frac{1 - Z^j}{1 - Z^N}$, $0 \leq j \leq N$, where $Z = \frac{(2 + R\Delta x)}{(2 - R\Delta x)}$. This solution suffers non-physical oscillation if $\Delta x > 2/R$.

To avoid this non-physical oscillation, use **upwind differencing** for the first derivative, and retain centered difference for the second derivative; we obtain

$$\begin{aligned} u \frac{\phi_j - \phi_{j-1}}{\Delta x} - D \frac{\phi_{j-1} - 2\phi_j + \phi_{j+1}}{\Delta x^2} &= 0, u > 0, 1 \leq j \leq N-1 \\ u \frac{\phi_{j+1} - \phi_j}{\Delta x} - D \frac{\phi_{j-1} - 2\phi_j + \phi_{j+1}}{\Delta x^2} &= 0, u < 0, 1 \leq j \leq N-1 \end{aligned} \quad (\text{Eq 2 - 1.2.5})$$

That is, the gradient of ϕ is approximated by "its gradient between j and the mesh-point

1. Here we use subscript notation for partial differentiation, $\phi_x = \frac{\partial \phi}{\partial x}$, and $\phi_{xx} = \frac{\partial^2 \phi}{\partial x^2}$. We will use this notation whenever it is convenient.

upwind of j ".

The solution subject to (Eq 2 - 1.2.5) is $\phi_j = \frac{1-Y^j}{1-Y^N}$, $0 \leq j \leq N$, where, $Y = 1 + R\Delta x$, which is oscillation free.

2 - 1.2.2 Temporal Discretization:

Returning to the time-dependent problem (Eq 2 - 1.1.1) ---- (Eq 2 - 1.1.3), assume $u > 0$ constant.

The *explicit* (forward time difference) discretization is:

$$\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} + u \left(\frac{\phi_j^n - \phi_{j-1}^n}{\Delta x} \right) - D \frac{(\phi_{j-1}^n - 2\phi_j^n + \phi_{j+1}^n)}{\Delta x^2} = 0 \quad (\text{Eq 2 - 1.2.6})$$

The stability condition for such discretization is: $\Delta t \leq \frac{\Delta x^2}{2D + u\Delta x}$

The *implicit* (backward time difference) discretization, which is unconditional stable, is:

$$\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} + u \left(\frac{\phi_j^{n+1} - \phi_{j-1}^{n+1}}{\Delta x} \right) - D \frac{(\phi_{j-1}^{n+1} - 2\phi_j^{n+1} + \phi_{j+1}^{n+1})}{\Delta x^2} = 0 \quad (\text{Eq 2 - 1.2.7})$$

2 - 1.3 Control Volume Approach

The basic idea of the control volume approach (Patañkar, 1980) is: the calculation domain is divided into a number of non-overlapping control volumes such that there is one control volume surrounding each grid point, see Figure 2 - 1.3.1. The differential equation is integrated over each control volume. The result equation is the discretization equation containing the values of ϕ for a group of grid points.

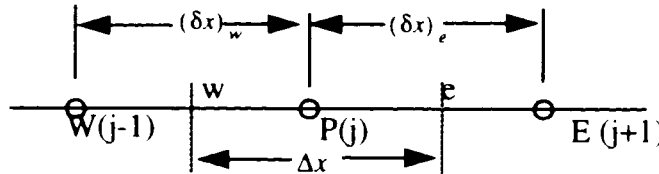


Figure 2 - 1.3.1 Grid-point cluster for the one-dimensional problem

Consider one-dimensional convective-diffusion governed by (Eq 2 - 1.1.1),

$$\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} - D \frac{\partial^2 \phi}{\partial x^2} = 0, 0 < x < 1, t > 0 \quad (\text{Eq 2 - 1.3.1})$$

Preparation: In Figure 2 - 1.3.1, the dashed lines show the face of the control volume. Integrating (Eq 2 - 1.3.1) over the control volume and over the time interval from t to $t + \Delta t$, we get

$$\int_w^e \int_t^{t+\Delta t} \frac{\partial \phi}{\partial t} dt dx = \int_t^{t+\Delta t} \int_w^e \left(-u \frac{\partial \phi}{\partial x} + D \frac{\partial^2 \phi}{\partial x^2} \right) dx dt \quad (\text{Eq 2 - 1.3.2})$$

It can be shown that difference schemes can be applied to the control volume method.

For the explicit scheme,

$$(\phi_p^{n+1} - \phi_p^n) \Delta x = \Delta t \left(-u (\phi_e^n - \phi_w^n) + D \left(\left(\frac{\partial \phi}{\partial x} \right)_e^n - \left(\frac{\partial \phi}{\partial x} \right)_w^n \right) \right) \quad (\text{Eq 2 - 1.3.3})$$

For the fully implicit scheme,

$$(\phi_p^{n+1} - \phi_p^n) \Delta x = \Delta t \left(-u (\phi_e^{n+1} - \phi_w^{n+1}) + D \left(\left(\frac{\partial \phi}{\partial x} \right)_e^{n+1} - \left(\frac{\partial \phi}{\partial x} \right)_w^{n+1} \right) \right) \quad (\text{Eq 2 - 1.3.4})$$

Profile assumption: (1) stepwise profile, (2) piecewise-linear profile.

For the convection term, if a piecewise-linear profile is used, $\phi_e = \frac{1}{2}(\phi_E + \phi_P)$ and $\phi_w = \frac{1}{2}(\phi_P + \phi_W)$. Then $\phi_e^n - \phi_w^n = \frac{1}{2}(\phi_E^n - \phi_W^n)$, which is a centered difference scheme.

Assuming $u > 0$, if a stepwise profile is used, $\phi_e = \phi_P$ and $\phi_w = \phi_W$. Thus, $\phi_e^n - \phi_w^n = (\phi_P^n - \phi_W^n)$, which is an upwind scheme.

Using a piecewise-linear profile assumption for the derivative term, set $(\delta x)_e = (\delta x)_w = \Delta x$. Since $\left(\frac{\partial \phi}{\partial x} \right)_e^n = \frac{\phi_E^n - \phi_P^n}{(\delta x)_e}$, $\left(\frac{\partial \phi}{\partial x} \right)_w^n = \frac{\phi_P^n - \phi_W^n}{(\delta x)_w}$, then $\left(\frac{\partial \phi}{\partial x} \right)_e^n - \left(\frac{\partial \phi}{\partial x} \right)_w^n = \frac{\phi_E^n - 2\phi_P^n + \phi_W^n}{\Delta x}$, which is second centered difference scheme.

The discretization equation:

$$\text{explicit: } (\phi_p^{n+1} - \phi_p^n) \Delta x = \Delta t \left(-u (\phi_p^n - \phi_w^n) + D \left(\frac{\phi_E^n - \phi_P^n}{(\delta x)_e} - \frac{\phi_P^n - \phi_W^n}{(\delta x)_w} \right) \right) \quad (\text{Eq 2 - 1.3.5})$$

For better understanding, set $(\delta x)_e = (\delta x)_w = \Delta x$, substitute j for P , $j-1$ for W , $j+1$ for E , then the above equation can be rewritten as:

$$\frac{(\phi_j^{n+1} - \phi_j^n)}{\Delta t} = -u \frac{(\phi_j^n - \phi_{j-1}^n)}{\Delta x} + D \left(\frac{(\phi_{j+1}^n - \phi_j^n) - (\phi_j^n - \phi_{j-1}^n)}{(\Delta x)^2} \right) \quad (\text{Eq 2 - 1.3.6})$$

which is exact same as (Eq 2 - 1.2.6)

$$\text{implicit: } (\phi_p^{n+1} - \phi_p^n) \Delta x = \Delta t \left(-u (\phi_p^{n+1} - \phi_w^{n+1}) + D \left(\frac{\phi_E^{n+1} - \phi_P^{n+1}}{(\delta x)_e} - \frac{\phi_P^{n+1} - \phi_W^{n+1}}{(\delta x)_w} \right) \right) \quad (\text{Eq 2 - 1.3.7})$$

Again, it can be rewritten as:

$$\frac{(\phi_p^{n+1} - \phi_p^n)}{\Delta t} = -u \frac{(\phi_p^{n+1} - \phi_w^{n+1})}{\Delta x} + D \left(\frac{(\phi_E^{n+1} - \phi_P^{n+1}) - (\phi_P^{n+1} - \phi_W^{n+1})}{(\Delta x)^2} \right) \quad (\text{Eq 2 - 1.3.8})$$

which is exact the same as (Eq 2 - 1.2.7).

It has been shown that the finite difference method and control volume have similar features, and both methods suffer from numerical diffusion.

2 - 1.4 Numerical Diffusion:

The actual effect of numerical diffusion on the solution of a PDE can be best illustrated by

setting physical diffusion D to zero in (Eq 2 - 1.1.1):

$$\phi_t + u\phi_x = 0 \quad (\text{Eq 2 - 1.4.1})$$

subject to initial and boundary conditions:

$$\phi(0, t) = f(t), \quad \phi(x, 0) = g(x) \quad (\text{Eq 2 - 1.4.2})$$

The analytical solution to (Eq 2 - 1.4.1) for the special case $g(x) = 0$ and $f(t) = 0$ for $t < 0$ is

$$\phi(x, t) = f(t - x/u) \quad (\text{Eq 2 - 1.4.3})$$

Further, if we consider the additional special case $f(t) = 1, t > 0$, so that at $x = 0$, the entering temperature undergoes a unit step change at $t = 0$

$$f(t) = U(t) = \begin{cases} 0, & t < 0; \\ 1, & t \geq 0; \end{cases} \quad (\text{Eq 2 - 1.4.4})$$

then (Eq 2 - 1.4.3)

$$\phi(x, t) = U(t - x/u) \quad (\text{Eq 2 - 1.4.5})$$

But in reality, if $u = 1$, at $x = 0.5$, the numerical solution may look like:

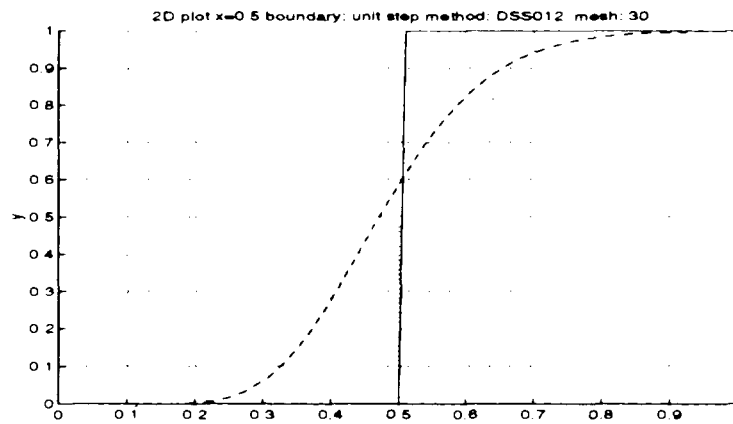


Figure 2 - 1.4.1 Illustration of numerical diffusion.

Comparing the analytical solution (solid line) with the numerical solution (dash line), since there is no physical diffusion in this model, the smoothing of the step function is purely a result of the numerical approximation, which is called numerical diffusion, or artificial viscosity. Since it will distort the breakthrough curve in PSA, numerical diffusion must be minimized.

2 - 2 Numerical Analysis of the Existing Finite Difference Schemes:

Preliminary numerical analyses of existing finite difference methods has been done in the following literature papers. The common feature of these papers is that they use Taylor's series expansion to analyze numerical diffusion based on analysis of the truncation error generated by finite difference schemes.

2 - 2.1 Quantitative Evaluation of Numerical Diffusion

With a Taylor series expansion, Lantz (Lantz 1971) analyzes the truncation error of basic

finite difference schemes, and states that the truncation error results in numerical diffusion. The latter are also quantitative, depending on the block size and time step. Therefore, the numerical diffusivity expressions can provide guidelines for choosing block size and time step combinations that minimize the effect of numerical diffusion.

2 - 2.1.1 Development:

The convective-diffusion equation under consideration is (Eq 2 - 1.1.1) with unit velocity:

$$\frac{\partial \phi}{\partial t} = D \frac{\partial^2 \phi}{\partial x^2} - \frac{\partial \phi}{\partial x} \quad (\text{Eq 2 - 2.1.1})$$

The explicit finite-difference approximation to the differential equation (Eq 2 - 2.1.1) using backward difference (BD) approximation to the first-order space derivative can be written:

$$\frac{\phi_{x,t+\Delta t} - \phi_{x,t}}{\Delta t} = D \frac{\phi_{x+\Delta x,t} - 2\phi_{x,t} + \phi_{x-\Delta x,t}}{(\Delta x)^2} - \frac{\phi_{x,t} - \phi_{x-\Delta x,t}}{\Delta x} \quad (\text{Eq 2 - 2.1.2})$$

The corresponding differential equation being solved by (Eq 2 - 2.1.2), retained only through second-order differentials, is then:

$$\frac{\partial \phi}{\partial t}(x,t) + \frac{\Delta t}{2} \frac{\partial^2 \phi}{\partial t^2}(x,t) = D \frac{\partial^2 \phi}{\partial x^2}(x,t) - \frac{\partial \phi}{\partial x}(x,t) + \frac{\Delta x}{2} \frac{\partial^2 \phi}{\partial x^2}(x,t) + \dots \quad (\text{Eq 2 - 2.1.3})$$

An equivalence between time and space derivative can be established by differentiating (Eq 2 - 2.1.1) with respect to time and with respect to space. The equivalence can be expressed as:

$$\frac{\partial^2 \phi}{\partial t^2}(x,t) = \frac{\partial^2 \phi}{\partial x^2}(x,t) - D \left[\frac{\partial^3 \phi}{\partial x^3}(x,t) - \frac{\partial}{\partial x} \left(\frac{\partial^2 \phi}{\partial x^2}(x,t) \right) \right] + \dots \quad (\text{Eq 2 - 2.1.4})$$

Consequently, the PDE being solved by a BD representation (Eq 2 - 2.1.2) is:

$$\frac{\partial \phi}{\partial t}(x,t) = - \frac{\partial \phi}{\partial x}(x,t) + \left(D + \frac{\Delta x - \Delta t}{2} \right) \frac{\partial^2 \phi}{\partial x^2}(x,t) + \dots \quad (\text{Eq 2 - 2.1.5})$$

Hence the total diffusivity is the quantity $D + (\Delta x - \Delta t)/2$. The term $(\Delta x - \Delta t)/2$ will be called the numerical diffusivity. D is the physical diffusivity.

2 - 2.1.2 Summary of Truncation Error of Basic Finite Difference Schemes

With the above analysis, Lantz shows that numerical diffusivity indeed results from truncation error. A summary of the numerical diffusivity resulting from the various finite difference schemes based on analyses of (Eq 2 - 2.1.1) is listed in Table 2 - 2.1.1¹.

Table 2 - 2.1.1 Summary of truncation error expressions

Difference Form		Error Form
spatial	time	
Backward differencing	Explicit	$(\Delta x - \Delta t)/2$

Table 2 - 2.1.1 Summary of truncation error expressions

Difference Form		Error Form
spatial	time	
Centered differencing	Explicit	$(-\Delta t) / 2$
Backward differencing	Implicit	$(\Delta x + \Delta t) / 2$
Centered differencing	Implicit	$(\Delta t) / 2$

Lantz also compared the observed and calculated numerical diffusivity for the above difference formulations of convective-diffusion equation, and the agreement is good. We don't quote it here. He suggested that the researcher can refer to this table to choose the block size and time step to keep the numerical diffusion at an acceptable level, in the way that either the numerical diffusivity can be made negligibly small compared with physical diffusivity, or numerical plus input diffusivity should be equal to the desired level.

2 - 2.2 Transient and Steady-State Analysis On Artificial Viscosity

Roache (Roache 1972) shows that the usual analysis of the implicit artificial viscosity in finite difference analogs of the linear advection equation is ambiguous, with different results obtained for transient and steady-state problems. For the advection-diffusion equation, the steady-state analysis is shown to be more appropriate for steady-state problems. With both transient and steady-state analysis, he also demonstrated that some popular methods, touted as having no artificial viscosity, actually do have such when applied to steady-state problems.

2 - 2.2.1 Analysis on Convection Equation (usual analysis)

Roache cited: "Artificial viscosity" is a particular kind of truncation error exhibited by some finite difference analogs of advection equations. He credits Noh and Protter (Noh and Protter 1963) with having first presented an analysis of the artificial viscosity of the upwind differencing method applied to the convection equation:

$$\phi_i = -u\phi_x \quad (\text{Eq 2 - 2.2.1})$$

For $u > 0$, the upwind differencing method for (Eq 2 - 2.2.1) gives:

$$(\phi_i^{n+1} - \phi_i^n) / (\Delta t) = -u [(\phi_i^n - \phi_{i-1}^n) / (\Delta x)] \quad (\text{Eq 2 - 2.2.2})$$

Rewriting the above equation using Courant number $c = (u\Delta t) / (\Delta x)$,

$$\phi_i^{n+1} = \phi_i^n - c (\phi_i^n - \phi_{i-1}^n) \quad (\text{Eq 2 - 2.2.3})$$

With a Taylor series expansion, analysis shows

1. We omit the immiscible case.

$$\phi_t = -u\phi_x + D_{numt}\phi_{xx} + O(\Delta x^2, \Delta t^2) \quad (\text{Eq 2 - 2.2.4})$$

where

$$D_{numt} = u(\Delta x)/2 - u^2(\Delta t)/2 = (1/2)u\Delta x(1-c) \quad (\text{Eq 2 - 2.2.5})$$

The non-physical coefficient D_{numt} of $(\partial^2\phi)/(\partial x^2)$, introduced by the method, is referred as the artificial viscosity or numerical diffusion.

2 - 2.2.2 Transient vs. Steady-State Analyses:

The interpretation of D_{numt} in multidimensional, viscous and/or steady-state problems is not as straightforward as it might appear. Suspicion arises when one considers (Eq 2 - 2.2.5), which shows D_{numt} dependent on Δt through the Courant number c .

Consider a problem in which a steady-state has developed $\phi_i^{n+1} = \phi_i^n$. Once this condition is reached, both (Eq 2 - 2.2.1) and computational experience with the upwind differencing method in multidimensional problems indicate that a change in Δt does not change the steady-state solution. Thus Roache moves to a steady-state analysis.

For the steady-state problem, setting $\phi_i^{n+1} = \phi_i^n$, expanding in a Taylor series, a steady-state numerical viscosity is obtained,

$$D_{numS} = (1/2)u\Delta x \quad (\text{Eq 2 - 2.2.6})$$

In this formulation, $D_{numS} \neq f(\Delta t)$, and the steady-state independence of Δt is not suspect.

Roache then extended steady-state analysis to the problem with a diffusion term present:

$$\phi_t = -u\phi_x + D\phi_{xx} \quad (\text{Eq 2 - 2.2.7})$$

Again, the steady-state analysis is (Eq 2 - 2.2.6) while the transient analysis is (Eq 2 - 2.2.5). Unlike the situation for the convection equation, the distinction between the D_{numt} of (Eq 2 - 2.2.5) and D_{numS} of (Eq 2 - 2.2.6) is then important, and the steady-state analysis is more appropriate for this convective-diffusion problem.

For the multidimensional problems with nonlinear coefficients, the resolution of the transient and steady-state analyses is not so neat. Now, consider the application of upwind differencing with (physical) diffusion terms in two-dimensional flow. For constant $u, v_i > 0$, this gives

$$\begin{aligned} (\phi_{ij}^{n+1} - \phi_{ij}^n) / (\Delta t) = & -(u\phi_{ij}^n - u\phi_{i-1,j}^n) / (\Delta x) - (v\phi_{ij}^n - v\phi_{i,j-1}^n) / (\Delta y) \\ & + D((\phi_{i+1,j}^n - 2\phi_{ij}^n + \phi_{i-1,j}^n) / (\Delta x^2) + (\phi_{i,j+1}^n - 2\phi_{ij}^n + \phi_{i,j-1}^n) / (\Delta y^2)) \end{aligned} \quad (\text{Eq 2 - 2.2.8})$$

for which stability requires

$$\Delta t \leq \frac{1}{2D(1/(\Delta x^2) + 1/(\Delta y^2)) + |u|/(\Delta x) + |v|/(\Delta y)} \quad (\text{Eq 2 - 2.2.9})$$

where, in the transient analysis: $D_{numtx} = (1/2)u\Delta x(1-c_x)$, $D_{numty} = (1/2)v\Delta y(1-c_v)$

but in the steady-state analysis, $D_{numSx} = (1/2)u\Delta x$, $D_{numSy} = (1/2)v\Delta y$

Both analyses give different values of D_{numt} or D_{numx} in different directions, each of the form (Eq 2 - 2.2.5) or (Eq 2 - 2.2.6). The transient analysis predicts that the steady-state solution for the upwind differencing method is a function of Δt , which disagrees with computational experience. Thus, the steady-state analysis does appear to be appropriate for multidimensional and/or nonlinear steady-state problems.

2 - 2.2.3 Transient and Steady-state Analysis of Other Methods

Roache summarizes the analysis of other methods in Table 2 - 2.2.1. Comparing this table to Table 2 - 2.1.1, the earlier one does not contain u because (Eq 2 - 2.1.1) takes $u = 1$.

Table 2 - 2.2.1 Implicit artificial viscosities from transient and steady-state analyses for various finite difference methods applied to $\phi_t = -u\phi_x$, with $c = (u\Delta t) / (\Delta x)$

Description	Method	Transient D_{numt}	Steady D_{numx}	Formal Truncation Error
1. Upwind	$\phi_i^{n+1} = \phi_i^n - c(\phi_i^n - \phi_{i-1}^n)$	$\frac{u\Delta x}{2}(1-c)$	$\frac{u\Delta x}{2}$	$O(\Delta t, \Delta x)$
2. FTCS	$\phi_i^{n+1} = \phi_i^n - (c/2)(\phi_{i+1}^n - \phi_{i-1}^n)$	$-(u^2\Delta t/2)$	0	$O(\Delta t, \Delta x^2)$
3. Lax	$\phi_i^{n+1} = (1/2)(\phi_{i+1}^n + \phi_{i-1}^n) - (c/2)(\phi_{i+1}^n - \phi_{i-1}^n)$	$\frac{\Delta x^2}{2\Delta t}(1-c^2)$	$\left(\frac{\Delta x^2}{2\Delta t}\right)$	$O\left(\Delta t, \Delta x^2, \frac{\Delta x^2}{\Delta t}\right)$
4. Leith	$\phi_i^{n+1} = \phi_i^n - (c/2)(\phi_i^n - \phi_{i-1}^n) + c^2/2(\phi_{i+1}^n + 2\phi_i^n + \phi_{i-1}^n)$	0	$\left(\frac{u^2\Delta t}{2}\right)$	$O(\Delta t, \Delta x^2)$
5. Matsuno	$\overline{\phi_i^{n+1}} = \phi_i^n - (c/2)(\phi_i^n - \phi_{i-1}^n)$ $\phi_i^{n+1} = \phi_i^n - (c/2)(\overline{\phi_{i+1}^{n+1}} - \overline{\phi_{i-1}^{n+1}})$	$u^2\Delta t$	0	$O(\Delta t, \Delta x^2)$

2 - 2.3 The Upwind Difference Schemes

In "incompressible flow", the author summarizes the common finite difference methods used in simulation of incompressible flow. The first upwind differencing method has advantages among other finite difference methods, although it suffers from numerical diffusion. For example, compared with the Forward-time, centered-space (FTCS) method (see also Table 2 - 2.2.1), upwind differencing is not stability limited by $\Delta x > 2/R$ (see also section 2 - 1.2.1).

Based on the first upwind differencing method, the second upwind differencing method is developed, and the latter can have better accuracy.

2 - 2.3.1 The Second Upwind Differencing Method

In the second upwind differencing method, or "donor cell" method (Gentry, Martin, and Daly, 1966) some sort of average interface velocities on each side of the mesh point are defined; the sign of these velocities determines, by upwind differencing, which cell values of ϕ to use. In one-

dimensional notation,

$$\frac{\Delta\phi_i}{\Delta t} = -\frac{u_e\phi_e - u_w\phi_w}{\Delta x} \quad (\text{Eq 2 - 2.3.1})$$

Where

$$u_e = \frac{1}{2}(u_{i+1} + u_i), u_w = \frac{1}{2}(u_i + u_{i-1}) \quad (\text{Eq 2 - 2.3.2})$$

or perhaps some other averaging scheme.

And using the upwinding scheme,

$$\begin{aligned} \phi_e &= \phi_i, (u_e > 0); \phi_e = \phi_{i+1}, (u_e < 0) \\ \phi_w &= \phi_{i-1}, (u_w > 0); \phi_w = \phi_i, (u_w < 0) \end{aligned} \quad (\text{Eq 2 - 2.3.3})$$

Consider ϕ constant with $\phi_{i-1} = \phi_i = \phi_{i+1} = \hat{\phi}$, but spatially varying u . Then,

$$\frac{\Delta\phi_i}{\Delta t} = -\frac{u_e\phi_e - u_w\phi_w}{\Delta x} = -\frac{\hat{\phi}\frac{1}{2}(u_{i+1} + u_i) - \hat{\phi}\frac{1}{2}(u_i + u_{i-1})}{\Delta x} = -\hat{\phi}\left(\frac{u_{i+1} - u_i}{2\Delta x}\right) \quad (\text{Eq 2 - 2.3.4})$$

which is second-order accurate for the convection field.

Thus, it can be considered that the second upwind differencing method is more accurate than the first method, since it retains second order accuracy of $\partial(u\phi)/\partial x$ possessed by the centered difference scheme; as a proof, the author of "incompressible flow" refers to Torrance (1968), that it is indeed superior to the first upwind difference method in an actual two-dimensional computation.

2 - 2.3.2 Comments

However, I conjecture that this second upwind method will not be second order accurate if the solution has the pattern of a shock wave, since it is quite contrary to the assumption of (Eq 2 - 2.3.4). Therefore, it can not be the solution to the numerical diffusion problem in PSA modelling. As shown in chapter 1, the concentration profile has the pattern of a breakthrough curve.

2 - 2.4 A Consistency Check for Estimating Truncation Error due to Upstream Differencing

It has been shown that upstream differencing seems quite an attractive method because of its inherent algorithmic simplicity and the fact it is oscillation free (see section 2 - 1.2.1). However, the associated numerical diffusion is an extremely serious handicap which may outweigh its positive features. For this reason, it is important to have a *direct* method for estimating the error incurred when upstream differencing is used for convection. Leonard (1978) outlines a simple procedure for achieving this.

2 - 2.4.1 Introduction

Consider a convective-diffusive equation with a local source term in one dimension:

$$\frac{\partial \phi}{\partial t} = -\frac{\partial(u\phi)}{\partial x} + D\frac{\partial^2 \phi}{\partial x^2} + S \quad (\text{Eq 2 - 2.4.1})$$

A popular solution has been to use upstream differencing for the convective term, while still using central differencing for the diffusion terms. The artificial diffusion takes the form $(u\Delta x)/2$, (see section 2 - 2.3.1)

(Eq 2 - 2.4.1) can be rearranged into:

$$\frac{\partial(u\phi)}{\partial x} = -\frac{\partial \phi}{\partial t} + D\frac{\partial^2 \phi}{\partial x^2} + S \quad (\text{Eq 2 - 2.4.2})$$

$$\text{or } \frac{\partial(u\phi)}{\partial x} = S^* \quad (\text{Eq 2 - 2.4.3})$$

The basic idea of the method is as follows: assume that an exact solution ϕ^E is known and that from this, S^* can be computed to any desired degree of accuracy; then at given fixed values of y , z , and t , (Eq 2 - 2.4.3) represents an ODE which can be solved using upstream differencing for the x -derivative. Clearly, a comparison of ϕ computed this way with the original ϕ^E gives a direct measure of the truncation error committed in using upstream differencing in the x direction.

In practice, ϕ^E and the corresponding $S^*(x)$ will not be known exactly. Instead, ϕ will be available from a computer simulation of the particular problem. But for fixed y , z , and t , a S^* can be computed with the given ϕ function directly from (Eq 2 - 2.4.3) using any desired numerical accuracy. Then using the consistent value of $S^*(x)$, ϕ is again recomputed from (Eq 2 - 2.4.3) using upstream differencing for the x -derivative. The difference between the original and recomputed ϕ functions now represents the inconsistency due to upstream differencing in the x direction in comparison with the accuracy used in computing S^* . To the extent that computed S^* will be of the same order as the (unknown) 'exact' S^* , this inconsistency will be of the same order as the 'true' truncation error due to using upstream differencing in the particular coordinate direction. The final error estimate can be taken simply as the largest of the individual estimates.

2 - 2.4.2 Basic formula

Suppose a numerical solution of (Eq 2 - 2.4.1) has been made called $\phi^I(x, y, z, t)$ and $u^I(x, y, z, t)$, and plot as function of x . In a given computational cell, the numerical model of (Eq 2 - 2.4.3) is, as defined in Figure 2 - 2.4.1

$$\frac{u_r \phi_c - u_l \phi_L}{\Delta x_c} = S^*, u > 0 \quad (\text{Eq 2 - 2.4.4})$$

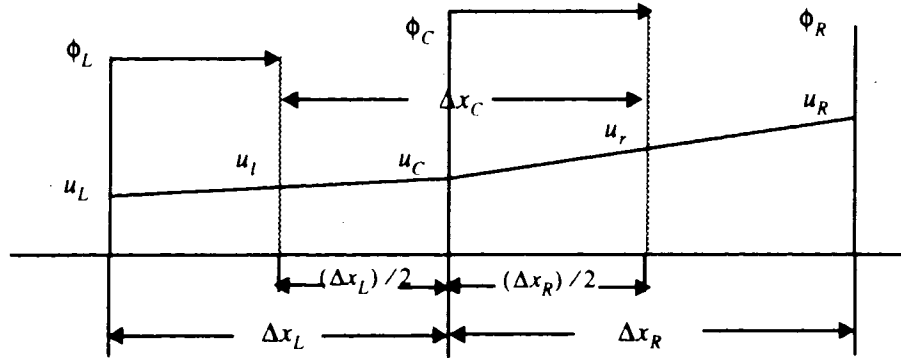


Figure 2 - 2.4.1 Definition of terms in equation (Eq 2 - 2.4.4)

(Eq 2 - 2.4.4) could be used to recompute ϕ as function of x , e.g.,

$$\phi_C = (u_L' \phi_L + \Delta x_C \bar{S}^*) / u_C' \quad (\text{Eq 2 - 2.4.5})$$

From (Eq 2 - 2.4.3), the consistent control-volume average is given by:

$$\bar{S}^* = \frac{1}{\Delta x_C} \int_{-(\Delta x_L)/2}^{(\Delta x_R)/2} \frac{\partial(u\phi)}{\partial x} dx = \frac{1}{\Delta x_C} (u_R \phi_R - u_L \phi_L) \quad (\text{Eq 2 - 2.4.6})$$

Therefore, the recomputed function ϕ'' at each cell:

$$\phi_C'' = u_L' (\phi_L'' - \phi_L') / u_C' + \phi_C' \quad (\text{Eq 2 - 2.4.7})$$

for $u_L, u_R > 0$, thus generating a downstream-marching algorithm for ϕ'' . In most cases, a fully graphical procedure is adequate.

Graphical procedure for constant velocity

Rewrite the equation for constant velocity: $\phi_C'' = \phi_C'' + (\phi_r' - \phi_l')$ This is the basis for the graphical construction of ϕ'' from ϕ' .

It can be seen that, at any station:

$$\phi_r' - \phi_C'' = \phi_l' - \phi_L'' = \Delta \phi_{Ref} = const \quad (\text{Eq 2 - 2.4.8})$$

Thus, **a simpler construction procedure:**

1. compute $\Delta \phi_{Ref} = \phi_r' - \phi_C'$ at the reference point;
2. find the ϕ_C'' value from $\phi_C'' = \phi_r' - \Delta \phi_{Ref}$.

Note that the local inconsistency, $\phi_C' - \phi_C''$, is a direct estimate of the truncation error involved in the original computation generating ϕ' .

If the local inconsistency between ϕ'' and ϕ' is written:

$$\varepsilon = \phi_C' - \phi_C'' = (\phi_r' - \phi_C'') - (\phi_r' - \phi_C') = \Delta \phi_{Ref} - \Delta \phi \quad (\text{Eq 2 - 2.4.9})$$

Locally, an estimate of ε , correct to second order in Δx , can be made by noting that:

$$\Delta\phi = \phi'_r - \phi'_c = \frac{\Delta x}{2} \frac{\partial\phi'}{\partial x} - \frac{1}{8} \frac{\partial^2\phi'}{\partial x^2} \Delta x^2 + O(\Delta x^3) \quad (\text{Eq 2 - 2.4.10})$$

Therefore,

$$\varepsilon \approx \frac{1}{2} \Delta x_{Ref} \left(\frac{\partial\phi'}{\partial x} \right)_{Ref} - \frac{1}{2} \Delta x \left(\frac{\partial\phi'}{\partial x} \right) \quad (\text{Eq 2 - 2.4.11})$$

It should be clear from the construction of graphic or (Eq 2 - 2.4.9) and (Eq 2 - 2.4.11) that the only upstream-difference computations which are free of first-order truncation error (in regions of approximately constant velocity) are those for which ϕ varies, at most, linearly with the streamwise grid index. If there is any curvature in ϕ , there will be a non-zero first-order truncation error. In order to minimize ε , a finer grid spacing is needed in regions of larger gradient. In fact, if an estimate of the solution is available (from rough calculation or from experimental measurements), (Eq 2 - 2.4.11) can be used to define the appropriate local grid size necessary for acceptable values of ε . This procedure may need to be iterated (not usually more than once or twice) to check that the inconsistency in the final upstream-difference computation is within acceptable bounds.

An approximate check for the flux consistency is also developed in (Leonard 1978). The same graphical construction can be used in the general case to make an adequate first-order consistency check on the convective flux computation. Leonard claims that method is particularly useful in rationalizing empirical tuning procedures used to calibrate upstream-difference numerical models in terms of measured results.

2 - 2.5 Comments on Section 2-2

In this section, we have studied the analyses of numerical diffusion in various finite difference methods, especially for upwind difference schemes. The basic idea is to use the expression of numerical diffusion as guideline to choose proper time step and block size to minimize it. This is one way of solving the problem of numerical diffusion. However, it is not easy to implement this idea, especially, for example, when the transient and steady-state solutions for P.D.E are both important or the truncation error bounds check may require impractical fine mesh. Alternatively, researchers are seeking generally more accurate numerical methods, such as higher-order difference formulation advanced on upwind difference schemes, which will be explored in the next section.

2 - 3 Advanced Study on Numerical Diffusion and Improvement of Upwind Difference Methods

2 - 3.1 A Survey of Finite Differences with Upwinding for Numerical Modelling of the Incompressible Convective Diffusion Equation

Leonard (1980), analyzes the available finite difference schemes, and emphasizes that a straight-forward upstream shifted third-order convective differencing scheme can automatically combine inherent stability and accuracy, and is algorithmically consistent with second-order diffusive differencing. Since a central difference method may lead to unphysical oscillation or computational nonconvergence and non-centered upstream-shifted convective differencing schemes may cause artificial diffusion, the introduced third-order finite difference method seems a remedy to these problems.

2 - 3.1.1 Development:

The model convective diffusion equation:

$$\frac{\partial \phi}{\partial t} = -u \frac{\partial \phi}{\partial x} + D \frac{\partial^2 \phi}{\partial x^2} + S \quad (\text{Eq 2 - 3.1.1})$$

$$\text{or } u \frac{\partial \phi}{\partial x} = D \frac{\partial^2 \phi}{\partial x^2} + S^*, \text{ where, } S^* = S - \frac{\partial \phi}{\partial t} \quad (\text{Eq 2 - 3.1.2})$$

Analysis of Finite-Difference Operators

Through the feedback sensitivity of various finite difference operators, Leonard demonstrates the stabilities of these operators.

$$(\partial \phi_i) / (\partial t) = RHS, \text{ RHS: the numerically modelled terms on the right-hand side.} \quad (\text{Eq 2 - 3.1.3})$$

$$\Sigma = (\partial RHS) / (\partial \phi_i), \Sigma: \text{feedback sensitivity.} \quad (\text{Eq 2 - 3.1.4})$$

If $\Sigma > 0$, the algorithm would lead to exponential growth of any perturbations and is therefore undesirable. If $\Sigma = 0$, the algorithm has neutral sensitivity and perturbations can be superimposed on the solution without affecting the RHS---the algorithm is insensitive to these errors and no automatic corrective actions are taken. This type of neutral sensitivity is often associated with temporal and spatial oscillations. It is directly analogous to the oscillatory nature of marginal stability in dynamic analyses. If $\Sigma < 0$, the algorithm will damp out random fluctuations. This is clearly a highly desirable property of any numerical algorithm.

The diffusion term

For the second central difference approximation to the diffusion term

$$D \left[\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} \right] = D \left[\left(\frac{\partial^2 \phi}{\partial x^2} \right)_i + \frac{1}{12} \phi_i^{(iv)} (\Delta x^2) + \dots \right] \quad (\text{Eq 2 - 3.1.5})$$

For this operator, $\Sigma = -2D / (\Delta x^2)$. Therefore this operator is stable.

The convection term

1. The second central difference

$$-u \left[\frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} \right] = -u \left[\frac{\partial \phi}{\partial x} + \frac{1}{6} \phi_i^{(iii)} \Delta x^2 + \dots \right] \quad (\text{Eq 2 - 3.1.6})$$

So for this operator $\Sigma = 0$, a major problem of this operator. Therefore, it is unstable.

2. The upwinding Strategy

i. First order upwinding

$$-u \left[\frac{\phi_i - \phi_{i-1}}{\Delta x} \right] = -u \left[\left(\frac{\partial \phi}{\partial x} \right)_i - \frac{1}{2} \phi_i^{(ii)} \Delta x + \dots \right], u > 0 \quad (\text{Eq 2 - 3.1.7})$$

$$-u \left[\frac{\phi_{i+1} - \phi_i}{\Delta x} \right] = -u \left[\left(\frac{\partial \phi}{\partial x} \right)_i - \frac{1}{2} \phi_i^{(ii)} \Delta x + \dots \right], u < 0 \quad (\text{Eq 2 - 3.1.8})$$

$$D_{nums} = |u| \Delta x / 2, D_{nums} \ll D \quad (\text{Eq 2 - 3.1.9})$$

$$\Sigma = -|u| / (\Delta x), \text{ stable} \quad (\text{Eq 2 - 3.1.10})$$

ii. Second order upwinding

$$-u \left[\frac{3\phi_i - 4\phi_{i-1} + \phi_{i-2}}{2\Delta x} \right] = -u \left[\left(\frac{\partial \phi}{\partial x} \right)_i - \frac{1}{3} \phi_i^{(iii)} \Delta x^2 + \frac{1}{4} \phi_i^{(iv)} \Delta x^3 + \dots \right], \text{ for } u > 0 \quad (\text{Eq 2 - 3.1.11})$$

$$-u \left[\frac{3\phi_{i+2} - 4\phi_{i+1} + \phi_{i+1}}{2\Delta x} \right] = -u \left[\left(\frac{\partial \phi}{\partial x} \right)_i - \frac{1}{3} \phi_i^{(iii)} \Delta x^2 - \frac{1}{4} \phi_i^{(iv)} \Delta x^3 + \dots \right], \text{ for } u < 0 \quad (\text{Eq 2 - 3.1.12})$$

$$\Sigma = -(3|u|) / (2\Delta x) \quad (\text{Eq 2 - 3.1.13})$$

which is of course stabilizing. However, the discretization error is potentially oscillatory.

iii. Third order upwinding

$$-u \left[\frac{-\phi_{i+2} + 8\phi_{i+1} - 8\phi_{i-1} + \phi_{i-2}}{12\Delta x} \right] - |u| \left[\frac{\phi_{i+2} - 4\phi_{i+1} + 6\phi_i - 4\phi_{i-1} + \phi_{i-2}}{12\Delta x} \right] \quad (\text{Eq 2 - 3.1.14})$$

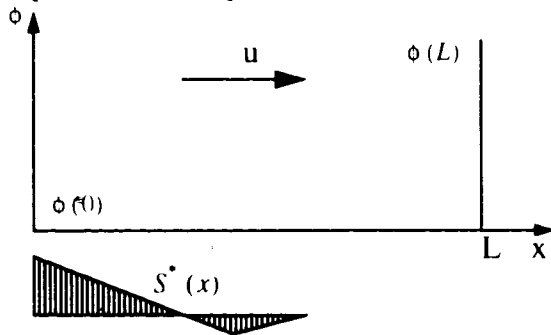
$$= -u \left(\frac{\partial \phi}{\partial x} \right)_i - \frac{1}{12} |u| \phi_i^{(iv)} \Delta x^3 + \frac{1}{4} u \phi_i^{(v)} \Delta x^4 + \dots$$

$$\Sigma = -|u| / (2\Delta x) - 2D\Delta x^2 \quad (\text{Eq 2 - 3.1.15})$$

which, being unconditionally negative for all (physical) values of u and D , represents very good damping qualities.

2 - 3.1.2 Model Test Problem

For comparative studies of the performance of different spatial differencing methods, a simple model test problem is set:



$$u \frac{d\phi}{dx} = D \frac{d^2\phi}{dx^2} + S^*(x), 0 \leq x \leq L$$

$$\phi(0) = 0, \phi(L) = const$$

where u and Γ are positive constants

Figure 2 - 3.1.1 Formulation of model problem

Second-order central differencing

$$u \left[\frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} \right] = D \left[\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} \right] + S^* \quad (\text{Eq 2 - 3.1.16})$$

which can be rearranged into

$$\frac{1}{2} [\phi_{i+1} - \phi_{i-1}] = \frac{1}{P_\Delta^*} [\phi_{i+1} - 2\phi_i + \phi_{i-1}] + \bar{S}^* \quad (\text{Eq 2 - 3.1.17})$$

where $\bar{S}^* = (S^* \Delta x) / u$ and P_Δ^* is the effective grid Peclet number. ---in this case $P_\Delta^* = P_\Delta$.

As noted, this method is relatively well-behaved for $P_\Delta < 2$, but may lead to oscillatory result when $P_\Delta > 2$.

First-order upwinding and hybrid methods

First order upwinding

$$u \left[\frac{\phi_i - \phi_{i-1}}{\Delta x} \right] = D \left(\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} \right) + S^* \quad (\text{Eq 2 - 3.1.18})$$

can be written as

$$\left(\frac{1}{2} [\phi_{i+1} - \phi_{i-1}] = \frac{1}{P_\Delta^*} [\phi_{i+1} - 2\phi_i + \phi_{i-1}] \right) + \bar{S}^* \quad (\text{Eq 2 - 3.1.19})$$

where $P_\Delta^* \equiv 2$. Thus, the result is inaccurate, therefore unacceptable.

This defect is partially corrected by a hybrid technique in which second-order central differencing is used for both convection and diffusion when $P_\Delta \leq 2$, and full upwinding when $P_\Delta \geq 2$. The governing equation can still be written in the form in (Eq 2 - 3.1.19), except, $P_\Delta^* = P_\Delta$ for $P_\Delta \leq 2$, and $P_\Delta^* \equiv 2$, for $P_\Delta \geq 2$. However, for $P_\Delta \leq 2$, there is no improvement over central differencing and for $P_\Delta \geq 2$, the method is so inaccurate that it is useless for any practical purposes.

“Optimal” upwinding

$$u \left[\alpha \frac{\phi_i - \phi_{i-1}}{\Delta x} + (1 - \alpha) \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} \right] = D \left[\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} \right] + S^* \quad (\text{Eq 2 - 3.1.20})$$

If written in the standard form “optimal” upwind is equivalent to second order central differencing using an effective grid Peclet number $P_\Delta^* = 2 \tanh(P_\Delta/2)$.

Since the effective diffusion coefficient is $D_{eff} = u\Delta x / P_\Delta^*$, it can be seen that:

$$D_{eff} = D, \text{ for } P_\Delta < 0.5 \text{ and } D_{eff} = (u\Delta x) / 2 \text{ for } P_\Delta > 5.$$

As P_Δ is increased, the artificial diffusion becomes relatively more dominant. From the computed result, we can see that “optimal” upwinding is actually worse than the (already unacceptable) hybrid scheme over the complete range of Peclet numbers.

The QUICK method

A control-volume formulation named QUICK (Quadratic Upstream interpolation for Convective Kinematics) (Leonard, 1979), as shown in Figure 2 - 3.1.2, uses quadratic interpolation through three points: two points straddling the control-volume face together with an additional adjacent upstream points.

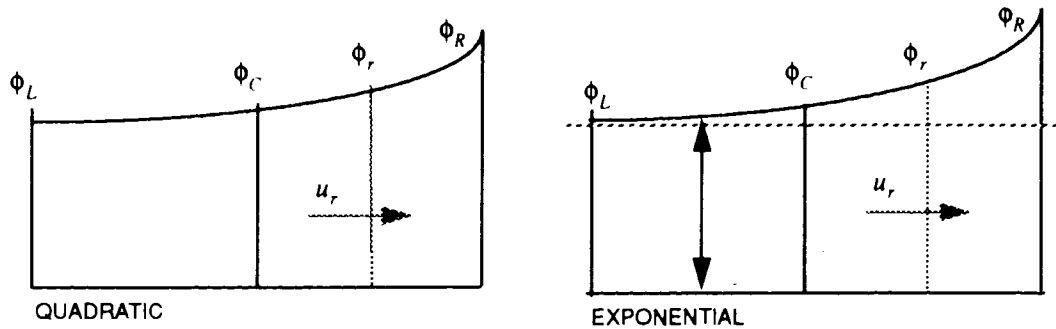


Figure 2 - 3.1.2 Quadratic and exponential three-point interpolation

For the QUICK method, the right-face value is

$$\phi_r = \frac{1}{2} (\phi_i + \phi_{i+1}) - \frac{1}{8} CURV \quad (\text{Eq 2 - 3.1.21})$$

where the curvature term is given by:

$$\phi_{i-1} - 2\phi_i + \phi_{i+1}, \text{ if } u > 0 \quad (\text{Eq 2 - 3.1.22})$$

$$\text{or } \phi_i - 2\phi_{i+1} + \phi_{i+2}, \text{ if } u < 0 \quad (\text{Eq 2 - 3.1.23})$$

$$\left(\frac{\partial \phi}{\partial x}\right)_r = \frac{\phi_{i+1} - \phi_i}{\Delta x} \quad (\text{Eq 2 - 3.1.24})$$

The model equation can be written

$$\frac{1}{2} [\phi_{i+1} - \phi_{i-1}] - \frac{1}{8} [\phi_{i+1} - 3\phi_i + 3\phi_{i-1} - \phi_{i-2}] = \frac{1}{P_\Delta^*} [\phi_{i+1} - 2\phi_i + \phi_{i-1}] + \bar{S}^* \quad (\text{Eq 2 - 3.1.25})$$

in which $P_\Delta^* = P_\Delta$.

In simulations of practical problems the basic QUICK method is extremely “manageable” from the user’s point of view, generating solutions of high accuracy over most of the flow domain in a stable and algorithmically efficient manner.

EXQUISITE method

The problem of resolving a sudden jump in value in the flow direction over a few grid points can be solved by using a more appropriate interpolation function than the quadratic polynomial used in the QUICK method. One needs an interpolation function which can change rapidly and monotonically (when necessary). For this, the exponential function is quite appropriate. Figure 2 - 3.1.2 shows how the same three node values can be interpolated by either quadratic function or an exponential.

For the right-face value, quadratic interpolation gives (as before)

$$\phi_r = \frac{1}{2}(\phi_C + \phi_R) - \frac{1}{8}CURV \quad (\text{Eq 2 - 3.1.26})$$

where for $u > 0$ $CURV = \phi_{i-1} - 2\phi_i + \phi_{i+1}$,

where a three-parameter exponential of the form $\phi = A + Be^{C\xi}$, where $\xi = x - x_i$, leads to

$$\phi_r = A \pm \sqrt{(\phi_R - A)(\phi_C - A)} \quad (\text{Eq 2 - 3.1.27})$$

$$\bar{\phi} = (\phi - \phi_L) / (\phi_R - \phi_L) \quad (\text{normalized variable}) \quad (\text{Eq 2 - 3.1.28})$$

$$\bar{\phi}_r = \frac{\sqrt{\bar{\phi}_C(1 - \bar{\phi}_C)^3} - \bar{\phi}_C}{(1 - 2\bar{\phi}_C)} \quad (\text{Eq 2 - 3.1.29})$$

$$\bar{\phi}_{RG} = \bar{\phi}_C + \frac{\sqrt{\bar{\phi}_C(1 - \bar{\phi}_C)^3}}{(1 - 2\bar{\phi}_C)} \ln\left(\frac{1 - \bar{\phi}_C}{\bar{\phi}_C}\right) \quad (\text{Eq 2 - 3.1.30})$$

$$\left(\frac{\partial \phi}{\partial x}\right)_r = \frac{\bar{\phi}_{RG} - \bar{\phi}_C}{\Delta x} \quad (\text{Eq 2 - 3.1.31})$$

The resulting method is naturally referred to as EXponential or Quadratic Upstream Interpolation for Solution of the Incompressible Transport Equation (EXQUISITE). Clearly, this method can handle source terms and boundary jumps with high accuracy.

2 - 3.2 Beyond First-Order Upwinding: The ULTRA-SHARP Alternative for Non-oscillatory Steady-State Simulation of Convection

Leonard (1990) analyses the shortcomings of Hybrid and PLD method (Power Law Differencing scheme of Patankar (1980)) (also see Leonard 1980, or section 2 - 3.1)----They suffer numerical diffusion. Then he analyses the problems with high-order methods which introduces other problems---usually in the form of overshoots, undershoots or severe oscillation. ----Finally, Leonard presents the ULTRA-SHARP alternative method, which is the advance version of QUICK implemented with a new universal limiter. This method is featured as a high-resolution nonoscillatory multidimensional steady-state high-speed convective modelling.

2 - 3.2.1 Shortcomings of Hybrid and PLDs

The first order upwind scheme and Hybrid and PLD scheme has been studied in (Leonard 1980). In (Leonard 1990), comparing the solution from Hybrid and PLD schemes with the exact solution, one can see the numerical diffusion present in these first-order upwind method.

2 - 3.2.2 Problems with High-Order Methods.

1. spurious oscillation of symmetrical schemes.
2. non-monotonicity of higher-order multidimensional upwinding.

2 - 3.2.3 ULTRA-SHARP Alternative:

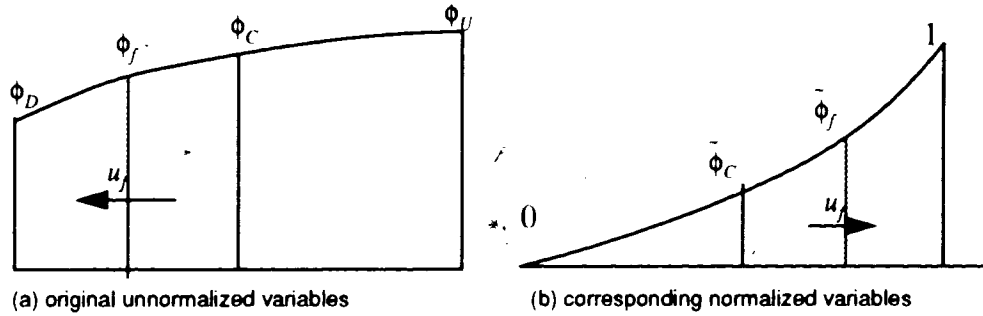


Figure 2 - 3.2.1 Normalized variable diagram for the universal limiter

Define normalized variables:

$$\bar{\phi}(x, y, z) = \frac{\phi(x, y, z) - \phi_U}{\phi_D - \phi_U} \quad (\text{Eq 2 - 3.2.1})$$

The universal limiter constraints on $\bar{\phi}_f$ can be written

$$\begin{aligned} \bar{\phi}_C \leq \bar{\phi}_f \leq 1, \text{ for } (0 \leq \bar{\phi}_C \leq 1) \\ \bar{\phi}_f = 1 + 0.5(\bar{\phi}_C - 1), \text{ for } (\bar{\phi}_C > 1) \\ \bar{\phi}_f = 1.5\bar{\phi}_C, \text{ for } (\bar{\phi}_C < 0) \\ \bar{\phi}_f \leq \text{const}\bar{\phi}_C, \text{ near } (\bar{\phi}_C \rightarrow 0_+) \end{aligned} \quad (\text{Eq 2 - 3.2.2})$$

For the explicit control-volume time-marching algorithm (Appendix II in (Leonard 1990)), Leonard outlines a procedure to apply the universal limiter constraints based on (Eq 2 - 3.2.2) at each stage of a pseudo-time-marching or iterative solution.

The Downwind Weighting Factor

Alternatively, an iterative implicit solution can be implemented by introducing the downwind weighting factor, as now described. Instead of limiting the interface variable directly, the Downwind Weighting Factor (DWF) is introduced as an auxiliary variable, thereby generating a compact implicit scheme suitable for tridiagonal solution methods. After explicitly computing the high-order multidimensional upwind-biased estimate, $\bar{\phi}_f$, define

$$DWF = \frac{\phi_f - \phi_C}{\phi_D - \phi_C} = \frac{\bar{\phi}_f - \bar{\phi}_C}{1 - \bar{\phi}_C} \quad (\text{Eq 2 - 3.2.3})$$

The universal limiter constraints, in terms of DWF, become:

$$\begin{aligned} 0 \leq DWF \leq 1, \text{ for } (0 \leq \bar{\phi}_C \leq 1) \\ DWF \leq ((\text{const} - 1)\bar{\phi}_C) / (1 - \bar{\phi}_C), \text{ near } (\bar{\phi}_C \rightarrow 0_+) \\ DWF = \bar{\phi} / (2(1 - \bar{\phi})), \text{ for } (\bar{\phi} \leq 0) \\ DWF = 0.5, \text{ for } (\bar{\phi} > 1) \end{aligned} \quad (\text{Eq 2 - 3.2.4})$$

Now rewrite the face-value as

$$\phi_f = DWF\phi_U^{TBC} + (1 - DWF)\phi_C^{TBC} \quad (\text{Eq 2 - 3.2.5})$$

where TBC stands for 'to-be-computed' in the next iteration of an implicit line-sweep update.

If ϕ_w^* is the initial higher-order estimate on the control volume west face, the DWF is first computed according to:

$$DWF_w^+ = \frac{\phi_w^* - \phi_w}{\phi_C - \phi_w}, \text{ if } (u_w \geq 0)$$

$$DWF_w^- = \frac{\phi_w^* - \phi_C}{\phi_w - \phi_C}, \text{ if } (u_w \leq 0)$$
(Eq 2 - 3.2.6)

Then, the appropriate DWF is limited according to (Eq 2 - 3.2.4).

The face value in the implicit update is then

$$\phi_w = DWF_w^+ \phi_p + (1 - DWF_w^+) \phi_w, \text{ for } (u_w \geq 0)$$

$$\phi_w = DWF_w^- \phi_w + (1 - DWF_w^-) \phi_p, \text{ for } (u_w < 0)$$
(Eq 2 - 3.2.7)

Similarly, the other faces of this particular control volume cell are approximated. This results in an update of the form

$$a_p \phi_p = a_w \phi_w + a_s \phi_s + a_b \phi_b + a_e \phi_e + a_n \phi_n + a_t \phi_t + b \quad (\text{Eq 2 - 3.2.8})$$

It should be stressed that the higher order multidimensional information and the non-oscillatory universal limiter constraints are implicitly contained in the DWF's of each face rather than involving 'outlying' node-values that are then lumped into the explicit source term. The limiter constraints inherent in the DWF's guarantee non-oscillatory results, with stability and convergence properties similar to the first-order method (DWF=0)----but without introducing artificial numerical diffusion.

Non-oscillatory Multidimensional Result

In the Appendix III (Leonard 1990), Leonard develops a series of higher order upwinding schemes for the convection term. In this experimental test, he shows the simulation results of applying the universal limiter constraints to the QUICK-2D scheme for the oblique-step test at $P_\lambda = 100$. Comparing these results to the unlimited QUICK (reflect third order accuracy) results, the resolution remains sharp (reflecting third-order accuracy without overshoots or undershoot). Applying this universal limiter to higher-order convection schemes can dramatically increase the sharpness in the simulation of the near-discontinuity.

Cost-effectiveness and Adaptive Stencil Expansion

Question: For a prescribed accuracy, which is more cost-effective in terms of overall computer usage, a fine-grid computation using a low-order method, or a coarse grid calculation using a higher-order scheme?

Answer: Gaskell and Lau (1988) compared first-order upwinding with the third-order QUICK scheme and a non-oscillatory version (similar to ULTRA-QUICK) using successive grid refinement on the two-dimensional oblique-step test. For a prescribed accuracy, first-order methods require such a fine grid, compared with the practical grid of third-order, that CPU time was larger by three orders of magnitude!

Question: What order is more 'optimal' in the above sense of cost-effectiveness?

Answer: A simple strategy of *adaptive stencil expansion* is proving to be extremely cost-effective in both steady-state and time-accurate transient simulation: using third order ULTRA-QUICK in 'smooth' regions of the flow, on the basis of some 'non-smoothness' monitor, switch to a higher-order (ULTRA) scheme locally, as needed.

Some monitors such as the CURAV (the absolute average 'curvature') and the GRAD (local absolute normal 'gradient') can be used:

$$CURAVE = 0.5 | (\phi_{i+2} - \phi_{i+1}) - (\phi_i - \phi_{i-1}) | \quad (\text{Eq 2 - 3.2.9})$$

$$GRAD = | \phi_{i+1} - \phi_i | \quad (\text{Eq 2 - 3.2.10})$$

The order-switching strategy is shown

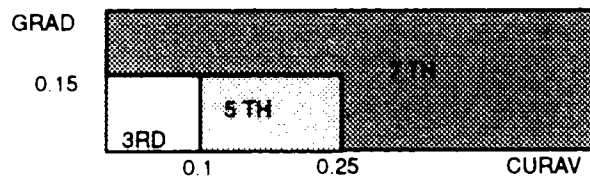


Figure 2 - 3.2.2 Schematic Diagram of order-switching strategy used in the locally adaptive stencil expansion algorithm

For steady-state calculations, it appears that an adaptive ULTRA-QUICK/5th/7th-order convection scheme has a number of attractive attributes, including cost-effectiveness (high coarse-grid accuracy), reliability (excellent stability and convergence properties) and ease of coding.

2 - 3.2.4 Comments on Section 2-3

Leonard has presented a relatively new achievement, viz, using a higher-order upwinding scheme in the region where the solution changes rapidly. These higher-order upwinding schemes have the problem of non-monotonicity. To solve this problem, the author uses the universal limiter scheme to achieve a monotonic but sharp solution. The results are impressive.

2 - 3.3 A Method for Predicting and Minimizing Numerical Diffusion

Peterson (1992) addresses from a Lagrangian viewpoint the problem of inaccuracy of interpolation in space and time resulting numerical diffusion. He uses a Taylor's series expansion for streamwise and cross-stream interpolation processes to give numerical diffusion coefficients, and states that these simple coefficients can be used to adjust physical diffusion coefficients, and

provide second-order accuracy for convection in control volume solutions.

2 - 3.3.1 "FAST" solution for transient convection

Peterson develops "FAST" diffusion correction function, which he provides significantly improved performance for large time step and Peclet numbers, something highly desirable feature for transient solutions.

The conservation equation under consideration by Peterson is

$$\rho \frac{\partial \phi}{\partial t} + \rho \frac{\partial (u\phi)}{\partial x} + \rho \frac{\partial (v\phi)}{\partial y} = \frac{\partial}{\partial x} \left(D \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(D \frac{\partial \phi}{\partial y} \right) + S \quad (\text{Eq 2 - 3.3.1})$$

From a Lagrangian viewpoint, with Taylor's series expansion, Peterson gives both streamwise and cross-stream numerical diffusions results from various difference schemes in the control volume method.

For the one-dimensional problem, only streamwise numerical diffusion is considered, viz,

$$\rho \frac{\partial \phi}{\partial t} + \rho \frac{\partial (u\phi)}{\partial x} = \frac{\partial}{\partial x} \left(D \frac{\partial \phi}{\partial x} \right) + S \quad (\text{Eq 2 - 3.3.2})$$

The implicit and explicit upwind numerical diffusion coefficient can be combined as

$$D_{\text{num}}/D = \frac{1}{2} |1 - c| Pe \quad (\text{Eq 2 - 3.3.3})$$

where c is Courant number, defined as $c = u\Delta t/\Delta x$, and Pe is the grid Peclet number, defined as $Pe = (\rho u \Delta x)/\Gamma$

Peterson argues that, in general, convection terms should be evaluated explicitly to avoid dispersion effects and diffusion terms should be evaluated implicitly.

Therefore, the equations becomes,

$$\rho \left(\frac{\phi_P^{n+1} - \phi_P^n}{\Delta t} \right) + \rho u_w \frac{(\phi_P^n - \phi_W^n)}{\Delta x} = D \left(\frac{\phi_E^{n+1} - 2\phi_P^{n+1} + \phi_W^{n+1}}{\Delta x^2} \right) + S \quad (\text{Eq 2 - 3.3.4})$$

Using numerical diffusion coefficient derived previously to adjust the physical diffusion, the following equation is obtained:

$$\frac{\rho \Delta x}{\Delta t} (\phi_P^{n+1} - \phi_P^n) + \rho u_w (\phi_P^n - \phi_W^n) = A_e D_e \frac{\phi_E^{n+1} - \phi_P^{n+1}}{\Delta x} - A_w D_w \frac{\phi_P^{n+1} - \phi_W^{n+1}}{\Delta x} + S \Delta x \quad (\text{Eq 2 - 3.3.5})$$

where, the FAST diffusion correction functions A can be written as simple functions of the Courant numbers and Peclet numbers,

$$\begin{aligned} A_e &= \max [0, 1 - 0.5 |1 - |c_e|| Pe_e] \\ A_w &= \max [0, 1 - 0.5 |1 - |c_w|| Pe_w] \end{aligned} \quad (\text{Eq 2 - 3.3.6})$$

The Courant numbers are defined as $c_w = \Delta t u_w / \Delta x$ and $c_e = \Delta t u_e / \Delta x$, and $Pe_w = \rho u_w (\Delta x) / \Gamma_w$, $Pe_e = \rho u_e (\Delta x) / \Gamma_e$.

2 - 3.3.2 One-Dimensional Transient Flow Problems

A test problem has been studied by Peterson, some findings are:

1. Without physical diffusion, $Pe = \infty$, only choosing $c = 1$, results preserves the exact solution, and other values of c will result severe numerical diffusion, a higher order scheme is desired.
2. With physical diffusion $Pe < 5$, the FAST diffusion correction functions can be employed, good agreement is obtained and it is superior than Hybrid scheme. For $Pe > 5$, higher order methods are required for accurate solution.
3. For transient solutions, convective terms must be evaluated explicitly. Implicit evaluation of convective terms introduces strong diffusive effect. This diffusion will strongly degrade the quality of any transient solution.

2 - 3.4 Comments on Section 2-3

In this section, we explored more advanced higher-order upwind schemes, first the QUICK method, and then the ULTRA-SHARP which overcomes the oscillation problem----it is the QUICK method implemented with a universal limiter. Although Peterson(1992) presented an inexpensive method in which a diffusion correction coefficient is developed to improve the accuracy of control volume solution, this method can only be employed with a relatively small Peclet number. Since in PSA modelling the Peclet number usually has large value, this diffusion correction function won't resolve the numerical diffusion in our PSA modelling. All the researchers have agreed that the first upwind scheme suffers numerical diffusion, and higher-order upwind schemes seems a remedy for numerical diffusion, and noticing that it suffers oscillation in a certain degree.

2 - 4 Explicit Artificial Viscosity

Artificial viscosity which is explicitly added to the equations in numerical method is referred to as explicit artificial viscosity. von Neumann and Richtmyer(1950), explicitly added a viscosity-like term to the convection equation in order to allow the calculation of shock waves. Their explicit artificial viscosity term was deliberately made proportional to Δx^2 . Approach is given below.

2 - 4.1 An Artificial Dissipation Scheme for the Navier-Stokes Equations

Kaniel, Mond and Ben-Dor (1993) use the finite element method to solve the Navier-Stokes Equations. Isotropic artificial dissipation is added to the Navier-Stokes equations along with a correction term which cancels the artificial dissipation terms in the limit when the mesh size is zero. For a finite mesh size, the correction term replaces the artificial viscosity terms with a hyperviscosity term, i.e., with an artificial dissipation which depends on the fourth derivatives of the velocity.

Development:

Spurious oscillations often occur in numerical solutions of the Navier-Stokes equations which for incompressible flow are

$$\nabla \cdot V = 0 \quad (\text{Eq 2 - 4.1.1})$$

$$\frac{\partial V}{\partial t} + V \cdot \nabla V = -\frac{1}{\rho} \nabla P + \nu \nabla^2 V \quad (\text{Eq 2 - 4.1.2})$$

The Numerical Scheme

Artificial viscosity is added to (Eq 2 - 4.1.2) as well as a correction term:

$$\frac{\partial V}{\partial t} + V \cdot \nabla V = -\frac{1}{\rho} \nabla P + (\nu + \nu_a) \nabla^2 V + \nu_a \nabla \times \Omega, \quad (\text{Eq 2 - 4.1.3})$$

where the artificial viscosity ν_a is assumed to constant, or

$$\frac{\partial V}{\partial t} + V \cdot \nabla V = -\frac{1}{\rho} \nabla P + \nabla [(\nu + \nu_a) \nabla \cdot V] - \nabla \times [(\nu + \nu_a) \nabla \times (\Omega - \nu_a \Omega)] \quad (\text{Eq 2 - 4.1.4})$$

if the artificial viscosity is variable.

To demonstrate this, a finite difference discretization is used for which $\Omega_{i,j}$ is given by:

$$\Omega_{i,j} = \frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x} - \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} \quad (\text{Eq 2 - 4.1.5})$$

The x component of the last term on the right hand side of (Eq 2 - 4.1.4) is therefore given by:

$$-\frac{\partial}{\partial y} [\nabla \times V - \Omega] \equiv \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} - \frac{u_{i,j+2} - 2u_{i,j} + u_{i,j-2}}{\Delta y^2} = O(\Delta y^2) \quad (\text{Eq 2 - 4.1.6})$$

It is obvious that the sum of the added terms [the artificial dissipation $\nabla \times (\nu_a \nabla \times V)$ plus the correction term $-\nabla \times (\nu_a \Omega)$ is not zero, but it is small whenever the flow is smooth with respect to the mesh size.

To summarize, instead of directly solving the Navier-Stokes equations; standard Galerkin integrations are performed on the following equations:

$$u_x + v_y = 0 \quad (\text{Eq 2 - 4.1.7})$$

$$u_x + u_y = 0 \quad (\text{Eq 2 - 4.1.8})$$

$$\frac{u - u^0}{\Delta t} + (uu)_x + (uv)_y = -\frac{1}{\rho} P_x + [(\nu + \nu_a)(u_x + v_y)]_x + [(\nu + \nu_a)(u_y - v_x) - \nu_a \Omega]_y \quad (\text{Eq 2 - 4.1.9})$$

$$\frac{v - v^0}{\Delta t} + (uv)_x + (vv)_y = -\frac{1}{\rho} P_y + [(\nu + \nu_a)(u_x + v_y)]_x + [(\nu + \nu_a)(u_y - v_x) - \nu_a \Omega]_x \quad (\text{Eq 2 - 4.1.10})$$

Artificial Viscosity Coefficient

For the discussed problem, four different artificial viscosity coefficients are presented in the order of increasing accuracy.

These are

$$\frac{v_a}{v} = Pe \quad (\text{Eq 2 - 4.1.11})$$

$$\frac{v_a}{v} = \left| 1 - \left(\frac{v_x - u_y}{\Omega} \right) \right|^d Pe, d = 0.2 \quad (\text{Eq 2 - 4.1.12})$$

$$\frac{v_a}{v} = \frac{(Pe)^2}{Pe + f}, f = 10 \quad (\text{Eq 2 - 4.1.13})$$

$$\frac{v_a}{v} = \frac{(Pe)^2}{Pe + f} \left| 1 - \left(\frac{v_x - u_y}{\Omega} \right) \right|^d \quad (\text{Eq 2 - 4.1.14})$$

where Pe is an elemental Peclet number, which is based on the analysis on a triangular element,

$$Pe = \max_i \left(0, \frac{u(y_i - y_{i^*}) + v(x_{i^*} - x_i)}{6\alpha AB^2} - 1 \right) \quad (\text{Eq 2 - 4.1.15})$$

where i^* is the vertex next to i the counterclockwise direction. A is the area of the triangle, $B^2 = \sum l_i^2 / (2A^2)$ and l_i is the length of the i th side of the triangular element.

Numerical results derived using a coarse grid

The artificial viscosity coefficients were employed in the simulation of the flow over a backward facing step---a realistic internal flow situation which has a variety of boundary conditions and a complicated flow behavior. The present results compared favorably with the experimental result and are by far more accurate than the numerical results obtained by using a finite difference code.

2 - 5 Conclusion for Chapter 2

From this chapter, we have learned that numerical diffusion results from truncation error exhibited by upwind difference schemes either applied in a finite difference method or a control volume approach. Higher order upwind schemes have a better performance on numerical diffusion, but with a certain degree of oscillation. A Universal limiter has been implemented to improve the oscillation generated by higher order upwind schemes. An inexpensive diffusion correction function has been developed to improve numerical diffusion in the control volume formulation. Finally, we include the possibility of solving difficult P.D.Es by explicitly adding artificial viscosity, which can be minimized when the mesh is refined. One important common conclusion to be drawn from the paper studied in this chapter is that higher order upwind schemes provide a solution for numerical diffusion when the Peclet number is high, which is the case of our modelling of a PSA process.

Chapter 3 Convection Equation

From the study on numerical diffusion in the last chapter, an important conclusion drawn is that a higher order upwind scheme is the solution to numerical diffusion. In this chapter, we will explore the Numerical Method of Lines (NUMOL) and the Orthogonal Collocation Method, in which the higher order upwind schemes can be embedded.

Convection equations, so called first-order hyperbolic PDEs, can be solved analytically, and this appears to be very simple. However, they are also a difficult class of partial differential equations to integrate numerically. They are the major governing equations to be solved in our modelling of a PSA process. In this chapter, we will discuss some physical properties of first-order hyperbolic PDEs, and as a demonstration of each of the above methods, we will solve the convection equation with each method. We then study the effect of numerical diffusion as well as oscillation from them by comparing the results.

3 - 1 Convection Equation

In this section, we first study the physical properties of convective equation.

The convection equation is:

$$\frac{\partial \phi}{\partial t} = -v \frac{\partial \phi}{\partial x} \quad (\text{Eq 3 - 1.1})$$

where ϕ can represent any intensive properties of the fluid, such as temperature. Since it is first order in t and x , it requires one initial condition and one boundary condition:

$$\phi(0, t) = f(t), \quad \phi(x, 0) = g(x) \quad (\text{Eq 3 - 1.2})$$

It is very easy to verify that the analytical solution to (Eq 3 - 1.1) to (Eq 3 - 1.2) for the special case $g(x) = 0$ and $f(t) = 0$ for $t < 0$:

$$\phi(x, t) = f\left(t - \frac{x}{v}\right) \quad (\text{Eq 3 - 1.3})$$

Further, if we consider the additional special case $f(t) = 1$, $t > 0$, so that at $x = 0$, the entering temperature undergoes a unit step change at $t = 0$ is

$$f(t) = u(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases} \quad (\text{Eq 3 - 1.4})$$

Following the (Eq 3 - 1.3),

$$\phi(x, t) = u\left(t - \frac{x}{v}\right) \quad (\text{Eq 3 - 1.5})$$

i.e., the unit step in temperature at $x = 0$ propagates along the tube at velocity v , and eventually, at $t = L/v$, the unit-step will reach the other end of the tube (L is the length of the tube). At any position along the tube when $t = x/v$, an observer would see a unit step function pass by. In other words, the convection equation propagates a finite discontinuity along the tube for this special case.

The above property of a first-order hyperbolic PDE causes a problem in computing a numerical solution. Consider the slope of the solution at any point x along the tube when $t = x/v$; the slope ϕ_x in (Eq 3 - 1.1) is infinite. Clearly, any numerical procedure based on well-behaved functions will fail under this condition. Therefore the general methods which work well on parabolic PDEs or elliptic PDEs will not succeed in dealing with first-order hyperbolic PDEs. Unlike the parabolic or elliptic PDEs, the first-order hyperbolic PDE can represent propagation discontinuity or shocks.

3 - 2 Solving Convection Equation with the Numerical Method of Lines

3 - 2.1 A Brief Introduction to the Numerical Method of Lines

Generally, we need to replace PDEs with algebraic approximations, and then to obtain approximate numerical solutions to PDEs with the auxiliary conditions. This procedure is the basis for well-known classical finite difference, finite element, and control volume methods for PDEs. The Numerical Method of Lines (NUMOL) is really just a small departure from this basic approach.

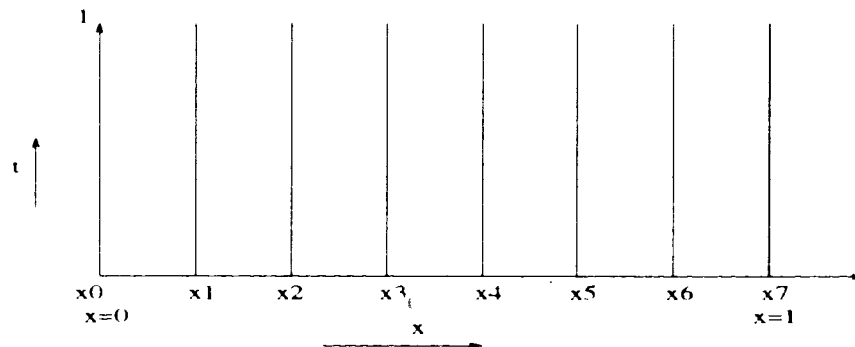


Figure 3 - 2.1.1 sketch of numerical method of lines

In NUMOL, we retain the index i to account for variations of the dependent variable ϕ with x , but we treat t as a continuous variable. Thus, we keep the derivative ϕ_t term, and substitute the derivatives ϕ_x or ϕ_{xx} terms with an difference approximations; this will lead a system of ODEs in t . This is the essence of the NUMOL. One of the advantages of NUMOL is that there are many ODEs solvers available either in the commercial market or research field.

Therefore, the basic steps of NUMOL:

1. Space differencing
2. ODE integration

Next, we will follow the NUMOL procedure to solve the convection equation.

3 - 2.2 Spatial differentiation

Space discretization of the PDEs produces a system of ODEs. As studied in Chapter 2, for spatial discretization of convection term, first upwind difference suffers numerical diffusion, while centered difference has serious oscillation, and higher-order upwind difference seems a remedy for numerical diffusion. Several finite difference formulations (Schiesser 1994) have been used to approximate the convection term, and similar conclusions can be drawn from comparison of the results. These approximation formulas are listed in the following:

1. DSS002----one-dimensional, three-point centered approximations for first-order derivatives, suffer oscillation problems.
2. DSS004----one-dimensional, five-point centered approximations for first-order derivatives, suffer serious oscillation problems.
3. DSS012----one-dimensional, two-point upwind approximations for first-order derivatives, suffer serious numerical diffusion.
4. DSS014----one dimensional, three-point upwind approximations for first-order derivatives, gives an improved result on oscillation and diffusion.
5. DSS020----one-dimensional, five-point upwind approximations for first-order derivatives, gives the best performance among these five methods.

Schiesser (1994) also shows that among higher-order upwind difference formulations, five-point upwind DSS020 is the optimal choice between accuracy and oscillation.

In our experiments, DSS012 and DSS020 for the spatial discretization of convection term are implemented. Here we briefly explain these two methods:

DSS012---First-order upwind approximation to u_x :

for $v > 0$:

$$\frac{d\phi(x_1)}{dx} = \frac{\phi(x_2) - \phi(x_1)}{dx} \quad (\text{Eq 3 - 2.2.1})$$

$$\frac{d\phi(x_i)}{dx} = \frac{\phi(x_i) - \phi(x_{i-1})}{dx} \quad (\text{Eq 3 - 2.2.2})$$

for $v < 0$:

$$\frac{d\phi(x_i)}{dx} = \frac{\phi(x_{i+1}) - \phi(x_i)}{dx} \quad (\text{Eq 3 - 2.2.3})$$

$$\frac{d\phi(x_n)}{dx} = \frac{\phi(x_n) - \phi(x_{n-1})}{dx} \quad (\text{Eq 3 - 2.2.4})$$

DSS020---Fourth-order upwind approximation to u_x

for $v > 0$:

$$\frac{d\phi(x_1)}{dx} = \frac{1}{12dx} (-25\phi(x_1) + 48\phi(x_2) - 36\phi(x_3) + 16\phi(x_4) - 3\phi(x_5)) \quad (\text{Eq 3 - 2.2.5})$$

$$\frac{d\phi(x_2)}{dx} = \frac{1}{12dx} (-3\phi(x_1) - 10\phi(x_2) + 18\phi(x_3) - 6\phi(x_4) + \phi(x_5)) \quad (\text{Eq 3 - 2.2.6})$$

$$\frac{d\phi(x_3)}{dx} = \frac{1}{12dx} (\phi(x_1) - 8\phi(x_2) + 0\phi(x_3) + 8\phi(x_4) - \phi(x_5)) \quad (\text{Eq 3 - 2.2.7})$$

$$\frac{d\phi(x_i)}{dx} = \frac{1}{12dx} (-\phi(x_{i-3}) + 6\phi(x_{i-2}) - 18\phi(x_{i-1}) + 10\phi(x_i) + 3\phi(x_{i+1})) \quad (\text{Eq 3 - 2.2.8})$$

$$\frac{d\phi(x_n)}{dx} = \frac{1}{12dx} (3\phi(x_{n-4}) - 16\phi(x_{n-3}) + 36\phi(x_{n-2}) - 48\phi(x_{n-1}) + 25\phi(x_n)) \quad (\text{Eq 3 - 2.2.9})$$

for $v < 0$:

$$\frac{d\phi(x_1)}{dx} = \frac{1}{12dx} (-25\phi(x_1) + 48\phi(x_2) - 36\phi(x_3) + 16\phi(x_4) - 3\phi(x_5)) \quad (\text{Eq 3 - 2.2.10})$$

$$\frac{d\phi(x_i)}{dx} = \frac{1}{12dx} (-3\phi(x_{i-1}) - 10\phi(x_i) + 18\phi(x_{i+1}) - 6\phi(x_{i+2}) + \phi(x_{i+3})) \quad (\text{Eq 3 - 2.2.11})$$

$$\frac{d\phi(x_{n-2})}{dx} = \frac{1}{12dx} (\phi(x_{n-4}) - 8\phi(x_{n-3}) + 0\phi(x_{n-2}) + 8\phi(x_{n-1}) - \phi(x_n)) \quad (\text{Eq 3 - 2.2.12})$$

$$\frac{d\phi(x_{n-1})}{dx} = \frac{1}{12dx} (-\phi(x_{n-4}) + 6\phi(x_{n-3}) - 18\phi(x_{n-2}) + 10\phi(x_{n-1}) + 3\phi(x_n)) \quad (\text{Eq 3 - 2.2.13})$$

$$\frac{d\phi(x_n)}{dx} = \frac{1}{12dx} (3\phi(x_{n-4}) - 16\phi(x_{n-3}) + 36\phi(x_{n-2}) - 48\phi(x_{n-1}) + 25\phi(x_n)) \quad (\text{Eq 3 - 2.2.14})$$

The performance of DSS020 is better than centered difference methods DSS002 and DSS004 on oscillation, and better than upwind difference methods DSS012 and DSS014 on numerical diffusion. However it still produces some oscillation. It has been found that if we can use a slope step function, the so called “ramp function”, to approximate the unit step for the boundary, the numerical results will agree with the analytical results better. That is also true in reality, in a realistic model, we would not expect propagating discontinuities; generally, there will be some physical phenomenon that will produce a finite rate of change, for example, molecular diffusion of temperature, mass, or momentum. In the implementation, we use $r(t) = \phi(0, t)$, which has a steep but finite slope

$$r(t) = \begin{cases} 0, & t < 0; \\ s \times t, & 0 \leq t < (1/s) \\ 1, & t > (1/s) \end{cases} \quad (\text{Eq 3 - 2.2.15})$$

Here slope can be taken $s=5, 10, 20$.

The exact solution of the convective equation corresponding to this boundary condition is:

$$\phi(x, t) = r\left(t - \frac{x}{v}\right) \quad (\text{Eq 3 - 2.2.16})$$

3 - 2.3 ODE integration

After space-discretizing the PDE, the problem has been converted to solve a system of ODEs. The system of ODEs generated by the spacing differencing of the convective equation is stiff. Therefore we need an implicit method to solve ODEs.

There are many implicit methods for integrating stiff ODEs. In our implementation, we

employed the semi-implicit extrapolation method (Press, Teukolsky, Vetterling 1995) for our ODE integration.

Semi-implicit Extrapolation Method

Complete information about the Semi-implicit Extrapolation Method can be found in (Press, Teukolsky, Vetterling 1995). We just excerpt some necessary part to make this chapter self-contained.

First look at the first order implicit method.

For the system of ODEs:

$$\dot{\phi} = f(\phi) \quad (\text{Eq 3 - 2.3.1})$$

implicit differencing gives

$$\phi^{n+1} = \phi^n + hf(\phi^{n+1}) \quad (\text{Eq 3 - 2.3.2})$$

Suppose we try linearizing the equations:

$$\phi^{n+1} = \phi^n + h \left[f(\phi^n) + \left. \frac{\partial f}{\partial \phi} \right|_{\phi^n} (\phi^{n+1} - \phi^n) \right] \quad (\text{Eq 3 - 2.3.3})$$

Here $\frac{\partial f}{\partial \phi}$ is the matrix of the partial derivatives of the right-hand side (the Jacobian matrix). Rearrange the equation into the form

$$\phi^{n+1} = \phi^n + h \left[I - h \frac{\partial f}{\partial \phi} \right]^{-1} f(\phi^n) \quad (\text{Eq 3 - 2.3.4})$$

If h is not too big, only one iteration of Newton's method may be accurate enough to solve (Eq 3 - 2.3.1) using (Eq 3 - 2.3.4). In other words, at each step we have to invert the matrix $I - h \frac{\partial f}{\partial \phi}$ to find ϕ^{n+1} . Solving implicit methods by linearization is called a "semi-implicit" method, so (Eq 3 - 2.3.4) is the semi-implicit Euler method.

The above method is only first-order accurate. Most problems will benefit from higher-order methods. Here we use the implementation of a semi-implicit extrapolation method due to Bader and Deuffhard (1983).

The starting point is an implicit form of the midpoint rule:

$$\phi^{n+1} - \phi^{n-1} = 2hf \left(\frac{\phi^{n+1} + \phi^{n-1}}{2} \right) \quad (\text{Eq 3 - 2.3.5})$$

Convert this equation into semi-implicit form by linearizing the right-hand side about $f(\phi^n)$; the result is the semi-implicit midpoint rule

$$\left[1 - h \frac{\partial f}{\partial \phi} \right] \phi^{n+1} = \left[1 + h \frac{\partial f}{\partial \phi} \right] \phi^{n-1} + 2h \left[f(\phi^n) - \frac{\partial f}{\partial \phi} \phi^n \right] \quad (\text{Eq 3 - 2.3.6})$$

It is used with a special first step, the semi-implicit Euler step (Eq 3 - 2.3.4) and a special "smoothing" last step in which the last ϕ is replaced by

$$\bar{\phi}^n = \frac{1}{2}(\phi^{n+1} - \phi^{n-1}) \quad (\text{Eq 3 - 2.3.7})$$

There are also other methods for stiff systems, such as the Rosenbrock methods, which is a generalization of the Runge-Kutta method, and the Gear's backward differentiation method. Here we chose the semi-implicit extrapolation method after comparing these methods.

3 - 2.4 The Computation Results

The experiments has been done based on four cases.

1. DSS012 method with unit step boundary condition.

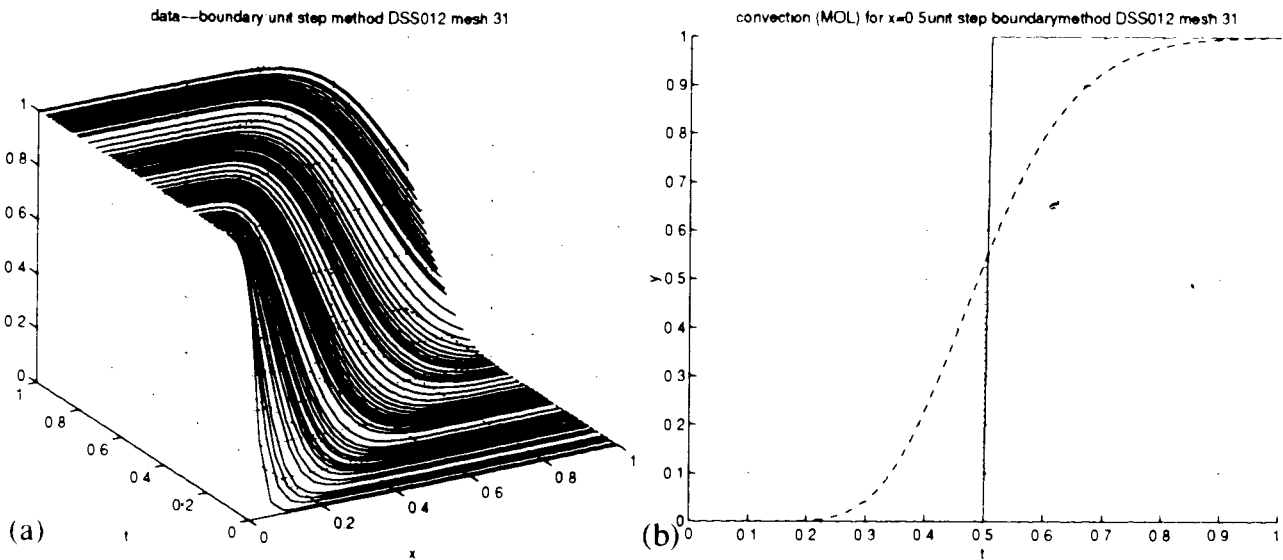


Figure 3 - 2.4.1 Computation result by NUMOL (DSS012) with unit step boundary condition, mesh=31 (a) 3D plot, (b) 2d plot at x=0.5 against exact solution (solid line)

2. DSS020 method with unit step boundary condition.

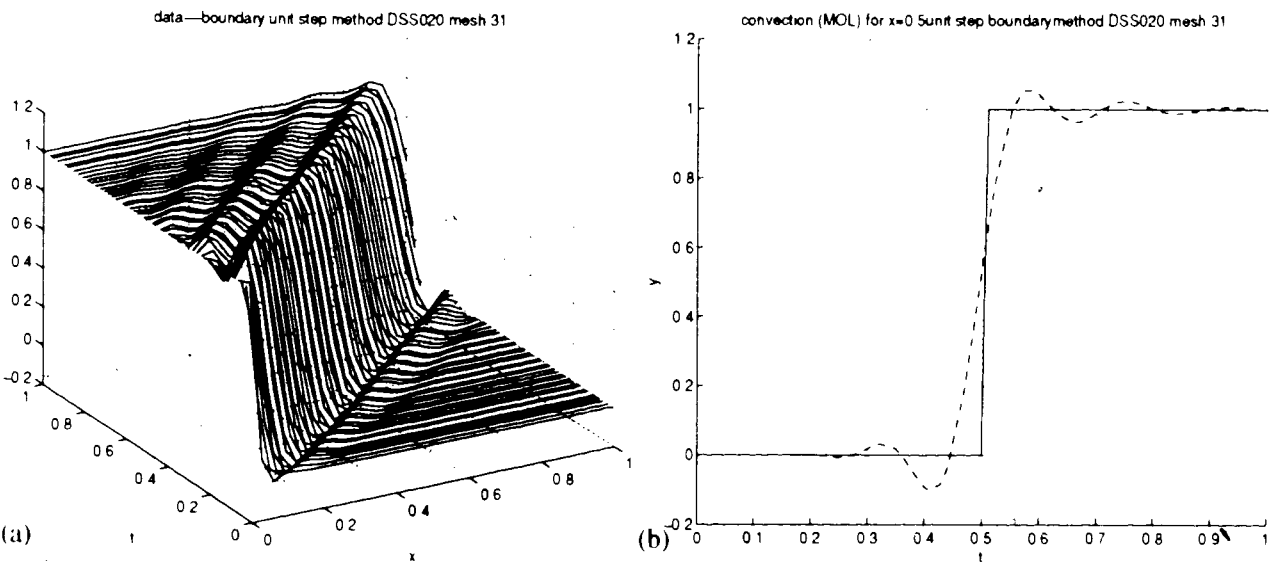


Figure 3 - 2.4.2 Computation result by NUMOL (DSS020) with unit step boundary condition mesh=31, (a) 3D plot, (b) 2D plot against exact solution (solid line)

3. DSS012 method with slope step boundary condition.

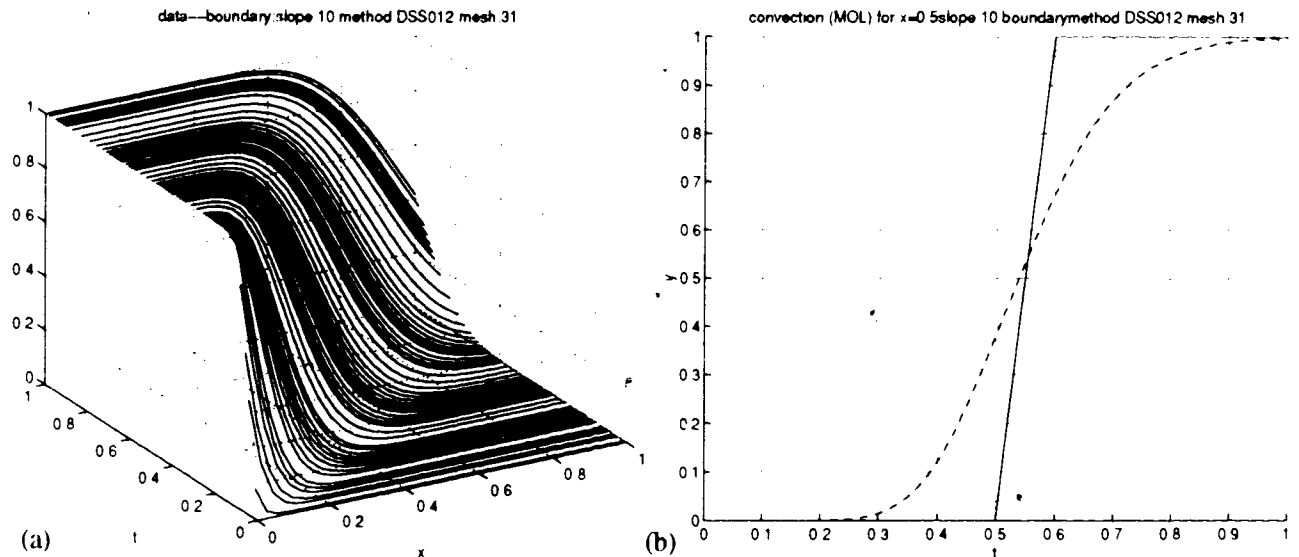


Figure 3 - 2.4.3 Computation result by NUMOL(DSS012) with steep slope (slope=10) step boundary condition mesh=31, (a) 3D plot, (b) 2D plot against exact solution (solid line)

4. DSS020 method with slope step boundary condition

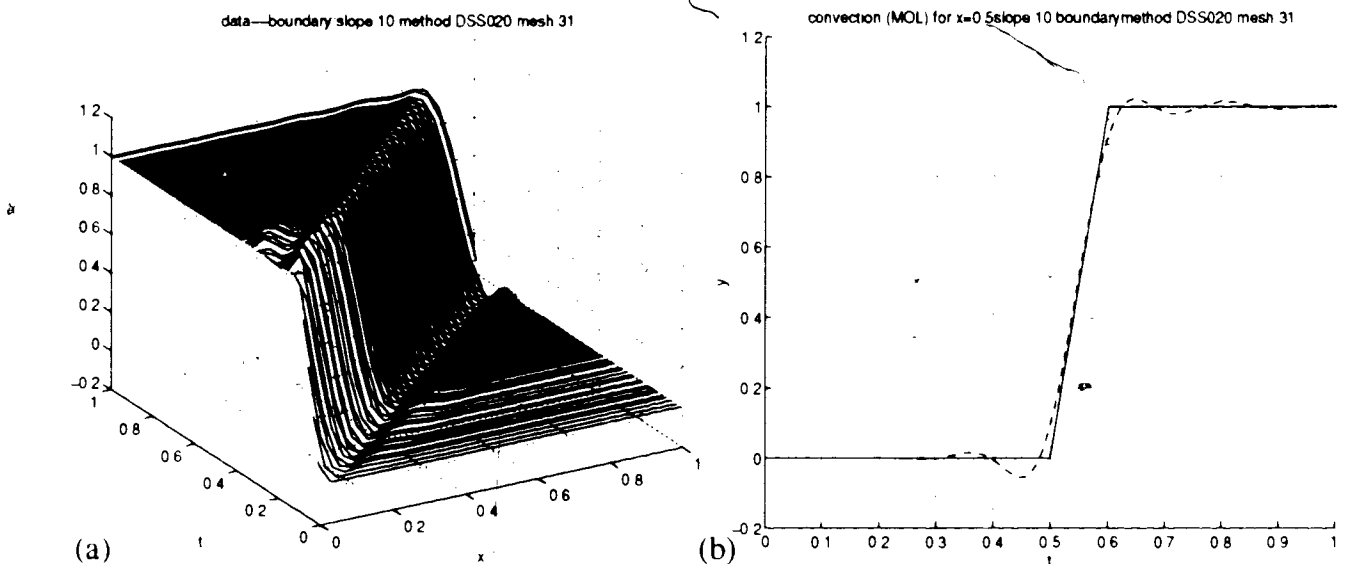


Figure 3 - 2.4.4 Computation result by NUMOL(DSS020) with steep slope (slope=10) step boundary condition mesh=31, (a) 3D plot, (b) 2D plot against exact solution (solid line).

The purpose for these experiments is to examine the performance of the method of DSS012 and DSS020, including the effect of boundary condition. To examine the performance, we observe the numerical diffusion and oscillation among the above different situations.

At first, we look at the results of case (1-4). The findings are listed here:

1. The method of DSS012 (first-order upwind) with a unit step or slope step boundary condition, suffers numerical diffusion but without oscillation (Figure 3 - 2.4.1 & Figure 3 - 2.4.3). As the mesh gets finer, the effect of numerical diffusion is getting less.
2. The method of DSS020 (fourth-order biased upwind) with unit step or slope step

boundary condition, gives much improvement on numerical diffusion, but suffers some oscillation (Figure 3 - 2.4.2 & Figure 3 - 2.4.4) The effect of oscillation and diffusion becomes less as the mesh gets finer (Figure 3 - 2.4.5).

3. The effect of oscillation and diffusion in the DSS020 method with slope step boundary condition becomes less than with unit step boundary condition, and the results have better agreement with the exact solution (Figure 3 - 2.4.4).

5. DSS020 method with slope step boundary condition with much finer mesh

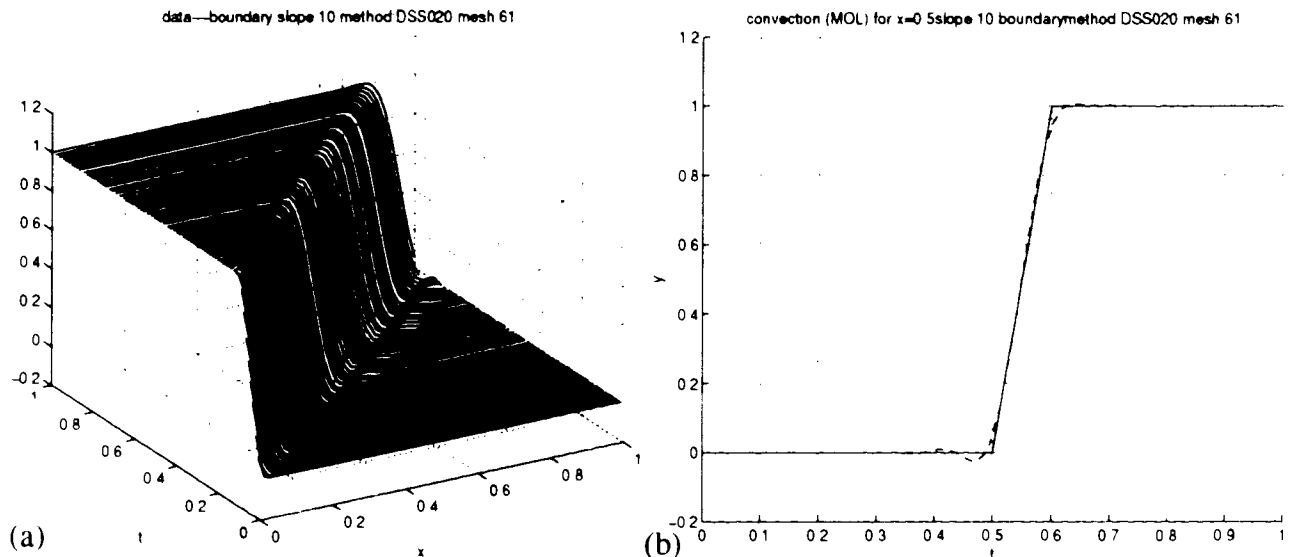


Figure 3 - 2.4.5 Computation result by NUMOL(DSS020) with steep slope (slope=10) step boundary condition mesh=61, (a) 3D plot, (b) 2D plot against exact solution (solid line).

3 - 2.5 Comments

The numerical method of lines, along with five-point upwind spacing differencing for the first-order space derivative, and semi-implicit extrapolation integration for the O.D.E, can be used to solve the convective equation with steep slope boundary condition, and acceptable results are achieved. The number of mesh points in the method of lines is not restrictive, but the computations are expensive when the mesh gets finer (case 5). The higher order upwind schemes indeed offers a solution to numerical diffusion.

3 - 3 Solving Convection Equation With the Orthogonal Collocation Method

3 - 3.1 A Brief Introduction to the Orthogonal Collocation Method

Another method embedded with higher order schemes is the orthogonal collocation method. In this section, we briefly introduce this method; for detail we refer to Finlayson (1972) or Appendix.

In a similar manner as NUMOL, by the orthogonal collocation method, the original P.D.E can be reduced to a system of O.D.Es, and then the resulting system of O.D.Es. can be solved by

standard subroutine.

Trial Function & Orthogonal Polynomials

Consider an unknown function $\phi(x, t)$, which satisfies partial differential equations and boundary condition. an Approximating function called a trial function can be chosen, such that the boundary conditions are satisfied. Depending on the problem, non-symmetric or symmetric trial functions should be used.

To solve the convection equation, a non-symmetric trial function is used, as defined by

$$\phi(x, \tau) = (1-x)\phi(0, \tau) + x\phi(1, \tau) + x(1-x) \sum_{i=1}^M a_i(\tau) P_{i-1}(x) \quad (\text{Eq 3 - 3.1.1})$$

where $a_i(\tau)$ are functions of time or constants and P_i are orthogonal polynomials defined by

$$\int_0^1 w(x) P_n(x) P_m(x) dx = 0, n = 0, 1, \dots, m-1 \quad (\text{Eq 3 - 3.1.2})$$

where $w(x)$ is weight function, such as $w(x) = 1$ in this work. The polynomials defined by (Eq 3 - 3.1.2) are shifted Legendre polynomials.

The collocation points x_1, \dots, x_{M+2} are the roots of $P_n(x) = 0$, and $x_1 = 0, x_{M+2} = 1$.

The resulting orthogonal polynomials and corresponding collocation points can be found in the literatures e.g. in Finlayson (1972) for order up to 7. However, in many situations, we need higher order polynomials. Indeed, these orthogonal polynomials can be computed in a simple manner by the Gram-Schmidt Orthogonalization procedure using inner product (Dorny 1975). The symbolic computation can be implemented in Maple V. The order of resulting polynomials can be as high as 20, and different polynomials can be easily obtained by changing the definition of inner product corresponding to the definition of each individual polynomial, limited only by the computer capability. For detail we refer to the Appendix.

Interior formulas based on ordinates

The Trial function can be formulated to an equivalent set of equations in terms of the unknown ordinates $\phi(x_1), \phi(x_2), \dots, \phi(x_N)$.

The corresponding set of collocation matrices can be written for the nonsymmetric trial functions (Eq 3 - 3.1.1) based on the shifted Legendre polynomials.

Rewrite (Eq 3 - 3.1.1) as,

$$\phi(x) = \sum_{j=1}^{M+2} d_j x^{j-1} \quad (\text{Eq 3 - 3.1.3})$$

Evaluate at collocation points, i.e.,

$$\phi(x_i) = \sum_{j=1}^{M+2} d_j x_i^{j-1} \quad (\text{Eq 3 - 3.1.4})$$

Take the first derivative and the second derivative of this expression and evaluate them at the collocation points:

$$\left. \frac{d\phi}{dx} \right|_{x_i} = \sum_{j=1}^{M+2} d_j \left. \frac{dx^{j-1}}{dx} \right|_{x_i} \quad (\text{Eq 3 - 3.1.5})$$

$$\left. \frac{d^2\phi}{dx^2} \right|_{x_i} = \sum_{j=1}^{M+2} d_j \left. \frac{d^2(x^{j-1})}{dx^2} \right|_{x_i} \quad (\text{Eq 3 - 3.1.6})$$

These can be written in matrix notation as

$$\phi = Qd, \quad Q_{ij} = x_i^{j-1} \quad (\text{Eq 3 - 3.1.7})$$

$$\left. \frac{d\phi}{dx} \right|_{x_i} = Cd, \quad C_{ij} = \left. \frac{dx^{j-1}}{dx} \right|_{x_i} = (j-1)x_i^{j-2} \quad (\text{Eq 3 - 3.1.8})$$

$$\left. \frac{d^2\phi}{dx^2} \right|_{x_i} = Dd, \quad D_{ij} = \left. \frac{d^2(x^{j-1})}{dx^2} \right|_{x_i} = (j-1)(j-2)x_i^{j-3} \quad (\text{Eq 3 - 3.1.9})$$

Since $d = Q^{-1}\phi$

$$\left. \frac{d\phi}{dx} \right|_{x_i} = Cd = CQ^{-1}\phi = A^x\phi, \quad (\text{Eq 3 - 3.1.10})$$

$$\left. \frac{d^2\phi}{dx^2} \right|_{x_i} = Dd = DQ^{-1}\phi = B^x\phi \quad (\text{Eq 3 - 3.1.11})$$

Matrices A^x, B^x are called collocation matrices.

Where,

$$[A_{ij}^x]^{(M+2)} = \begin{bmatrix} \left. \frac{dx^0}{dx} \right|_{x_1} & \dots & \left. \frac{dx^{M+1}}{dx} \right|_{x_1} \\ \vdots & \dots & \vdots \\ \left. \frac{dx^0}{dx} \right|_{x_{M+2}} & \dots & \left. \frac{dx^{M+1}}{dx} \right|_{x_{M+2}} \end{bmatrix} [Q]^{-1} \quad (\text{Eq 3 - 3.1.12})$$

$$[B_{ij}^x]^{(M+2)} = \begin{bmatrix} \nabla^2(x^0)|_{x_1} & \dots & \nabla^2(x^{M+1})|_{x_1} \\ \vdots & \dots & \vdots \\ \nabla^2(x^0)|_{x_{M+2}} & \dots & \nabla^2(x^{M+1})|_{x_{M+2}} \end{bmatrix} [Q]^{-1} \quad (\text{Eq 3 - 3.1.13})$$

$$[W_i^{(n)}] = \begin{bmatrix} 1 & \dots & 1 \\ \int_0^1 x^0 dx & \dots & \int_0^1 x^{M+1} dx \\ 0 & \dots & 0 \end{bmatrix} [Q]^{-1} \quad (\text{Eq 3 - 3.1.14})$$

$$[Q] = \begin{bmatrix} 1 & x_1^1 & \dots & x_1^{M+1} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{M+2}^1 & \dots & x_{M+2}^{2n} \end{bmatrix} \quad (\text{Eq 3 - 3.1.15})$$

The original P.D.E. can then be written in collocation form, which gives a system of O.D.E.s.

3 - 3.2 Solving Convection Equation with the Orthogonal Collocation Method

Rewrite the convection equation which is subject to initial and boundary conditions:

$$\frac{\partial \phi}{\partial t} = -v \frac{\partial \phi}{\partial x} \quad (\text{Eq 3 - 3.2.1})$$

$$\phi(0, t) = f(t), \quad \phi(x, 0) = g(x) \quad (\text{Eq 3 - 3.2.2})$$

where

$$f(t) = u(t) = \begin{cases} 0, & t < 0; \\ 1, & t \geq 0; \end{cases} \quad (\text{Eq 3 - 3.2.3})$$

$$\text{or } r(t) = \begin{cases} 0, & t < 0; \\ s \times t, & 0 \leq t < (1/s) \\ 1, & t > (1/s) \end{cases} \quad (\text{Eq 3 - 3.2.4})$$

The profile of ϕ is approximated by the following nonsymmetric trial function (Eq 3 - 3.1.1) based on the shifted Legendre polynomials:

$$\phi(x, \tau) = (1-x)\phi(0, \tau) + x\phi(1, \tau) + x(1-x) \sum_{i=1}^M a_i(\tau) P_{i-1}(x) \quad (\text{Eq 3 - 3.2.5})$$

where the polynomials are defined by:

$$\int_0^1 w(x) P_n(x) P_m(x) dx = 0, \quad n = 0, 1, \dots, m-1 \quad (\text{Eq 3 - 3.2.6})$$

where $w(x) = 1$.

Then, (Eq 3 - 3.2.1) is reduced to a system of O.D.Es by writing it in the collocation form:

$$\frac{d\phi(x_j)}{dt} = -v \sum_{i=1}^{m+2} A_{j,i}^x \phi(x_i), \quad j = 2, \dots, m+2 \quad (\text{Eq 3 - 3.2.7})$$

and $\phi(x_1, t) = f(t)$, or $\phi(x_1, t) = r(t)$.

3 - 3.3 The Computation Results

1. Orthogonal collocation with unit step boundary condition

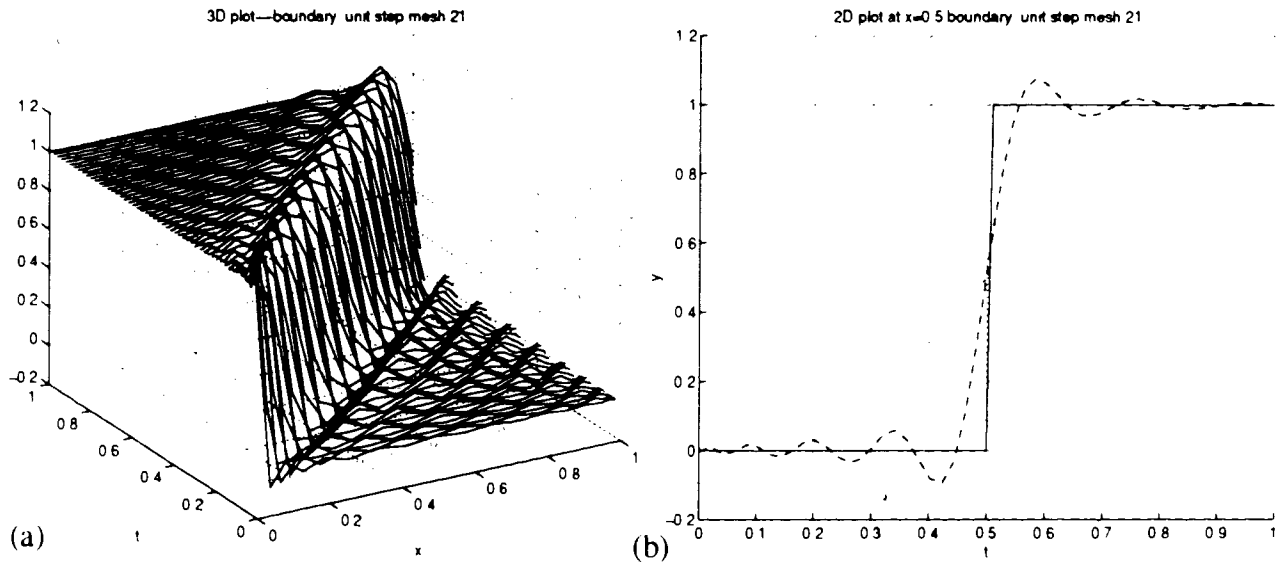


Figure 3 - 3.3.1 Computation result by orthogonal collocation method with unit step boundary condition, mesh=21, (a) 3D plot (b) 2D plot against exact solution (solid line).

2. Orthogonal collocation with slope boundary condition

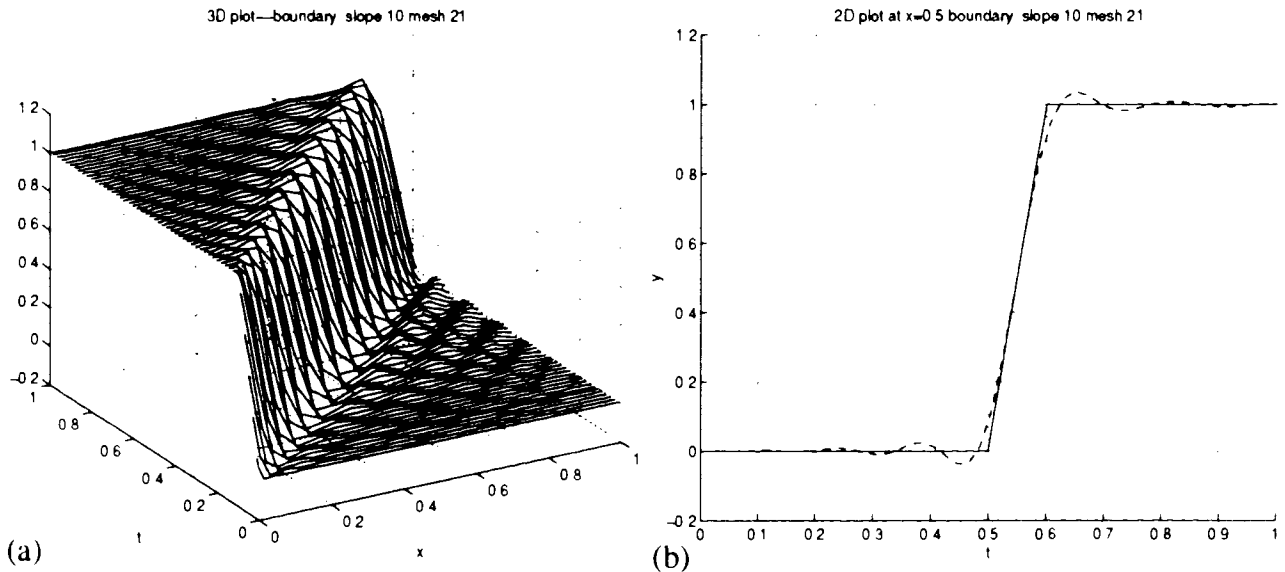


Figure 3 - 3.3.2 Orthogonal collocation method with slope (slope=10) boundary condition mesh=21 (a) 3D plot (b) 2D plot against exact solution (solid line)

3 - 3.4 Comments:

From the above Figure 3 - 3.3.1 & Figure 3 - 3.3.2 we can see that the orthogonal collocation method also has improved performance on numerical diffusion. With a unit step boundary condition, it suffers nonphysical oscillation; however, with a steep slope boundary condition, the effect of oscillation becomes less, and the numerical solution agrees with analytical solution well. Since the orthogonal polynomials can be computed to order 20 at present, so the mesh

number is restrict to no more than 22, which gives sufficiently accurate results.

3 - 4 Comparison between NUMOL and Orthogonal Collocation

Comparison between the numerical method of lines and orthogonal collocation can be done by observing the performance of numerical diffusion and oscillation when solving the convection equation. In this section, we are going to present the computation results based on the same mesh number, and compare the results and computation time. The upwind difference method used in NUMOL is DSS020 which we have concluded to be the best upwind scheme among the finite difference formulas for first spatial derivative. Also, in the computation with both methods, we allow the program to choose an integration step with loose restriction of 0.05 as an upper limit.

3 - 4.1 observation of performance on numerical diffusion and oscillation

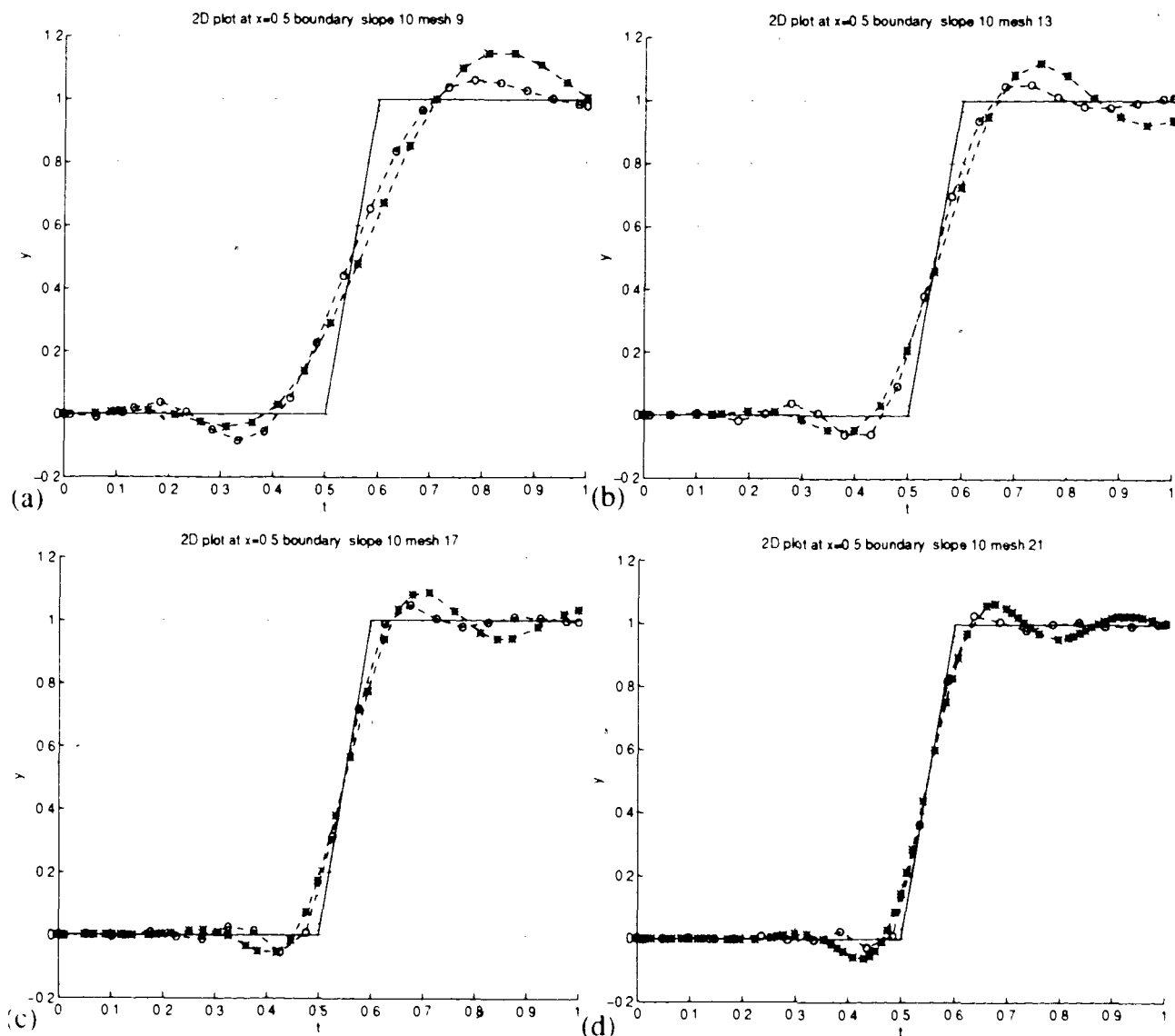


Figure 3 - 4.1.1 2D plot of profiles at $x=0.5$ with steep slope step 10, maxstep=0.05 (*---NUMOL, o---orthogonal collocation), (a)mesh=9, (b) mesh=13, (c)mesh=17 (d) mesh=21

3 - 4.2 Comparison of Computation Time

Table 3 - 4.2.1 Comparison of Computation Time

mesh numbers	numerical method of lines		orthogonal collocation	
	computation time (second)	integration steps	computation time (second)	integration steps
7	0.2	38	0.1	26
9	0.3	37	0.2	26
11	0.4	41	0.3	26
13	0.4	27	0.4	26
15	0.7	38	0.5	26
17	1.2	47	0.7	26
19	1.8	69	0.8	26
21	2.6	95	1.1	26

3 - 4.3 Conclusion for Chapter 3

In this chapter, we have used the numerical method of lines and the orthogonal collocation method to solve the convection equation. In NUMOL, the higher order upwinding scheme DSS020 is chosen for first-order spatial derivative approximation. And in orthogonal collocation, we use inherited global higher order collocation forms for space derivatives. Therefore, in both methods, higher order formulations have been embedded in the methods to handle the numerical diffusion. From Figure 3 - 4.1.1, it can be seen that the orthogonal collocation method has better performance on numerical diffusion and less oscillation for the same mesh number. Also from Table 3 - 4.2.1 the computation time for same number of mesh points, orthogonal collocation take less computation time, which is result of larger integration step allowable in orthogonal collocation method because more accurate formulation than in numerical method of lines. Therefore, orthogonal collocation has a better performance than method of lines for convection simulation for the same number of mesh points. The number of mesh points in method of lines is not restrict, but computation are more expensive when mesh get finer.

Chapter 4 Convective Diffusion Equation

4 - 1 Convective Diffusion Equation

Consider the extension of convection equation (Eq 3-1.1) to include axial diffusion:

$$\frac{\partial \phi}{\partial t} = -v \frac{\partial \phi}{\partial x} + D \frac{\partial^2 \phi}{\partial x^2} \quad (\text{Eq 4 - 1.1})$$

The above equation is a convective diffusion equation or parabolic-hyperbolic PDE; D is an axial diffusivity.

One initial condition is:

$$\phi(0, t) = f(t) \quad (\text{Eq 4 - 1.2})$$

Boundary conditions are:

$$\phi(x, 0) = g(x) \quad (\text{Eq 4 - 1.3})$$

$$\phi_x(L, t) = 0 \quad (\text{Eq 4 - 1.4})$$

Since (Eq 4 - 1.1) is second-order in x, it requires two boundary conditions.

(Eq 4 - 1.2) specifies only convection takes place at the exit of the tube.

Since (Eq 4 - 1.1) reduces to Fourier's second law for $v = 0$, and the convection equation for $D=0$, the numerical solution should have the properties we would expect (i.e., properties of the solutions for both Fourier's second law and the convection equation)

4 - 2 The NUMOL Solution of Convective Diffusion Equation

Compute a NUMOL solution to equations (Eq 4 - 1.1) to (Eq 4 - 1.4) for $f(t)=1$, $g(x)=0$, $v=L=1$. We will use DSS020 to calculate the convective derivative $\frac{\partial \phi}{\partial x}$ and DSS044 to calculate the diffusion derivative $\frac{\partial^2 \phi}{\partial x^2}$ in (Eq 4 - 1.1)

4 - 2.1 Space Discretization

In chapter 3, we described DSS020, which is the optimal choice of approximation to the first-order space derivative. Here, we only give the approximation form DSS044 to second-order space derivative.

In the implementation of NUMOL, two different boundary conditions are considered. They are: Dirichlet boundary (the value of boundary point is specified) or Neumann boundary (the derivative of boundary is specified).

The approximation to second-order space derivative:

with Dirichlet boundary at inlet,

$$\frac{d^2 \phi_1}{dx^2} = \frac{2}{4! dx^2} (45\phi(1) - 154\phi(2) + 214\phi(3) - 156\phi(4) + 61\phi(5) - 10\phi(6)) \quad (\text{Eq 4 - 2.1.1})$$

at outlet,

$$\frac{d^2\phi_n}{dx^2} = \frac{2}{4! dx^2} (45\phi(n) - 154\phi(n-1) + 214\phi(n-2) - 156\phi(n-3) + 61\phi(n-4) - 10\phi(n-5)) \quad (\text{Eq 4 - 2.1.2})$$

with Neumann boundary condition, at inlet,

$$\frac{d^2\phi_1}{dx^2} = \frac{2}{4! dx^2} \left(\left(-\frac{415}{6}\right)\phi(1) + 96\phi(2) - 36\phi(3) + \frac{32}{3}\phi(4) - \frac{3}{2}\phi(5) - 50\frac{d\phi_1}{dx} \right) \quad (\text{Eq 4 - 2.1.3})$$

at outlet,

$$\frac{d^2\phi_n}{dx^2} = \frac{2}{4! dx^2} \left(\left(-\frac{415}{6}\right)\phi(n) + 96\phi(n-1) - 36\phi(n-2) + \frac{32}{3}\phi(n-3) - \frac{3}{2}\phi(n-4) + 50\frac{d\phi_n}{dx} \right) \quad (\text{Eq 4 - 2.1.4})$$

for all interior points

$$\frac{d^2\phi_2}{dx^2} = \frac{2}{4! dx^2} (10\phi(1) - 15\phi(2) - 4\phi(3) + 14\phi(4) - 6\phi(5) + \phi(6)) \quad (\text{Eq 4 - 2.1.5})$$

$$\frac{d^2u_i}{dx^2} = \frac{2}{4! dx^2} (-1.0\phi(i-2) + 16\phi(i-1) - 30\phi(i) + 16\phi(i+1) - 1.0\phi(i+2)) \quad (\text{Eq 4 - 2.1.6})$$

$$\frac{d^2u_{n-1}}{dx^2} = \frac{2}{4! dx^2} (10\phi(n) - 15\phi(n-1) - 4\phi(n-2) + 14\phi(n-3) - 6\phi(n-4) + \phi(n-5)) \quad (\text{Eq 4 - 2.1.7})$$

4 - 2.2 ODE Integration

After space discretization for the first-order derivative and second-order derivative, we obtain a system of ODEs, we used the same ODE integrator as used in chapter 3. We have known that the ODE integrator is step-variable at this point.

4 - 2.3 The Computation Result

Tests have been conducted for various values of diffusion coefficient. Maximum integration step size is 0.05.

1. Pure convection case by setting $v=1.0, D=0$

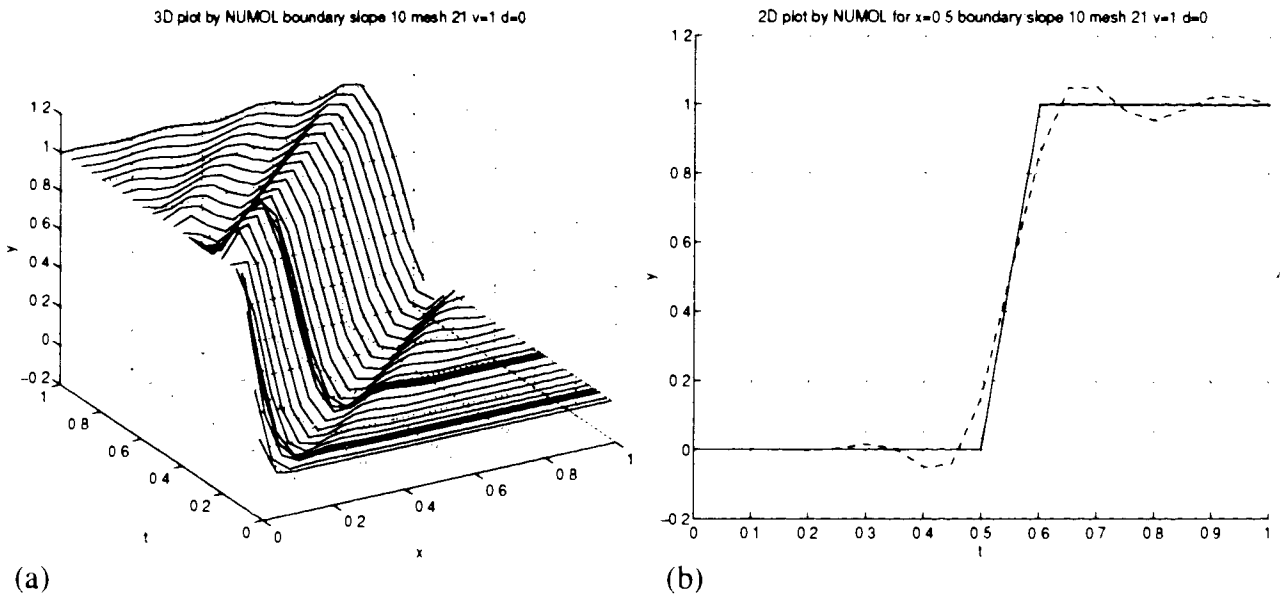


Figure 4 - 2.3.1 Simulation for convective diffusion by NUMOL with steep slope (=10) boundary condition, mesh=21, pure convection case (a) 3D plot (b) 2D profile at $x=0.5$ against exact solution (solid line)

2. Convective Diffusion with $v=1, D=0.01$

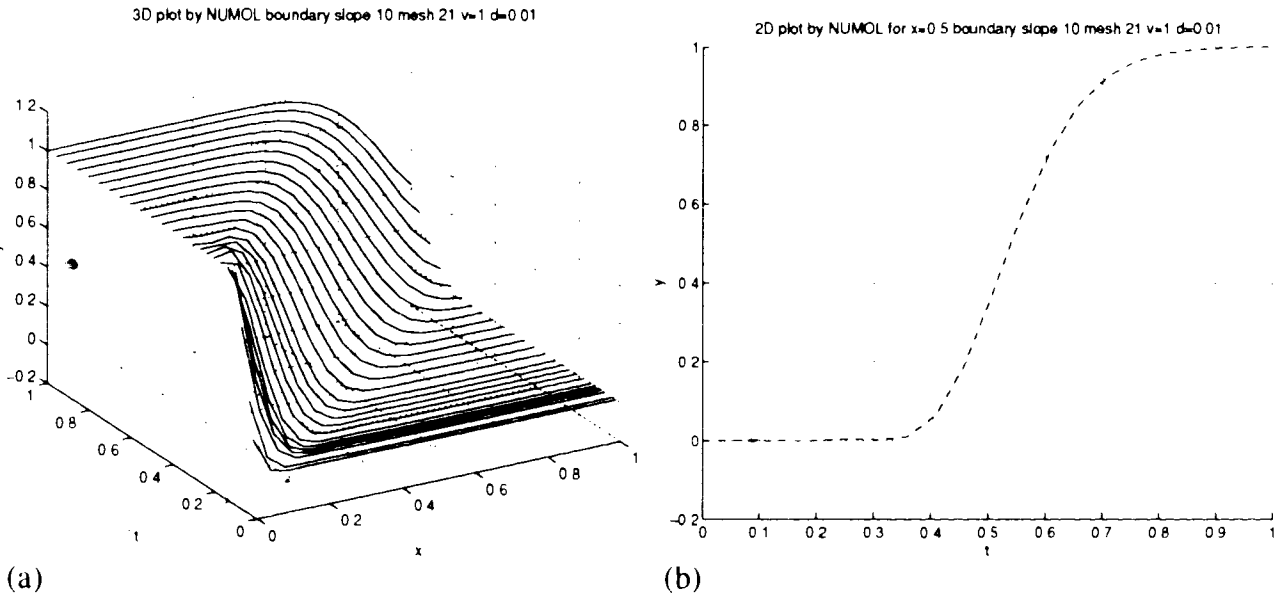


Figure 4 - 2.3.2 Simulation for convective diffusion by NUMOL with steep slope (=10) boundary condition, mesh=21, convective diffusion case (a) 3D plot (b) 2D profile at $x=0.5$

3. Pure diffusion case with $v=0, D=1.0$

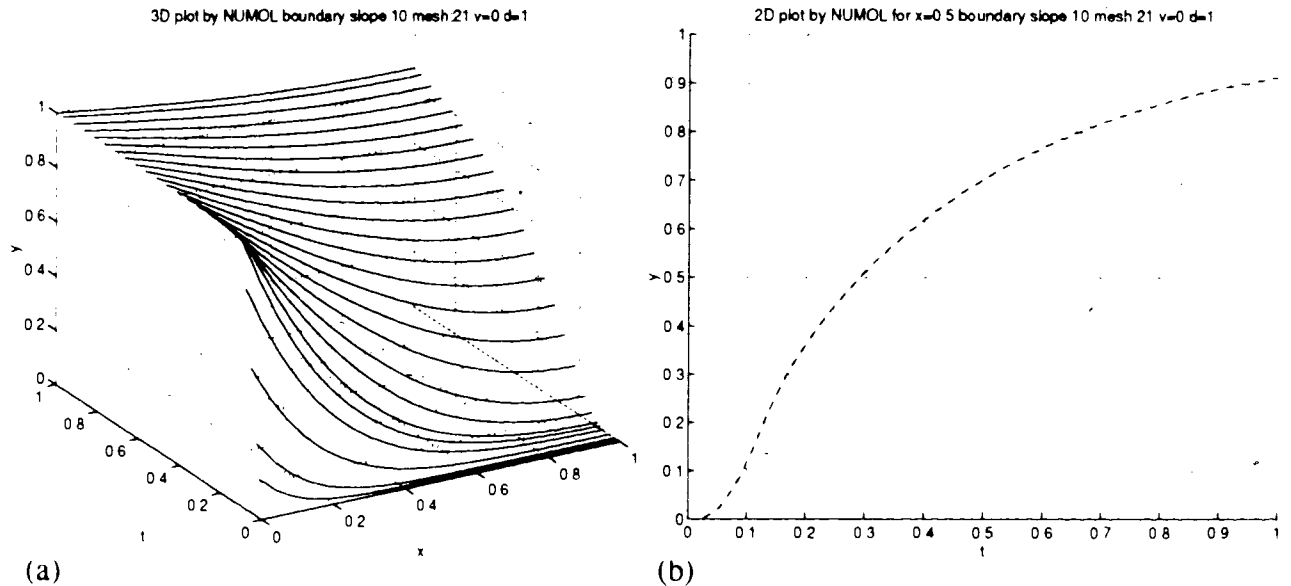


Figure 4 - 2.3.3 Simulation for convective diffusion by NUMOL with steep slope (=10) boundary condition, mesh=21, pure diffusion case (a) 3D plot (b) 2D profile at $x=0.5$

4 - 2.4 Comments

It is evident that the solution is dispersed as the physical diffusion coefficient increased, and with $v=1, D=0$, the solution resembles that of convective equation; with $v=0, D=1$, the solution resembles the diffusion equation. It is also noticeable that as D increases, the oscillation of solution disappears but with strong physical diffusion effect instead.

From the experiments we have conducted so far, we can see the method of lines can handle both the convective and convective-diffusion equation.

4 - 3 Solving Convective Diffusion Equation with the Orthogonal Collocation Method.

4 - 3.1 Collocation Form

Again as in last chapter, we use the nonsymmetric trial function based on the shifted Legendre polynomials. Then collocation form of (Eq 4 - 1.1) can be:

$$\frac{d\phi}{dt}(x_j) = -v \sum_{i=1}^{m+2} A_{j,i}^x \phi(x_i) + D \sum_{j=1}^{m+2} B_{j,i}^x \phi(x_i), j = 2, \dots, m+2 \quad (\text{Eq 4 - 3.1.1})$$

The collocation form of boundary condition at outlet is:

$$\sum_{i=1}^{m+2} A_{m+2,i}^x \phi(x_i) = 0 \quad (\text{Eq 4 - 3.1.2})$$

(Eq 4 - 3.1.2) can be used to deduce the form of boundary point:

$$\phi(x_{m+2}) = \sum_{i=1}^{m+2} A_{m+2,i}^x \phi(x_i) / A_{m+2,m+2}^x \quad (\text{Eq 4 - 3.1.3})$$

Substitute the above equation into (Eq 4 - 3.1.1), we have:

$$\begin{aligned} \frac{d\phi}{dt}(x_j) &= -v \sum_{i=1}^{m+1} A_{j,i}^x \phi(x_i) + D \sum_{i=1}^{m+1} B_{j,i}^x \phi(x_i) \\ &- v \frac{A_{j,m+2}^x}{A_{m+2,m+2}^x} \sum_{i=1}^{m+1} A_{m+2,i}^x \phi(x_i) + D \frac{B_{j,m+2}^x}{A_{m+2,m+2}^x} \sum_{i=1}^{m+1} B_{m+2,i}^x \phi(x_i) \end{aligned} \quad (\text{Eq 4 - 3.1.4})$$

Thus, (Eq 4 - 1.1) has been reduced to a system of O.D.Es by the orthogonal collocation method, the resulting O.D.E's will be solved by standard O.D.E solver.

4 - 3.2 The Computation Result

Tests has been conducted based on various value of physical diffusion by orthogonal collocation method, maximum integration step is 0.05:

1. Pure convection case with $v=1.0, D=0$

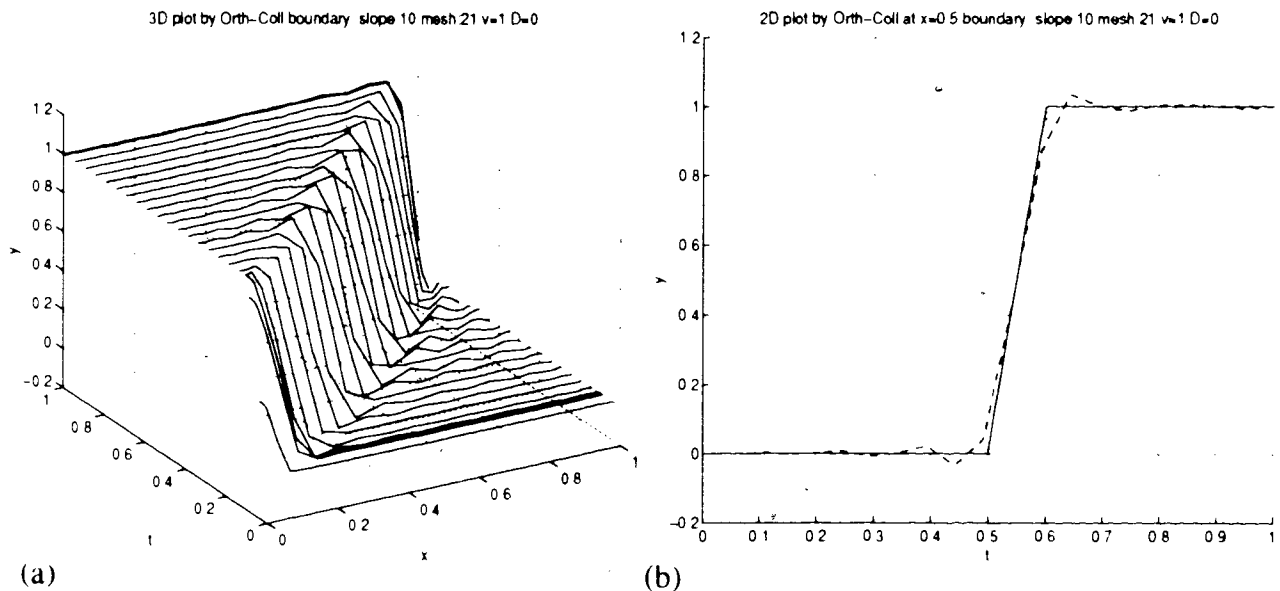


Figure 4 - 3.2.1 Simulation for convective diffusion by the orthogonal collocation with steep slope (=10) boundary condition, mesh=21, pure convection case (a) 3D plot (b) 2D profile at $x=0.5$ against exact solution (solid line)

2. Convective Diffusion with $v=1, D=0.01$

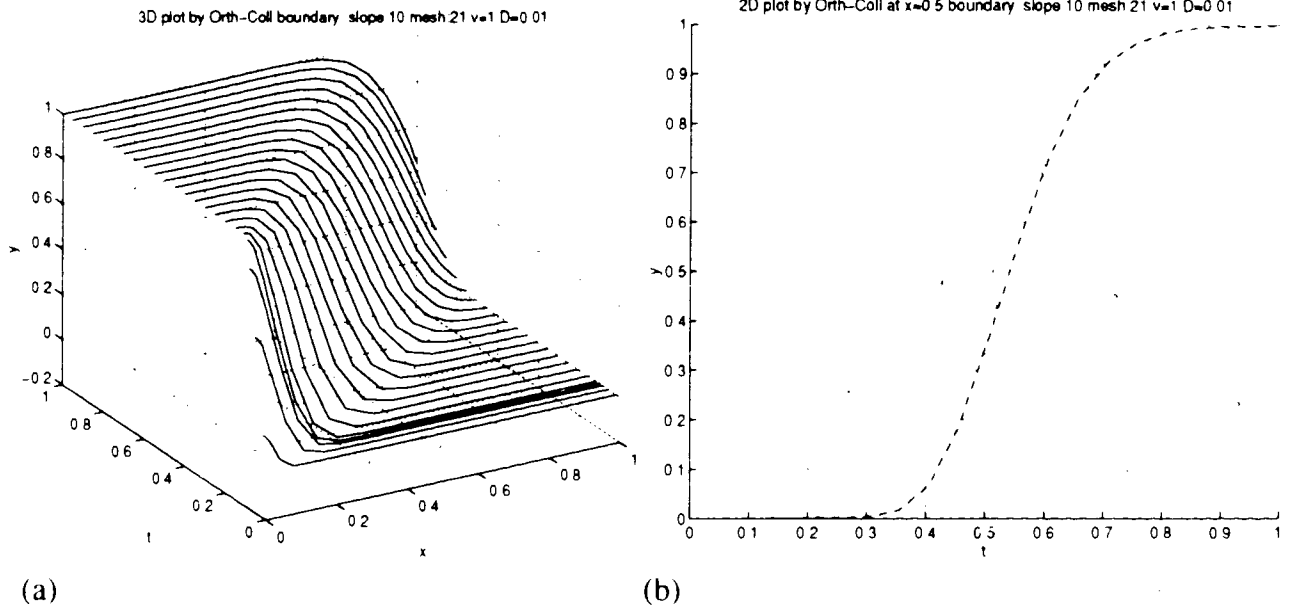


Figure 4 - 3.2.2 Simulation for convective diffusion by orthogonal collocation with steep slope ($=10$) boundary condition, mesh=21, convective diffusion case (a) 3D plot (b) 2D profile at $x=0.5$

3. Pure diffusion case with $v=0, D=1.0$

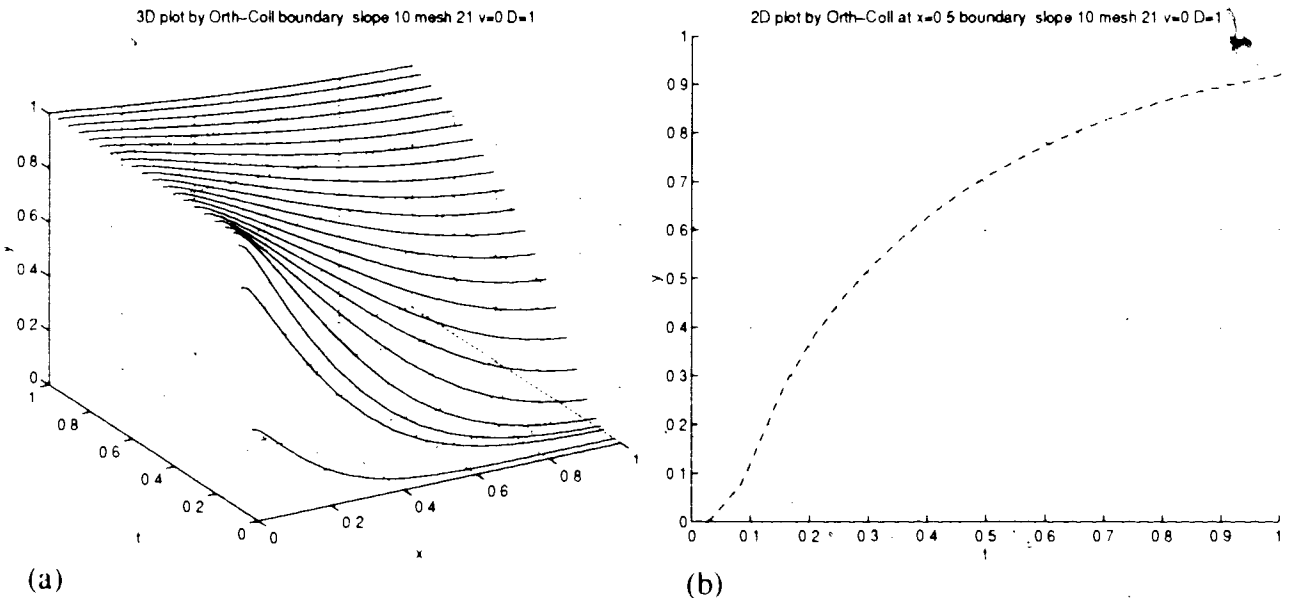


Figure 4 - 3.2.3 Simulation for convective diffusion by orthogonal collocation with steep slope ($=10$) boundary condition, mesh=21, pure diffusion case (a) 3D plot (b) 2D profile at $x=0.5$

4 - 3.3 Comments

Similar comments can be made for the results by the orthogonal collocation method: the solution is dispersed as the physical diffusion coefficient increased, and with $v=1, D=0$, the solution resembles that of convective equation; with $v=0, D=1$, the solution resembles the diffusion equation. It is also noticeable that as D increases, the oscillation of solution disappears

but with strong physical diffusion effect instead. Generally, the result by the orthogonal collocation method is more accurate than method of lines at same mesh number.

4 - 4 Comparison of the Results and Conclusion

A comparison between results by numerical method of lines and orthogonal collocation method, based on the simulation of convective diffusion equation with various value of diffusion parameter.

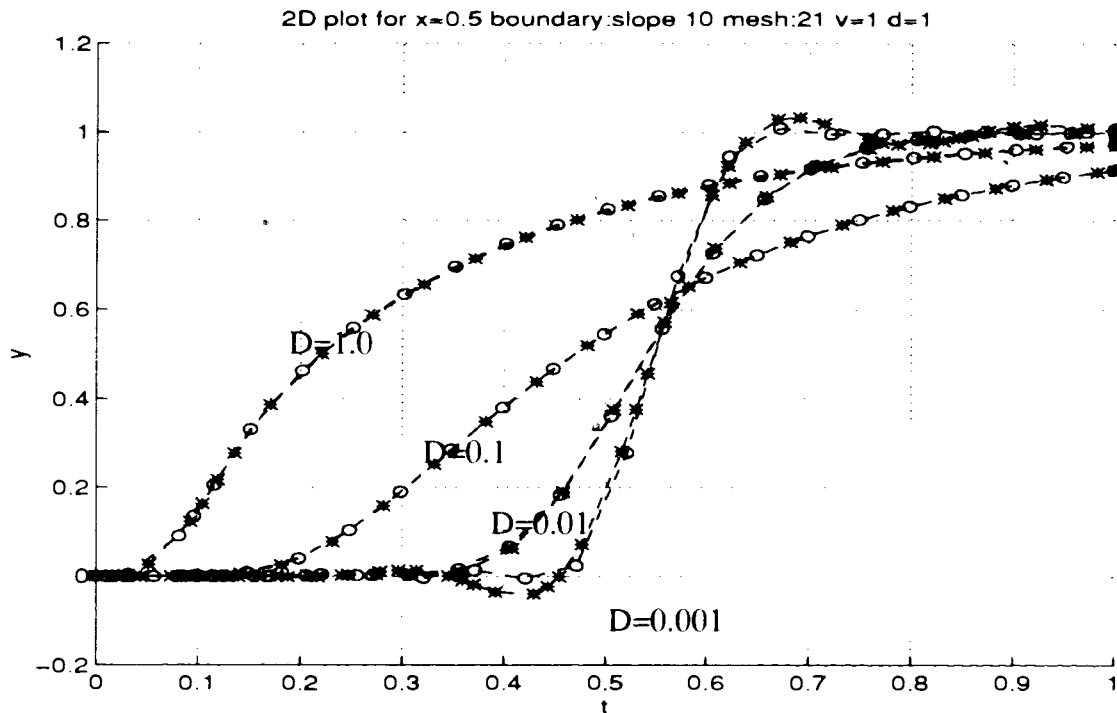


Figure 4 - 4.1 Comparison between NUMOL and orthogonal collocation, mesh=11. (*--NUMOL results, o--orthogonal collocation result).

We can see from the above Figure 4 - 4.1, that the orthogonal collocation can handle convective diffusion equation which has small physical diffusion better than the numerical method of lines, with less numerical diffusion and nonphysical oscillation ($D=0.001$). With increasing of physical diffusion, the results of two methods agree very well. Since in PSA modelling, we are interested in a method which can handle strong convection, and a self-sharpening curve, therefore, as the conclusion of this chapter, we are in favor of the orthogonal collocation method.

Chapter 5 Simulation of a Fixed-Bed Adsorption Column

Ruthven (Raghavan & Ruthven 1983) presents the orthogonal collocation method solution to a typical problem of adsorption in which a fixed-bed adsorption column is described. The problem presents a rather difficult mathematical model to solve numerically, and more important it has an analytical solution (Rasmuson and Neretnieks 1980), therefore, we choose it as a basis for the comparison of NUMOL and orthogonal collocation in order to give us an insight into the performances of the two numerical methods.

5 - 1 Description of the Fixed-Bed Adsorption column

We consider an isothermal adsorption column, packed with porous spherical particles, which is subjected, at time zero, to a small step change in the concentration of an adsorbable species in the carrier. The flow pattern is described by the axial dispersed plug-flow model, and both external and internal diffusional resistances to mass transfer are included.

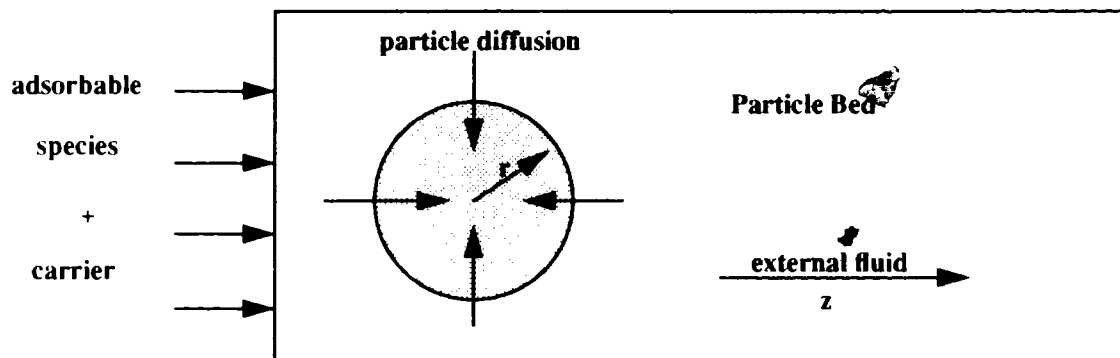


Figure 5 - 1.1 diagram of adsorption bed

5 - 2 Mathematical Model:

The system may be described by the following set of equations.

5 - 2.1 Dimensional Form:

Particle Diffusion:

$$\frac{\partial q}{\partial t} = D \left(\frac{\partial^2 q}{\partial r^2} + \frac{2}{r} \frac{\partial q}{\partial r} \right) \quad (\text{Eq 5 - 2.1.1})$$

Initial and boundary Conditions:

$$q(r, t) \Big|_{t=0} = 0 \quad (\text{Eq 5 - 2.1.2})$$

$$\frac{\partial q}{\partial r} \Big|_{r=0} = 0 \quad (\text{Eq 5 - 2.1.3})$$

$$D \frac{\partial q}{\partial r} \Big|_{r=R} = k \left[c(z, t) - \frac{q|_{r=R}}{K} \right] \quad (\text{Eq 5 - 2.1.4})$$

External fluid:

$$-D_L \frac{\partial^2 C}{\partial z^2} + v \frac{\partial C}{\partial z} + \frac{\partial C}{\partial t} = -\frac{(1-\epsilon)}{\epsilon} \left(\frac{3k}{r} (C(z,t) - \frac{q|_{r=R}}{K}) \right) \quad (\text{Eq 5 - 2.1.5})$$

Boundary conditions:

$$D_L \frac{\partial C}{\partial z} \Big|_{z=0} = -v(C|_{z=0} - C|_{z=0'}) \quad (\text{Eq 5 - 2.1.6})$$

$$\frac{\partial C}{\partial z} \Big|_{z=L} = 0 \quad (\text{Eq 5 - 2.1.7})$$

5 - 2.2 Dimensionless Form:

In our numerical simulation, we use the dimensionless forms of equation (Eq 5 - 2.1.1) - (Eq 5 - 2.1.7), to compute the solution. The dimensionless forms of above equations can be written:

Particle Diffusion:

$$\frac{\partial Q}{\partial \tau} = \nabla^2 Q = \frac{\partial^2 Q}{\partial \eta^2} + \frac{2}{\eta} \frac{\partial Q}{\partial \eta} \quad (\text{Eq 5 - 2.2.1})$$

Initial and boundary Conditions:

$$Q(\eta, \tau) \Big|_{\tau=0} = 0 \quad (\text{Eq 5 - 2.2.2})$$

$$\frac{\partial Q}{\partial \eta} \Big|_{\eta=0} = 0 \quad (\text{Eq 5 - 2.2.3})$$

$$\frac{1}{K} \frac{\partial Q}{\partial \eta} \Big|_{\eta=1} = \xi \left[U - \frac{Q|_{\eta=1}}{K} \right] \quad (\text{Eq 5 - 2.2.4})$$

External fluid:

$$\frac{\partial U}{\partial \tau} = \frac{1}{Pe} \psi \theta \frac{\partial^2 U}{\partial x^2} - \psi \theta \frac{\partial U}{\partial x} - 3\psi \xi \left(U - \frac{Q|_{\eta=1}}{K} \right) \quad (\text{Eq 5 - 2.2.5})$$

Boundary conditions:

$$\frac{\partial U}{\partial x} \Big|_{x=0} = -Pe (U|_{x=0} - U|_{x=0'}) \quad (\text{Eq 5 - 2.2.6})$$

$$\frac{\partial U}{\partial x} \Big|_{x=1} = 0 \quad (\text{Eq 5 - 2.2.7})$$

Table 5 - 2.2.1 Parameters in dimensionless form:

Parameter	Definition
Peclet number:	$Pe = (Lv)/D_L$
distribution ratio:	$\psi = K \left(\frac{1-\epsilon}{\epsilon} \right)$
bed length parameter:	$\theta = \frac{vR^2\epsilon}{LDK(1-\epsilon)}$
film resistance parameter:	$\xi = kR/DK$

Table 5 - 2.2.2 Dimensionless Variables

Dimensionless variables	Definition
Q : dimensionless internal concentration in particles	$Q = q/C_0$
η : dimensionless radial distance in particle	$\eta = r/R$
U : dimensionless fluid-phase concentration	$U = C/C_0$
x : dimensionless axial distance	$x = z/L$
τ : dimensionless contact time	$\tau = (Dt)/R^2$

Since (Eq 5 - 2.2.1) and (Eq 5 - 2.2.5) are coupled by (Eq 5 - 2.2.4), we need to solve these equations simultaneously. Also, (Eq 5 - 2.2.1) implies that the particles in the bed are being treated as a cylinder along the length of the bed, therefore, the internal concentration in a particle can be affected not only by (Eq 5 - 2.2.1), but also by the fluid-phase concentration at its axial position; meanwhile, the external fluid concentration at an axial position can be computed by both (Eq 5 - 2.2.5) and the surface concentration of the particle at that axial position. We will solve this problem with both NUMOL and the orthogonal collocation method and compare the results against exact solution (Rasmuson and Neretnieks 1980).

5 - 2.3 Space Discretization of the Computational Domain

In either of the methods, we need first to space discretize the computational domain Figure 5 - 2.3.1, the mesh points in external fluid domain and internal fluid domain can be finite difference mesh points or collocation points.

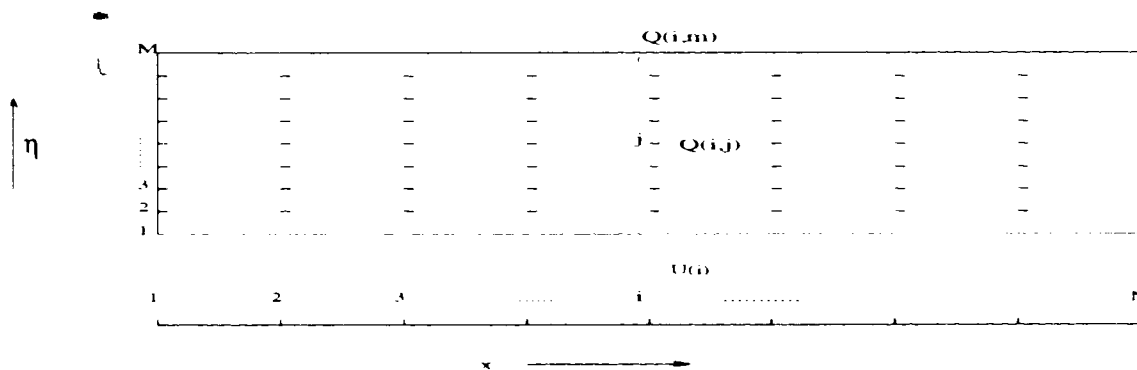


Figure 5 - 2.3.1 Space discretizing of the computational region

Since we treat the collection of spherical particles in the bed as a cylinder along the length of the bed, the computational region is not identical to the physical region. To compute the external fluid-phase concentration, we not only need to know the concentration at axial positions, but also need the internal concentration at radial positions. For example, to compute $U(i)$, we need the information of $U(1), U(2), \dots, U(N)$ as well as $Q(i, m)$; to compute $Q(i, m)$, we need to know the

information of $U(i)$, as well as $Q(i,1), Q(i,2), \dots, Q(i,m)$. Although we treat the computational region as a two dimensional region, actually it is simpler than that, since $Q(i,j)$ has no direct relation with $Q(i-1,j)$ or $Q(i+1,j)$. Therefore, we can arrange variables of the computational region into a computational matrix in the way as shown in Table 5 - 2.3.1

Table 5 - 2.3.1 relation between Q,U and Y

y[1].....	U[1]	dydt[1].....	dUdt[1]
:		:	
y[n].....	U[n]	dydt[n].....	dUdt[n]
y[n+1].....	Q[1,1]	dydt[n+1].....	dQdt[1,1]
:		:	
y[n+m].....	Q[1,m]	dydt[n+m].....	dQdt[1,m]
y[n+m+1].....	Q[2,1]	dydt[n+m+1].....	dQdt[2,1]
:		:	
y[n+m+m].....	Q[2,m]	dydt[n+m].....	dQdt[1,m]
:		:	
y[n+(i-1)*m+j].....	Q[i,j]	dydt[n+(i-1)*m+j].....	dQdt[i,j]
:		:	
y[n+n*m].....	Q[n,m]	dydt[n+n*m].....	dQdt[n,m]

The advantage of arranging these variables in such a way is that the derivative subroutines and Jacobian subroutine are easy to implement and to plug into the ODE solver, and the variables can be solved simultaneously. This arrangement is applicable to both methods.

5 - 3 Simulation By the Numerical Method of Lines(NUMOL)

Generally, by method of lines, we replace first space derivatives and second space derivatives in (Eq 5 - 2.2.1) to (Eq 5 - 2.2.7) by finite difference formulas, then the set of PDE's is converted into a system of ODEs in time; by standard ODE integrator, we integrate the system of ODEs and solve the problem.

5 - 3.1 Space differentiation:

First, for (Eq 5 - 2.2.5)

$$\frac{\partial U}{\partial \tau} = \frac{1}{Pe} \psi \theta \frac{\partial^2 U}{\partial x^2} - \psi \theta \frac{\partial U}{\partial x} - 3\psi \xi \left(U - \frac{Q|_{\eta=1}}{K} \right) \quad (\text{Eq 5 - 3.1.1})$$

This equation is similar to convective diffusion equation except that it includes the adsorption term, which relate to internal concentration at the surface of particles. After space discretization, for the typical point i along the bed, the above equation becomes:

$$\frac{dU}{d\tau} [i] = \frac{1}{Pe} \psi \theta \frac{d^2 U}{dx^2} [i] - \psi \theta \frac{dU}{dx} [i] - 3\psi \xi \left\{ U [i] - \frac{Q [i, m]}{K} \right\} \quad (\text{Eq 5 - 3.1.2})$$

In chapter 4, we have concluded, for convective-diffusional differential equation, the first space derivative can be replaced by DSS020 (fourth order upwind differencing), and the second space derivative can be replaced by DSS044 (fourth order centered differencing).

In our problem, the boundary condition are known for $\frac{dU}{dx} [1]$ and $\frac{dU}{dx} [n]$

$$\frac{dU}{dx} [1] = \left. \frac{\partial U}{\partial x} \right|_{x=0} = (-Pe \{ U|_{x=0^+} - U|_{x=0^-} \}) = -Pe (1.0 - U [1]) \quad (\text{Eq 5 - 3.1.3})$$

$$\frac{dU}{dx} [n] = \left. \frac{\partial U}{\partial x} \right|_{x=1} = 0 \quad (\text{Eq 5 - 3.1.4})$$

which can be used to substitute in DSS044 for solving $\frac{d^2 U}{dx^2} [1]$ and $\frac{d^2 U}{dx^2} [n]$ (see section 4-2.1)

Second, we study the (Eq 5 - 2.2.1)

$$\frac{\partial Q}{\partial \tau} = \frac{\partial^2 Q}{\partial \tau^2} + \frac{2}{\eta} \frac{\partial Q}{\partial \tau} \quad (\text{Eq 5 - 3.1.5})$$

To study the particles at point i along the bed, and we subdivide the radial domain into M points. For $Q(i, j)$, Eq. 5-2.1.5 can then be written as:

$$\frac{dQ}{dt} [i, j] = \frac{d^2 Q}{d\eta^2} [i, j] + \frac{2}{\eta_j} \frac{dQ}{d\eta} [i, j] \quad (\text{Eq 5 - 3.1.6})$$

where,

$$\eta_j = (j - 1) d\eta, \quad d\eta = \frac{1.0}{(m - 1)} \quad (\text{Eq 5 - 3.1.7})$$

The boundary conditions are:

$$\frac{dQ}{d\eta} [i, 1] = \left. \frac{\partial Q}{\partial \eta} \right|_{\eta=0} = 0 \quad (\text{Eq 5 - 3.1.8})$$

$$\frac{dQ}{d\eta} [i, m] = \left. \frac{\partial Q}{\partial \eta} \right|_{\eta=1} = K\xi \left\{ U [i] - \frac{Q [i, m]}{K} \right\} \quad (\text{Eq 5 - 3.1.9})$$

$$\frac{\partial q}{\partial \eta} [i, m] = K\xi \left\{ U [i] - \frac{Q [i, m]}{K} \right\} \quad (\text{Eq 5 - 3.1.10})$$

Special care is needed for $\frac{dQ}{dt} [i, 1]$

$$\lim_{\eta \rightarrow 0} \frac{1}{\eta} Q_{\eta} = \lim_{r \rightarrow 0} \frac{1}{r} Q_{r\eta} = Q_{r\eta} \quad (\text{Eq 5 - 3.1.11})$$

$$\frac{dQ}{dt} [i, 1] = \frac{d^2 Q}{d\eta^2} [i, 1] + \frac{2}{\eta_1} \frac{dQ}{d\eta} [i, 1] = \frac{d^2 Q}{d\eta^2} [i, 1] + 2 \frac{d^2 Q}{d\eta^2} [i, 1] = 3.0 \frac{d^2 Q}{d\eta^2} [i, 1] \quad (\text{Eq 5 - 3.1.12})$$

Either combination of DSS004 and DSS044 (fourth-order accuracy), or DSS002 and DSS042 (second-order accuracy) can be used for space differencing for the first and second space derivatives respectively.

DSS004, fourth-order centered space differencing, can be used for the first space derivative $\frac{dQ}{d\eta}$. The scheme is following.

$$dx = \frac{x_u - x_l}{(m-1)}, \quad r4fdx = \frac{1.0}{24dx} \quad (\text{Eq 5 - 3.1.13})$$

$$\frac{d\phi}{dx} [1] = r4fdx (-50\phi [1] + 96\phi [2] - 72\phi [3] + 32\phi [4] - 6\phi [5]) \quad (\text{Eq 5 - 3.1.14})$$

$$\frac{d\phi}{dx} [2] = r4fdx (-6\phi [1] - 20\phi [2] + 36\phi [3] - 12\phi [4] + 2\phi [5]) \quad (\text{Eq 5 - 3.1.15})$$

$$\frac{d\phi}{dx} [i] = r4fdx (2\phi [i-2] - 16\phi [i-1] + 0\phi [i] + 16\phi [i+1] - 2\phi [i+2]) \quad (\text{Eq 5 - 3.1.16})$$

$$\frac{d\phi}{dx} [m-1] = r4fdx (-2\phi [m-4] + 12\phi [m-3] - 36\phi [m-2] + 20\phi [m-1] + 6\phi [m]) \quad (\text{Eq 5 - 3.1.17})$$

$$\frac{d\phi}{dx} [m] = r4fdx (6\phi [m-4] - 32\phi [m-3] + 72\phi [m-2] - 96\phi [i, m-1] + 50\phi [m]) \quad (\text{Eq 5 - 3.1.18})$$

DSS044 is fourth order space differencing for the second space derivative. (see section 4-2.1) and boundary conditions $\frac{dQ}{d\eta} [i, 1]$ and $\frac{dQ}{d\eta} [i, m]$ are known.

Also, DSS002 second-order centered space differencing for the first space derivative, and DSS042 second-order centered space differencing for the second space derivative can be implemented if less accuracy is required in the internal simulation in exchange for higher speed.

$\frac{dU}{dt}$ and $\frac{dQ}{dt}$ are then assembled into matrix *dydt* according the arrangement gave in Table 5-2.3.1.

With present stiff ODE solver, not only do we need to supply the problem dependent subroutine *derivs*, we also need to supply the *jacobn* subroutine which evaluate Jacobian matrix of the differential equation with respect to the dependent variables. Implementation of subroutine *jacobn* fully depends on the implementation of subroutine *derivs*. The matching of *derivs* and *jacobn* is the main factor to reduce the integrating time of ODE.

5 - 3.2 The result

The numerical simulation of a fixed-bed adsorption column by NUMOL with a particular set of parameters are presented as follows:

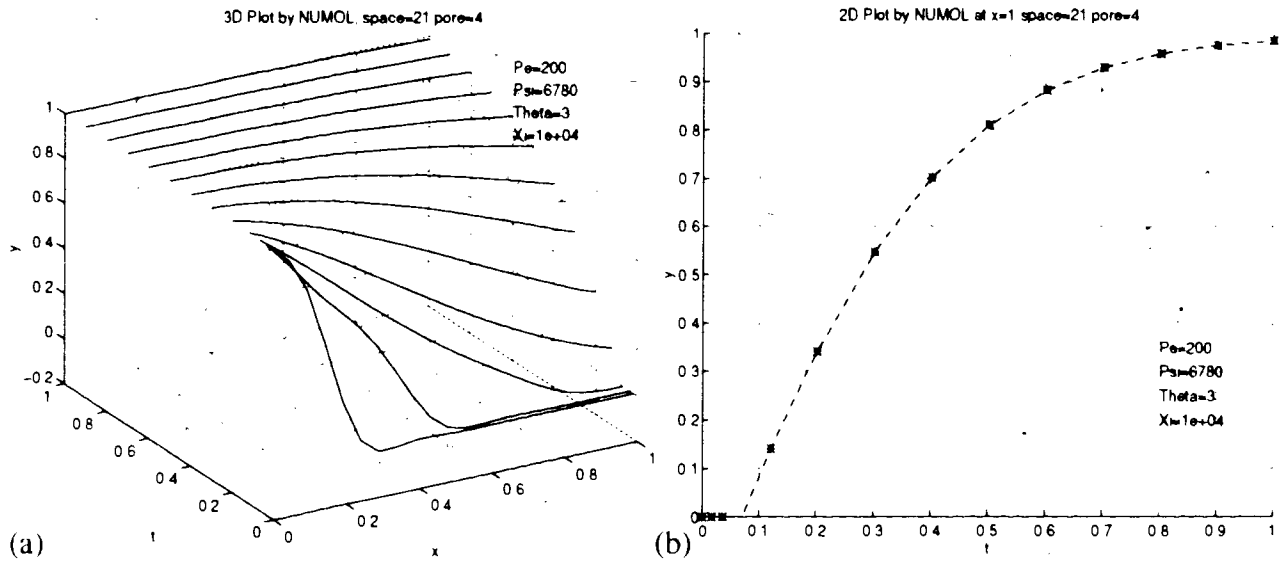


Figure 5 - 3.2.1 Simulation Result by NUMOL with mesh space=21, pore=4 (a) 3D plot (b) 2D plot at outlet

5 - 4 Numerical Simulation by the Orthogonal Collocation Method

5 - 4.1 Collocation Form

(Eq 5 - 2.2.1) to (Eq 5 - 2.2.7) can be reduced to a set of ordinary differential equations by writing in collocation form.

The concentration profile in the spherical particle is approximated by the symmetric trial function defined as:

$$Q(\eta, \tau) = Q(1, \tau) + (1 - \eta^2) \sum_{i=1}^N a_i(\tau) P_{i-1}(\eta^2) \quad (\text{Eq 5 - 4.1.1})$$

where $a_i(\tau)$ are functions of time (or constants) and P_i are the orthogonal polynomials (Jacobi polynomials) defined by:

$$\int_0^1 w(\eta^2) P_j(\eta^2) \eta^{a-1} d\eta = C_i \delta_{ij} \quad (\text{Eq 5 - 4.1.2})$$

where $w(\eta^2) = 1 - \eta^2, j = 1, 2$

(Eq 5 - 2.2.1) can be written in collocation form as:

$$\frac{1}{K} \frac{dQ_k}{d\tau} = \sum_{i=1}^{N+1} B_{k,i} \frac{Q_i}{K}, k = 1, 2, \dots, N \quad (\text{Eq 5 - 4.1.3})$$

(Eq 5 - 2.2.4) is written in collocation form as:

$$\frac{1}{K} \sum_{i=1}^{N+1} A_{N+1,i} Q_i(j) = \xi(U(j) - \frac{Q_{N+1}}{K}(j)) \quad (\text{Eq 5 - 4.1.4})$$

where N is the number of collocation points for the particle and N+1 refers to the external

surface of the particle. The above equation can also be written after some simple manipulation:

$$\frac{Q_{N+1}(j)}{K} = \frac{U(j) - \frac{1}{K\xi} \sum_{i=1}^N A_{N+1,i} Q_i(j)}{\frac{1}{\xi} A_{N+1,N+1} + 1} = \frac{\xi U(j) - \frac{1}{K} \sum_{i=1}^N A_{N+1,i} Q_i(j)}{A_{N+1,N+1} + \xi} \quad (\text{Eq 5 - 4.1.5})$$

thus, (Eq 5 - 4.1.3) becomes:

$$\begin{aligned} \frac{1}{K} \frac{dQ_k(j)}{d\tau} &= \sum_{i=1}^N B_{k,i} \frac{Q_i(j)}{K} - \frac{B_{k,N+1} \frac{1}{K} \sum_{i=1}^N A_{N+1,i} Q_i(j)}{A_{N+1,N+1} + \xi} + \frac{B_{k,N+1} \xi U(j)}{A_{N+1,N+1} + \xi} \\ &= \sum_{i=1}^N \left(B_{k,i} - \frac{B_{k,N+1} \frac{1}{K} \sum_{i=1}^N A_{N+1,i}}{A_{N+1,N+1} + \xi} \right) \frac{Q_i(j)}{K} + \frac{B_{k,N+1} \xi U(j)}{A_{N+1,N+1} + \xi} \end{aligned} \quad (\text{Eq 5 - 4.1.6})$$

The concentration profile in the external fluid phase is similarly approximated by the following trial function based on nonsymmetric polynomials:

$$U(x, \tau) = (1-x)U(0, \tau) + xU(1, \tau) + x(1-x) \sum_{i=1}^M a_i(\tau) P_{i-1}(x) \quad (\text{Eq 5 - 4.1.7})$$

where $a_i(\tau)$ are, as before, functions of time or constants and P_i are the non symmetric polynomials (Legendre polynomials) defined by the condition.

$$\int_0^1 w(x) P_n(x) P_m(x) dx = 0, n = 0, 1, \dots, m-1 \quad (\text{Eq 5 - 4.1.8})$$

where $w(x) = 1$ in this work.

(Eq 5 - 2.2.5) to (Eq 5 - 2.2.7) can be written in collocation form as:

$$\frac{dU(j)}{d\tau} = \frac{1}{Pe} \phi \theta \sum_{i=1}^{M+2} B_{j,i}^x U(i) - \phi \theta \sum_{i=1}^{M+2} A_{j,i}^x U(i) - 3\phi \xi (U(j) - \frac{Q_{N+1}^j}{K}), j = 2, 3, \dots, M+1 \quad (\text{Eq 5 - 4.1.9})$$

$$\sum_{i=1}^{M+2} A_{1,i}^x U(i) = -Pe (U|_{x=0} - U(1)) \quad (\text{Eq 5 - 4.1.10})$$

$$\sum_{i=1}^{M+2} (A_{M+2,i}^x) U(i) = 0 \quad (\text{Eq 5 - 4.1.11})$$

The above two boundary conditions can be used to eliminate $U(1)$ and $U(M+2)$ from (Eq 5 - 4.1.9). If we define:

$$\Phi = A_{M+2, M+2}^x (A_{1,1}^x - Pe) - A_{1, M+2}^x A_{M+2,1}^x \quad (\text{Eq 5 - 4.1.12})$$

$$R_1 = \frac{A_{M+2,1}^x}{\Phi}, R_2 = \frac{A_{1,1}^x}{\Phi}, R_3 = \frac{A_{1,M+2}^x}{\Phi}, R_4 = \frac{A_{M+2,M+2}^x}{\Phi} \quad (\text{Eq 5 - 4.1.13})$$

$$U(1) = (-PeU|_{x=0})R_4 - \left(\sum_{i=2}^{M+1} A_{1,i}^x U(i) \right) R_4 + \left(\sum_{i=2}^{M+1} A_{M+2,i}^x U(i) \right) R_3 \quad (\text{Eq 5 - 4.1.14})$$

$$U(M+2) = PeU|_{x=0} R_1 + \left(\sum_{i=2}^{M+1} A_{1,i}^x U(i) \right) R_1 + \left(\sum_{i=2}^{M+1} A_{M+2,i}^x U(i) \right) R_2 \quad (\text{Eq 5 - 4.1.15})$$

Thus, (Eq 5 - 4.1.9) becomes:

$$\begin{aligned} \frac{dU}{d\tau}(j) = & \sum_{i=2}^{M+1} \left[\left(\frac{\phi\theta}{Pe} B_{j,i}^x - \phi\theta A_{j,i}^x \right) + \left(\frac{\phi\theta}{Pe} B_{j,1}^x - \phi\theta A_{j,1}^x \right) (R_3 A_{M+2,i}^x - R_4 A_{1,i}^x) \right. \\ & \left. + \left(\frac{\phi\theta}{Pe} B_{j,M+2}^x - \phi\theta A_{j,M+2}^x \right) (R_1 A_{1,i}^x - R_2 A_{M+2,i}^x) \right] U(i) \\ & - PeU|_{x=0} \left[R_4 \left(\frac{\phi\theta}{Pe} B_{j,1}^x - \phi\theta A_{j,1}^x \right) - R_1 \left(\frac{\phi\theta}{Pe} B_{j,M+2}^x - \phi\theta A_{j,M+2}^x \right) \right] \\ & + 3\phi\xi \left(\frac{\xi}{\xi + A_{N+1,N+1}} - 1 \right) U(j) - \frac{3\phi\xi \sum_{i=1}^N A_{N+1,i} (Q_i U) / K}{\xi + A_{N+1,N+1}} \end{aligned} \quad (\text{Eq 5 - 4.1.16})$$

The procedures for generating the A, B and A^x, B^x matrices of (Eq 5 - 4.1.11) to (Eq 5 - 4.1.16) are described in detail in Appendix or Finlayson (1972)

5 - 4.2 Simulation Result

A simulation result by orthogonal collocation method with a set of parameters is present as follows:

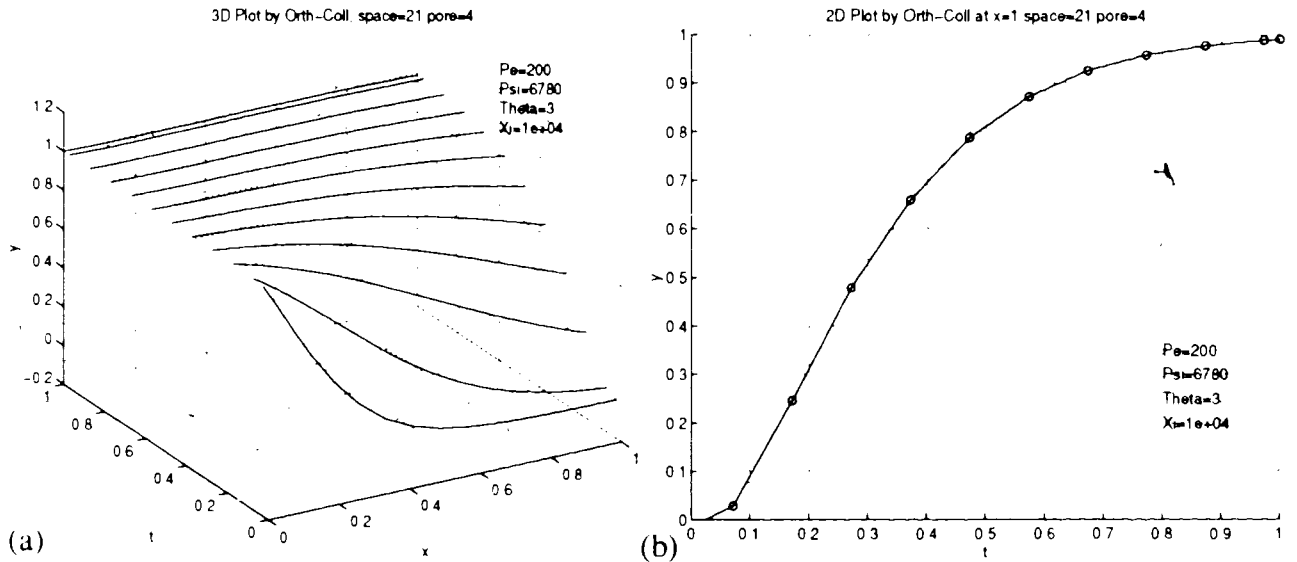


Figure 5 - 4.2.1 Simulation Result by Orthogonal Collocation method, with mesh space=21, pore=4, (a) 3D plot, (b) 2D plot at outlet.

5 - 5 Experiments and Comparison of Results

In this section, we are going through several experiments to test the two models implemented by NUMOL and Orthogonal Collocation method.

5 - 5.1 Comparison between Numerical and Analytical results

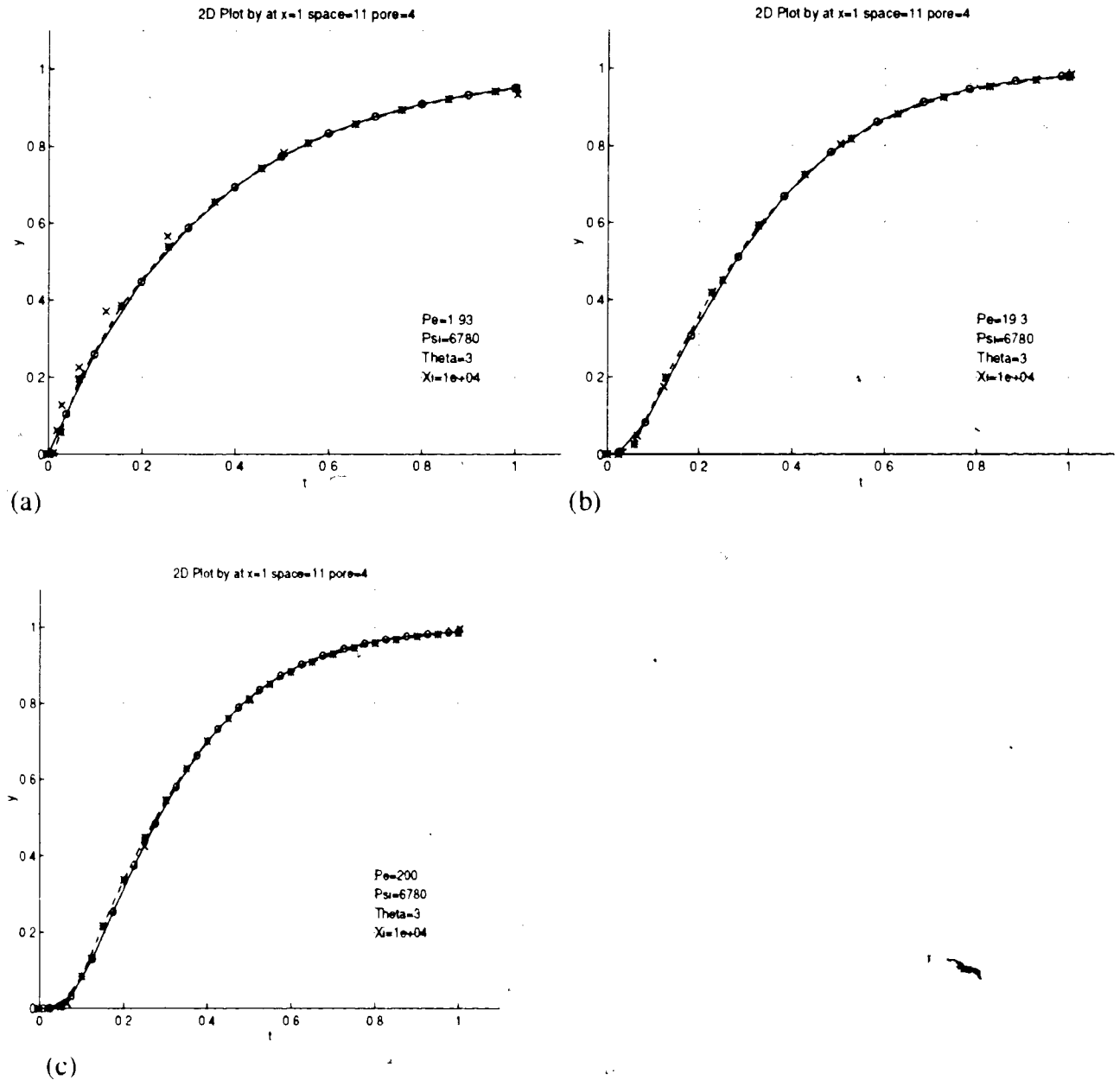


Figure 5 - 5.1.1 Comparison between analytical results and numerical results from NUMOL and orthogonal collocation, mesh space=11, pore=4, x---analytical result, *---NUMOL result, o---orth-coll result. (a) Pe=1.93, (b) Pe=19.3, (c) Pe=200

From the above Figure 5 - 5.1.1, for three different values of Peclet number, both numerical results from NUMOL and orthogonal collocation method agree with the analytical results, therefore, the results of both numerical methods are verified. We can see that even using a coarse mesh, both numerical methods can handle this model problem.

5 - 5.2 Effects of Axial Dispersion and External Mass Transfer Resistance

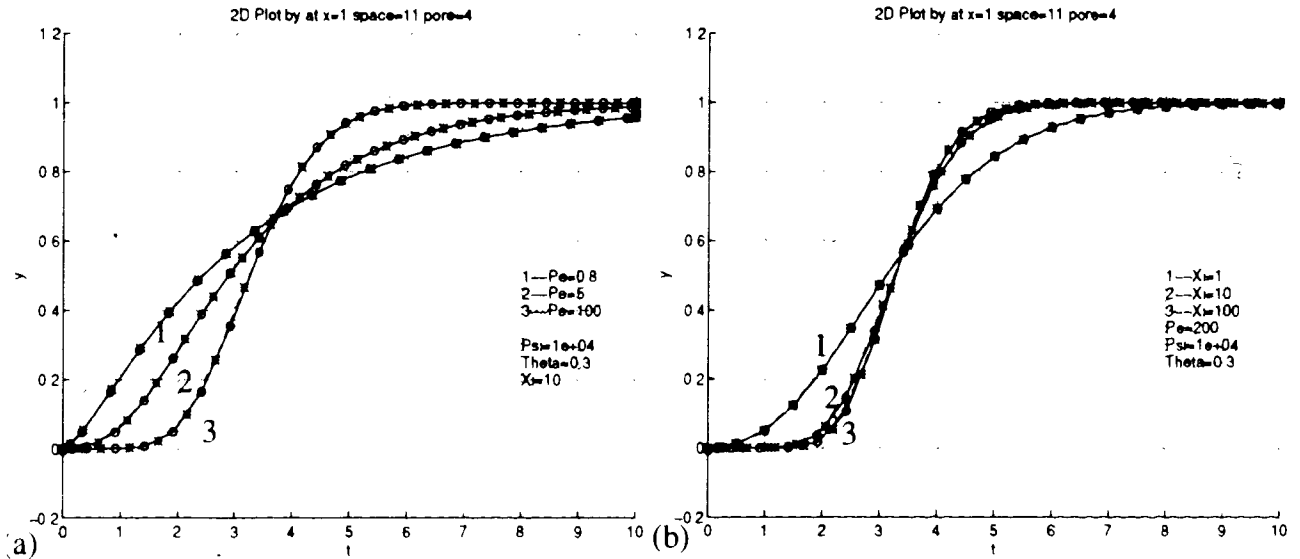


Figure 5 - 5.2.1 Comparison of simulations conducted by (*---)NUMOL and (o---)Orth-coll (a) effect of Peclet Number, (b) effect of external film resistance

It has also been shown that simulations from both methods give us similar prediction about the effect of Peclet number and external film resistance.

5 - 5.3 Effects of Distribution Ratio

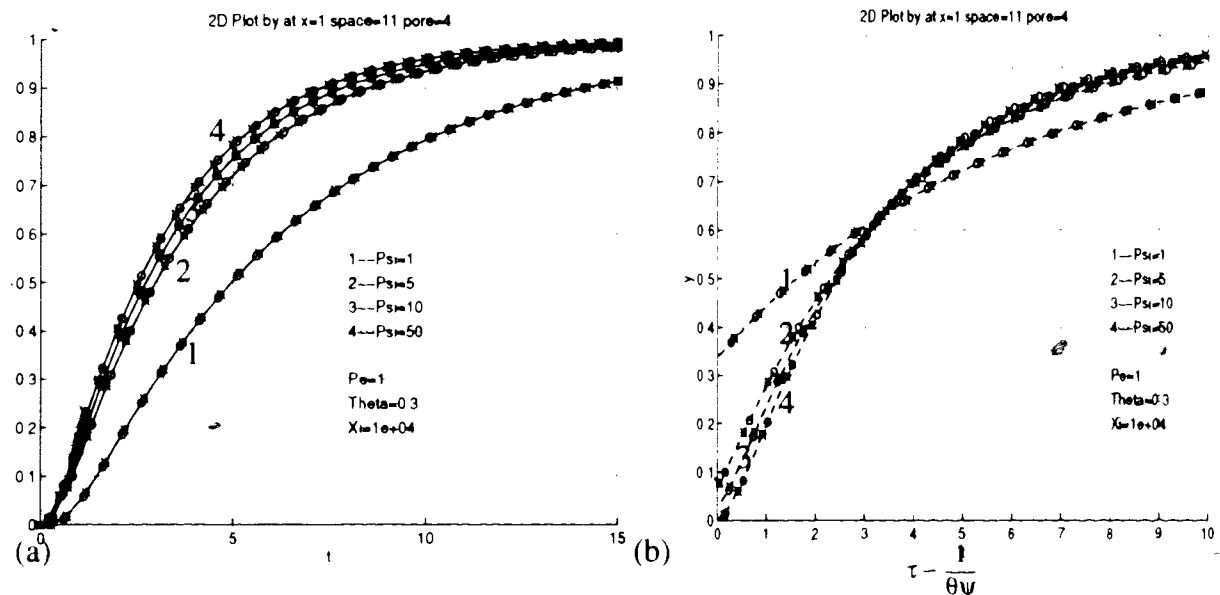


Figure 5 - 5.3.1 Comparison of simulations conducted by (*---)NUMOL and (o---)Orth-coll effects of Distribution Ratio. $Pe=1.0$ (a) plotted versus τ , (b) plotted versus $\tau - \frac{1}{\theta\psi}$

Other experiments such as effect of distribution ratio have also been conducted, see Figure 5 - 5.3.1 for $Pe=1.0$ and Figure 5 - 5.3.1 for $Pe=20000.0$, Figure 5 - 5.3.1(a) and Figure 5 - 5.3.1 (a) are plotted against dimensionless time variable τ , and Figure 5 - 5.3.1(b) and Figure 5 - 5.3.1 (b) plotted against modified time variable $\tau - \frac{1}{\theta\psi}$. Simulations from both methods predict

same behavior of adsorption bed. We can see that the effect of the distribution ratio on the theoretical breakthrough curves to be significant only when the effect of axial dispersion is important.

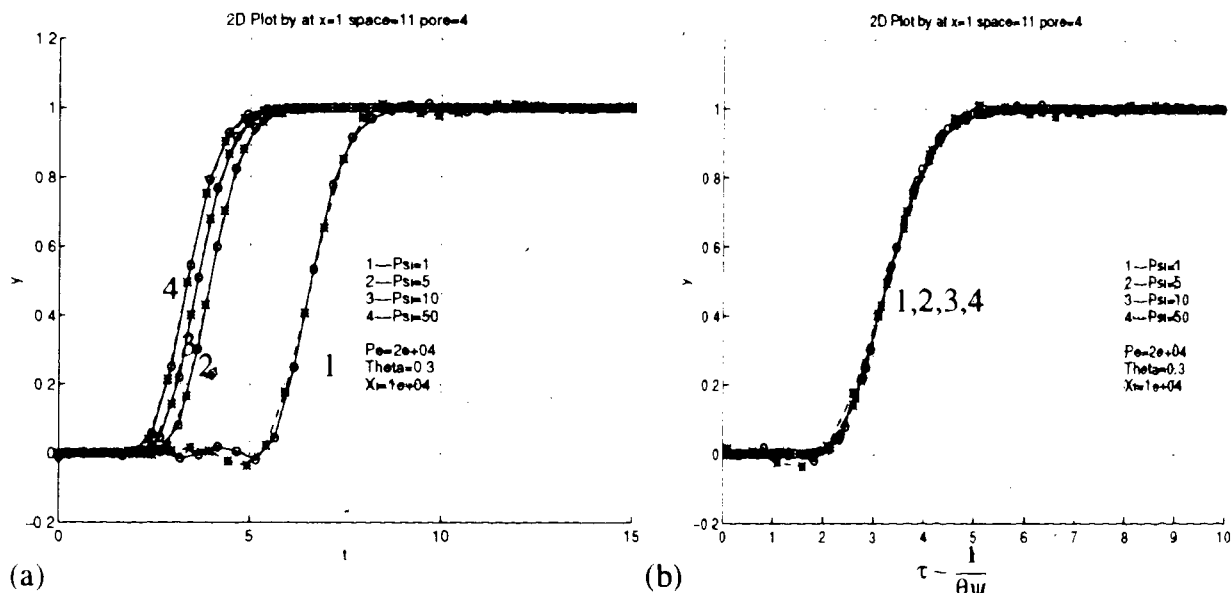


Figure 5 - 5.3.2 Comparison of simulations conducted by (*---)NUMOL and (o---)Orth-coll effects of Distribution Ratio. $Pe=20000.0$ (a) plotted verse τ (b) $\tau - \frac{1}{\theta\psi}$

All the results produced so far agree with that from Dr. Ruthven's paper.

However, as we look at Figure 5 - 5.3.1 (a) for high Peclet number (20000.0), it can be seen that the results of orthogonal collocation have better performance on numerical diffusion and nonphysical oscillation.

5 - 5.4 The Computation Time

Table 5 - 5.4.1 Comparison of computational time (CPU time, unit: second) by NUMOL and Orthogonal collocation method with mesh space=11, pore=4, maxstep=0.05

Parameters	Computational time for NUMOL	Computational time for Orth-Coll
$Pe = 1.93, \psi = 6780, \theta = 3.0, \xi = 10000.0$	6.6	3.1
$Pe = 19.3, \psi = 6780, \theta = 3.0, \xi = 10000.0$	6.9	3.3
$Pe = 200.00, \psi = 6780, \theta = 3.0, \xi = 10000.0$	7.5	3.3
$Pe = 0.8, \psi = 10000.0, \theta = 0.3, \xi = 10000.0$	7.4	3.1
$Pe \cong 5.0, \psi = 10000.0, \theta = 0.3, \xi = 10000.0$	8.7	3.1
$Pe = 100.0, \psi = 10000.0, \theta = 0.3, \xi = 10000.0$	8.5	3.7

Therefore, we have shown that the orthogonal collocation method is superior to NUMOL on

computational time with similar performance as shown in Figure 5-4.1.1 and Figure 5-4.2.1.

5 - 6 Conclusion for Chapter 5

From this chapter, we have used NUMOL and Orthogonal Collocation method studied in last two chapters to deal with a fairly complex problem in PSA modelling. It has been shown that both methods can handle this task, and similar results are achieved and agree with analytical results. However, orthogonal collocation method is superior to NUMOL on computational time, and has less numerical diffusion and nonphysical oscillation when Peclet number is large. The reason would be that, for the same mesh number, the formulation for the first and second derivatives (DSS020 and DSS044) in NUMOL are locally fourth-order accurate, however, the formulation in orthogonal collocation method is globally more accurate, since number of collocation points means same number of orthogonal polynomials in the formulation, and more collocation points means a higher order of formulation globally. The comparison between the formulation of these two methods is not straightforward, but generally we can say that for the same mesh number l (≥ 4), the formulation in orthogonal collocation method is more accurate. The accurate formulations in orthogonal collocation method allows large integration step and less times of evaluation of *derivs* and *jacobn* in O.D.E integrator, which results less computation time. Therefore, we prefer to use the Orthogonal Collocation method in our future study when we have to be content with coarse mesh.

Chapter 6 PSA Simulation by Double Collocation Method

6 - 1 Introduction

A simple two-bed PSA system for air separation, in which kinetic effects are important and the changes in flow rate due to adsorption are significant, has been studied. Under these circumstances, the assumption which have been traditionally employed in modeling of the PSA cycle (adsorption equilibrium, constant velocity) are no longer valid. In the paper of Raghavan and Ruthven (1985), a more general mathematical model was developed without these simplifying assumptions. Since the corresponding mathematical model for simulation is more general and complicated, and no longer fits in the procedure of PDE-ODE, double collocation method is developed and used to solve it numerically. In this paper, we will present this mathematical model and double collocation method as well as the simulation results. The work in this chapter will serve as a base for our modelling of the conventional PSA process.

6 - 2 A System of PSA Process Description

The basic PSA cycle involves four distinct steps: (as shown in Figure 6 - 2.1).

1. Feed step
2. Blowdown step
3. Purge step
4. Repressurization step

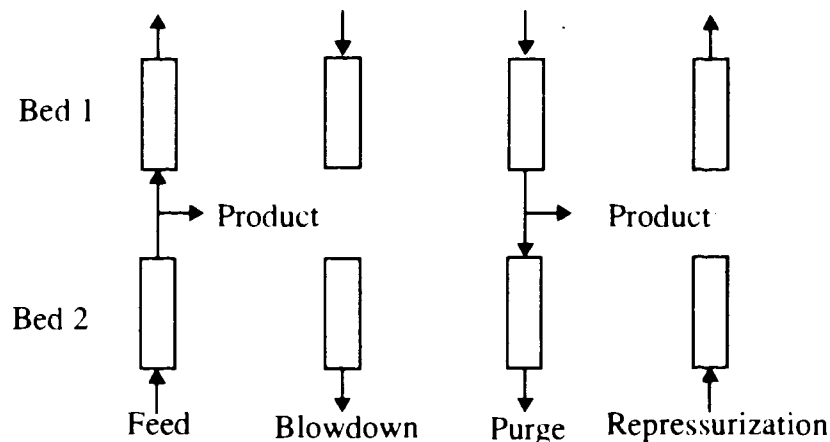


Figure 6 - 2.1 Steps involved in a PSA cycle

During one cycle, time distribution is shown on Figure 6 - 2.2:

1. t_{max} : duration of adsorption at step 1 or 3.
2. t_b : duration of blowdown at step 2 or step 4

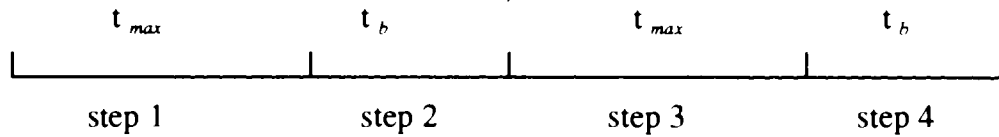


Figure 6 - 2.2 Time distribution of one cycle

During step 1 a high-pressure feed (in our case air) is supplied continuously to bed 2, which is packed with carbon molecular sieve (CMS) pellets in which oxygen is adsorbed more rapidly than nitrogen. The nitrogen which remains in the gas phase passes through the bed and is removed as pure raffinate product. A small fraction of this stream is expanded to low pressure and used to purge bed 1 (which is also packed with CMS pellets). In step 2, bed 1 is pressurized with feed while bed 2 is subjected to a pressure reduction (blowdown). The same cycle is repeated in steps 3 and 4 with high-pressure flow and adsorption occurring in bed 1 and purging occurring in bed 2.

In order to develop a mathematical model the following approximations are introduced:

1. The system is considered isothermal with total pressure remaining constant throughout the bed during high-pressure and low-pressure flow operations (step 1 and 3).
2. The flow pattern is described by the axial-dispersed plug flow model.
3. The equilibrium relationships for both oxygen and nitrogen are assumed linear.
4. Mass transfer rates are represented by linear driving force expressions and the rate coefficient is the same for both high-pressure and low-pressure steps.

Subject to these assumptions, the dynamic behavior of the system may be described by the following set of equations ($A=O_2, B=N_2$).

6 - 3 Mathematical Model

6 - 3.1 Step 1: High-Pressure Flow in Bed 2 and Low-Pressure Flow in Bed 1

6 - 3.1.1 Dimensional form:

External fluid phase in bed 2:

$$\frac{\partial C_{A2}}{\partial t} - D_{L2} \frac{\partial^2 C_{A2}}{\partial z^2} + v_2 \frac{\partial C_{A2}}{\partial z} + C_{A2} \frac{\partial v_2}{\partial z} + \left(\frac{1 - \epsilon}{\epsilon} \right) \frac{\partial q_{A2}}{\partial t} = 0 \quad (\text{Eq 6 - 3.1.1})$$

$$\frac{\partial C_{B2}}{\partial t} - D_{L2} \frac{\partial^2 C_{B2}}{\partial z^2} + v_2 \frac{\partial C_{B2}}{\partial z} + C_{B2} \frac{\partial v_2}{\partial z} + \left(\frac{1 - \epsilon}{\epsilon} \right) \frac{\partial q_{B2}}{\partial t} = 0 \quad (\text{Eq 6 - 3.1.2})$$

$$C_{A2} + C_{B2} = C_{HP} (\text{const}) \quad (\text{Eq 6 - 3.1.3})$$

adding (Eq 6 - 3.1.1) and (Eq 6 - 3.1.2) and considering (Eq 6 - 3.1.3)

$$C_{HP} \frac{\partial v_2}{\partial z} + \frac{1-\varepsilon}{\varepsilon} \frac{\partial q_{A2}}{\partial t} + \frac{1-\varepsilon}{\varepsilon} \frac{\partial q_{B2}}{\partial t} = 0 \quad (\text{Eq 6 - 3.1.4})$$

Solid phase in bed 2:

$$\frac{\partial q_{A2}}{\partial t} = k_A (q_{A2}^* - q_{A2}) \quad (\text{Eq 6 - 3.1.5})$$

$$\frac{\partial q_{B2}}{\partial t} = k_B (q_{B2}^* - q_{B2}) \quad (\text{Eq 6 - 3.1.6})$$

$$\text{where: } q_{A2}^* = K_A C_{A2}, \quad q_{B2}^* = K_B C_{B2}. \quad (\text{Eq 6 - 3.1.7})$$

Boundary conditions:

$$D_{L2} \frac{\partial C_{A2}}{\partial z} \Big|_{z=0} = -v_{OH} (C_{A2} \Big|_{z=0} - C_{A2} \Big|_{0'}) \quad (\text{Eq 6 - 3.1.8})$$

$$\frac{\partial C_{A2}}{\partial z} \Big|_{z=L} = 0 \quad (\text{Eq 6 - 3.1.9})$$

External fluid phase in bed 1:

$$\frac{\partial C_{A1}}{\partial t} - D_{L1} \frac{\partial^2 C_{A1}}{\partial z^2} + v_1 \frac{\partial C_{A1}}{\partial z} + C_{A1} \frac{\partial v_1}{\partial z} + \frac{1-\varepsilon}{\varepsilon} \frac{\partial q_{A1}}{\partial t} = 0 \quad (\text{Eq 6 - 3.1.10})$$

$$\frac{\partial C_{B1}}{\partial t} - D_{L2} \frac{\partial^2 C_{B1}}{\partial z^2} + v_1 \frac{\partial C_{B1}}{\partial z} + C_{B1} \frac{\partial v_1}{\partial z} + \frac{1-\varepsilon}{\varepsilon} \frac{\partial q_{B1}}{\partial t} = 0 \quad (\text{Eq 6 - 3.1.11})$$

$$C_{A1} + C_{B1} = C_{LP} (\text{const}) \quad (\text{Eq 6 - 3.1.12})$$

Adding (Eq 6 - 3.1.10) and (Eq 6 - 3.1.11), and considering (Eq 6 - 3.1.12),

$$C_{LP} \frac{\partial v_1}{\partial z} + \frac{1-\varepsilon}{\varepsilon} \frac{\partial q_{A1}}{\partial t} + \frac{1-\varepsilon}{\varepsilon} \frac{\partial q_{B1}}{\partial t} = 0 \quad (\text{Eq 6 - 3.1.13})$$

Solid phase in bed 1:

$$\frac{\partial q_{A1}}{\partial t} = k_A (q_{A1}^* - q_{A1}) \quad (\text{Eq 6 - 3.1.14})$$

$$\frac{\partial q_{B1}}{\partial t} = k_B (q_{B1}^* - q_{B1}) \quad (\text{Eq 6 - 3.1.15})$$

$$\text{where: } q_{A1}^* = K_A C_{A1}, \quad q_{B1}^* = K_B C_{B1}. \quad (\text{Eq 6 - 3.1.16})$$

Boundary conditions:

$$D_{L1} \frac{\partial C_{A1}}{\partial z} \Big|_{z=0} = -v_{OL} (C_{A1} \Big|_{z=0} - C_{A1} \Big|_{0'}) \quad (\text{Eq 6 - 3.1.17})$$

where:

$$C_{A1} \Big|_{z=0} = C_{A2} \Big|_{z=L, P=P_L} \quad (\text{Eq 6 - 3.1.18})$$

$$\frac{\partial C_{A1}}{\partial z} \Big|_{z=L} = 0 \quad (\text{Eq 6 - 3.1.19})$$

The initial conditions for the start-up of the cyclic operation with two clean beds are the following two sets of equations:

$$\begin{aligned}
 C_{A2}(z, t = 0) &= 0; C_{B2}(z, t = 0) = 0 \\
 q_{A2}(z, t = 0) &= 0; q_{B2}(z, t = 0) = 0 \\
 C_{A1}(z, t = 0) &= 0; C_{B1}(z, t = 0) = 0 \\
 q_{A1}(z, t = 0) &= 0; q_{B1}(z, t = 0) = 0
 \end{aligned}
 \tag{Eq 6 - 3.1.20}$$

Since C_{B2} can be found from C_{HP} and C_{A2} (for high-pressure flow in bed 2), and C_{B1} can be found from C_{LP} and C_{A1} (for low-pressure flow in bed 1), the equations which must be solved simultaneously along with the relevant boundary conditions are (Eq 6 - 3.1.1) (Eq 6 - 3.1.4) (Eq 6 - 3.1.5) (Eq 6 - 3.1.6) (Eq 6 - 3.1.10) (Eq 6 - 3.1.13) (Eq 6 - 3.1.14) (Eq 6 - 3.1.15).

6 - 3.1.2 Dimensionless Forms

The above equations can be written in dimensionless form for calculation. First we list the necessary dimensionless variables and parameters for future reference.

Table 6 - 3.1.1 Dimensionless Variables

dimensionless variables	definition
X_{Ai}	C_{Ai}/C_{HP} or C_{Ai}/C_{LP} depending on whether high-pressure flow adsorption or low-pressure flow desorption occurs in the <i>i</i> th bed
X_{Bi}	C_{Bi}/C_{HP} or C_{Bi}/C_{LP} depending on whether high-pressure flow adsorption or low-pressure flow desorption occurs in the <i>i</i> th bed, also equal to $(1 - X_{Ai})$
Y_{Ai}	$q_{Ai}/K_A C_{HP}$, adsorbate concentration of component A in solid phase in bed <i>i</i>
Y_{Bi}	$q_{Bi}/K_B C_{HP}$, adsorbate concentration of component B in solid phase in bed <i>i</i>
x	$x = z/L$, dimensionless distance
τ	$\tau = t/h$, dimensionless time, h : dimensional time interval (see Figure 6 - 4.1)
V_i	$V_i = v_i/v_{OH}$, interstitial fluid velocity in bed <i>i</i>

Table 6 - 3.1.2 Dimensionless Parameters

dimensionless parameters	definition
α/h	$\alpha/h = L/(v_{OH}h)$, where $\alpha = L/v_{OH}$, ratio of bed length to high pressure feed inlet velocity, s
Pe_H	$Pe_H = Lv_{OH}/D_{L2}$, Peclet number for high pressure flow in bed 2 during step 1

Table 6 - 3.1.2 Dimensionless Parameters

dimensionless parameters	definition
Pe_L	$Pe_L = Lv_{OL}/D_{L1}$, Peclet number for high pressure flow in bed 1 during step 1
$k_A h$	dimensionless overall mass transfer coefficient for Component A, where k_A is overall mass transfer coefficient for component A, s^{-1} .
$k_B h$	dimensionless overall mass transfer coefficient for Component B, where k_B is overall mass transfer coefficient for component B, s^{-1} .
K_A	adsorption equilibrium constant for component A.
K_B	adsorption equilibrium constant for component B.
G	$G = v_{OL}/v_{OH}$, purge to feed velocity ratio.
β	$\beta = C_{HP}/C_{LP} = P_H/P_L$, total concentration at high-pressure to total concentration at low-pressure ratio.

Dimensionless forms for bed 2:

$$\frac{\alpha}{h} \frac{\partial X_{A2}}{\partial \tau} - \frac{1}{Pe_H} \frac{\partial^2 X_{A2}}{\partial x^2} + V_2 \frac{\partial X_{A2}}{\partial x} + \frac{1-\epsilon}{\epsilon} \alpha k_A K_A (X_{A2} - Y_{A2}) (1 - X_{A2}) - \frac{1-\epsilon}{\epsilon} \alpha k_B K_B X_{A2} (1 - X_{A2} - Y_{B2}) = 0 \quad (\text{Eq 6 - 3.1.21})$$

$$\frac{\partial Y_{A2}}{\partial \tau} = h k_A (X_{A2} - Y_{A2}) \quad (\text{Eq 6 - 3.1.22})$$

$$\frac{\partial Y_{B2}}{\partial \tau} = h k_B (1 - X_{A2} - Y_{A2}) \quad (\text{Eq 6 - 3.1.23})$$

$$\frac{\partial V_2}{\partial x} + \frac{1-\epsilon}{\epsilon} \alpha k_A K_A (X_{A2} - Y_{A2}) + \frac{1-\epsilon}{\epsilon} \alpha k_B K_B (1 - X_{A2} - Y_{B2}) = 0 \quad (\text{Eq 6 - 3.1.24})$$

Boundary condition in dimensionless form for bed 2:

$$\left. \frac{\partial X_{A2}}{\partial x} \right|_{x=0} = -Pe_H (X_{A2}|_{x=0} - X_{A2}|_{x=0'}) \quad (\text{Eq 6 - 3.1.25})$$

$$\left. \frac{\partial X_{A2}}{\partial x} \right|_{x=1} = 0 \quad (\text{Eq 6 - 3.1.26})$$

Dimensionless form for bed 1:

$$\frac{\alpha}{h} \frac{\partial X_{A1}}{\partial \tau} - \frac{1}{Pe_L} \frac{\partial^2 X_{A1}}{\partial x^2} + V_1 \frac{\partial X_{A1}}{\partial x} + \beta \frac{1-\epsilon}{\epsilon} \alpha k_A K_A \left(\frac{X_{A1}}{\beta} - Y_{A1} \right) (1 - X_{A1}) - \beta \frac{1-\epsilon}{\epsilon} \alpha k_B K_B X_{A1} \left(\frac{1 - X_{A1}}{\beta} - Y_{B1} \right) = 0 \quad (\text{Eq 6 - 3.1.27})$$

$$\frac{\partial Y_{A1}}{\partial \tau} = h k_A \left(\frac{X_{A1}}{\beta} - Y_{A1} \right) \quad (\text{Eq 6 - 3.1.28})$$

$$\frac{\partial Y_{B1}}{\partial \tau} = hk_B \left(\frac{1 - X_{A1}}{\beta} - Y_{B1} \right) \quad (\text{Eq 6 - 3.1.29})$$

$$\frac{\partial V_1}{\partial x} + \beta \frac{1 - \varepsilon}{\varepsilon} \alpha k_A K_A \left(\frac{X_{A1}}{\beta} - Y_{A1} \right) + \beta \frac{1 - \varepsilon}{\varepsilon} \alpha k_B K_B \left(\frac{1 - X_{A1}}{\beta} - Y_{B1} \right) = 0 \quad (\text{Eq 6 - 3.1.30})$$

Boundary condition in dimensionless form for bed 1:

$$\left. \frac{\partial X_{A1}}{\partial x} \right|_{x=0} = -Pe_L G (X_{A1}|_{x=0} - X_{A1}|_{x=0'}) \quad (\text{Eq 6 - 3.1.31})$$

where:

$$X_{A1}|_{x=0} = X_{A2}|_{x=1} \quad (\text{Eq 6 - 3.1.32})$$

$$\left. \frac{\partial X_{A1}}{\partial x} \right|_{x=1} = 0 \quad (\text{Eq 6 - 3.1.33})$$

6 - 3.2 Step 2: Blowdown of Bed 2 and Pressurization of Bed 1

6 - 3.2.1) For blowdown of bed 2, the following two approximations are introduced:

1. At each and every position in the bed, the gas phase concentrations of O₂ and N₂ at the end of blowdown correspond to the concentrations at the end of the preceding high-pressure flow adsorption step, multiplied by the pressure ratio, P_L/P_H, where P_H is the pressure at which adsorption is carried out and P_L is the purge or desorption pressure.

$$C_{A2}(z, t = t_{max} + t_b) = C_{A2}(z, t = t_{max}) \frac{P_L}{P_H} \quad (\text{Eq 6 - 3.2.1})$$

$$C_{B2}(z, t = t_{max} + t_b) = C_{B2}(z, t = t_{max}) \frac{P_L}{P_H} \quad (\text{Eq 6 - 3.2.2})$$

Since the concentrations of O₂ and N₂ at the end of blowdown are equal to the concentration at the beginning of step 3, i.e., at the beginning of purge(desorption), we can write the above equations in dimensionless form:

$$X_{A2}(x, \tau)|_{P=P_L, t=t_{max}+t_b} = X_{A2}(x, \tau)|_{P=P_H, t=t_{max}} \quad (\text{Eq 6 - 3.2.3})$$

$$X_{B2}(x, \tau)|_{P=P_L, t=t_{max}+t_b} = X_{B2}(x, \tau)|_{P=P_H, t=t_{max}} \quad (\text{Eq 6 - 3.2.4})$$

2. The solid phase concentrations of O₂ and N₂ are assumed to remain frozen.

$$q_{A2}(z, t = t_{max} + t_b) = q_{A2}(z, t = t_{max}) \quad (\text{Eq 6 - 3.2.5})$$

$$q_{B2}(z, t = t_{max} + t_b) = q_{B2}(z, t = t_{max}) \quad (\text{Eq 6 - 3.2.6})$$

Similarly, we can write the above equations in dimensionless form:

$$Y_{A2}(x, \tau)|_{P=P_L, t=t_{max}+t_b} = Y_{A2}(x, \tau)|_{P=P_H, t=t_{max}} \quad (\text{Eq 6 - 3.2.7})$$

$$Y_{B2}(x, \tau)|_{P=P_L, t=t_{max}+t_b} = Y_{B2}(x, \tau)|_{P=P_H, t=t_{max}} \quad (\text{Eq 6 - 3.2.8})$$

6 - 3.2.2 For pressurization of bed 1, the following two approximations are introduced:

1. After low pressure purge step, there are still gas phase O₂ and N₂ remaining in the bed. During pressurization by feed air, these gas phase remnants are pushed toward the closed end of

the bed through a distance $(1-P_L/P_H)$ from the feed inlet. Thus, for example, if the pressure ratio, P_L/P_H is 0.33, then the concentrations of O_2 and N_2 at the end of pressurization are the average of the respective concentrations of the remaining O_2 and N_2 at the end of purge step. However, the average concentrations exist only through a dimensionless length of 0.33 from the closed end of the bed, the rest of bed being filled with the pressurizing feed air. We think this approximation is clearly more realistic than assuming a uniform composition in the gas phase. The above approximations can be written in mathematical form:

$$C_{A1}(z, t = t_{max} + t_b) = \begin{cases} \frac{\int_0^L C_{A1}(z, t = t_{max}) dz}{(P_L/P_H)}, & z \in ((1 - P_L/P_H)L, L) \\ C_{A1}(z = 0), & z \in (0, (1 - P_L/P_H)L) \end{cases} \quad (\text{Eq 6 - 3.2.9})$$

$$C_{B1}(z, t = t_{max} + t_b) = \begin{cases} \frac{\int_0^L C_{B1}(z, t = t_{max}) dz}{(P_L/P_H)}, & z \in ((1 - P_L/P_H)L, L) \\ C_{B1}(z = 0), & z \in (0, (1 - P_L/P_H)L) \end{cases} \quad (\text{Eq 6 - 3.2.10})$$

write in dimensionless form:

$$X_{A1}(x, \tau) \Big|_{P=P_H, t=t_{max}+t_b} = \begin{cases} \int_0^1 X_{B1}(x, \tau) \Big|_{P=P_L, t=t_{max}} dx, & x \in ((1 - P_L/P_H), 1) \\ X_{B1}(x = 0), & x \in (0, (1 - P_L/P_H)) \end{cases} \quad (\text{Eq 6 - 3.2.11})$$

2. The solid phase concentrations of O_2 and N_2 remain unchanged at the values of the end of the preceding low pressure purge flow.

$$q_{A1}(z, t = t_{max} + t_b) = q_{A1}(z, t = t_{max}) \quad (\text{Eq 6 - 3.2.12})$$

$$q_{B1}(z, t = t_{max} + t_b) = q_{B1}(z, t = t_{max}) \quad (\text{Eq 6 - 3.2.13})$$

write in dimensionless form:

$$Y_{A1}(x, \tau) \Big|_{P=P_H, t=t_{max}+t_b} = Y_{A1}(x, \tau) \Big|_{P=P_L, t=t_{max}} \quad (\text{Eq 6 - 3.2.14})$$

$$Y_{B1}(x, \tau) \Big|_{P=P_H, t=t_{max}+t_b} = Y_{B1(1)}(x, \tau) \Big|_{P=P_L, t=t_{max}} \quad (\text{Eq 6 - 3.2.15})$$

6 - 3.3 Step 3

This step is virtually the same as step 1, except that here the high-pressure feed flows through bed 1 while a portion of the exit product (N_2) purges bed 2 at the low operation pressure. The equations which have been solved to describe this step are identical to those solved for step 1, but with the change in the direction of flow taken into account.

6 - 3.4 Step 4

The approximations made for step 2 are valid for step 4, the difference being that bed 1 is now subjected to pressure reduction (Blowdown) and bed 2 is pressurized.

6 - 3.5 Cyclic Steady-State:

The above equations were written in dimensionless form and solved to give the O_2 composition, C_{A1}/C_{HP} , C_{A1}/C_{LP} (step 1), C_{A2}/C_{LP} , C_{A2}/C_{HP} (step 3) in the fluid phase, and O_2 and N_2 concentration in solid phase, $q_{A2}/K_A C_{HP}$, $q_{B2}/K_B C_{HP}$ (step 1), $q_{A1}/K_A C_{HP}$, $q_{B1}/K_B C_{HP}$ (step 3), as functions of dimensionless bed distance (z/L) at various times t . The final cyclic steady-state profiles were determined by continuing the simulation for sufficient cycles until no significant further changes in the profiles occurred. Mathematically, we use formula:

$$\frac{\left| \sum_i^N X_{A2}(x_i, \tau) \Big|_{\text{cycle}=j, t=t_{max}} - \sum_i^N X_{A2}(x_i, \tau) \Big|_{\text{cycle}=j-1, t=t_{max}} \right|}{\sum_i^N X_{A2}(x_i, \tau) \Big|_{\text{cycle}=j, t=t_{max}}} < \delta \quad (\text{Eq 6 - 3.5.1})$$

here, we set $\delta = 0.01$, N is the total number of collocation points in x domain.

6 - 3.6 Simulation:

By repeating the simulation with different sets of parameters the effects of the purge/feed ratio ($G=v_{OL}/v_{OH}$), the bed length/inlet feed velocity ($\alpha=L/v_{OH}$), the time duration (t_{max}) of the adsorption and desorption steps and the axial Peclet number ($Pe=v_{OH}L/D_L$) were studied.

6 - 4 Double Collocation method:

The current mathematical model no longer fits in PDE-ODE procedure, since application of this procedure would lead to sets of ODEs together with sets of nonlinear algebraic equations. Such a combination of ODE and nonlinear algebraic equations is likely to cause stiffness and convergence problems. Therefore, the method of double collocation was applied so that all equations would be reduced to sets of nonlinear algebraic equations. We first studied this method.

The method of double collocation involves discretization of the spatial as well as the time derivatives in the equations (as shown in Figure 6 - 4.1). The resulting nonlinear algebraic equations are then solved using a standard routine. Since our problem is non-symmetric, we use nonsymmetric trial function of shifted Legendre polynomials to approximate the dimensionless variables.

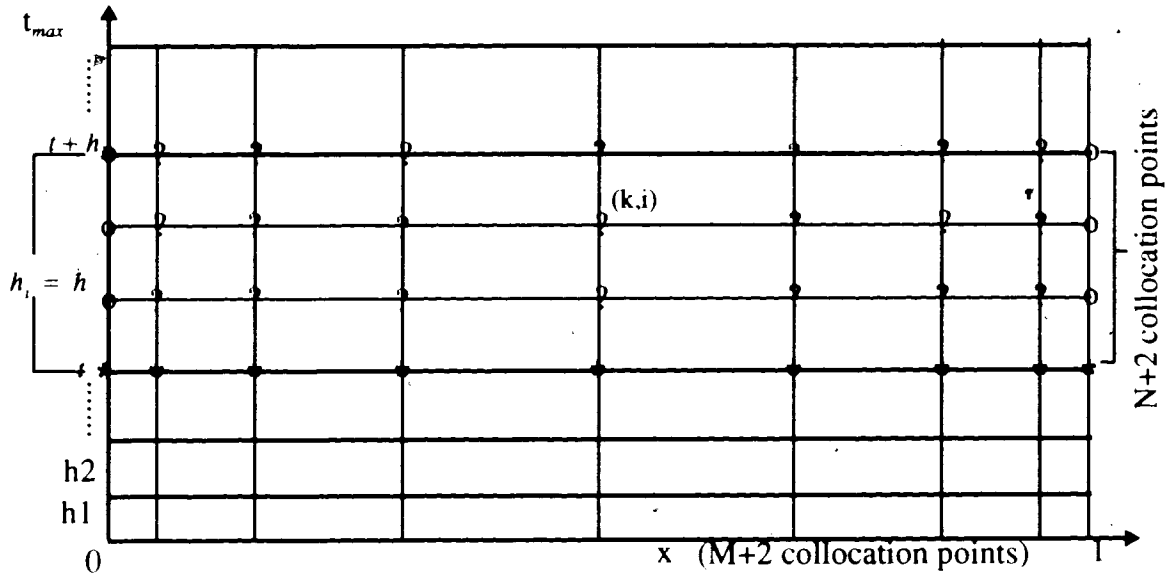


Figure 6 - 4.1 space difference and time discretization

In cyclic operation, the computation will run from step 1, through step 2, step 3, to step 4 repetitively. From the above mathematical model description, we know that step 1 and step 3 are basically identical, and step 2 and step 4 are identical. Since step 2 and step 4 are only one step calculation, the major computational task will be step 1 and step 3.

Therefore, we will focus on step 1. With dimensionless form, we will solve set of equations from (Eq 6 - 3.1.21) to (Eq 6 - 3.1.33) for time domain 0 to t_{max} . As shown in **Figure 6 - 4.1**, we subdivide the time domain into various time steps h_1, h_2, \dots, h_n . At each time step, h_i , we derive the above equations. And it is in this time period, we apply the double collocation method. First we subdivide the time period h_i into $N+2$ collocation points, and then subdivide the space domain into $M+2$ collocation points. The first time layer, i.e. $i = 1$, the variables on the points of this layer are the initial condition (marked as *), which are also the results from last step h_{i-1} . With the boundary condition (marked as O), we can solve the variables on the points at the rest of layers (marked as ?) by the double collocation method. In the following section, we describe the use of double collocation method for solving the sets of equations at this time period.

6 - 4.1 Preparation:

Assuming the symbol ϕ stands for any one of the variables, $X_{A1}, X_{A2}, Y_{A1}, Y_{A2}, Y_{B1}, Y_{B2}, V_1, V_2$, we may write,

$$\left. \frac{\partial \phi}{\partial x} \right|_{k_i} = \sum_{j=1}^{M+2} A^x_{kj} \phi_{j_i} \quad (\text{Eq 6 - 4.1.1})$$

$$\left. \frac{\partial^2 \phi}{\partial x^2} \right|_{k_i} = \sum_{j=1}^{M+2} B^x_{kj} \phi_{j_i} \quad (\text{Eq 6 - 4.1.2})$$

$$\left. \frac{\partial \phi}{\partial \tau} \right|_{ki} = \sum_{j=1}^{N+2} A_{ij}^{\tau} \phi_{ki} \quad (\text{Eq 6 - 4.1.3})$$

$$\int_0^1 \phi dx \Big|_i = \sum_{j=1}^{M+2} W_j \phi_{ji} \quad (\text{Eq 6 - 4.1.4})$$

Here, $A_{k,j}^x$ is the collocation coefficient for the gradient, $A_{k,j}^{\tau}$ is the collocation coefficient for the gradient in the time direction, $B_{k,j}^x$ is the collocation coefficient for the Laplacian, and W_j is the weight coefficient for integration. For the procedure of obtaining the coefficient matrices refer to Appendix.

The initial condition are given as:

$$\begin{aligned} X_{A2}(k, 1), k = 2 \dots M+1 \\ Y_{A2}(k, 1), k = 1 \dots M+1 \\ Y_{B2}(k, 1), k = 1 \dots M+1 \\ V_2(k, 1), k = 1 \dots M+1 \end{aligned} \quad (\text{Eq 6 - 4.1.5})$$

$$\begin{aligned} X_{A1}(k, 1), k = 2 \dots M+1 \\ Y_{A1}(k, 1), k = 1 \dots M+1 \\ Y_{B1}(k, 1), k = 1 \dots M+1 \\ V_1(k, 1), k = 1 \dots M+1 \end{aligned} \quad (\text{Eq 6 - 4.1.6})$$

6 - 4.2 Computation for one time step by double collocation:

The computation for one time step by double collocation involves four steps.

Step 1: we assume (initial guess) the values of the unknown variables. That is we provide initial guess for,

$$\begin{aligned} X_{A2}(k, i), k = 2 \dots M+1, i = 2 \dots N+2 \\ Y_{A2}(k, 1), k = 1 \dots M+1, i = 2 \dots N+2 \\ Y_{B2}(k, 1), k = 1 \dots M+1, i = 2 \dots N+2 \\ V_2(k, 1), k = 1 \dots M+1, i = 2 \dots N+2 \end{aligned} \quad (\text{Eq 6 - 4.2.1})$$

$$\begin{aligned} X_{A1}(k, 1), k = 2 \dots M+1, i = 2 \dots N+2 \\ Y_{A1}(k, 1), k = 1 \dots M+1, i = 2 \dots N+2 \\ Y_{B1}(k, 1), k = 1 \dots M+1, i = 2 \dots N+2 \\ V_1(k, 1), k = 1 \dots M+1, i = 2 \dots N+2 \end{aligned} \quad (\text{Eq 6 - 4.2.2})$$

Since the number of unknowns is equal to the number of equations, we have in all $(4*M+5)*(N+1)*2$ equations to solve simultaneously for steps 1 and 3 of the PSA simulation.

Step 2: the computation involves the evaluation of $X_{A2}(1,i)$, $X_{A2}(M+2)$, $X_{A1}(1,i)$, $X_{A2}(M+2)$. These values can be extracted based on the values of $X_{A2}(k,i)$ and $X_{A1}(k,i)$ for $k=2 \dots M+1$ and

$i=2..N+2$, for the dimensionless form of the boundary conditions given by (Eq 6 - 3.1.25)--(Eq 6 - 3.1.26) and (Eq 6 - 3.1.31)--(Eq 6 - 3.1.33) which written in collocation form would be:

For high-pressure flow in bed 2 during step 1,

$$\sum_{j=1}^{M+2} A^x(1, j) X_{A_2}(j, i) = (-Pe_H) [X_{A_2}|_{x=0^-} - X_{A_2}(1, i)] \quad (\text{Eq 6 - 4.2.3})$$

$$\sum_{j=1}^{M+2} A^x(M+2, j) X_{A_2}(j, i) = 0 \quad (\text{Eq 6 - 4.2.4})$$

Solving (Eq 6 - 4.2.3) and (Eq 6 - 4.2.4), we get the expressions for the boundary variables,

$$\Phi_H = A^x(M+2, M+2) (A^x(1, 1) - Pe_H) - A^x(1, M+2) A^x(M+2, 1) \quad (\text{Eq 6 - 4.2.5})$$

$$R_{1H} = \frac{A^x(M+2, 1)}{\Phi_H}, \quad R_{2H} = \frac{A^x(1, 1) - Pe_H}{\Phi_H} \quad (\text{Eq 6 - 4.2.6})$$

$$R_{3H} = \frac{A^x(1, M+2)}{\Phi_H}, \quad R_{4H} = \frac{A^x(M+2, M+2)}{\Phi_H} \quad (\text{Eq 6 - 4.2.7})$$

$$X_{A_2}(1, i) = -Pe_H X_{A_2}|_{x=0^-} R_{4H} - \sum_{j=2}^{M+1} A^x(1, j) X_{A_2}(j, i) R_{4H} + \sum_{j=2}^{M+1} A^x(M+2, j) X_{A_2}(j, i) R_{3H} \quad (\text{Eq 6 - 4.2.8})$$

$$X_{A_2}(M+2, i) = Pe_H X_{A_2}|_{x=0^-} R_{1H} + \sum_{j=2}^{M+1} A^x(1, j) X_{A_2}(j, i) R_{1H} - \sum_{j=2}^{M+1} A^x(M+2, j) X_{A_2}(j, i) R_{2H} \quad (\text{Eq 6 - 4.2.9})$$

and for low-pressure flow in bed 1,

$$\sum_{j=1}^{M+2} A^x(1, j) X_{A_1}(j, i) = (-GPe_L) [X_{A_1}|_{x=0^-} - X_{A_1}(1, i)] \quad (\text{Eq 6 - 4.2.10})$$

$$\sum_{j=1}^{M+2} A^x(M+2, j) X_{A_1}(j, i) = 0 \quad (\text{Eq 6 - 4.2.11})$$

solving (Eq 6 - 4.2.10) and (Eq 6 - 4.2.11), we get,

$$\Phi_L = A^x(M+2, M+2) (A^x(1, 1) - GPe_L) - A^x(1, M+2) A^x(M+2, 1) \quad (\text{Eq 6 - 4.2.12})$$

$$R_{1L} = \frac{A^x(M+2, 1)}{\Phi_L}, \quad R_{2L} = \frac{A^x(1, 1) - GPe_L}{\Phi_L} \quad (\text{Eq 6 - 4.2.13})$$

$$R_{3L} = \frac{A^x(1, M+2)}{\Phi_L}, \quad R_{4L} = \frac{A^x(M+2, M+2)}{\Phi_L} \quad (\text{Eq 6 - 4.2.14})$$

$$X_{A_1}(1, i) = (-GPe_L) X_{A_1}|_{x=0^-} R_{4L} - \sum_{j=2}^{M+1} A^x(1, j) X_{A_1}(j, i) R_{4L} + \sum_{j=2}^{M+1} A^x(M+2, j) X_{A_1}(j, i) R_{3L} \quad (\text{Eq 6 - 4.2.15})$$

$$X_{A1}(M+2, i) = GPe_L X_{A1}|_{x=0^-} R_{1L} + \sum_{j=2}^{M+1} A^x(1; j) X_{A1}(j, i) R_{1L} - \sum_{j=2}^{M+1} A^x(M+2, j) X_{A1}(j, i) R_{2L} \quad (\text{Eq 6 - 4.2.16})$$

where, by (Eq 6 - 3.1.32)

$$X_{A1}|_{x=0^-} = X_{A2}(M+2, i) \quad (\text{Eq 6 - 4.2.17})$$

Step 3: knowing now the values of the variables at all points in the two beds, we compute $\frac{\partial \phi}{\partial x}|_{k,i}$, $\frac{\partial^2 \phi}{\partial x^2}|_{k,i}$ and $\frac{\partial \phi}{\partial \tau}|_{k,i}$.

Step 4: involves setting up the residuals $R^{\phi}_{k,i}$ and updating the initial guess to force the $(4*M+5)*4*2$ residuals to become zero.

For $X_{A2}, Y_{A2}, Y_{B2}, V_2$ during high-pressure flow adsorption in bed 2 (step 1) we write,

$$R^{X_{A2}}_{k,i} = \frac{\alpha}{h} \sum_{j=1}^{N+2} A^{\tau}_{i,j} X_{A2}(k, j) - \frac{1}{Pe_H} \sum_{j=1}^{M+2} B^{\tau}_{k,j} X_{A2}(j, i) + V_2(k, i) \sum_{j=1}^{M+2} A^x_{k,j} X_{A2}(j, i) + \frac{1-\epsilon}{\epsilon} \alpha k_A K_A [X_{A2}(k, i) - Y_{A2}(k, i)] (1 - X_{A2}(k, i)) - \frac{1-\epsilon}{\epsilon} \alpha k_B K_B X_{A2}(k, i) [(1 - X_{A2}(k, i)) - Y_{B2}(k, i)] = 0 \quad (\text{Eq 6 - 4.2.18})$$

where $k = 2 \dots M+1; i = 2 \dots N+2$

$$R^{Y_{A2}}_{k,i} = \sum_{j=1}^{N+2} A^{\tau}_{i,j} Y_{A2}(k, j) - h k_A [X_{A2}(k, i) - Y_{A2}(k, i)] = 0 \quad (\text{Eq 6 - 4.2.19})$$

where $k = 1 \dots M+2, i = 2 \dots N+2$

$$R^{Y_{B2}}_{k,i} = \sum_{j=1}^{N+2} A^{\tau}_{i,j} Y_{B2}(k, j) - h k_B [(1 - X_{A2}(k, i)) - Y_{A2}(k, i)] = 0 \quad (\text{Eq 6 - 4.2.20})$$

where $k = 1 \dots M+2, i = 2 \dots N+2$

$$R^{V_2}_{k,i} = \sum_{j=1}^{M+2} A^x_{k,j} V_2(j, i) + \frac{1-\epsilon}{\epsilon} \alpha k_A K_A [X_{A2}(k, i) - Y_{A2}(k, i)] + \frac{1-\epsilon}{\epsilon} \alpha k_B K_B [(1 - X_{A2}(k, i)) - Y_{A2}(k, i)] = 0 \quad (\text{Eq 6 - 4.2.21})$$

where $k = 2 \dots M+2; i = 2 \dots N+2$.

Similarly, equations for $X_{A1}, Y_{A1}, Y_{B2}, V_1$ are written in collocation form for the low-pressure purge flow in bed 1 (step 1).

$$\begin{aligned}
R_{k,i}^{X_{A1}} = & \frac{\alpha}{h} \sum_{j=1}^{N+2} A_{i,j}^{\tau} X_{A1}(k,j) - \frac{1}{Pe_L} \sum_{j=1}^{M+2} B_{k,j}^x X_{A1}(j,i) + V_1(k,i) \sum_{j=1}^{M+2} A_{k,j}^x X_{A1}(j,i) \\
& + \frac{1-\varepsilon}{\varepsilon} \alpha k_A K_A \beta \left(\frac{X_{A1}(k,i)}{\beta} - Y_{A1}(k,i) \right) (1.0 - X_{A1}(k,i)) \\
& - \frac{1-\varepsilon}{\varepsilon} \alpha k_B K_B \beta X_{A1}(k,i) \left(\frac{(1.0 - X_{A2}(k,i))}{\beta} - Y_{B1}(k,i) \right) = 0
\end{aligned} \tag{Eq 6 - 4.2.22}$$

where $k = 2 \dots M+1$; $i = 2 \dots N+2$

$$R_{k,i}^{Y_{A1}} = \sum_{j=1}^{N+2} A_{i,j}^{\tau} Y_{A1}(k,j) - h k_A \left(\frac{X_{A1}(k,i)}{\beta} - Y_{A1}(k,i) \right) = 0 \tag{Eq 6 - 4.2.23}$$

where $k = 1 \dots M+2$, $i = 2 \dots N+2$

$$R_{k,i}^{Y_{B1}} = \sum_{j=1}^{N+2} A_{i,j}^{\tau} Y_{B1}(k,j) - h k_B \left(\frac{1 - X_{A1}(k,i)}{\beta} - Y_{A1}(k,i) \right) = 0 \tag{Eq 6 - 4.2.24}$$

where $k = 1 \dots M+2$, $i = 2 \dots N+2$

$$\begin{aligned}
R_{k,i}^{V_1} = & \sum_{j=1}^{M+2} A_{k,j}^x V_1(j,i) + \frac{1-\varepsilon}{\varepsilon} \alpha k_A K_A \beta \left(\frac{X_{A1}(k,i)}{\beta} - Y_{A1}(k,i) \right) \\
& + \frac{1-\varepsilon}{\varepsilon} \alpha k_B K_B \beta \left(\frac{1 - X_{A2}(k,i)}{\beta} - Y_{A2}(k,i) \right) = 0
\end{aligned} \tag{Eq 6 - 4.2.25}$$

where $k = 2 \dots M+2$; $i = 2 \dots N+2$.

A standard subroutine such as Newton method is used to solve the nonlinear algebraic equations to get the distribution of X_{A2} , Y_{A2} , Y_{B2} , V_2 and X_{A1} , Y_{A1} , Y_{B2} , V_1 in the two beds at $\tau = 1$.

The procedure from step 1 to step 4 is repeated until $t = t_{max}$ = duration of time for the first step (adsorption) in bed 2 and desorption (purging in bed 1). So far, we finish the simulation of step 1 of PSA cycle.

After the approximations for step 2 of the PSA, the nonlinear algebraic equations for the step 3 [adsorption in bed 1 and desorption (purging) in bed 2] are solved, which is followed by the approximations for step 4. This essentially completes the computations for one cycle of the PSA operation. The computation for the next cycle involves the solution of the nonlinear algebraic equations for step 1 with the initial conditions based on the profile in the beds at the end of the previous cycle. The computation continues until cyclic steady state is reached, when there is no further change in the composition profiles in the beds between two successive cycles.

6 - 5 The Computation Results and Discussion

6 - 5.1 The sensitivity of the assumptions for repressurization to the number of collocation points

The simulation was run with various number of collocation points, and it was found that the assumptions for repressurization are sensitive to the number of collocation points. Since the

result of repressurization will affect the next step of adsorption, therefore, it will affect the result of cyclic operation. To gain more understanding of this phenomenon, we did an experiment. Assume the concentration of repressurized oxygen at inlet is 0.2, and the concentration of oxygen at the rest of bed is 0.05. The total mass after exact integration would be 0.15; comparing with the integration result by collocation method, we find the difference, listed in Table 6 - 5.1.

Table 6 - 5.1 Collocation points with Repressurization

number of collocation points	difference in total mass
m=6	-0.025
m=7	-0.0037
m=8	0.01
m=9	-0.00925
m=10	0.0022
m=11	0.0107
m=12	-0.002836

It can be seen from the above table, the sampling by some numbers of collocation points can best represent the assumption of repressurization step such as 7 and 12. It would be the best if we could do simulation with m=12. However, for reason of economy, we can also take m=7 as compromise between accuracy and speed. Generally, we will run simulations with m=7.

6 - 5.2 Observation of Mass balance

One feature of this program is the implementation of observation of mass balance during the adsorption step. The idea of the mass balance monitor is:

$$\begin{aligned} &(\text{change in mass of oxygen in the bed since time } t=0) = \\ &(\text{mass of oxygen entering since } t=0) - (\text{mass of oxygen leaving since } t=0) \end{aligned}$$

Therefore, if we set variable R as oxygen mass balance monitor,

$$\begin{aligned} R = &(\text{change in mass of oxygen in the bed since time } t=0) - \\ &(\text{mass of oxygen entering since } t=0) + (\text{mass of oxygen leaving since } t=0) = 0 \end{aligned}$$

The detailed implementation of mass monitor involves integrals variables such as X_1, X_2, Y_1, Y_2 along the adsorption bed. Refer Appendix for the computation of integral.

From observation of the ratio of R to total mass of oxygen entering after the cyclic operation reaches steady state, we are able to verify the simulation result. For example, with mesh=7, run simulation with data listed in row 2 of Table 6 - 5.1, the ratio is 0.12%; again with mesh=12, for the same simulation, the ratio is 0.0059%, which also means more accurate results.

6 - 5.3 Output of the Simulation by Polynomials

The outputs of our simulation are the values at collocation points. However, we sometime would prefer to see the evolution inside the bed, therefore, we also express the profile of various variables in the form of polynomials.

$$\phi(x, \tau) = (1-x)\phi(0, \tau) + x\phi(1, \tau) + x(1-x) \sum_{i=1}^M a_i(\tau) P_{i-1}(x) \quad (\text{Eq 6 - 5.1})$$

where $P_i(x)$ are the shifted polynomials. $\phi(x, \tau)$ can be variables such as, $X_{A2}, Y_{A2}, Y_{B2}, V_2$. For details of the implementation, refer to Appendix.

In Figure 6 - 5.1, Figure 6 - 5.2, Figure 6 - 5.3, we have shown the profiles of these variables at the end of step 1 of each cycle. The simulation of cyclic operation takes 9 cycles to reach the steady state, and the total computation time is 19:59 minutes.

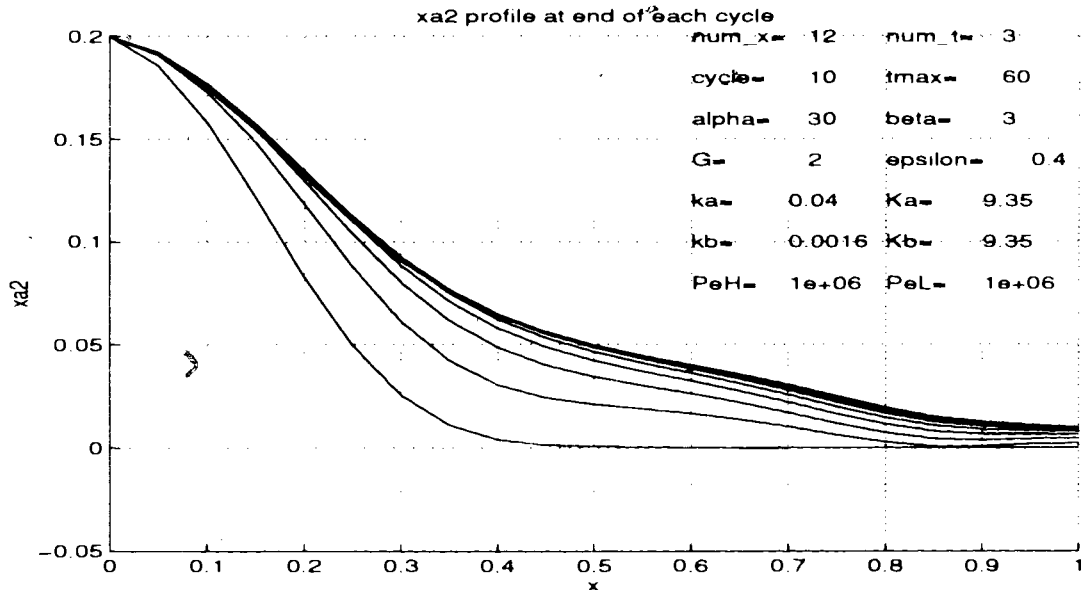


Figure 6 - 5.1 Gas phase composition profiles of oxygen in bed 2 at end of high-pressure step.
Parameter values as for ratio G (row 2, Table 6 - 5.1)

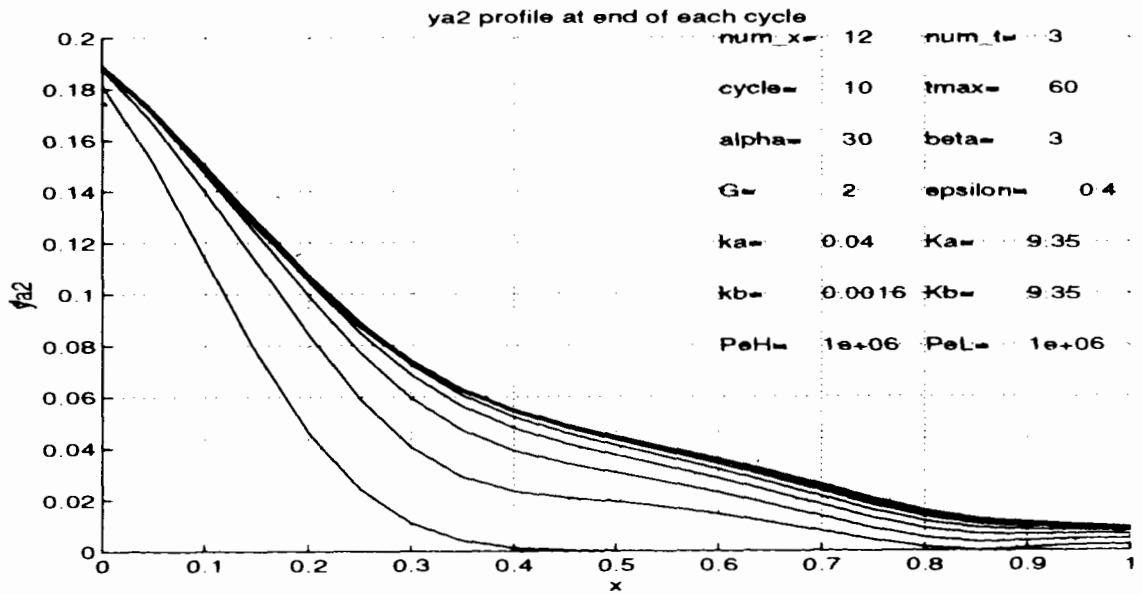


Figure 6 - 5.2 Solid-phase concentration profiles of oxygen in bed 2 at the end of high-pressure. Parameter values as for ratio G (row 2, Table 6 - 5.1)

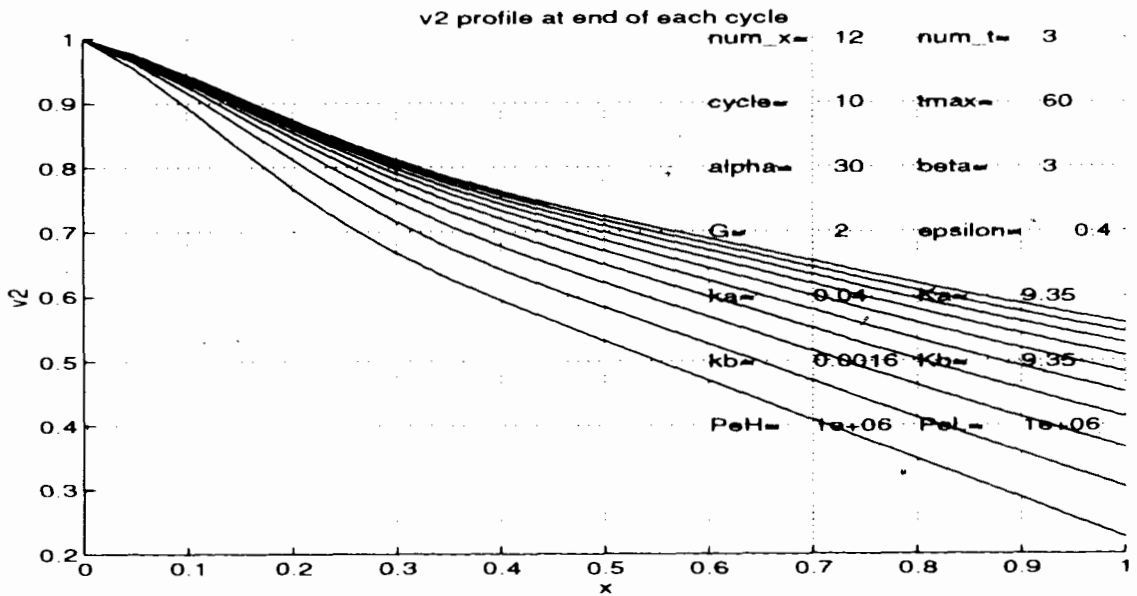


Figure 6 - 5.3 Fluid velocity as a function of distance in bed 2 at end of high-pressure step. Parameter values as for ratio G (row 2, Table 6 - 5.1).

6 - 5.4 Result of Steady-state PSA operation

In the next table, we list the result at the end of step 1 of steady-state PSA operation.

Table 6 - 5.1 Exit Composition of Oxygen, and Exit Fluid Velocity in Bed 2 at End of Step 1 of Steady-State PSA operation Assuming Solid Phase Concentrations of O₂ and N₂ Frozen During Blowdown

α	G	t_{max}	k_A	k_B	K_A	K_B	Pe_H	Pe_L	X_{A2}	V_2
20	2	60	0.040	0.0016	9.35	9.35	1.0e6	1.0e6	0.015998	0.654973
30	2	60	0.040	0.0016	9.35	9.35	1.0e6	1.0e6	0.009403	0.578583
40	2	60	0.040	0.0016	9.35	9.35	1.0e6	1.0e6	0.008360	0.510112
30	1	60	0.040	0.0016	9.35	9.35	1.0e6	1.0e6	0.026307	0.622295
30	2	60	0.040	0.0016	9.35	9.35	1.0e6	1.0e6	0.009403	0.578583
30	3	60	0.040	0.0016	9.35	9.35	1.0e6	1.0e6	0.006109	0.478815
30	2	30	0.040	0.0016	9.35	9.35	1.0e6	1.0e6	0.026484	0.523984
30	2	60	0.040	0.0016	9.35	9.35	1.0e6	1.0e6	0.009403	0.587583
30	2	90	0.040	0.0016	9.35	9.35	1.0e6	1.0e6	0.006071	0.570014
30	2	120	0.040	0.0016	9.35	9.35	1.0e6	1.0e6	0.005570	0.619394
30	2	180	0.040	0.0016	9.35	9.35	1.0e6	1.0e6	0.008284	0.655468
30	2	240	0.040	0.0016	9.35	9.35	1.0e6	1.0e6	0.014452	0.683554
30	2	60	0.040	0.0016	9.35	9.35	100	100	0.011505	0.545612
30	2	60	0.040	0.0016	9.35	9.35	10	10	0.029940	0.556392
30	2	60	0.0026	0.000106	9.35	9.35	1.0e6	1.0e6	0.145532	0.881673

6 - 5.5 Computation Time.

For the parameters listed in row 2 of Table 6-5.2, we run the simulation with different mesh number and list the computation time.

Table 6 - 5.1 Collocation points with Repressurization

number of mesh	computation time (minute: second)	number of cycles to reach steady state
m=6	2:40	11
m=7	3:31	9
m=8	5:29	9
m=9	9:27	10
m=10	12:40	10
m=11	17:00	10
m=12	19:35	9
m=13	26:40	9
m=14	41:24	10
m=15	48:21	10

6 - 6 Conclusion:

As conclusion of this chapter, double collocation method can be used to solve the complex mathematical model of combination of P.D.Es and O.D.Es. Comparing our results with Raghavan's, there is good agreement between them, therefore, we have successfully reproduced the numerical simulation of a PSA cyclic operation with double collocation method. The purpose of the work in this chapter is to serve a base for our modelling of PSA process in next chapter, in which we will also need to observe the variation of temperature of PSA operation.

Chapter 7 Simulation of A Conventional PSA Process

7 - 1 Theoretical Model

In this chapter, we will simulate the adsorption column of a conventional PSA process. In order to build a more realistic model, we will make fewer assumptions than in last chapter. Consequently, the mathematical model becomes more complicated, and more difficult to solve. We will continue to use the double collocation method, introduced in Chapter 6, to do the numerical simulation.

The following assumptions are made:

1. Conventional PSA, in which the flow boundary conditions at the end of the bed and the imposed pressure variations are specified.
2. Assume there are two adsorbable gases present, adsorbable to different extents.
3. We further assume that the bed may not be isothermal, so we must solve the energy equation to obtain the local temperature. Heat effect is considered in this model.
4. Pressure drop along x direction is negligible.
5. The flow velocity is not constant, but varies along the bed and with time.
6. The equilibrium relationships are represented by the linear isotherm.

Therefore, in this PSA model, there are three variations: concentration (bulk gas and solid phase), temperature and velocity of the flow.

This is the first stage of study of a conventional PSA cycle, since at present we're primarily interested in the adsorption step. The solution developed here can easily be extended to cyclic operation by changing the boundary conditions. Therefore, we have further description: the bed runs through a cycle; there are four or five distinct phases, each with particular flow and pressure boundary conditions.

7 - 2 Mathematical Model

7 - 2.1 Dimensional Form

Conservation of species:

$$-D_{11} \frac{\partial^2 c_1}{\partial z^2} + \frac{\partial v c_1}{\partial z} + \frac{\partial c_1}{\partial t} = -D_1 \frac{1-\epsilon}{\epsilon} (B_1 c_1 - q_1) \quad (\text{Eq 7 - 2.1.1})$$

$$-D_{12} \frac{\partial^2 c_2}{\partial z^2} + \frac{\partial v c_2}{\partial z} + \frac{\partial c_2}{\partial t} = -D_2 \frac{1-\epsilon}{\epsilon} (B_2 c_2 - q_2) \quad (\text{Eq 7 - 2.1.2})$$

$$\frac{\partial q_1}{\partial t} = D_1 (B_1 c_1 - q_1) \quad (\text{Eq 7 - 2.1.3})$$

$$\frac{\partial q_2}{\partial t} = D_2 (B_2 c_2 - q_2) \quad (\text{Eq 7 - 2.1.4})$$

Temperature dependence of isotherm equation:

$$B_1 = B_{10} \exp\left(\frac{H_1}{RT} - \frac{H_1}{RT_a}\right) \quad (\text{Eq 7 - 2.1.5})$$

$$B_2 = B_{20} \exp\left(\frac{H_2}{RT} - \frac{H_2}{RT_a}\right) \quad (\text{Eq 7 - 2.1.6})$$

Where T_a is the temperature at which the value of B_{10} and B_{20} was calculated (usually 300 K). Note that if T is greater than T_a then B_1 and B_2 will be less than B_{10} and B_{20} respectively. That is, adsorption gets poorer as the temperature rises. Since we are modelling a bed that is at about 300K, the values of B_1 and B_2 will be approximately equal to B_{10} and B_{20} respectively.

Conservation of energy

$$\sum_i \left(c_{pi} \frac{\partial v c_i T}{\partial z} + c_{vi} \frac{\partial c_i T}{\partial t} - H_i D_i \frac{1-\epsilon}{\epsilon} (B_i c_i - q_i) \right) + c_s \frac{1-\epsilon}{\epsilon} \rho_s \frac{\partial T}{\partial t} + k_{eq} (T - T_w) = 0 \quad (\text{Eq 7 - 2.1.7})$$

Where T_w is the bed wall temperature, it may be equal to T_a , but need not be, in this work we set $T_w = T_a$.

This equation may be simplified if the specific heats of the two species of gases are sufficiently close; if we write c_p for the mean specific heat at constant pressure, at c_v for the mean specific heat at constant volume, we have:

$$c_p \frac{\partial v c T}{\partial z} + c_v \frac{\partial c T}{\partial t} - \sum_i H_i D_i \frac{1-\epsilon}{\epsilon} (B_i c_i - q_i) + c_s \frac{1-\epsilon}{\epsilon} \rho_s \frac{\partial T}{\partial t} + k_{eq} (T - T_w) = 0 \quad (\text{Eq 7 - 2.1.8})$$

where $c = c_1 + c_2$ is the total concentration of the two species - that is, c is the density of the mixture. This equation can be further simplified by noting that the ideal gas equation gives us:

$$P = R c T \quad (\text{Eq 7 - 2.1.9})$$

and since

$$\frac{\partial P}{\partial z} = 0, \quad (\text{Eq 7 - 2.1.10})$$

$$c_p \frac{P}{R} \frac{\partial v}{\partial z} + \frac{c_v}{R} \frac{\partial p}{\partial t} - \sum_i H_i D_i \frac{1-\epsilon}{\epsilon} (B_i c_i - q_i) + c_s \frac{1-\epsilon}{\epsilon} \rho_s \frac{\partial T}{\partial t} + k_{eq} (T - T_w) = 0 \quad (\text{Eq 7 - 2.1.11})$$

where $\frac{\partial P}{\partial t}$ is supplied as a boundary condition.

Derivation based on the above equations:

Differentiate $P = R c T$ on both sides, we have:

$$\frac{\partial P}{\partial t} = R c \frac{\partial T}{\partial t} + R T \frac{\partial c}{\partial t} \quad (\text{Eq 7 - 2.1.12})$$

$$\frac{\partial c}{\partial t} = \frac{1}{R T} \frac{\partial P}{\partial t} - \frac{c}{T} \frac{\partial T}{\partial t} \quad (\text{Eq 7 - 2.1.13})$$

adding (Eq 7 - 2.1.1) and (Eq 7 - 2.1.2) and considering (Eq 7 - 2.1.3), (Eq 7 - 2.1.4), and the

above equation,

$$-D_{11} \frac{\partial^2 c_1}{\partial z^2} - D_{12} \frac{\partial^2 c_2}{\partial z^2} + \frac{\partial v c}{\partial z} + \left(\frac{1}{RT} \frac{\partial P}{\partial t} - \frac{c}{T} \frac{\partial T}{\partial t} \right) = -D_1 \frac{1-\epsilon}{\epsilon} (B_1 c_1 - q_1) - D_2 \frac{1-\epsilon}{\epsilon} (B_2 c_2 - q_2) \quad (\text{Eq 7 - 2.1.14})$$

Therefore, we will have the above set of six equations (Eq 7 - 2.1.1), (Eq 7 - 2.1.2), (Eq 7 - 2.1.3), (Eq 7 - 2.1.4) and (Eq 7 - 2.1.11)(Eq 7 - 2.1.14) for solving the variables c_1, c_2, q_1, q_2, T, v along with two auxiliary equations (Eq 7 - 2.1.5) and (Eq 7 - 2.1.6) to find B_1 and B_2 .

Initial condition

For an initial condition, we can assume that the fraction of nitrogen in the gas in bed varies linearly from 80% at the feed end to 0% at the other end.

The gases in the adsorbed phase are in local equilibrium with the gases in the gas phase.

$$c_1(z, 0) = c_{10} \left(\frac{L-z}{L} \right) \quad (\text{Eq 7 - 2.1.15})$$

$$c_2(z, 0) = c_0 - c_1(z, 0) \quad (\text{Eq 7 - 2.1.16})$$

$$q_1(z, 0) = B_{10} c_1(z, 0) \quad (\text{Eq 7 - 2.1.17})$$

$$q_2(z, 0) = B_{20} c_2(z, 0) \quad (\text{Eq 7 - 2.1.18})$$

$$v(z, 0) = v_0 \quad (\text{Eq 7 - 2.1.19})$$

$$T(z, 0) = T_a \quad (\text{Eq 7 - 2.1.20})$$

Boundary condition:

The boundary conditions for the variables in dimensional forms are listed as:

$$D_{11} \frac{\partial c_1}{\partial z} \Big|_{z=0,t} = -v_0 (c_1|_{z=0} - c_1|_{z=0'}) \quad (\text{Eq 7 - 2.1.21})$$

$$\frac{\partial c_1}{\partial z} \Big|_{z=L,t} = 0 \quad (\text{Eq 7 - 2.1.22})$$

$$D_{12} \frac{\partial c_2}{\partial z} \Big|_{z=0,t} = -v_0 (c_2|_{z=0} - c_2|_{z=0'}) \quad (\text{Eq 7 - 2.1.23})$$

$$\frac{\partial c_2}{\partial z} \Big|_{z=L,t} = 0 \quad (\text{Eq 7 - 2.1.24})$$

$$\frac{\partial P}{\partial t} = 0 \quad (\text{Eq 7 - 2.1.25})$$

$$T(z, t) \Big|_{z=0,t} = T_a \quad (\text{Eq 7 - 2.1.26})$$

$$v(z, t) \Big|_{z=0,t} = v_0 \quad (\text{Eq 7 - 2.1.27})$$

7 - 2.2 Dimensionless Form

The above-mentioned six equations can be converted into their dimensionless form.

7 - 2.2.1 Define dimensionless variables in Table 7-2.2.1:

Table 7 - 2.2.1 Dimensionless Variables

dimensionless variables	definition
X_1	c_1/c_0
X_2	c_2/c_0
Y_1	$q_1/B_{10}c_0$
Y_2	$q_2/B_{20}c_0$
\bar{P}	P/P_0 , where $P_0 = RT_a c_0$
\bar{T}	T/T_a
\bar{v}	v/v_0
x	z/L
τ	t/h , where h is dimensional time interval, refer Figure 6 - 4.1

7 - 2.2.2 Equations in dimensionless form:

(Eq 7 - 2.1.5) and (Eq 7 - 2.1.6) can be rewritten as:

$$\beta_1 = \frac{B_1}{B_{10}} = \exp\left(\frac{H_1}{RT_a} \left(\frac{1}{\bar{T}} - 1\right)\right) \quad (\text{Eq 7 - 2.2.1})$$

$$\beta_2 = \frac{B_2}{B_{20}} = \exp\left(\frac{H_2}{RT_a} \left(\frac{1}{\bar{T}} - 1\right)\right) \quad (\text{Eq 7 - 2.2.2})$$

(Eq 7 - 2.1.3) and (Eq 7 - 2.1.4) can be rewritten as:

$$\frac{\partial Y_1}{\partial \tau} = hD_1 (\beta_1 X_1 - Y_1) \quad (\text{Eq 7 - 2.2.3})$$

$$\frac{\partial Y_2}{\partial \tau} = hD_2 (\beta_2 X_2 - Y_2) \quad (\text{Eq 7 - 2.2.4})$$

(Eq 7 - 2.1.1) and (Eq 7 - 2.1.2) can be written as:

$$\frac{\alpha}{h} \frac{\partial X_1}{\partial \tau} - \frac{1}{Pe_1} \frac{\partial^2 X_1}{\partial x^2} + \frac{\partial \bar{v} X_1}{\partial x} = -\alpha D_1 \frac{1-\epsilon}{\epsilon} B_{10} (\beta_1 X_1 - Y_1) \quad (\text{Eq 7 - 2.2.5})$$

$$\frac{\alpha}{h} \frac{\partial X_2}{\partial \tau} - \frac{1}{Pe_2} \frac{\partial^2 X_2}{\partial x^2} + \frac{\partial \bar{v} X_2}{\partial x} = -\alpha D_2 \frac{1-\epsilon}{\epsilon} B_{20} (\beta_2 X_2 - Y_2) \quad (\text{Eq 7 - 2.2.6})$$

(Eq 7 - 2.1.11) can be written as:

$$c_p \bar{P} \frac{\partial \bar{v}}{\partial x} + \frac{\alpha}{h} c_v \frac{\partial \bar{P}}{\partial \tau} - \frac{H_1}{T_a} \alpha D_1 \frac{1-\epsilon}{\epsilon} B_{10} (\beta_1 X_1 - Y_1) - \frac{H_2}{T_a} \alpha D_2 \frac{1-\epsilon}{\epsilon} B_{20} (\beta_2 X_2 - Y_2) + \frac{\alpha}{h} \frac{1-\epsilon}{c_0} \frac{1-\epsilon}{\epsilon} \rho_s \frac{\partial \bar{T}}{\partial \tau} + \alpha k_{eq} \frac{1}{c_0} (\bar{T} - 1) = 0 \quad (\text{Eq 7 - 2.2.7})$$

(Eq 7 - 2.1.14) can be written as:

$$-\frac{1}{Pe_1} \frac{\partial^2 X_1}{\partial x^2} - \frac{1}{Pe_2} \frac{\partial^2 X_2}{\partial x^2} + \frac{\alpha}{h} \frac{1}{\bar{T}} \frac{\partial \bar{P}}{\partial \tau} - \frac{\alpha}{h} \frac{(X_1 + X_2)}{\bar{T}} \frac{\partial \bar{T}}{\partial \tau} + \frac{\partial \bar{v}(X_1 + X_2)}{\partial x} = -\alpha D_1 \frac{1-\epsilon}{\epsilon} B_{10} (\beta_1 X_1 - Y_1) - \alpha D_2 \frac{1-\epsilon}{\epsilon} B_{20} (\beta_2 X_2 - Y_2) \quad (\text{Eq 7 - 2.2.8})$$

So far, we have all the dimensionless equations from (Eq 7 - 2.2.1)--(Eq 7 - 2.2.8) for the dimensionless variables $X_1, X_2, Y_1, Y_2, \bar{T}, \bar{v}$.

7 - 2.2.3 Tables of dimensionless parameters

Here, we list the dimensionless parameters.

Table 7 - 2.2.2 dimensionless parameters

parameters	equations
Pe_1	$(Lv_0)/D_{11}$
Pe_2	$(Lv_0)/D_{12}$
α/h	$L/(v_0 h)$, where $\alpha = L/v_0$
hD_1	
hD_2	
ϕ_1	$\alpha D_1 \frac{1-\epsilon}{\epsilon} B_{10}$
ϕ_2	$\alpha D_2 \frac{1-\epsilon}{\epsilon} B_{20}$
α_s/c_0	$c_s \frac{1-\epsilon}{\epsilon} \rho_s/c_0$, where $\alpha_s = c_s \frac{1-\epsilon}{\epsilon} \rho_s$
$\alpha k_{eq}/c_0$	

The complete parameters used in this work are listed in section 7 - 7 at the end of this chapter. Refer (Singh & Jones, 1996) for more information.

Initial condition:

written in dimensionless form:

$$X_1(x, 0) = X_{10}(1.0 - x) \quad (\text{Eq 7 - 2.2.9})$$

$$X_2(x, 0) = 1.0 - X_1(x, 0) \quad (\text{Eq 7 - 2.2.10})$$

$$Y_1(x, 0) = X_1(x, 0) \quad (\text{Eq 7 - 2.2.11})$$

$$Y_2(x, 0) = X_2(x, 0) \quad (\text{Eq 7 - 2.2.12})$$

$$\bar{T}(x, 0) = 1.0 \quad (\text{Eq 7 - 2.2.13})$$

$$\bar{V}(x, 0) = 1.0 \quad (\text{Eq 7 - 2.2.14})$$

Boundary condition:

Rewritten in dimensionless form:

$$\left. \frac{\partial X_1}{\partial x} \right|_{x=0, \tau} = -Pe_1 (X_1|_{0^-, \tau} - X_1|_{0^+, \tau}) \quad (\text{Eq 7 - 2.2.15})$$

$$\left. \frac{\partial X_1}{\partial x} \right|_{x=1, \tau} = 0 \quad (\text{Eq 7 - 2.2.16})$$

$$\left. \frac{\partial X_2}{\partial x} \right|_{x=0, \tau} = -Pe_2 (X_2|_{0^-, \tau} - X_2|_{0^+, \tau}) \quad (\text{Eq 7 - 2.2.17})$$

$$\left. \frac{\partial X_2}{\partial x} \right|_{x=1, \tau} = 0 \quad (\text{Eq 7 - 2.2.18})$$

$$\frac{\partial \bar{P}}{\partial \tau} = 0 \quad (\text{Eq 7 - 2.2.19})$$

$$\bar{T}(x, \tau)|_{x=0, \tau} = 1.0 \quad (\text{Eq 7 - 2.2.20})$$

$$\bar{v}(x, \tau)|_{x=0, \tau} = 1.0 \quad (\text{Eq 7 - 2.2.21})$$

7 - 3 Implementation of double collocation

Four steps nonlinear calculation:

7 - 3.1 Step 1

Initial guess, based on the result of last step.

$$\begin{aligned} X_1(k, i), k = 2, \dots, M+1; i = 2, \dots, N+2; \{M(N+1)\} \\ X_2(k, i), k = 2, \dots, M+1; i = 2, \dots, N+2; \{M(N+1)\} \\ Y_1(k, i), k = 1, \dots, M+2; i = 2, \dots, N+2; \{(M+2)(N+1)\} \\ Y_2(k, i), k = 1, \dots, M+2; i = 2, \dots, N+2; \{(M+2)(N+1)\} \\ \bar{T}(k, i), k = 1, \dots, M+2; i = 2, \dots, N+2; \{(M+2)(N+1)\} \\ \bar{v}(k, i), k = 1, \dots, M+2; i = 2, \dots, N+2; \{(M+2)(N+1)\} \end{aligned} \quad (\text{Eq 7 - 3.1.1})$$

7 - 3.2 Step 2

Solve the boundary condition for X_1, X_2 and \bar{T}, \bar{v} according to the above nondimensional boundary condition equations which are written in matrix form.

$$\sum_{j=1}^{M+2} A^x(1, j) X_1(j, i) = (-Pe_1) (X_1|_0 - X_1(1, i)) \quad (\text{Eq 7 - 3.2.1})$$

$$\sum_{j=1}^{M+2} A^x(M+2, j) X_1(j, i) = 0 \quad (\text{Eq 7 - 3.2.2})$$

$$\sum_{j=1}^{M+2} A^x(1, j) X_2(j, i) = (-Pe_2) (X_2|_0 - X_2(1, i)) \quad (\text{Eq 7 - 3.2.3})$$

$$\sum_{j=1}^{M+2} A^x(M+2, j) X_2(j, i) = 0 \quad (\text{Eq 7 - 3.2.4})$$

7 - 3.3 Step 3

• Compute the derivatives for the following variables:

$$\left. \frac{\partial \phi}{\partial x} \right|_{k,i} = \sum_{j=1}^{M+2} A^x_{k,j} \phi_{j,i} \quad (\text{Eq 7 - 3.3.1})$$

$$\left. \frac{\partial^2 \phi}{\partial x^2} \right|_{k,i} = \sum_{j=1}^{M+2} B^x_{k,j} \phi_{j,i} \quad (\text{Eq 7 - 3.3.2})$$

$$\left. \frac{\partial \phi}{\partial \tau} \right|_{k,i} = \sum_{j=1}^{N+2} A^T_{i,j} \phi_{k,j} \quad (\text{Eq 7 - 3.3.3})$$

where ϕ stands for the following variables:

Table 7 - 3.3.1 Variables

ϕ	$\frac{\partial \phi}{\partial x}$	$\frac{\partial^2 \phi}{\partial x^2}$	$\frac{\partial \phi}{\partial \tau}$
X_1	$\frac{\partial X_1}{\partial x}$	$\frac{\partial^2 X_1}{\partial x^2}$	$\frac{\partial X_1}{\partial \tau}$
X_2	$\frac{\partial X_2}{\partial x}$	$\frac{\partial^2 X_2}{\partial x^2}$	$\frac{\partial X_2}{\partial \tau}$
Y_1			$\frac{\partial Y_1}{\partial \tau}$
Y_2			$\frac{\partial Y_2}{\partial \tau}$
\bar{v}	$\frac{\partial \bar{v}}{\partial x}$		
\bar{T}			$\frac{\partial \bar{T}}{\partial \tau}$

7 - 3.4 Step 4

Setting up the residuals $R_{k,i}^0$ and updating the initial guess to force the $(6 \cdot M + 6) \cdot (N + 1)$ unknown residuals to become zero. To simplify the calculation, we compute β_1 and β_2 with the local temperature at last step.

$$\beta_1(k) = \exp\left(\frac{H_1}{RT_a} \left(\frac{1}{\bar{T}(k, 1)} - 1\right)\right) \quad (\text{Eq 7 - 3.4.1})$$

$$\beta_2(k) = \exp\left(\frac{H_2}{RT_a} \left(\frac{1}{\bar{T}(k,1)} - 1\right)\right) \quad (\text{Eq 7 - 3.4.2})$$

$$R_{k,i}^{X_1} = \frac{\alpha}{h} \frac{\partial X_1}{\partial \tau} \Big|_{k,i} - \frac{1}{Pe_1} \frac{\partial^2 X_1}{\partial x^2} \Big|_{k,i} + \bar{v}_{k,i} \frac{\partial X_1}{\partial x} \Big|_{k,i} + X_1(k,i) \frac{\partial v}{\partial x} \Big|_{k,i} + \phi_1(\beta_1(k) X_1 - Y_1) = 0 \quad (\text{Eq 7 - 3.4.3})$$

$$R_{k,i}^{X_2} = \frac{\alpha}{h} \frac{\partial X_2}{\partial \tau} \Big|_{k,i} - \frac{1}{Pe_2} \frac{\partial^2 X_2}{\partial x^2} \Big|_{k,i} + \bar{v}_{k,i} \frac{\partial X_2}{\partial x} \Big|_{k,i} + X_2(k,i) \frac{\partial v}{\partial x} \Big|_{k,i} + \phi_2(\beta_2(k) X_2 - Y_2) = 0 \quad (\text{Eq 7 - 3.4.4})$$

$$R_{k,i}^{Y_1} = \frac{\partial Y_1}{\partial \tau} \Big|_{k,i} - hD_1 \{ \beta_1(k) X_1(k,i) - Y_1(k,i) \} \quad (\text{Eq 7 - 3.4.5})$$

$$R_{k,i}^{Y_2} = \frac{\partial Y_2}{\partial \tau} \Big|_{k,i} - hD_2 \{ \beta_2(k) X_2(k,i) - Y_2(k,i) \} \quad (\text{Eq 7 - 3.4.6})$$

And (Eq 7 - 2.2.7) and (Eq 7 - 2.2.8) are implemented as the following:

$$R_{k,i}^T = \frac{\alpha}{h} \frac{\partial T}{\partial \tau} + \frac{c_0}{\alpha_s} \left(\begin{array}{l} \lambda c_p \bar{P} \frac{\partial \bar{v}}{\partial x} - \frac{H_1}{T_a} \phi_1(\beta_1(k) X_1(k,i) - Y_1(k,i)) \\ - \frac{H_2}{T_a} \phi_2(\beta_2(k) X_2(k,i) - Y_2(k,i)) + \alpha k_{eq} \frac{1}{c_0} (\bar{T} - 1) \end{array} \right) = 0 \quad (\text{Eq 7 - 3.4.7})$$

$$R_{k,i}^{\bar{v}} = - \frac{1}{Pe_1} \frac{\partial^2 X_1}{\partial x^2} \Big|_{k,i} - \frac{1}{Pe_2} \frac{\partial^2 X_2}{\partial x^2} \Big|_{k,i} + (X_1(k,i) + X_2(k,i)) \frac{\partial \bar{v}}{\partial x} + \bar{v}(k,i) \left(\frac{\partial X_1}{\partial x} \Big|_{k,i} + \frac{\partial X_2}{\partial x} \Big|_{k,i} \right) + \phi_1(\beta_1(k) X_1(k,i) - Y_1(k,i)) + \phi_2(\beta_2(k) X_2(k,i) - Y_2(k,i)) + \frac{X_1(k,i) + X_2(k,i)}{\bar{T}(k,i)} \frac{c_0}{\alpha_s} \left(\begin{array}{l} c_p \bar{P} \frac{\partial \bar{v}}{\partial x} - \frac{H_1}{T_a} \phi_1(\beta_1(k) X_1(k,i) - Y_1(k,i)) \\ - \frac{H_2}{T_a} \phi_2(\beta_2(k) X_2(k,i) - Y_2(k,i)) + \alpha k_{eq} \frac{1}{c_0} (\bar{T} - 1) \end{array} \right) = 0 \quad (\text{Eq 7 - 3.4.8})$$

In this step we will use standard subroutine to solve the resulting systems of nonlinear equations, in our program, we will also supply full Jacobian analysis in order to reduce computation expenses. The simulation will repeat the four steps until time reach to the time length of adsorption period.

7 - 4 The Simulation Result:

With the boundary and initial conditions and the parameters abovementioned, we set:

1. space mesh=13 (including 2 boundary points)
2. time layers=3 (including 2 boundary layers)
3. adsorption duration tmax = 120 seconds,
4. velocity = 0.02 m/second.

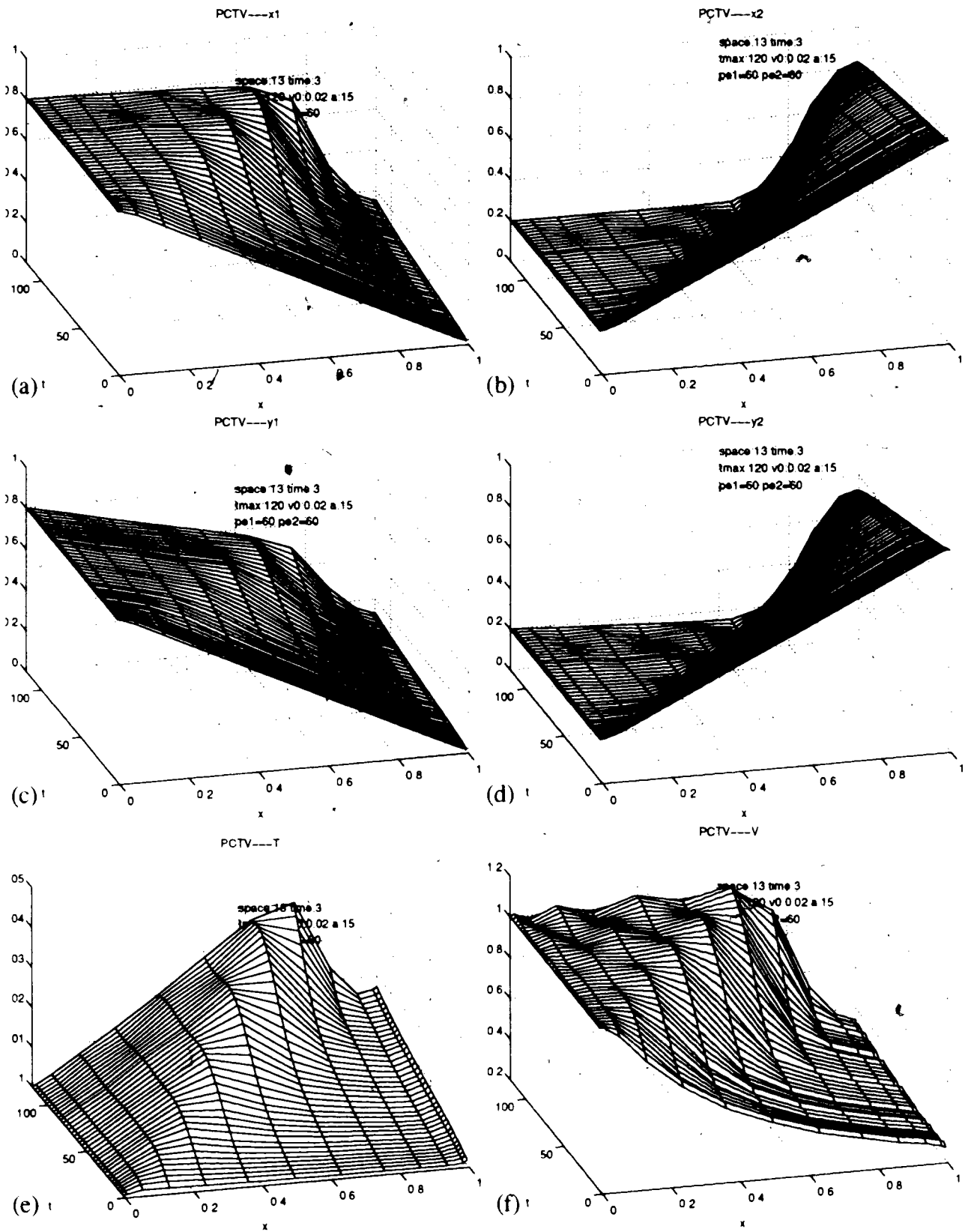


Figure 7 - 4.1 3D plot of variation of variables, (a) concentration of nitrogen, (b) concentration of oxygen, (c) concentration of nitrogen in the solid phase (d) concentration of oxygen in the solid phase (e) temperature (f) velocity of flow

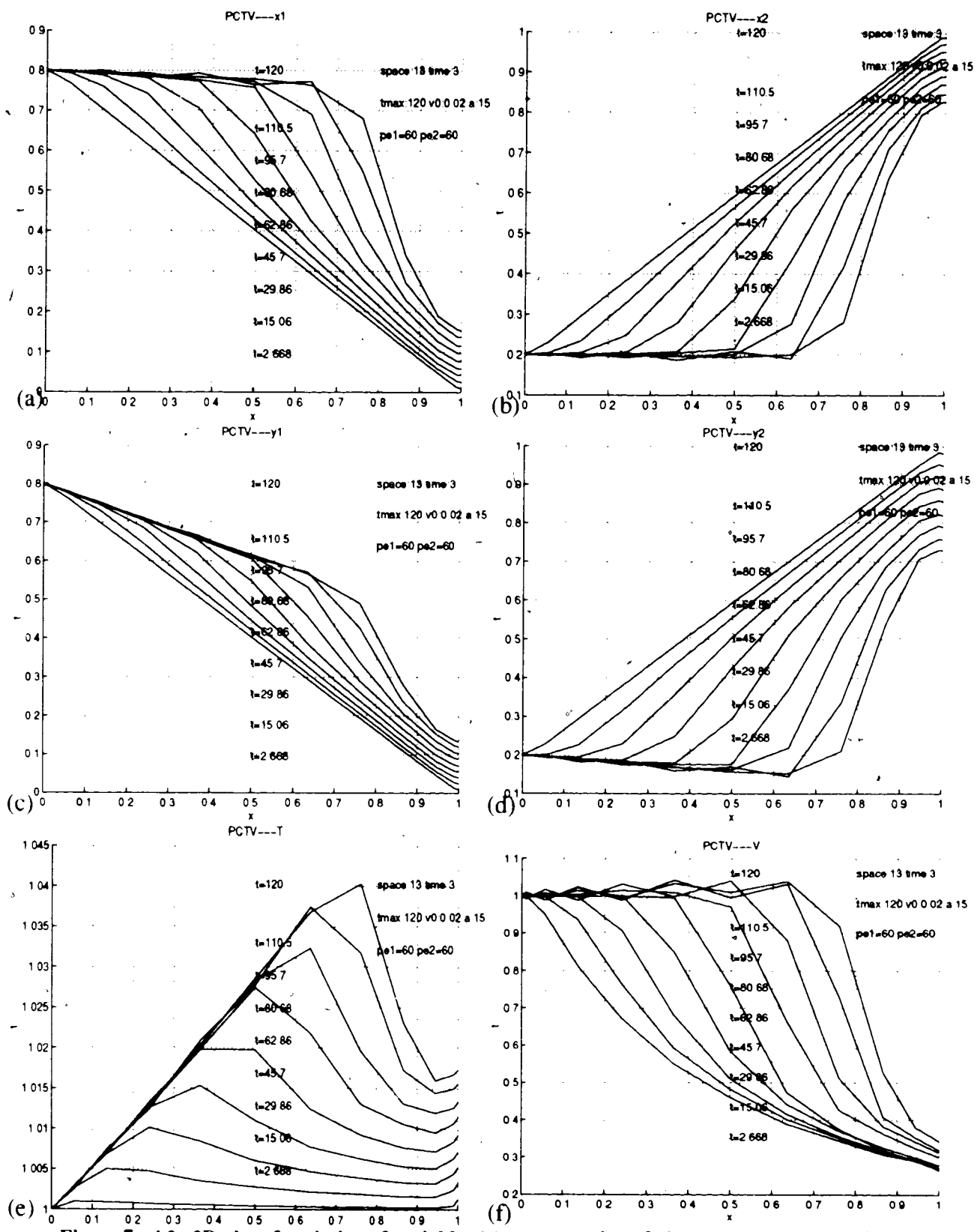


Figure 7-4.2 2D plot of variation of variables (a) concentration of nitrogen, (b) concentration of oxygen, (c) concentration of nitrogen in the solid phase (d) concentration of oxygen in the solid phase (e) temperature (f) velocity of flow

From the above figures, we can get more insight about the evolution of the absorption step of a conventional PSA cycle.

1. Figure 7 - 4.1 (a) and Figure 7 - 4.2 (a) clearly shows that the curve representing the variation of concentration of adsorbable (nitrogen) species is self-sharpening as adsorption goes on. This confirms the theoretical prediction in chapter 1, that when the adsorption is favorable and significant, the concentration curve is self-sharpening. It also shows that collocation method can handle this problem.
2. Figure 7 - 4.1 (b) and Figure 7 - 4.2 (b) shows that as the adsorption proceeds, the concentration breakthrough curve of product (oxygen) in the bed is getting steeper.
3. Figure 7 - 4.1 (c) and Figure 7 - 4.2 (c) shows development of the concentration of nitrogen in the solid phase.
4. Figure 7 - 4.1 (d) and Figure 7 - 4.2 (d) shows development of the concentration of oxygen in the solid phase.
5. Figure 7 - 4.1 (e) and Figure 7 - 4.2 (e) shows that as flow moves from inlet to the outlet of adsorption column, the adsorption also moves along the bed, the temperature rises at the places where the adsorption takes place intensively.
6. Figure 7 - 4.1 (f) and Figure 7 - 4.2 (f) shows that the velocity of fluid flow is really not constant, it slows down where the adsorption take place; it recovers to the inlet velocity where the bed gets saturated.

All the above figures have shown that adsorption in this conventional PSA simulation is favorable and significant, the temperature and velocity of fluid vary as the adsorption takes place, a self-sharpening breakthrough curve develops as the fluid flows through the adsorption column.

7 - 5 The Computation Time

Table 7 - 5.1 Collocation points with Repressurization

number of mesh	computation time (minute : second)
m=6	0:05
m=7	0:11
m=8	0:27
m=9	1:02
m=10	1:58
m=11	3:35
m=12	6:23
m=13	10:22
m=14	15:46
m=15	27:03

7 - 6 Conclusion for Chapter 7

With double collocation method, we can solve the conventional PSA simulation model, and it has shown that the method can handle the self-sharpening simulation, and the results verify our prediction in Chapter 1.

7 - 7 List of Parameters

Table 7 - 7.1: Parameters Used in Simulation

Data	parameter	value	unit, explanation
operation data	cycle	1	number of cycle of psa operation
	tmax	120	second, flowing period
bed properties	length	30.0	cm, bed length
	diameter	1.66	cm, bed diameter
	c_s	1000	J/kg.K specific heat of zeolite
	ϵ	0.3	Porosity of bed
	ρ_s	640	kg/m ³
	k_{eq}	100.0	W/(m ³ K), equivalent radial conductance
Gas properties	R	8.314	J/gm-mol/K universal gas constant
	patm	101325.0	1atm=101325 N/m ²
	p_0	2.0	atm.
	T_a	300	K, Temperature at initial condition
	v_0	0.02	m/s, velocity of gas flow at inlet
Nitrogen:	x10	0.8	dimensionless gas component (percentage)
	c_{v1}	20.8	J/gm-mol/K, specific heat of bulk gas (constant volume)
	c_{p1}	18.0	J/gm-mol/K, specific heat of bulk gas (constant pressure)
	D_1	6.2	Linear driving force coefficient or gas diffusivity (per second)
	H_1	21000	J/gm-mol. Enthalpy of adsorption
	B_{10}	15.0	Langmuir constant at standard temperature m ³ /gm-mol
	rho1	28.0	molecular weight
	D_{11}	0.0001	coefficient of axial diffusion
Oxygen:	x20	0.2	dimensionless gas component (percentage)
	c_{v2}	21.1	J/gm-mol/K specific heat of bulk gas (constant volume)
	c_{p2}	29.5	J/gm-mol/K specific heat of bulk gas (constant pressure)
	D_2	35.0	Linear driving force coefficient or gas diffusivity (per second)
	H_2	19000	J/gm-mol. Enthalpy of adsorption
	B_{20}	2.0	Langmuir constant at standard temperature, m ³ /gm-mol
	rho2	32.0	Molecular weight
	D_{12}	0.0001	coefficient of axial diffusion

Table 7 - 7.2: Derived Parameters

	parameters	definition
gas	c_0	$p_0 \cdot p_{atm} / (R \cdot T_a)$
	c_v	$(c_{v1} + c_{v2}) / 2.0$
	c_p	$(c_{p1} + c_{p2}) / 2.0$

Chapter 8 General Conclusion and Future Work

8 - 1 The Present Study Has the Following Findings:

1. The object of this thesis is to build realistic models for the design of PSA cycles and PSA units. For the realistic modelling of a conventional PSA cycle, the adsorption column can not be assumed to be isothermal and to have constant gas velocity. When the benchtop apparatus is scaled up to industrial size, the thermal condition of the adsorption column changes from near-isothermal to near-adiabatic; therefore, thermal effect has to be considered in our model, as well as the variation of fluid flow due to bulk adsorption. In Chapter 1, we have shown that, in this circumstance, the breakthrough curve of the concentration profile is self-sharpening as it propagates along the adsorption column. Therefore, it is very important that we can represent this self-sharpening breakthrough curve accurately, otherwise, we will get low prediction of either total productivity or purity from the computer modelling. This also means the numerical methods of computer modelling must be free from false diffusion (numerical diffusion), which is a non-physical effect. As another object of our research, we have discussed how to reduce or eliminate this effect of false diffusion.

2. As noted in Chapter 2, the truncation error of applying the finite difference method to solve partial differential equation, which is the reason for numerical diffusion. Earlier researchers tried to relate the truncation error to block and step size, and through adjusting the block and step size to minimize the truncation error. There has been some success in minimizing false diffusion this way; however this solution to false diffusion is limited to the specific problem, and sometimes results in prohibitively small block and step size. Later researchers started to investigate new methods which have higher order accuracy, in order to minimize the truncation error. The typical new methods would still be upwinding, the order of accuracy can be third, fourth, fifth, or even seventh, ninth order. In the scope of this thesis, I study two methods: the numerical method of lines, and the orthogonal collocation method, in either of which higher order formulas can be embedded. These two methods have been applied to solve the convection equation, the convective-diffusion equation, and a practical PSA model. It has been shown that numerical method of lines with fourth-order formula and orthogonal collocation can both give solutions to the numerical diffusion; also orthogonal collocation method is superior to the numerical method of lines in computation time.

3. For the mathematical model of the conventional PSA, which is the combination of P.D.Es and O.D.Es, solution is not possible by P.D.E-O.D.E. procedure. Therefore, we extend the orthogonal collocation method to the double collocation method to solve the mathematical model. In Chapter 6, we employ this method to solve a simplified version of PSA model, in

which the thermal condition is assumed isothermal, however, we take consideration of variation of velocity of fluid flow due to bulk adsorption. Based on the work of Chapter 6, we solve our realistic model of conventional PSA model, in which we take consideration of thermal effect as well as the variation of velocity of fluid flow due to bulk adsorption. The results agree with the theoretical analysis made in chapter 1. It has been shown that the breakthrough curve of the concentration profile is self-sharpening as it propagates along the adsorption column. The computer model also present the temperature variation inside adsorption column, as well as variation of velocity of fluid flow. The temperature rises at the places where the adsorption take place, while velocity of fluid flow slow down at the places where the adsorption take place. Therefore, the computer model has provided us with insight into the adsorption column. The extension of the double collocation method to a semi-adiabatic bed is and original contribution of this thesis.

8 - 2 Recommendation for Future Work

Several future tasks are recommended:

1. Extending the current two-component adsorption column into three-component adsorption column.

The present model can be adapted to any other two gases by changing the input parameters. However, some separation problems includes three or more gases, in the future, it is also desired to include third component such as carbon dioxide.

2. Realization of cyclic operation.

Current work of simulation of the conventional PSA cycle is only the simulation of adsorption step. The solution we've developed can easily be extended to cyclic operation by changing the boundary conditions. Therefore, we have further description: the bed runs through a cycle; there are four or five distinct phases, each with particular flow and pressure boundary conditions.

There are two ways of simulating bed behavior; we can try to simulate the operation of the bed from initial conditions, through a "warm-up" phase, to cyclically steady state; or we can make it a boundary condition that the state of the bed at the end of cycle must be identical to its state at the beginning. (This could be guaranteed by choosing the solution to be a sum of sinusoids with period a multiple of the cycle length.)

3. Simulation of more advanced cyclic operation such as TCPSA.

Upon the completion of these simulations, we will obtain realistic models to help on the design of PSA unit as well as PSA cyclic.

Appendix: The Orthogonal Collocation Method and Orthogonal Polynomials

The Orthogonal collocation method has been widely employed in problems of chemical engineering. The fundamental step of this method is the solution of the collocation matrices which are derived directly from associated orthogonal polynomials. Approaches based on two categories of orthogonal polynomials (Jacobi and shifted Legendre polynomials) are presented in detail in this section by symbolic computation. A general approach based on the Gram-Schmidt orthogonalization procedure can be employed to obtain variously defined orthogonal polynomials.

A - 1 A Brief Introduction to The Orthogonal Collocation Method

In the orthogonal collocation method, an original PDE can be reduced to a system of ODEs by writing it in collocation form, and then the resulting system of ODEs can be solved by a standard subroutine.

A - 1.1 Trial Function & Orthogonal Polynomials

Consider an unknown function $\phi(x, t)$, which satisfies partial differential equations and boundary conditions. An approximating function called trial function can be chosen, such that the boundary conditions are satisfied. Depending on nature of the problem, non-symmetric or symmetric trial function should be used.

An example of a symmetric trial function is:

$$\phi(x, t) = \phi(1, t) + (1-x^2) \sum_1^N a_i(t) P_{i-1}(x^2) \quad (\text{Eq A - 1.1.1})$$

where $a_i(t)$ are functions of time (or constants) and P_i are the orthogonal polynomials defined by:

$$\int_0^1 w(x^2) P_j(x^2) P_i(x^2) x^{a-1} dx = C_i \delta_{ij} \quad (\text{Eq A - 1.1.2})$$

where $w(x^2) = 1-x^2, j = 1, 2, \dots$

Here the volume element dV has been replaced by the proportional quantity $x^{a-1} dx$; thus for slabs $a=1$, for cylinders $a=2$ and for spheres $a=3$. The polynomials defined by (Eq A - 1.1.2) are Jacobi polynomials.

An example of a nonsymmetric trial function is:

$$\phi(x, \tau) = (1-x)\phi(0, \tau) + x\phi(1, \tau) + x(1-x) \sum_{i=1}^M a_i(\tau) P_{i-1}(x) \quad (\text{Eq A - 1.1.3})$$

where $a_i(\tau)$ are functions of time or constants and P_i are the non symmetric polynomials defined by the condition:

$$\int_0^1 w(x) P_n(x) P_m(x) dx = 0, n = 0, 1, \dots, m-1 \quad (\text{Eq A - 1.1.4})$$

where $w(x)$ is weight function, such as $w(x) = 1$. The polynomials defined by (Eq A - 1.1.4) are shifted Legendre polynomials.

The collocation points x_1, \dots, x_N are given by the zeroes of $P_n(x^2)$ or $p_n(x)$

A - 1.2 Interior formulas based on ordinates

The trial function can be translated to an equivalent set of equations in terms of the unknown ordinates $\phi(x_1), \phi(x_2), \dots, \phi(x_N)$

For example, rewrite (Eq A - 1.1.1)

$$\phi(x) = \sum_{j=1}^{N+1} d_j x^{2j-2} \quad (\text{Eq A - 1.2.1})$$

Evaluate at collocation points,

$$\phi(x_i) = \sum_{j=1}^{N+1} d_j x_i^{2j-2} \quad (\text{Eq A - 1.2.2})$$

Take the first derivative and the Laplacian of this expression and evaluate them at the collocation points:

$$\left. \frac{d\phi}{dx} \right|_{x_i} = \sum_{j=1}^{N+1} d_j \left. \frac{dx^{2j-2}}{dx} \right|_{x_i} \quad (\text{Eq A - 1.2.3})$$

$$\nabla^2 \phi|_{x_i} = \sum_{j=1}^{N+1} d_j \nabla^2 (x^{2j-2})|_{x_i} \quad (\text{Eq A - 1.2.4})$$

These can be written in matrix notation

$$\phi = Qd, \quad Q_{ij} = x_i^{2j-2} \quad (\text{Eq A - 1.2.5})$$

$$\left. \frac{d\phi}{dx} \right|_{x_i} = Cd, \quad C_{ij} = \left. \frac{dx^{2j-2}}{dx} \right|_{x_i} \quad (\text{Eq A - 1.2.6})$$

$$\nabla^2 \phi = Dd, \quad D_{ij} = \nabla^2 (x^{2j-2})|_{x_i} \quad (\text{Eq A - 1.2.7})$$

Since $d = Q^{-1}\phi$

$$\left. \frac{d\phi}{dx} \right|_{x_i} = Cd = CQ^{-1}\phi = A\phi \quad (\text{Eq A - 1.2.8})$$

$$\nabla^2 \phi = Dd = DQ^{-1}\phi = B\phi \quad (\text{Eq A - 1.2.9})$$

Integrals of the solution can be calculated with high accuracy via the summation formula

$$\int_0^1 \phi(x) x^{a-1} dx = \int_0^1 \sum_{j=1}^{N+1} d_j x^{2j-2} x^{a-1} dx = \sum_{j=1}^{N+1} d_j \int_0^1 x^{2j-2} x^{a-1} dx = \sum_{i=1}^{N+1} W_i \phi(x_i) \quad (\text{Eq A - 1.2.10})$$

which requires the value of solution only at the collocation points x_1, \dots, x_N, x_{N+1}

The collocation points x_1, \dots, x_N that appear here are the roots of $P_n(x^2) = 0$ and x_{N+1} is unity.

Matrices A, B are called collocation matrices.

Where,

$$[A_{ij}^{(N)}] = \begin{bmatrix} \left(\frac{dx^0}{dx}\right)\Big|_{x_1} & \dots & \left(\frac{dx^{2N}}{dx}\right)\Big|_{x_1} \\ \vdots & \dots & \vdots \\ \left(\frac{dx^0}{dx}\right)\Big|_{x_{N+1}} & \dots & \left(\frac{dx^{2N}}{dx}\right)\Big|_{x_{N+1}} \end{bmatrix} [Q]^{-1} \quad (\text{Eq A - 1.2.11})$$

$$[B_{ij}^{(N)}] = \begin{bmatrix} \nabla^2(x^0)\Big|_{x_1} & \dots & \nabla^2(x^{2N})\Big|_{x_1} \\ \vdots & \dots & \vdots \\ \nabla^2(x^0)\Big|_{x_{N+1}} & \dots & \nabla^2(x^{2N})\Big|_{x_{N+1}} \end{bmatrix} [Q]^{-1} \quad (\text{Eq A - 1.2.12})$$

$$[W_i^{(N)}] = \left[\int_0^1 x^{0+a-1} dx \dots \int_0^1 x^{2N+a-1} dx \right] [Q]^{-1} \quad (\text{Eq A - 1.2.13})$$

$$[Q] = \begin{bmatrix} 1 & x_1^2 & \dots & x_1^{2N} \\ \vdots & \vdots & \dots & \vdots \\ 1 & x_{N+1}^2 & \dots & x_{N+1}^{2N} \end{bmatrix} \quad (\text{Eq A - 1.2.14})$$

Similarly, the corresponding set of collocation matrices can be written for the nonsymmetric trial function (Eq A - 1.1.3) based on the shifted Legendre polynomials.

Rewrite (Eq A - 1.1.3),

$$\phi(x) = \sum_{j=1}^{M+2} d_j x^{j-1} \quad (\text{Eq A - 1.2.15})$$

Evaluate at collocation points,

$$\phi(x_i) = \sum_{j=1}^{M+2} d_j x_i^{j-1} \quad (\text{Eq A - 1.2.16})$$

Take the first derivative and the Laplacian of this expression and evaluate them at the collocation points:

$$\left. \frac{d\phi}{dx} \right|_{x_i} = \sum_{j=1}^{M+2} d_j \left. \frac{dx^{j-1}}{dx} \right|_{x_i} \quad (\text{Eq A - 1.2.17})$$

$$\nabla^2 \phi_{x_i} = \sum_{j=1}^{M+2} d_j \nabla^2 (x^{j-1}) \Big|_{x_i} \quad (\text{Eq A - 1.2.18})$$

These can be written in matrix notation

$$y = Qd, \quad Q_{ij} = \left. x^{j-1} \right|_{x_i} \quad (\text{Eq A - 1.2.19})$$

$$\frac{dy}{dx} = Cd, \quad C_{ij} = \left. \frac{dx^{j-1}}{dx} \right|_{x_i} = (j-1)x^{j-2} \Big|_{x_i} \quad (\text{Eq A - 1.2.20})$$

$$\nabla^2 y = Dd, \quad D_{ij} = \left. \nabla^2 (x^{j-1}) \right|_{x_i} = (j-1)(j-2)x^{j-3} \quad (\text{Eq A - 1.2.21})$$

Since $d = Q^{-1}\phi$

$$\frac{dy}{dx} = Cd = CQ^{-1}\phi = A^x\phi \quad (\text{Eq A - 1.2.22})$$

$$\nabla^2 y = Dd = DQ^{-1}\phi = B^x\phi \quad (\text{Eq A - 1.2.23})$$

Integral of the solution can be obtained by

$$\int_0^1 \phi(x) dx = \int_0^1 \sum_{j=1}^{M+2} d_j x^{j-1} dx = \sum_{j=1}^{M+2} d_j \int_0^1 x^{j-1} dx = \sum_{j=1}^{M+2} W_j \phi(x_j) \quad (\text{Eq A - 1.2.24})$$

Matrices A^x, B^x are called collocation matrices.

The collocation points x_1, \dots, x_{M+2} that appear here are the roots of $P_n(x) = 0$ $x_1 = 0$ and x_{M+2} is unity.

Where,

$$[A_{ij}^{x(M+2)}] = \begin{bmatrix} \left. \frac{dx^0}{dx} \right|_{x_1} & \dots & \left. \frac{dx^{M+1}}{dx} \right|_{x_1} \\ \vdots & \dots & \vdots \\ \left. \frac{dx^0}{dx} \right|_{x_{M+2}} & \dots & \left. \frac{dx^{M+1}}{dx} \right|_{x_{M+2}} \end{bmatrix} [Q]^{-1} \quad (\text{Eq A - 1.2.25})$$

$$[B_{ij}^{x(M+2)}] = \begin{bmatrix} \nabla^2 (x^0) \Big|_{x_1} & \dots & \nabla^2 (x^{M+1}) \Big|_{x_1} \\ \vdots & \dots & \vdots \\ \nabla^2 (x^0) \Big|_{x_{M+2}} & \dots & \nabla^2 (x^{M+1}) \Big|_{x_{M+2}} \end{bmatrix} [Q]^{-1} \quad (\text{Eq A - 1.2.26})$$

$$[W_i^{(M+2)}] = \begin{bmatrix} \int_0^1 x^0 dx & \dots & \int_0^1 x^{M+1} dx \end{bmatrix} [Q]^{-1} \quad (\text{Eq A - 1.2.27})$$

$$[Q] = \begin{bmatrix} 1 & x_1^1 & \dots & x_1^{M+1} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{M+2}^1 & \dots & x_{M+2}^{M+1} \end{bmatrix} \quad (\text{Eq A - 1.2.28})$$

A - 2 Orthogonal polynomials

Orthogonal polynomials such as the above mentioned Jacobi polynomials and shifted Legendre polynomials and corresponding collocation points can be found in the literature, such as Finlayson(1972), and Villadsen & Stewart (1967) for order from 3 to 7, already resulting in very complicated calculation. However, in many situation, we need higher order polynomials. In this section, we provide method to obtain higher order of orthogonal polynomials by symbolic computing. The symbolic computation can be implemented in Maple V.

A - 2.1 Jacobi Polynomials

The polynomials defined by (Eq A - 1.1.2) are Jacobi polynomials (Courant & Hilbert 1953, Morse & Feschbach 1953). Rewrite it:

$$\int_0^1 w(x^2) P_j(x^2) x^{a-1} dx = C_i \delta_{ij} \quad (\text{Eq A - 2.1.1})$$

where $w(x^2) = 1-x^2$, $j = 1, 2, \dots$, and $a = 1, 2, 3$.

The Jacobi polynomials are given explicitly by:

$$P_i(x^2) = F\left(-i, i + \frac{a}{2} + 1, \frac{a}{2}, x^2\right) \quad (\text{Eq A - 2.1.2})$$

The constant C_i is correspondingly given by:

$$C_i = \frac{\left[\Gamma\left(\frac{a}{2}\right)\right]^2 \Gamma(i+1) \Gamma(i+2)}{(4i+a+2) \Gamma\left(i + \frac{a}{2}\right) \Gamma\left(i + \frac{a}{2} + 1\right)} \quad (\text{Eq A - 2.1.3})$$

Here, F is the hypergeometric function, defined by

$$F\left(-i, i + \frac{a}{2} + 1, \frac{a}{2}, x^2\right) = \sum_{s=0}^i (-1)^s \left[\frac{i! \Gamma\left(\frac{a}{2}\right) \Gamma\left(i + \frac{a}{2} + 1 + s\right)}{(i-s)! s! \Gamma\left(i + \frac{a}{2} + 1\right) \Gamma\left(\frac{a}{2} + s\right)} \right] x^{2s} \quad (\text{Eq A - 2.1.4})$$

The above equation can be used for symbolic computing of Jacobi polynomials in Maple V. Γ function is defined as GAMMA in Maple V. The program has been listed in A - 5.1, where a can be 1, 2 and 3, the order of resulting Jacobi polynomials and their roots can be as high as 20. However, here for the reason of space, we can only list jacobi polynomial and their roots for order up to 10 in the Table A - 4.1 and Table A - 4.2 for $a=3$.

A - 2.2 Shifted Legendre Polynomials

Another important set of orthogonal polynomials used to construct nonsymmetric trial function are shifted Legendre polynomials, defined as:

$$\int_0^1 w(x) P_n(x) P_m(x) dx = 0, n = 0, 1, \dots, m-1 \quad (\text{Eq A - 2.2.1})$$

where $w(x)$ is weight function, such as $w(x) = 1$. Indeed, these orthogonal polynomials can be computed in a simple manner by Gram-Schmidt Orthogonalization procedure with an understanding of the concept of inner product (Dorny 1975).

Define inner product:

$$\langle f, g \rangle = \int_0^1 w(x) f(x) g(x) dx \quad (\text{Eq A - 2.2.2})$$

therefore, we can see that the inner product of

$$\langle P_n, P_m \rangle = \int_0^1 w(x) P_n(x) P_m(x) dx = 0, \quad (\text{Eq A - 2.2.3})$$

which is the definition of orthogonal polynomials. By Gram-Schmidt orthogonalization procedure, a natural basis polynomials $F_0(x) = 1, F_1(x) = x, F_2(x) = x^2, \dots$ can be orthogonalize into:

$$\begin{aligned} P_0(x) &= F_0(x) = 1 \\ P_1(x) &= F_1(x) - \frac{\langle F_1, P_0 \rangle}{\langle P_0, P_0 \rangle} P_0(x) \\ P_2(x) &= F_2(x) - \frac{\langle F_2, P_0 \rangle}{\langle P_0, P_0 \rangle} P_0(x) - \frac{\langle F_2, P_1 \rangle}{\langle P_1, P_1 \rangle} P_1(x) \\ P_3(x) &= F_3(x) - \frac{\langle F_3, P_0 \rangle}{\langle P_0, P_0 \rangle} P_0(x) - \frac{\langle F_3, P_1 \rangle}{\langle P_1, P_1 \rangle} P_1(x) - \frac{\langle F_3, P_2 \rangle}{\langle P_2, P_2 \rangle} P_2(x) \end{aligned} \quad (\text{Eq A - 2.2.4})$$

Accordingly, we can obtain orthogonal polynomials: $P_1(x), P_2(x), P_3(x), \dots$, We listed the shifted Legendre polynomials and their roots in Table A - 4.3 and Table A - 4.4

We can also take advantages of symbolic computation in Maple V to perform the orthogonalization procedure described in (Eq A - 2.1.4). The program is listed in section A - 5.2. The order of resulting polynomials can be as high as 20.

The advantage of above procedure is that different polynomials can be easily obtained by just changing definition of inner product corresponding to definition of each individual polynomial. For example, if we define:

$$\langle f, g \rangle = \langle P_n, P_m \rangle = \int_{-1}^1 w(x) P_n(x) P_m(x) dx = 0 \quad (\text{Eq A - 2.2.5})$$

and then utilize the orthogonalization procedure described above, the resulting orthogonal

polynomials are Legendre polynomials.

Upon obtaining the orthogonal polynomials, the procedure for obtaining collocation matrices is very simple.

1. Read N , and $x(j)$ roots.
2. Calculate Q , C , D matrices using for (Eq A - 1.2.5)-(Eq A - 1.2.7) Jacobi polynomials or (Eq A - 1.2.19) -(Eq A - 1.2.21) for shift Legendre polynomials.
3. Invert Q .
4. Calculate A , B , A^x , B^x using (Eq A - 1.2.11) - (Eq A - 1.2.14) or (Eq A - 1.2.25) -(Eq A - 1.2.28) respectively.

After we have obtained the collocation matrices, the P.D.E's can be written in collocation form which will result in a system of O.D.Es or a system of algebraic equations which can be solved with standard subroutines.

A - 3 Solution in the form of polynomials

Sometimes, it is desired to express the computation results in the form of polynomials.

A - 3.1 Solution in the form of Jacobi polynomials

Rewrite (Eq A - 1.1.1)

$$\phi(x, t) = \phi(1, t) + (1-x^2) \sum_1^N a_i(t) P_{i-1}(x^2) \quad (\text{Eq A - 3.1.1})$$

where

$$a_k(t) = \frac{1}{C_k} \int_0^1 [\phi(x, t) - \phi(1, t)] p_k(x^2) x^{a-1} dx, \quad k = 1, \dots, N \quad (\text{Eq A - 3.1.2})$$

The integration can be exactly done by

$$a_k(t) = \frac{1}{C_k} \sum_{i=1}^N W_i [\phi(x_i, t) - \phi(1, t)] p_k(x_i^2), \quad k = 0, \dots, N \quad (\text{Eq A - 3.1.3})$$

A - 3.2 Solution in the form of shifted Legendre Polynomials

Rewrite (Eq A - 1.1.3),

$$\phi(x, \tau) = (1-x)\phi(0, \tau) + x\phi(1, \tau) + x(1-x) \sum_{i=1}^M a_i(\tau) P_{i-1}(x) \quad (\text{Eq A - 3.2.1})$$

By definition of shifted Legendre orthogonal polynomials,

$$\int_0^1 p_i(x) p_j(x) dx = C_i \delta_{ij}, \quad \text{where } \delta_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases} \quad (\text{Eq A - 3.2.2})$$

therefore,

$$C_i = \int_0^1 p_i(x) p_i(x) dx \quad (\text{Eq A - 3.2.3})$$

This calculation can be done by symbolic computation in Maple V with the definition of inner product. See section A - 5.2.

Since,

$$\int_0^1 \left(\frac{1}{x(1-x)} \phi(x) - \frac{1}{x} \phi(0) - \frac{1}{1-x} \phi(1) \right) p_k(x) dx = \sum_{i=1}^M a_i \int_0^1 p_{i-1}(x) p_k(x) dx = a_k C_k \quad (\text{Eq A - 3.2.4})$$

$$k = 1, \dots, M, i = 1, \dots, M+2$$

therefore,

$$a_k = \frac{1}{C_k} \int_0^1 \left(\frac{1}{x(1-x)} \phi(x) - \frac{1}{x} \phi(0) - \frac{1}{1-x} \phi(1) \right) p_k(x) dx \quad (\text{Eq A - 3.2.5})$$

$$= \frac{1}{C_k} \sum_{i=1}^{M+2} W_i \left(\frac{1}{x_i(1-x_i)} \phi(x_i) - \frac{1}{x_i} \phi(0) - \frac{1}{1-x_i} \phi(1) \right) p_k(x_i)$$

A - 4 Table of Polynomials and Their Roots

Table A - 4.1 Jacobi polynomials a=3

Order	Polynomials
n=1	$P_1 = 1 - \frac{7}{3}x^2$
n=2	$P_2 = 1 - 6x^2 + \frac{33}{5}x^4$
n=3	$P_3 = 1 - 11x^2 + \frac{143}{5}x^4 - \frac{143}{7}x^6$
n=4	$P_4 = 1 - \frac{52}{3}x^2 + 78x^4 - \frac{884}{7}x^6 + \frac{4199}{63}x^8$
n=5	$P_5 = 1 - 25x^2 + 170x^4 - \frac{3230}{7}x^6 + \frac{1615}{3}x^8 - \frac{7429}{33}x^{10}$
n=6	$P_6 = 1 - 34x^2 + 323x^4 - 1292x^6 + \frac{7429}{3}x^8 - \frac{74290}{33}x^{10} + \frac{111435}{143}x^{12}$
n=7	$P_7 = 1 - \frac{133}{3}x^2 + \frac{2793}{5}x^4 - 3059x^6 + \frac{76475}{9}x^8 - \frac{137655}{11}x^{10} + \frac{130665}{143}x^{12} - \frac{392863}{143}x^{14}$
n=8	$P_8 = 1 - 56x^2 + \frac{4508}{5}x^4 + 6440x^6 + 24150x^8 - \frac{560280}{11}x^{10} + \frac{8684340}{143}x^{12} - \frac{496248}{13}x^{14} + \frac{2171086}{221}x^{16}$
n=9	$P_9 = 1 - 69x^2 + 1380x^4 - 12420x^6 + 60030x^8 - \frac{1860930}{11}x^{10} + \frac{3721860}{12}x^{12}$ $- \frac{3721860}{12}x^{14} + \frac{34427205}{221}x^{16} - \frac{11475735}{323}x^{18}$
n=10	$P_{10} = 1 - \frac{250}{3}x^2 + 2025x^4 - \frac{156600}{7}x^6 + 134850x^8 - 485460x^{10} + \frac{14159250}{12}x^{12}$ $- \frac{19957800}{13}x^{14} + \frac{22452525}{17}x^{16} - \frac{204567450}{323}x^{18} + \frac{293213345}{2261}x^{20}$

Table A - 4.2 The roots of Jacobi polynomials a=3

Order	Roots of polynomials
n=1	0.6546536707
n=2	0.4688487935 0.8302238963
n=3	0.3631174638 0.6771862795 0.8997579954
n=4	0.2957581356 0.5652353270 0.7844834737 0.9340014304
n=5	0.2492869301 0.4829098211 0.6861884691 0.8463475647 0.9533098466
n=6	0.2153539554 0.4206380547 0.6062532055 0.7635196900 0.8850820442 0.9652459265
n=7	0.1895119735 0.3721744336 0.5413853993 0.6910289806 0.8156962512 0.9108799959 0.9731321766
n=8	0.1691860234 0.3335048478 0.4882292857 0.6289081373 0.7514942026 0.8524605778 0.9289015282 0.9786117662
n=9	0.1527855158 0.3019898565 0.4441157833 0.5758319603 0.6940510261 0.7960019261 0.8792947553 0.9419762970 0.9825722966
n=10	0.1392762040 0.2758415489 0.4070379379 0.5303117711 0.6432636445 0.7436950412 0.8296510966 0.8994585580 0.9517579557 0.9855271559

Table A - 4.3 Shifted legendre polynomials

Order	Polynomials
n=1	$P_1 = x - \frac{1}{2}$
n=2	$P_2 = x^2 - x + \frac{1}{6}$
n=3	$P_3 = x^3 - \frac{1}{20} + \frac{3}{5}x - \frac{3}{2}x^2$
n=4	$P_4 = x^4 + \frac{1}{70} - \frac{2}{7}x + \frac{9}{7}x^2 - 2x^3$
n=5	$P_5 = x^5 - \frac{1}{252} + \frac{5}{42}x - \frac{5}{6}x^2 + \frac{20}{9}x^3 - \frac{5}{2}x^4$
n=6	$P_6 = x^6 + \frac{1}{924} - \frac{1}{22}x + \frac{5}{11}x^2 - \frac{20}{11}x^3 + \frac{75}{22}x^4 - 3x^5$
n=7	$P_7 = x^7 - \frac{1}{3432} + \frac{7}{429}x - \frac{63}{286}x^2 + \frac{175}{143}x^3 - \frac{175}{52}x^4 + \frac{63}{13}x^5 - \frac{7}{2}x^6$
n=8	$P_8 = x^8 + \frac{1}{12870} - \frac{4}{715}x + \frac{14}{143}x^2 - \frac{28}{39}x^3 + \frac{35}{13}x^4 - \frac{28}{5}x^5 + \frac{98}{15}x^6$
n=9	$P_9 = x^9 - \frac{1}{48620} + \frac{9}{4862}x - \frac{9}{221}x^2 + \frac{84}{221}x^3 - \frac{63}{34}x^4 + \frac{441}{85}x^5 - \frac{147}{17}x^6 + \frac{144}{17}x^7 - \frac{9}{2}x^8$
n=10	$P_{10} = x^{10} + \frac{1}{184756} - \frac{5}{8398}x + \frac{135}{8398}x^2 - \frac{60}{323}x^3 + \frac{735}{646}x^4 - \frac{1323}{323}x^5 + \frac{2940}{323}x^6 - \frac{240}{19}x^7 + \frac{405}{38}x^8 - 5x^9$

Table A - 4.4 The roots of shifted legendre polynomials

Order	Roots
N=1	0.5000000000
N=2	0.2113248654 0.7886751346
N=3	0.1127016654 0.5000000000 0.8872983346
N=4	0.6943184420 0.3300094782 0.6699905218 0.9305681558
N=5	0.04691007703 0.2307653449 0.5000000000 0.7692346551 0.9530899230
N=6	0.03376524290 0.1693953068 0.3806904070 0.6193095930 0.8306046932 0.9662347571
N=7	0.02544604383 0.1292344072 0.2970774243 0.5000000000 0.7029225757 0.8707655928 0.9745539562
N=8	0.01985507175 0.1016667613 0.2372337950 0.4082826788 0.5917173212 0.7627662050 0.8983332387 0.9801449282
N=9	0.01591988025 0.08198444634 0.1933142836 0.3378732883 0.5000000000 0.6621267117 0.8066857164 0.9180155537 0.9840801198
N=10	0.01304673574 0.06746831666 0.1602952159 0.2833023029 0.4255628305 0.5744371695 0.7166976971 0.8397047841 0.9325316833 0.9869532643

A - 5 Program List

A - 5.1 The program for symbolic computation of the Jacobi polynomials in Maple V:

```

> #solve the jacobi polynomials with the formulas in villadsen paper and their roots, here j is
the order of the polynomials, a=1 for slabs, a=2 for cylinders and a=3 for spheres.
> a:=3;
> p[i]:=proc(i,a,x)
> sum((-1)^s*(i!*GAMMA(a/2)*GAMMA(a/2+1+i+s))/((i-s)!*s!*GAMMA(a/
2+1+i)*GAMMA(a/2+s))*x^s,s=0..i);
> end;
> for j from 1 to 20 do
> p[j]:=p[i](j,a,x^2);
> fsolve(p[j]=0,x);
> solutions:='';
> for k from 1 to 2*j do
> evalf(solutions[k],20);
> od;
> od;

```

A - 5.2 The program for symbolic computation of the shifted Legendre polynomials in Maple V:

```
> #This maple program calculate the legendre polynomials and their roots by Gram-Schmidt
orthogonalization procedure
> #define inner product
> inner_product:=proc(f,g)
> int(f*g,x=0..1);
> end;
> #define a natural basis f[j](x)
> for j from 0 to 20 do
> f[j]:=x^j;
> od;
> #Gram-Schmidt orthogonalization procedure
> p[0]:=f[0];
> for k from 1 to 20 do
> ss:=0;
> for j from 0 to k-1 do
> ss:=ss+inner_product(f[k],p[j])/inner_product(p[j],p[j])*p[j];
> od;
> p[k]:=f[k]-ss;
> c[k]=inner_product(p[k],p[k]);
> od;
> #calculate their roots
> for k from 1 to 20 do
> k;
> fsolve(p[k],x);
> od;
```

A - 6 Reference:

1. R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Vol. 1, pp. 90-91, interscience 1953
2. P. M. Morse and H. Feschbach, *Methods of Theoretical Physics*, pp. 542, 780-81, 1754-55, McGraw-Hill 1953.
3. B.A. Finlayson, *The Method of Weighted Residuals and Variational Principles*, Academic Press(1972).
4. J.V. Villadsen, and W.E. Stewart, "Solution of Boundary Value Problems by Orthogonal collocation", *Chem. Eng. Sc.* 22, 1483 (1967).
5. C. Nelson Dorny, 1975, "A Vector Space Approach to Models and Optimizations." US: John Wiley & Sons Inc.
6. M.B. Monagan [et al.]; *Maple V programming guide*, Waterloo, ON, 1996

Bibliography

1. R. T. Cassidy, and E.S. Holmes, 1984, *Twenty-five years of progress in "adiabatic" adsorption processes*. A.I.Ch.E. Symp.Ser. 80, 68-75, 233.
2. D.M. Ruthven, S. Farooq, and K. Knaebel, *Pressure Swing Adsorption*, VCH publishers, Inc. 1994
3. C.W. Skarstrom , *U.S. patent No. 2,944,627* (1960), to Esso research and Engineering Company.
4. C.W. Skarstrom,, *Recent Developments in Separation Science*, Vol.2, P.95 CRC Press, Cleveland, Ohio, 1972.
5. S. Farooq, *Ph.D. thesis*, 1990. Department of Chemical Engineering, University of New Brunswick.
6. John Jones and Kulbir Singh, *Progress Report 3: Comparison of the Model with Experiment*, School of Engineering Science, Simon Fraser University, Feb, 1995
7. Charles A. Hall & Thomas A. Porsching, 1990, "*Numerical Analysis of Partial Differential Equations*". Prentice Hall, Englewood Cliffs, New Jersey.
8. W.E. Schiesser, 1994, "*The Numerical Method of Lines---Integration of Partial Differential Equations*", Academic Press, Inc.
9. Suhas V. Patankar, 1980, "*Numerical Heat Transfer and Fluid Flow*", Hemisphere Publishing Corporation.
10. R. B. Lantz, "Quantitative Evaluation of Numerical Diffusion (Truncation Error)", *Society of Petro Engineers Journal*, Sept.,1971
11. Patrick J. Roache, "On Artificial Viscosity", *Journal of Computational Physics*, 10, 1972
12. Incompressible Flow, 1972
13. B. P. Leonard, "A Consistency Check for Estimating Truncation Error due to Upstream Differencing", *Appl. Math. Modelling*, vol 2, December, 1978
14. B. P. Leonard, "A Survey of Finite Differences with Upwinding for Numerical Modelling of incompressible convective Diffusion Equation", 1980
15. B. P. Leonard and Simin Mokhtari, "Beyond First-order Upwinding: The ULTRA-SHARP Alternative for Non-oscillator Steady-state Simulation of Convection", *International Journal for Numerical Methods in Engineering*, vol.30, 1990
16. B.P. Leonard, "A stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation", *Computer Methods in Applied Mechanics and Engineering* 19, 59, 1979
17. P. F. Peterson, "A Method for Predicting and Minimizing Numerical Diffusion", *Numerical Heat Transfer, Part B*, vol 21, pp343-366, 1992
18. A. Kaniel, M. Mond and G.Ben-Dor, "An Artificial Dissipation Scheme for the Navier-Stokes Equations" *Int. J. Num. Meth Heat Fluid Flow*, vol. 3, 517-530,(1993).
19. W.F. Noh and M.H. Protter, *J.Math. & Mech.* 12 (1963), 149

20. P.H. Gaskell and A.K.C. Lau, "Curvature compensated convective transport: SMART, a new boundedness preserving transport algorithm", NASA TM 100916(ICOM-88-1), NASA-Lewis Research Center, 1988.
21. J. von Neumann and R.D. Richtmyer, "A method for the numerical calculations of hydrodynamical shocks", *J. Appl. Phys.* 21 (1950), 232
22. R. A. Gentry, R. E Martin, and B. J. Daly, . (1966). "An Eulerian Differencing Method for Unsteady compressible Flow Problem", *J. Comp. Phys.*, vol. 1, p. 87
23. William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, '*Numerical Recipes in C*', 1995.
24. G. Bader, and P. Deuffhard, 1983, *Numerische Mathematik*, vol. 41, pp. 373-398.
25. B.A. Finlayson , *The Method of Weighted Residuals and Variational Principles*, Academic Press (1972).
26. J.V. Villadsen, and W.E. Stewart, "Solution of Boundary Value Problems by Orthogonal collocation", *Chem. Eng. Sc.* 22, 1483 (1967).
27. C. Nelson Dorny, 1975, "A Vector Space Approach to Models and Optimizations." US: John Wiley & Sons Inc.
28. M.B. Monagan [et al.]; *Maple V programming guide*, Waterloo, ON, 1996
29. Anders Rasmuson and Ivars Neretnieks, "Exact Solution of a Model for Diffusion in Particles and Longitudinal Dispersion in Packed Beds", *AIChE Journal* (vol. 26, No.4 Page 686) July, 1980.
30. N.S. Raghavan and D.M. Ruthven, "Numerical Simulation of a Fixed-Bed Adsorption Column by the Method of Orthogonal Collocation", *AIChE Journal* (Vol. 29, No.6) Page 922, November 1983
31. N.S. Raghavan and D.M. Ruthven, "Numerical Simulation of a Kinetically Controlled Bulk Gas Separation", *AIChE Journal* (Vol. 31, No. 12) December, 1985 Page 2017
32. John Jones, 1996 "*Equations for PSA Simulation*", School of Engineering Science, Simon Fraser University.
33. Kulbir Singh and John Jones, "*Numerical Simulation of Air Separation by Piston-Driven Pressure Swing Adsorption*" School of Engineering Science, Simon Fraser University. July, 1996.