Fall 1995

Sensivity analysis and design calculations with biofiltration models

Michael Lawrence Cohen
New Jersey Institute of Technology

ABSTRACT

SENSITIVITY ANALYSIS AND DESIGN CALCULATIONS WITH BIOFILTRATION MODELS

by
Michael Lawrence Cohen

Biofiltration is a new technology for biological treatment of volatile organic compounds present in airstreams. It is a complex process and thus, engineering models which attempt to describe it are by necessity very involved and contain a large number of parameters.

In this study, two models describing biofiltration of airstreams carrying a single pollutant (VOC) were used in detailed parameter-sensitivity studies. One model concerned biofiltration under steady-state conditions, while the second described the transient behavior of the process. The intent of the sensitivity studies was to determine which model parameters need to be known with accuracy in order to allow for a good prediction of the size of a biofilter needed to achieve a given VOC-control objective. Studies with the steady-state biofiltration model have shown that accurate knowledge of the values of two kinetic parameters and the specific area of biofilm (and therefore the packing configuration) are essential. Studies with the transient model have revealed that in regards to transient behavior, the mass transfer coefficient is the most important design parameter.

Design calculations were also performed in this study for an integrated process involving soil venting and biofiltration for cleaning a contaminated aquifer. Preliminary

results (based on a number of simplifying assumptions) have shown that the proposed concept is plausible in the sense that a reasonable biofilter size is adequate for remediating a site in a relatively short period of time. It was also found that a given mass of contaminant can be treated more efficiently (shorter time, smaller biofilter volume) under constant venting rate if the volume of the aquifer is smaller (i.e., when the residence time of air in the aquifer is larger). This finding could be taken advantage of through faster remediation of a spill (before it spreads), or if seasonal variations affect the size of the aquifer.

SENSITIVITY ANALYSIS AND DESIGN CALCULATIONS WITH BIOFILTRATION MODELS

by
Michael Lawrence Cohen

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Environmental Engineering

Department of Civil and Environmental Engineering

January 1996

Blank Page

APPROVAL PAGE

SENSITIVITY ANALYSIS AND DESIGN CALCULATIONS WITH BIOFILTRATION MODELS

Michael Lawrence Cohen

Dr. Basil C. Baltzis, Thesis Advisor   Date
Professor of Chemical Engineering, NJIT

Dr. Hsin-Neng Hsieh, Committee Member   Date
Professor of Civil and Environmental Engineering, NJIT

Dr. Gordon A. Lewandowski, Committee Member   Date
Professor of Chemical Engineering and Chairman, NJIT

BIOGRAPHICAL SKETCH

Author: Michael Lawrence Cohen

Degree: Master of Science in Environmental Engineering

Date: January 1996

Undergraduate and Graduate Education:

- Master of Science in Environmental Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1996

- Bachelor of Science in Civil Engineering / Engineering and Public Policy,
  Carnegie Mellon, Pittsburgh, PA, 1994

Major: Environmental Engineering

This thesis is dedicated to
Dr. Indira Nair
without whom I would not have gotten this far.

ACKNOWLEDGMENT

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

A_S : biolayer surface area per unit volume of biofilter (m^{-1})

A_S^* : total surface area available for biolayer formation and adsorption per unit volume of biofilter (m^{-1})

c_G : concentration of pollutant in air entering the extraction well (g/m^3)

c_j : concentration of substance j in the air at a position h along the biofilter (g/m^3)

c_j^* : equilibrium concentration of a pollutant j at the gas/solid interface (g/m^3)

c_{ji} : value of c_j at h = 0 (g/m^3)

c_{jp} : concentration of substance j on the solid particle (g of pollutant j-absorbed/g particle)

c_L : concentration of pollutant in the contaminated aquifer (g/m^3)

$c_{L,0}$: concentration of pollutant in the contaminated aquifer at t = 0 (g/m^3)

c_O : oxygen concentration in the air at a position h along the biofilter (g/m^3)

c_{Oi} : oxygen concentration in the air at the inlet of the biofilter (g/m^3)

$\bar{c}_j$: dimensionless concentration of a compound j in the air ($\bar{c}_j = c_j/c_{ji}$)

$\bar{c}_{j,0}$: value of $\bar{c}_j$ at t = 0

$\bar{c}_{jp}$: dimensionless concentration of substance j on the solid particle defined as $(1-\upsilon)\rho_p c_{jp}/\upsilon c_{ji}$

$\bar{c}_{jP,0}$: value of $\bar{c}_{jP}$ at t = 0

$\bar{c}_O$: dimensionless concentration of oxygen in the air ($\bar{c}_O = c_O/c_{Oi}$)

$\bar{c}_{O,0}$: value of $\bar{c}_O$ at t = 0

D_{jw} : diffusion coefficient of a compound j in water (m^2/h)

D_{OW} : diffusion coefficient of oxygen in water (m^2/h)

e_j : effectiveness factor based on a pollutant j

e_O : effectiveness factor based on oxygen

F : flow rate of airstream (m^3/h)

$f(X_v)$: ratio of diffusivity of a compound in the biofilm to that in water

$g(\bar{c}_j, \bar{c}_O)$: dimensionless version of $\mu_j(s_j, s_O)$

h : position in the column (m); h = 0 at the entrance h = H at the exit

H : total height of the biofilter bed (m)

k_a : mass transfer coefficient between the gas and the solid surface (m/h)

k_d : Freundlich isotherm parameter

K_j : constant in the specific growth rate expression of a culture growing on compound j (g-compound j/m^3)

K_{Ij} : inhibition constant in the specific growth rate expression of a culture growing on compound j (g-compound j/m^3)

K_O : constant in the specific growth rate expression of a culture, expressing the effect of oxygen (g-oxygen/m^3)

L : rate of VOC-mass supply to the biofilter (g/h)

m_j : distribution coefficient for the substance j/water system

m_O : distribution coefficient for the oxygen-in-air/water system

n : Freundlich isotherm parameter

Q_G : flowrate of air entering the sparging point (m^3/h)

NOMENCLATURE
(Continued)

s_j : concentration of substance j at a position x in the biolayer at a point h along the column (g/m^3)

s_O : oxygen concentration at a position x in the biolayer, at a point h along the column (g/m^3)

$\bar{s}_j$: dimensionless concentration of a compound j in the biolayer ($\bar{s}_j = s_j/K_j$)

$\bar{s}_O$: dimensionless oxygen concentration in the biolayer ($\bar{s}_O = s_O/K_O$)

t : time (h)

u_g : superficial air velocity (m/h)

V_L : volume of liquid in the contaminated aquifer (m^3)

V_P : volume of the biofilter bed (m^3)

X_V : biofilm density (kg/m^3)

x : position in the biolayer (m)

Y_j : yield coefficient of a culture on compound j (g-biomass/g-compound j)

Y_{Oj} : yield coefficient of a culture on oxygen (g-biomass/g-oxygen), when VOC j is the carbon source

z : dimensionless position in the biofilter ($z = h/H$)

GREEK SYMBOLS

α : fraction of total surface area available for biofilm formation

β_1 : dimensionless parameter defined as $e_j \alpha \delta A_S^* X_V H \mu^* / Y_j u_g c_{ji} \upsilon$

β_2 : dimensionless parameter defined as $e_O \alpha \delta A_S^* X_V H \mu^* / Y_O u_g c_{Oi} \upsilon$

β_3 : dimensionless parameter defined as $k_a (1-\alpha) A_S^* H / u_g \upsilon$

γ : inverse dimensionless inhibition constant ($\gamma = K_j / K_{Ij}$)

δ : effective biolayer thickness (m)

ε_1 : dimensionless quantity defined as $c_{ji} / (K_j m_j)$

ε_2 : dimensionless quantity defined as $c_{Oi} / (K_O m_O)$

ζ : dimensionless time defined as t/τ

η : dimensionless quantity defined as $A_S f(X_V) D_{jw} K_j \tau / (\delta c_{ji})$

θ : dimensionless position in the biolayer ($\theta = x/\delta$)

λ : dimensionless quantity defined as $D_{jw} K_j Y_j / (D_{Ow} K_O Y_{Oj})$

$\mu_j(s_j, s_O)$: specific growth rate (h^{-1})

μ_j^* : constant in the specific growth rate expression for substance j (h^{-1})

ρ_p : density of the solid particles (g/m^3)

τ : space time defined as V / F

υ : void fraction of the biofilter bed

ϕ^2 : square of Thiele modulus based on compound j, defined as $\mu_j^* \delta^2 X_V / (f(X_V) D_{jw} K_j Y_j)$

ψ : dimensionless quantity defined as $1/c_{ji}[\upsilon c_{ji}/(1-\upsilon)\rho_p k_d]^{1/n}$

ω : dimensionless quantity defined as $K_O D_{Ow} c_{ji} / (K_j D_{jw} c_{Oi})$

SPECIAL SUBSCRIPTS

0   : refers to conditions at $t = 0$

i   : refers to conditions at $h = 0$, or $z = 0$

j   : compound j

$j = B$: compound is butanol

$j = T$: compound is toluene

CHAPTER 1

INTRODUCTION

Nature has the ability to decompose and recycle all substances produced by mankind. It can thus, treat all substances which are considered as wastes. However, environmental pollution is a serious problem. This arises from the fact that mankind produces wastes at rates much higher than those at which nature can decompose them. In addition, high levels of certain pollutants impact the health of humans and other living species while they destroy the natural ecosystem. Consequently, pollution control becomes an important issue and various laws regulate the presence of various pollutants in the air, water, and soil.

One of the most important groups of pollutants currently being regulated is that of Volatile Organic Compounds (VOCs). These compounds are among the major precursors, along with NO_X, of ground-level ozone. In an effort to reduce the occurrence of this ozone the Clean Air Act Amendments of 1990 (CAAA) have introduced restrictions on the amounts of VOCs that may be emitted. Presently, these restrictions are only on large-scale industrial operations. Eventually, these restrictions will be extended to include smaller scale contributors such as bakeries, dry cleaners, gasoline stations, and storage facilities [Baltzis and Wojdyla (1995a)].

In order to control VOC emissions many processes have been devised, including thermal and catalytic incineration, flares, condensers and adsorption [Moretti and Mukhopadhyay (1993)]. These processes are currently the most popular, but there are

several newer, possibly more effective and economical ones on the horizon. One of the more promising of these new technologies is biofiltration.

Biofiltration is a biological method for VOC treatment. It is based on the oxidation of VOCs by bacterial and fungal species which utilize the various compounds as a carbon or energy source. It can be argued that biofiltration is a naturally occurring process which is now used in a controlled manner as a specially designed technology. The first applications of biofiltration were in conjunction with wastewater [Fouhy and Grinthal (1994)], and later for odor control purposes. Presently biofilters are in use within many processes including aroma extraction, foundries, sewage (municipal and industrial), plastics processing, adhesives and rendering. They are used for the removal of many irritants and pollutants including odors, oxygen-, sulfur-, and nitrogen-containing organics and hydrogen sulfide [Fouhy, 1992]. In recent decades, there has arisen an interest in using biofiltration for dealing with emissions of hazardous and toxic substances.

There are a number of recent studies on biofiltration and they are reviewed in the next chapter of this thesis. Some of these studies have led to the development of mathematical models [e.g., Ottengraf and van den Oever (1983), Shareefdeen et al. (1993), Shareefdeen and Baltzis (1994)] with varying degrees of complexity. Models are very important for optimal engineering design of biofilters as well as for determining if biofiltration is an economically viable option for a given application.

Biofiltration models (both for steady-state and transient operation) involve a large number of parameters. Accurate knowledge of all these parameters requires extensive

and expensive experimental work, something which is not practical when biofiltration is considered in preliminary decision making. It thus becomes important to know which parameters are most essential for a good estimation of the size of biofilters. This determination can be made through sensitivity studies with the models, and this was the main objective of the study presented in this thesis.

Biofiltration is a process which was originally conceived for treatment of air which is unintentionally polluted. Conceivably though, it could be also used for treating air which is intentionally contaminated in order to resolve other pollution problems. For example, if soil or an aquifer is contaminated with VOCs air can be used for decontaminating them. In this case the pollution problem is intentionally removed from one phase to another; more specifically it is transferred to the air. This contaminated air could be subsequently treated in a biofilter where the pollutants are destroyed. This idea of an integrated soil venting/biofiltration process was also considered as part of the present thesis. Clearly the idea of soil venting is not new. Actually, a number of studies on soil venting or air sparging exist in the literature and are reviewed in the next chapter of this thesis. With regard to the integrated process, some preliminary calculations were performed based on detailed biofiltration models, but simple models describing air sparging (vapor extraction).

CHAPTER 2

LITERATURE REVIEW

2.1 Biofiltration

In recent years biological processes have been shown to be effective in treating VOC contaminated airstreams. There are two major biological systems for treating these airstreams: classical biofilters and biotrickling filters. These processes can be effective for a wide range of pollutants, are usually relatively inexpensive, produce environmentally harmless emissions and can be easily maintained for a relatively long period of time [Zilli et al. (1992)].

The classical biofilter (or simply biofilter) consists of a porous solid support placed either in an open or closed structure (reactor). Microorganisms having the ability to biodegrade volatile substances are immobilized on this solid support. The contaminated airstream passes through the classical biofilter, the contaminants enter the wet biofilm layer surrounding the particles, and are there degraded by the microorganisms. There is no liquid stream in a classical biofilter [Shareefdeen and Baltzis (1994), Hodge and Devinny (1994)]. The main factors affecting this process are the support (packing), the identity and properties of the contaminants, the microbial species and their characteristics, and the rates of VOC removal which can be achieved with a given unit.

The biotrickling filter is similar to the classical biofilter but it involves a recirculating liquid phase. This allows for continuous removal of chemicals such as

chlorinated compounds, which cause a change in the pH value. In these units pH-control is achieved by the addition of an acid or base when necessary. Biotrickling filters have also been shown to retain higher amounts of biomass in some instances, leading to higher degradation rates [Togna and Singh (1994)].

Due to the interest in biofiltration a lot of feasibility studies have been performed in the recent years. Detailed reviews of studies can be found in Shareefdeen (1994) and Androutsopoulou (1994). Removal rates vary with the compounds used, the size of biofilter, the concentration of pollutants in the airstream entering the biofilter, the packing material, etc. For example, Zilli et al. (1992) reported removal rates of phenol up to 124 $g/m^3/h$ with a packing material consisting of peat and glass beads in a 2:1 volume ratio. In their experiments they used *Pseudomonas* sp. Ottengraf and van den Oever (1983) studied the removal of mixtures (containing among others toluene and butanol) and reported maximum removal rates in the range of 20-40 $g/m^3/h$ depending on the identity of the pollutant. The packing material used was primarily a peat compost. Regarding ethanol, Baltzis and Androutsopoulou (1994) have reported removal rates up to 40 $g/m^3/h$ under steady-state conditions and much higher ones under transient conditions; they used a packing material consisting of peat and perlite (2:3 per volume). For the same substance (ethanol), Hodge and Devinny (1994) have reported removal rates ranging from 53 to 219 $g/m^3/h$ in biofilters packed with compost, or granular activated carbon (GAC), or a mixture of compost and diatomaceous earth. The foregoing review is not a complete account of existing studies, but it shows that various parameters affect the performance of biofilters.

Since the present thesis dealt with modeling of classical biofilters, following is a review of existing models.

The first model of biofiltration under steady-state conditions was published by Ottengraf and van den Oever (1983). Although the model was based on experiments with a mixture of VOCs, it essentially concerns removal of a single compound as it does not take into account potential interactions between pollutants. In addition, this model does not account for the potential impact of oxygen availability (although biofiltration is an aerobic process) and uses zero- or first-order kinetics with regard to the pollutant. Due to the extensive simplifying assumptions this model can be solved analytically, but is now considered as unrealistic.

The first detailed model describing steady-state biofiltration of a single VOC was published by Shareefdeen et al. (1993) and describes potential oxygen limitations of the process, while it accounts for more detailed (in fact complex) expressions for the degradation rate with regard to the VOC. Based on experiments with methanol, this model predicts that under most conditions oxygen is the limiting factor from the mass-transfer view point while the carbon source (methanol) is limiting from the kinetics point of view. The same model was used by Androutsopoulou (1994) who experimentally studied the removal of ethanol and butanol in two separate units. She reached the same conclusions regarding oxygen and VOC limitation as those of Shareefdeen et al. (1993). The same model was used in describing biofiltration of benzene and toluene in two separate columns [Shareefdeen (1994)]. In this case, it was found that although oxygen affects the process to a certain extent, limitation both from kinetics and mass-transfer

viewpoints is determined by the VOC. The conclusion from the foregoing studies is that although oxygen should be always considered it has to be definitely accounted for in cases where a hydrophilic compound is treated. This was not done by Hodge and Devinny (1994) who modeled ethanol biofiltration data along the lines of Ottengraf and van den Oever (1983). However, this new model incorporates rates of carbon dioxide evolution.

Models for steady-state biofiltration of VOC mixtures fall in two categories. Baltzis and Sharefdeen (1994) have proposed a model which accounts for competitive inhibition between pollutants and also accounts for oxygen effects. This model has been experimentally validated with mixtures of benzene and toluene. A model which accounts for competitive inhibition but neglects oxygen effects has been also used by Deshusses et al. (1995) based on experiments with MEK (methyl-ethyl-ketone) and MIBK (methyl-isobutyl-ketone). Recently, Baltzis and Wojdyla (1995b) have proposed a model which accounts for species differentiation in the biofilter bed. This model accounts for oxygen effects and assumes the formation of separate biofilm patches for each pollutant. This model has been used in explaining data of ethanol/butanol mixtures removal.

Transient biofiltration is a much more complex case since the process is complicated by the presence of adsorption/desorption effects. Shareefdeen and Baltzis (1994) were the first to propose a model for transient biofiltration of single VOCs. The model is an extension of the steady-state model proposed earlier by the same investigators [Shareefdeen et al. (1993)] and has been experimentally validated for the case of toluene removal under transient conditions. A transient model has also been

proposed by Deshusses et al. (1995) and used for describing MEK or MIBK transient removal. This model does not account for oxygen limitations and instead of adsorption it uses absorption (dissolution) of the VOCs in the water retained within the pores of the packing material.

The present thesis dealt with single VOCs and used the models of Shareefdeen et al. (1993) and Shareefdeen and Baltzis (1994) in sensitivity and calculational studies. It should also be added that since this thesis dealt with classical biofilters, studies of biotrickling filters have not been reviewed. These studies are few in number and the only one on modeling has been published by Diks and Ottengraf (1991).

2.2 Soil Venting

Soils can become contaminated with VOCs in many ways. The main sources of contamination are intentional dumping, accidental spills and leaks in underground storage tanks. There are a number of current practices for removing these VOCs. They include excavation of the entire site followed by treatment of the soil, and flushing of the aquifer followed by treatment of the water [Hutzler et al. (1991)]. These methods are very expensive and time consuming, and for this reason new approaches and technologies are currently being investigated.

Soil venting (also known as soil vapor extraction or air sparging) is a new remediation technique which is both significantly cheaper and less time consuming than the current methods. In this process, VOCs are removed from the contaminated aquifer by forcing (sparging) air through the soil, and into the aquifer. The VOCs are volatized

out of the water, and into the airstream. These VOCs are carried out of the ground, either into a vapor extraction system, or when permissible, into the atmosphere [Angell (1991)]. It has been found that soil venting can be effective in removing a wide range of VOCs, under numerous conditions [Hutzler et al. (1991)].

Wilson et al. (1987) have proposed a model for describing the removal of VOCs from the subsurface using forced venting. In this model a single component (gasoline) is forced upward through a uniform soil. Only diffusion through the soil is considered, with the only soil parameter used being porosity. This model was validated experimentally in the laboratory, and was found to need improvement.

Silka et al. (1991) proposed a simplistic mass balance model which assumed equilibrium between the liquid and gas. Transport through soil was described by a single variable, the effective diffusion coefficient (an overall coefficient accounting for partitioning, adsorption and tortuosity). This model has been found to have good qualitative agreement with data from vapor extraction of TCE.

Gierke et al. (1992) proposed a detailed model of vapor extraction. This model is more complex than previous ones, as it includes nonequilibrium effects and more soil parameters such as soil particle density, soil sorption capacity, aggregate radius and degree of saturation. These additional parameters make the model more realistic. Laboratory data on the extraction of toluene and methanol from Ottawa sand and an aggregated porous soil material were found to be in excellent agreement with the predictions of the model.

Soil venting is a complex process which appears to need further detailed investigation both from the experimental and the modeling viewpoint. In the present thesis equilibrium between air and water (regarding the VOC) was assumed while soil parameters were not considered.

CHAPTER 3

OBJECTIVES

There were two main objectives in this study. The first was to perform detailed parameter-sensitivity studies with steady-state and transient biofiltration models. The second objective was to perform design calculations for an integrated process involving air stripping (soil venting) and biofiltration for treatment of a contaminated aquifer.

All studies were based on existing models and computer codes which are applicable for cases where the contaminated airstream contains a single pollutant.

The intent of the sensitivity studies was to determine which model parameters need to be accurately known so that the size a biofilter needed for meeting a given VOC-control objective can be safely predicted.

The intent of the design calculations for the integrated soil venting/biofiltration process was to obtain some preliminary knowledge regarding the size of the required biofilter and the time frame within which a contaminated aquifer could be remediated.

Sensitivity studies with the steady-state biofiltration model were extensions of similar studies by Shareefdeen (1994) and Androutsopoulou (1994). They were performed as follows. Two basic sets of model-parameters were first selected: one for toluene (a hydrophobic compound) and one for butanol (a hydrophilic compound). Each model parameter -one at a time- was varied within a range from 0.1 to 10 times its base value and the biofilter size required for performing a given duty was calculated. In most instances, the space (residence) time of the airstream in the biofilter rather than the

11

volume of the biofilter itself was calculated. This was done because the volume (V) is related to the space time (τ) through the relation $\tau = V/F$, where F is the volumetric flowrate of the airstream supplied to the biofilter, and the model gives the same results for a given value of τ regardless of the individual values of V and F. The results of these studies are presented in Chapter 4 of the thesis.

Sensitivity studies with the transient biofiltration model were performed with a single set of basic parameter values (toluene). Only parameters appearing in the transient but not in the steady-state model were examined. These parameters refer to adsorption equilibrium and mass transfer characteristics between VOC and packing material of the biofilter. The objective here was to investigate the effect of parameters on the time required to achieve steady-state and on the size of the biofilter. The size of a biofilter can be affected by transient behavior parameters due to potential concentration overshoots. Results from these studies are presented in Chapter 5.

For the integrated soil venting/biofiltration process, it was assumed that the aquifer was contaminated with toluene only. One set of biofiltration parameters was used, while it was also assumed that the air passed through the aquifer carries toluene at equilibrium concentrations at all times. The rate of soil venting was changed so that the biofilter operates under relatively constant toluene concentrations over different time segments. Various criteria, such as the threshold limit value (TLV) and acceptable source impact level (ASIL), for toluene were taken into account for the design calculations. Results from this part of the study are presented in Chapter 6 of the thesis.

CHAPTER 4

STEADY-STATE BIOFILTRATION OF SINGLE VOCs

The work presented in this chapter refers to biofiltration under steady-state conditions when the contaminated airstream contains a single VOC. The process had been earlier modeled by Shareefdeen et al. (1993). This model was also used in the studies reported here. The model, which is presented in the next section of this chapter, contains a large number of parameters. These parameters fall into two different categories: model parameters and operating parameters. Model parameters are those which are specific to a given system; i.e., the packing material (e.g., biofilm specific area), characteristics of the biofilm (e.g., density), etc. Operating parameters are those which can be varied through design and they involve the concentration of the VOC in the airstream supplied to the biofilter, the rate of either air supply (volumetric) or VOC supply (mass) to the filter-bed, etc.

The work presented here falls into two categories. The first category of studies involved sensitivity studies with regard to the model parameters. The intent of these studies was to examine the impact of potential uncertainties in the values of a given model parameter on the predicted required size for a biofilter. In these studies, the biofilter was assumed to be supplied with a given VOC concentration and required to achieve a given percent removal of the pollutant. Sensitivity studies were performed by changing the value of one model parameter at a time.

As basis for the sensitivity studies, model parameter values from earlier studies were used. These studies concerned butanol which is a hydrophilic substance [Baltzis and Androutsopoulou (1994)], and toluene which is a hydrophobic substance [Shareefdeen and Baltzis (1994)]. The earlier reported/measured parameter values are referred to here as reference values.

The second category of studies reported here concern the operating parameters. Using the reference values of model parameters, the intent here was to find the required biofilter volume as a function of the inlet pollutant concentration for given values of rate of pollutant mass supply (load) to the biofilter bed. These studies are referred to as sizing calculations.

As mentioned earlier, two sets of reference parameter values were employed in the present study. The intent was to find if the same or different parameters are most important in the cases of hydrophilic and hydrophobic solvents. The use of parameters for butanol and toluene was due to the fact that these substances had been earlier used in detailed biofiltration experiments [Androutsopoulou (1994), Shareefdeen (1994)].

4.1 Model Equations and Numerical Methods

For airstreams carrying a single volatile organic compound (VOC) the biofiltration process in a classical biofilter can be described by the following mass balances, under steady-state conditions [Shareefdeen et al. (1993), Shareefdeen (1994)].

I. Mass balance for VOC j in the biolayer at a position h along the biofilter bed:

$$f(X_v)D_{jw}\frac{d^2s_j}{dx^2} = \frac{X_v}{Y_j}\mu_j(s_j,s_O)$$
(4.1)

with boundary conditions,

$$s_j = \frac{c_j}{m_j} \qquad \text{at} \quad x = 0$$
(4.2)

$$\frac{ds_j}{dx} = 0 \qquad \text{at} \quad x = \delta$$
(4.3)

Equation (4.1) implies that the rate of diffusion of VOC j in the biolayer is equal to the rate of its biodegradation.

II. Mass balance for oxygen in the biolayer at a position h along the biofilter bed:

$$f(X_v)D_{Ow}\frac{d^2s_O}{dx^2} = \frac{X_v}{Y_{Oj}}\mu_j(s_j,s_O)$$
(4.4)

with boundary conditions,

$$s_O = \frac{c_O}{m_O} \qquad \text{at} \quad x = 0$$
(4.5)

$$\frac{ds_O}{dx} = 0 \qquad \text{at} \quad x = \delta$$
(4.6)

Equation (4.4) implies that the rate of oxygen diffusion in the biolayer is equal to the rate of oxygen consumption in the biodegradation process.

III. Mass balance for VOC j in the airstream at a position h along the biofilter bed:

$$\frac{H}{\tau}\frac{dc_j}{dh} = A_s f(X_v)D_{jw}\left[\frac{ds_j}{dx}\right]_{x=0}$$
(4.7)

with boundary condition,

$$c_j = c_{ji} \qquad \text{at} \quad h = 0 \qquad (4.8)$$

Equation (4.7) implies that the rate of loss of VOC j from the air along the biofilter is equal to the rate at which VOC j is transferred to the biolayer. This transfer is expressed as a flux.

IV. Mass balance for oxygen in the airstream at a position h along the biofilter bed:

$$\frac{H}{\tau} \frac{dc_O}{dh} = A_s f(X_v) D_{Ow} \left[\frac{ds_O}{dx} \right]_{x=0} \qquad (4.9)$$

with boundary condition,

$$c_O = c_{Oi} \qquad \text{at} \quad h = 0 \qquad (4.10)$$

The terms in equation (4.9) have the same meaning for oxygen as those in equation (4.7) have for VOC j.

Function $\mu(s_j, s_O)$ which appears in equations (4.1) and (4.4) assumes the following form,

$$\mu_j(s_j, s_O) = \frac{\mu_j^* s_j s_O}{\left(K_j + s_j + s_j^2 / K_{1j} \right)\left(K_O + s_O \right)} \qquad (4.11)$$

when the degradation kinetics follow an Andrews expression with respect to the availability of the carbon source (VOC), and a Monod expression with respect to the availability of oxygen.

As has been shown by Shareefdeen et al. (1993) and Shareefdeen (1994) the model equations above can be brought in a dimensionless form once the following quantities are introduced,

$$\bar{s}_j = \frac{s_j}{K_j}, \qquad \bar{s}_O = \frac{s_O}{K_O}, \qquad \bar{c}_j = \frac{c_j}{c_{ji}}, \qquad \bar{c}_O = \frac{c_O}{c_{Oi}}, \qquad \gamma = \frac{K_j}{K_{1j}},$$

$$\lambda = \frac{D_{jw}K_jY_j}{D_{Ow}K_OY_{Oj}}, \qquad \phi^2 = \frac{\mu_j^*\delta X_v}{f(X_v)D_{jw}K_jY_j}, \qquad \eta = \frac{A_sf(X_v)D_{jw}K_j\tau}{\delta c_{ji}},$$

$$\omega = \frac{K_OD_{Ow}c_{ji}}{K_jD_{jw}c_{Oi}}, \qquad \varepsilon_1 = \frac{c_{ji}}{K_jm_j}, \qquad \varepsilon_2 = \frac{c_{Oi}}{K_Om_O}, \qquad \theta = \frac{x}{\delta}, \qquad z = \frac{h}{H}$$

Equations (4.1)-(4.10), when expression (4.11) is also taken into account, take correspondingly the following form.

$$\frac{d^2\bar{s}_j}{d\theta^2} = \phi^2 \frac{\bar{s}_j\bar{s}_O}{\left(1 + \bar{s}_j + \gamma\bar{s}_j^2\right)\left(1 + \bar{s}_O\right)} \tag{4.12}$$

$$\bar{s}_j = \varepsilon_1\bar{c}_j \qquad \text{at} \quad \theta = 0 \tag{4.13}$$

$$\frac{d\bar{s}_j}{d\theta} = 0 \qquad \text{at} \quad \theta = 1 \tag{4.14}$$

$$\frac{d^2\bar{s}_O}{d\theta^2} = \phi^2\lambda \frac{\bar{s}_j\bar{s}_O}{\left(1 + \bar{s}_j + \gamma\bar{s}_j^2\right)\left(1 + \bar{s}_O\right)} \tag{4.15}$$

$$\bar{s}_O = \varepsilon_2\bar{c}_O \qquad \text{at} \quad \theta = 0 \tag{4.16}$$

$$\frac{d\bar{s}_O}{d\theta} = 0 \qquad \text{at} \quad \theta = 1 \tag{4.17}$$

$$\frac{d\bar{c}_j}{dz} = \eta\left[\frac{d\bar{s}_j}{d\theta}\right]_{\theta=0} \tag{4.18}$$

$$\bar{c}_j = 1 \qquad \text{at} \quad z = 0 \tag{4.19}$$

$$\frac{d\bar{c}_O}{dz} = \eta\omega\left[\frac{d\bar{s}_O}{d\theta}\right]_{\theta=0} \tag{4.20}$$

$$\bar{c}_O = 1 \qquad\qquad \text{at } z = 0 \tag{4.21}$$

As has been shown by Shareefdeen et al. (1993), the four dependent variables $(\bar{s}_j, \bar{s}_O, \bar{c}_j, \bar{c}_O)$ are interrelated through the following two algebraic equations:

$$\bar{c}_j = \frac{\bar{c}_O - 1}{\lambda\omega} + 1 \tag{4.22}$$

$$\bar{s}_j = \frac{\bar{s}_O}{\lambda} - \varepsilon_1\left(\frac{1}{\lambda\omega} - 1\right) - \frac{1}{\lambda}\left(\varepsilon_2 - \frac{\varepsilon_1}{\omega}\right)\bar{c}_O \tag{4.23}$$

It is easy to show that equations (4.22) and (4.23) can be equivalently expressed as follows:

$$\bar{c}_O = \lambda\omega\left(\bar{c}_j - 1\right) + 1 \tag{4.24}$$

$$\bar{s}_O = \lambda\bar{s}_j - \lambda\left(\varepsilon_1 - \omega\varepsilon_2\right)\bar{c}_j - \varepsilon_2\left(\lambda\omega - 1\right) \tag{4.25}$$

Because of relations (4.22)-(4.25) one needs to solve two rather than four differential equations. There are two possible sets; set 1: equations (4.12)-(4.14), (4.18), (4.19), (4.24), and (4.25); set 2: equations (4.15)-(4.17), and (4.20)-(4.23). From the numerical point of view, one needs to keep differential equations referring to variables which exhibit the largest gradient. Hence, in cases where the VOC gets depleted faster than oxygen (in the biolayer) one needs to work with the equations of set 1, while in cases where oxygen is changing (in the biolayer) faster than the VOC one needs to work with the equations of set 2.

The volume (size) of a biofilter bed can be calculated via the following two formulas:

$$V = \tau F \tag{4.26}$$

$$V = \tau \frac{L}{c_{ji}} \tag{4.27}$$

where L is the rate of VOC-mass supply to the biofilter (mass of VOC supplied to the biofilter per unit time). Although L is called load here, it is recognized that the usual definition of load is "amount of VOC supplied to the biofilter per unit time and per unit volume of packing material."

The model equations have been solved through the use of computer codes which are based on the use of the orthogonal collocation method for solving equation (4.12) [or (4.15)], and the Runge-Kutta method for solving equation (4.18) [or (4.20)]. In cases where both set 1 and set 2 of the model equations had to be used, due to changes in what compound is consumed first in the biolayer at different locations of the biofilter bed, the location of the change was first determined and each segment was solved as a separate biofilter. The codes used were primarily those developed earlier [Shareefdeen et al. (1993); Shareefdeen (1994); Baltzis (1994)] while some parts were recently revised and/or refined by Tsangaris and Baltzis. These codes are given in Appendix C of this thesis.

4.2 Sensitivity Studies

As can be seen from equations (4.1)-(4.11) the model parameters are the following: kinetic parameters [Y_j, Y_{Oj}, and the four constants included in expression (4.11)]; diffusion coefficients (D_{jw}, D_{Ow}) of the VOC and oxygen in water; correction of diffusivities for biofilm [$f(X_V)$]; distribution coefficients or Henry's constants (m_j, m_O) for the VOC and oxygen; biofilm density (X_V); specific surface area of biofilm (A_S). The thickness of the active biofilm (δ) is a model parameter but varies with the location in the biofilter bed. It is calculated through a trial and error procedure through the computer code [Shareefdeen et al. (1993)]. The correction factor for diffusivities [$f(X_V)$] is a function of X_V as described by the correlation of Fan et al. (1990) and thus, it was not individually varied. With the exception of the yield coefficients, all other parameters were examined in sensitivity studies.

The results of the sensitivity studies are shown graphically in Figures A-1 through A-16. The x-axis in these graphs represents the relative value of the parameter studied. This relative value is defined as the ratio of the assumed value for that parameter to its reference one (reported in Table 4.1). The y-axis shows the residence (or space-) time required for achieving a given conversion of the pollutant. In all cases, two inlet concentrations of the pollutant (0.5 and 1.0 g/m^3) were considered and for each concentration two levels of removal (95% and 99%) were examined. Since the residence time, τ, is defined as F/V, once the flowrate is specified the volume can be calculated, or for a biofilter of a given size the value of τ dictates the value of the flowrate of the polluted airstream which can be treated (to the conversion indicated) in the given unit.

Table 4.1 Base-values of model parameters for solving the steady-state equations[*]

Parameter	Value	Units
A_S (butanol)	38	m^{-1}
A_S (toluene)	40	m^{-1}
c_{Oi}	275×10^{-3}	kg/m^3
D_{BW}	0.77×10^{-9}	m^2/s
D_{TW}	1.03×10^{-9}	m^2/s
D_O	2.41×10^{-9}	m^2/s
$f(X_V)$	0.195	—
μ_B^*	0.579	h^{-1}
μ_T^*	1.50	h^{-1}
K_B	0.952	kg/m^3
K_T	11.03×10^{-3}	kg/m^3
K_{IB}	0.857	kg/m^3
K_{IT}	78.94×10^{-3}	kg/m^3
K_O	0.26×10^{-3}	kg/m^3
m_B	0.00036	—
m_T	0.27	—
m_O	34.4	—
Y_B	0.458	kg/kg
Y_T	0.708	kg/kg
Y_{OB}	0.232	kg/kg
Y_{OT}	0.341	kg/kg
X_V	100	kg/m^3

[*] Values taken from Androutsopoulou (1994) and Shareefdeen (1994)

During the studies reported here, the value of each parameter was changed (by one-tenth increments) from one-tenth to two times its reference value. For the sensitivity studies, and for the inlet concentration values considered, the calculations showed that for the case of butanol it is always oxygen which is depleted first in the biolayer. Hence,equations of set 2 (section 4.1) were used in the calculations. In the case of toluene the calculations showed that toluene rather than oxygen gets depleted first in the biolayer. For this reason, all calculations for the toluene case are based on the solution of equations of set 1 (section 4.1).

4.2.1 Kinetic Parameters

From expression (4.11) one can see that the kinetic expression involves four constants. The sensitivity of the model to the values of these constants can be judged from the results shown in Figures A.1 through A.8.

Figures A-1 and A-2 show the sensitivity of the model to the values of parameter μ_j^*. As can be seen from the graphs, for relative μ_j^* values larger than 1 the predicted τ values change slightly while for relative values less than 1 there is a very substantial effect on the predicted τ values. The conclusion here is that for both hydrophilic and hydrophobic compounds, accurate knowledge of the value of μ_j^* is important. It is also interesting to notice that the curves representing 95% and 99% conversion are very close to one another in both cases considered for butanol (Figure A-1) while they are far apart in the case of toluene (Figure A-2). Although there is not enough evidence to be sure about it, this difference may be due to the different actual values of μ_B^* and μ_T^* (observe that μ_T^* is almost 3 times the value of μ_B^*).

Figures A-3 and A-4 show the sensitivity of the model to the value of kinetic parameter K_j. In both cases, as the value of K_j increases the value of τ also increases. However, for the case of butanol the changes in τ are insignificant, while for the case of toluene they are quite substantial. This difference could be possibly attributed to the actual values of K_B and K_T which are different by almost two orders of magnitude. As in the case of μ_j^* one can observe that the 95 and 99% conversion curves are very close to one another in the case of butanol as opposed to the case of toluene.

Figures A-5 and A-6 show the sensitivity of the model to the values of the inhibition constant K_{Ij}. In the case of butanol (Figure A-5) the model is sensitive only when K_{IB} is underestimated (relative values less than 1). In the case of toluene (Figure A-6) the model is totally insensitive to the value of parameter K_{IT}. It should be observed that the reference values of K_{IB} and K_{IT} differ by one order of magnitude. One could possibly argue that the more inhibitory a compound is (K_{Ij} small) the less sensitive the model is to the K_{Ij} value. It should be also observed that the value (reference) of K_{Ij} for toluene is so small that it essentially remains the same when the relative value is varied from 0.1 to 2. This may explain the completely flat profiles of Figure A-6.

Figures A-7 and A-8 show the sensitivity of the model to the value of the kinetic constant which is related to oxygen (K_O). In all cases the model is insensitive to the actual K_O value. This is an important result because the K_O values spanned during this study cover essentially the entire spectrum of K_O values which have been measured in various studies [Shareefdeen et al. (1993)].

If a general conclusion can be drawn from the sensitivity studies with the kinetic model parameters, it is the following. It appears that in all cases, two -not necessarily the same- kinetic parameters have to be known accurately or measured (μ_B^* and K_{IB} for butanol, μ_T^* and K_T for toluene). If two kinetic parameters are important, any effort to represent kinetics by a single parameter (zero- or first-order kinetics) is bound to lead to significant errors in sizing a biofilter unit.

4.2.2 Diffusion Coefficients

Figures A-9 and A-10 show the sensitivity of the model to the value of the diffusivity of the pollutant in water (or the biofilm since X_V is taken constant here). In the case of butanol the actual value of D_{BW} has absolutely no impact on the predicted value of τ, which is exactly the opposite to what happens in the case of toluene. It is worth observing from Figure A-10 that a serious mistake is made not only when D_{TW} is underestimated (relative value less than 1), but also when it is overestimated, as a slight overestimation leads to considerable underestimation of τ (observe that the τ values are large in the case of Figure A-10). This difference in behavior of butanol and toluene is hard to understand as the reference values of the two compounds are comparable. It may simply be the combined effect of the kinetics and mass transfer. This point needs further elaboration.

4.2.3 Distribution Coefficients

The sensitivity of the model to the value of Henry's constant, m_j, can be judged from the diagrams of Figures A-11 and A-12. For the case of toluene, m_T appears to be of considerable importance as there is an almost perfectly linear relation between the uncertainty in m_T and τ. In the case of butanol, for high inlet concentration values (Figure A-11b) severe underestimation of m_B creates problems. In all other cases the impact of m_B is minimal. It is interesting however to observe the shape of the constant conversion isoclines. They go through a minimum when the relative m_B value is about 0.7 in Figure A-11a and 1.4 in Figure A-11b. Although the existence of a minimum in

the case of Figure A-11b is not very clear from the graph, it does actually occur. It should be added here that since a lot of thermodynamic data are in fact available, it is unlikely that m_j is not known with certainty. What the graphs of Figures A-11 and A-12 actually (although indirectly) indicate, is the effect of uncertainty in the value of temperature which affects the m_j value.

4.2.4 Biofilm Density

Precise knowledge of the biofilm density, X_v, does not appear to affect biofilter size calculations unless the relative value is significantly less than 1. This can be seen from the diagrams of Figures A-13 and A-14. It should be also added that the reference (base) value for X_v is 100 kg/m^3. As has been discussed by Shareefdeen et al. (1993) the reported X_v values are between 23 and 220 kg/m^3. Hence, relative X_v values should be between 0.23 and 2.2. As can be seen from the graphs, the τ calculations are not sensitive to X_v for the aforementioned range of relative X_v values. A similar conclusion regarding toluene was reached by Shareefdeen and Baltzis (1994), who followed a different approach in sensitivity studies. It should be also mentioned here that since the correction for diffusivities [$f(X_v)$] is a function of X_v, the results of Figures A-13 and A-14 were actually obtained by varying both X_v and $f(X_v)$. The dependence of $f(X_v)$ on X_v was taken to be that reported by Fan et al. (1990).

4.2.5 Specific Surface Area of Biofilm

The parameter to which the model appears to be most sensitive is A_S, the specific area of biofilm. This can be seen from the graphs of Figures A-15 and A-16. For relative values larger than 0.5 there is an almost linear relationship of slope -1. That is, if the surface area is doubled, the required residence time is halved. For very small relative values, the relation between A_S and τ is almost exponential. The fact that A_S has such a major impact on predicting τ is not surprising as it directly relates to the amount of biomass (catalyst) which is in direct contact with the pollutant. This finding implies that the packing configuration is very important.

4.2.6 Conclusions from Sensitivity Studies

Based on the results obtained in the sensitivity studies discussed in the preceding sections, the following conclusions can be reached.

In all cases, parameters μ_j^* and A_S are very important. Also in all cases, a second kinetic parameter is also important, but its identity depends on the compound. If the compound is highly inhibitory this second kinetic parameter is K_j, while for non-strong inhibitors K_{Ij} is important. Finally, parameters m_j and D_{jw} appear to be significant only for highly volatile and inhibitory compounds.

4.3 Biofilter Sizing Calculations

As has been mentioned earlier, biofilter sizing calculations refer to studies with the operating parameters of the model. As can be seen from equations (4.1)-(4.11) these

parameters are the inlet concentrations of the VOC and oxygen (c_{ji}, c_{Oi}), the biofilter height (H), and the residence time (τ). Parameters H and τ always appear as a ratio (H/τ) and for this reason H was kept constant while τ was calculated. The inlet oxygen concentration was never varied since it was assumed that only atmospheric air is supplied to the biofilter. In these calculations the reference values of the model parameters were used (Table 4.1).

It should be mentioned here that for the case of butanol, as in the case of the sensitivity studies, it was found that oxygen is always depleted first. Thus, equations of set 2 were used (see preceding section). For the case of toluene, and for inlet concentration values exceeding 2.8 g/m^3, it was found that in a segment of the biofilter bed close to the entrance oxygen is depleted first (hence, set 2 was used), but in the remaining part of the reactor (till a desired exit concentration level or conversion is achieved) toluene is depleted first in the biofilm (hence, equations of set 1 were used). This switching requires a considerable amount of additional calculations in order to determine the exact location of the changeover.

Figures A-17 and A-18 show the required residence time for two conversion levels as a function of the inlet concentration. These diagrams can be used in determining the required volume of a unit in two different cases: when the flowrate of the contaminated airstream is specified, or when the load (rate of mass supply) is dictated. The conversion of τ to volume can be easily made via equations (4.26) and (4.27). Once again, it is worth noticing that in the butanol case the additional τ for getting 99% rather

than 95% conversion of the pollutant is not significant. However, for the case of toluene going from 95 to 99% conversion requires a significantly larger biofilter (Figure A-18).

Figures A-19 to A-22 show actual volume calculations when the load is specified. In this portion -for each value of load- four cases were considered: two based on conversion, and two based on exit concentrations. The most important feature of these diagrams is that there are cases (butanol, Figures A-19 and A-21) in which the isoclines go through a minimum. This is a finding originally reported by Baltzis (1994) for the case of ethanol. It suggests that if the inlet concentration is higher than that at which the minimum occurs one could -significantly- reduce the biofilter volume by diluting the contaminated airstream with pure air while keeping the load constant. The same is observed for the case of toluene, but it is only of mathematical/numerical importance as the minimum occurs at unrealistic concentration values in the order of 100 g/m^3.

CHAPTER 5

TRANSIENT BIOFILTRATION OF SINGLE VOCs

In this chapter, results from sensitivity studies with a transient biofiltration model are presented. This model was originally developed by Shareefdeen and Baltzis (1994) and is given in the next section of this chapter.

There is a substantial difference between steady-state and transient biofiltration and this is what makes the results presented here particularly important. This difference is that, with the materials used for packing classical biofilters, transient operation involves an extra process. This is the adsorption of VOCs to the packing material, a process which does not affect steady-state behavior since at steady-state the adsorption process is at equilibrium.

In general, the questions pertaining to transient operation refer to the time required for the system to (practically) reach steady-state and the form of the response. The form of the response may be such that concentrations exhibit an overshoot which is potentially unacceptable. Hence, the intent of the sensitivity studies with the transient biofiltration model was to examine the impact of various parameters on the key features of the response of a biofilter during transients.

5.1 Model Equations and Numerical Methodology

The basic model equations describing transient biofiltration of airstreams carrying a single pollutant (VOC) are mass balances written for three phases: biofilm, air, and solids (packing material). These equations, taken from Shareefdeen and Baltzis (1994), are as follows:

I. Mass balance for VOC j and for oxygen in the biofilm:

$$\frac{\partial s_j}{\partial t} = f(X_v)D_{jw}\frac{\partial^2 s_j}{\partial x^2} - \frac{X_v}{Y_j}\mu_j(s_j,s_O) \tag{5.1}$$

$$\frac{\partial s_O}{\partial t} = f(X_v)D_{Ow}\frac{\partial^2 s_O}{\partial x^2} - \frac{X_v}{Y_O}\mu_j(s_j,s_O) \tag{5.2}$$

II. Mass balances for VOC j and for oxygen in the gas phase:

$$\upsilon\frac{\partial c_j}{\partial t} = -u_g\frac{\partial c_j}{\partial h} + D_{jw}f(X_v)\alpha A_s^*\left(\frac{\partial s_j}{\partial x}\right)_{x=0} - k_a(1-\alpha)A_s^*(c_j - c_j^*) \tag{5.3}$$

$$\upsilon\frac{\partial c_O}{\partial t} = -u_g\frac{\partial c_O}{\partial h} + D_{Ow}f(X_v)\alpha A_s^*\left(\frac{\partial s_O}{\partial x}\right)_{x=0} \tag{5.4}$$

III. Mass balance for VOC j in the solid phase (particles):

$$(1-\upsilon)\rho_P\frac{\partial c_{jp}}{\partial t} = k_a(1-\alpha)A_s^*(c_j - c_j^*) \tag{5.5}$$

Equations (5.1) - (5.4) are the unsteady state versions of equations (4.1), (4.4), (4.7), and (4.9), respectively. The sole difference is the last term in equation (5.3) which stands for the rate of mass transfer of VOC j to the solid particles (excluding biolayer). Equation

(5.5) expresses that the rate of accumulation of VOC j in the solid particles is equal to the rate of mass transfer of VOC j to the particles. These equations are a set of partial differential equations, and their corresponding initial and boundary conditions can be found in Shareefdeen and Baltzis (1994) and Shareefdeen (1994).

Function $\mu_j(s_j, s_O)$ appearing in equations (5.1) and (5.2) expresses the kinetics of biodegradation and -as was also the case in Chapter 4- it is given by the following expression:

$$\mu_j(s_j, s_O) = \frac{\mu_j^* s_j}{K_j + s_j + \dfrac{s_j^2}{K_{Ij}}} \frac{s_O}{K_{Oj} + s_O} \tag{5.6}$$

The driving force for the mass transfer of the VOC from the air (gas phase) to the particles (solid phase) is $c_j - c_j^*$, as indicated in equations (5.3) and (5.5). Concentration c_j^* is related to the concentration of VOC in the solid phase through an adsorption isotherm. For the case of toluene, this isotherm was found [Shareefdeen (1994)] to follow the Freundlich equation. Hence, one can write

$$c_{jp} = k_d (c_j^*)^n \tag{5.7}$$

Equations (5.1) - (5.5) have not been exactly solved to date. They have been solved through an approximation which introduces the use of effectiveness factors defined as,

$$e_j = \frac{f(X_v)D_{jw}\left(\dfrac{\partial s_j}{\partial x}\right)_{x=0}}{\delta\dfrac{X_v}{Y_j}[\mu_j(s_j, s_0)]_{x=0}} \qquad (5.8)$$

$$e_0 = \frac{f(X_v)D_{0w}\left(\dfrac{\partial s_0}{\partial x}\right)_{x=0}}{\delta\dfrac{X_v}{Y_0}[\mu_j(s_j, s_0)]_{x=0}} \qquad (5.9)$$

Actually it can be easily shown that $e_j = e_0$. The use of the effectiveness factor allows for omission of equations (5.1) and (5.2). Details of this approximation can be found in the original references [Shareefdeen and Baltzis (1994) and Shareefdeen (1994)], where it is also shown that the problem reduces to the following set of equations

$$\frac{\partial \bar{c}_j}{\partial \zeta} = -\frac{1}{\upsilon}\frac{\partial \bar{c}_j}{\partial z} - \beta_1 g(\bar{c}_j, \bar{c}_0) - \beta_3(\bar{c}_j - \bar{c}_j^*) \qquad (5.10)$$

$$\frac{\partial \bar{c}_0}{\partial \zeta} = -\frac{1}{\upsilon}\frac{\partial \bar{c}_0}{\partial z} - \beta_2 g(\bar{c}_j, \bar{c}_0) \qquad (5.11)$$

$$\frac{\partial \bar{c}_{jP}}{\partial \zeta} = \beta_3(\bar{c}_j - \bar{c}_j^*) \qquad (5.12)$$

where

$$g(\bar{c}_j, \bar{c}_0) = \frac{\varepsilon_1 \bar{c}_j}{(1 + \varepsilon_1 \bar{c}_j + \varepsilon_1^2 \gamma \bar{c}_j^2)}\frac{\varepsilon_2 \bar{c}_0}{(1 + \varepsilon_2 \bar{c}_0)} \qquad (5.13)$$

and

$$\bar{c}_j^* = \psi(\bar{c}_{jP})^{\frac{1}{n}} \qquad (5.14)$$

The initial and boundary conditions for equations (5.10) - (5.14) are,

$$\overline{c}_j = 1, \qquad \overline{c}_O = 1, \qquad \overline{c}_{jP} = \overline{c}_{jP,0}(0) \qquad \text{at } \zeta = 0 \text{ and } z = 0 \qquad (5.15)$$

$$\overline{c}_j = \overline{c}_{j,0}(z), \qquad \overline{c}_O = \overline{c}_{O,0}(z), \qquad \overline{c}_{jP} = \overline{c}_{jP,0}(z) \quad \text{at } \zeta = 0 \text{ and } 0 < z \leq 1 \qquad (5.16)$$

$$\overline{c}_j = 1, \qquad \qquad \overline{c}_O = 1 \qquad \qquad \text{at } \zeta \geq 0 \text{ and } z = 0 \qquad (5.17)$$

Equations (5.10) - (5.12) are in dimensionless form, and the dimensionless quantities appearing in them are related to the dimensional quantities of the original equations through the following,

$$\overline{c}_j = \frac{c_j}{c_{ji}}, \qquad \overline{c}_j^* = \frac{c_j^*}{c_{ji}}, \qquad \overline{c}_O = \frac{c_O}{c_{Oi}}, \qquad \overline{c}_{jP} = \frac{(1-\upsilon)\rho_P c_{jP}}{\upsilon c_{ji}},$$

$$z = \frac{h}{H}, \qquad \zeta = \frac{t}{\tau}, \qquad \varepsilon_1 = \frac{c_{ji}}{m_j K_j}, \qquad \varepsilon_2 = \frac{c_{Oi}}{m_O K_O}, \qquad \gamma = \frac{K_j}{K_{Ij}},$$

$$\beta_1 = \frac{e_j \alpha \delta A_s^* X_v \mu^* \tau}{Y_j c_{ji} \upsilon}, \qquad \beta_2 = \frac{e_O \alpha \delta A_s^* X_v \mu^* \tau}{Y_O c_{Oi} \upsilon},$$

$$\beta_3 = \frac{k_a (1-\alpha) A_s^* \tau}{\upsilon}, \qquad \psi = \frac{1}{c_{ji}} \left[\frac{\upsilon c_{ji}}{(1-\upsilon)\rho_P k_d} \right]^{\frac{1}{n}}$$

Equations (5.10) - (5.12) cannot be solved unless expressions for δ and e are available. For the case of toluene, and for relatively low toluene concentrations in an airstream fed to a biofilter such expressions were derived by Shareefdeen (1994), and they have been used in the present study. Solution of equations (5.10) - (5.12) was based on a computer code which employs the method of finite differences in the z-direction and integration of the resulting set of ordinary differential equations via the ODESSA algorithm. The basic code was that of Shareefdeen (1994), modified by Tsangaris and updated for the purposes of this study. The code is given in Appendix D of this thesis.

5.2 Results and Discussion

An example of concentration profiles at various locations in the biofilter bed is given in Figure 5.1. From this figure it is interesting to observe that the exit concentration (which is the most important concentration from the practical point of view) overshoots (albeit slightly) the eventual steady-state value. For this reason, it was decided to perform sensitivity studies with regard to the degree of overshoot and the time at which it occurs. Another parameter which was investigated was the time needed to achieve the steady-state level of operation. Mathematically, it takes an infinite amount of time to reach steady-state. Here, the time for reaching 50 and 90% of the steady-state level was investigated. The steady-state exit concentrations (from which the 50 and 90% values were calculated) were obtained by solving the transient model, and allowing the model to run until no further changes occurred in the exit concentration (usually several days). This was done only once for each set of studies as transient parameters do not affect steady state values. Although the steady-state model from the previous chapter could had been utilized, it was found that there were slight differences between the steady-state concentrations determined by these two models. This is due to the approximation required for the determination of the effectiveness factor.

As can be seen from the basic model equations (5.1) - (5.7) the model contains a number of various parameters. All parameters discussed in Chapter 4 are also present in the transient model. Although some of these parameters may be affecting transient behavior (e.g., overshoot of exit concentration), they were not examined during the course of the present work. They were kept constant at the values (measured or
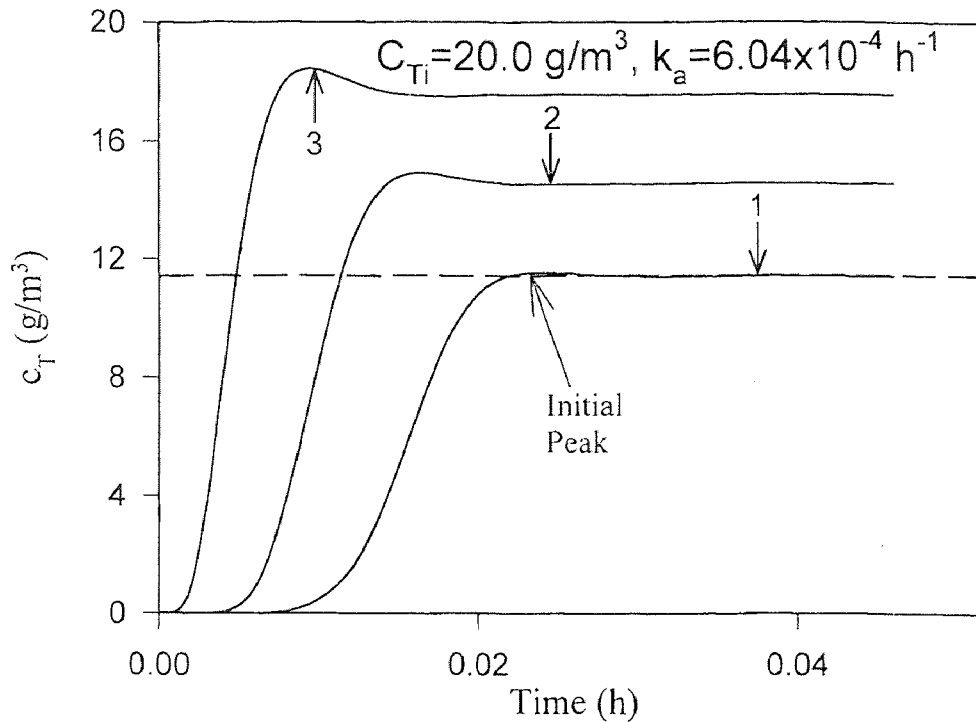
Figure 5-1 Transient concentration profile in a biofilter bed at the exit, 2/3 height, and 1/3 height (curves 1, 2, and 3, respectively). The dashed line represents the steady-state exit concentrations

estimated) by Shareefdeen (1994) and reported in Table 5.1. The parameters which were examined here were those which appear in the transient, but not in the steady-state model. These are, the void fraction of the bed (υ), the density of the packing material (ρ_p), the fraction of the surface area covered by biofilm (α), the mass transfer coefficient (k_a), and the Freundlich constant (k_d). Base values for these parameters are also given in Table 5.1. Parameters which affect the transient behavior but were not studied here are, the second Freundlich constant (n) and the initial distributions of VOC concentration on the solids and in the air [$c_{jp}(h)$ and $c_j(h)$ at t = 0]. The latter appear in the boundary conditions of the model equations and for the purposes of the studies reported here it was always assumed that $c_{jp}(h) = c_j(h) = 0$ at t = 0. The implication is that the results are valid

Table 5.1 Base-values of model parameters for the transient biofiltration studies[*]

Parameter	Value	Units
A_s^*	133.3	m^{-1}
c_{Oi}	275×10^{-3}	kg/m^3
D_{OW}	2.41×10^{-9}	m^2/s
D_{TW}	1.03×10^{-9}	m^2/s
$f(X_v)$	0.195	—
k_a	6.04×10^{-3}	h^{-1}
k_d	2.25×10^{-5}	kg/m^3
K_{IT}	78.94×10^{-3}	kg/m^3
K_O	0.26×10^{-3}	kg/m^3
K_T	11.03×10^{-3}	kg/m^3
m_O	34.4	—
m_T	0.27	—
n	1.04	—
τ	3.09	min
X_v	100	kg/m^3
Y_{OT}	0.341	kg/kg
Y_T	0.708	kg/kg
α	0.3	—
δ	$1.5c_j + 33.4$	μm (c_j in g/m^3)
e_T or e_O	$0.03c_j + 0.2$	(c_j in g/m^3)
υ	0.3	—
ρ_p	4.28×10^5	g/m^3
μ_T^*	1.50	h^{-1}

[*] From Shareefdeen (1994)

under process start-up conditions, i.e., when a biofilter unit is first put into operation with new packing material.

As in the case of the studies reported in Chapter 4, only one parameter was changed at a time. Changes were made relative to the reference (or base) values given in Table 5.1. A relative value of 0.2 for a parameter, means (as in Chapter 4) that the actual value of that parameter used in the calculations is equal to 0.2 times the base value reported in Table 5.1.

Results from the studies with the transient biofiltration model are given in a graphical form in the figures of Appendix B of the thesis.

Figures B-1 through B-5 show the time needed for the exit VOC concentration to reach 50% and 90% of its steady-state value as a function of various parameters. Calculations were performed at two different VOC (toluene) concentrations in the air stream fed to the biofilter; namely, 0.5 and 2.0 g/m^3.

Figure B-1 shows the effect of the void fraction of the biofilter bed (υ). Since this is a parameter that can be easily measured and varies from 0 to 1, actual rather than relative values of υ were used and are shown in the x-axis of the two graphs. As can be seen from the graphs, it appears that there is a perfectly linear relationship between time and υ. It also appears that for the two inlet concentrations tried, the time for reaching a given percentage of the steady-state level is essentially the same for a given value of υ. Values of υ very close to zero are meaningless because they imply that there is no room in the bed for the airstream to pass through. Values of υ very close to unity are also meaningless since they imply that there is no packing -and thus, no process- in the (empty) structure. In this case, the time needed to reach steady-state is equal to the space time, if the air passes in plug flow through the vessel (structure). For realistic values of υ, as υ increases the adsorption process is faster (since there are less solids) but the reaction is slower in the sense that the material passes through the bed without enough time to react. The overall effect is that it takes longer to reach steady-state as υ increases.

As Figure B-2 indicates, the value of the density of the packing (ρ_p) has no effect on the time needed to reach steady-state. Similar are the results with the Freundlich isotherm constant (k_d) as indicated by the graphs of Figure B-3.

The effect of parameter α (fraction of surface area covered with biofilm) on the time required to reach steady-state is shown in Figure B-4. As the graphs indicate, the effect of α is minimal. However, the following should be mentioned. The available correlations for the effectiveness factor and δ were for a value of $A_S = 40m^{-1}$. Since $A_S = \alpha A_S^*$, for the correlations to be valid the values of α and A_S^* were simultaneously changed so that their product remained constant. Thus, as α increases A_S^* decreases.

This also implies that the area of solids available for adsorption, i.e., $(1 - \alpha)A_S^*$, also decreases. One would then expect that -if adsorption was the only process- it would take longer to reach steady-state as there is less area through which a constant amount of VOC needs to be adsorbed onto the solids. However, there is also reaction taking place. A slow-down of the adsorption leads (during transients) to higher VOC concentrations in the gas phase and this makes the reaction faster. Overall the effect is minimal, showing a slight decrease in the time needed to reach steady-state. If the explanation above is correct, one should observe an increase in the time needed to reach steady-state for higher inlet toluene concentrations. High concentration brings the kinetics into the inhibitory regime; hence, the reaction is slowed down. Since a decrease in A_S^* also slows down the adsorption process, one should observe an overall increase in time. This hypothesis was not tested during the course of this work.

Figure B-5 shows the effect of the mass transfer coefficient on the time needed to reach steady-state. Here the effect can be very substantial as the logarithmic scale of the y-axis indicates. It should be mentioned here that the curve showing the time for reaching 50% of the steady-state value at the exit exhibits the same behavior as the one indicating time for reaching 90% of the steady-state value (curve 2 in the graphs). However, for the case of curve 1 the substantial increase in time occurs at k_a values which fall outside of the range of the graphs. The features of curve 2 could be potentially explained by the following arguments. At high values of k_a the pollutant is transferred fast to the solid packing and this leads to temporarily low VOC concentrations in the gas phase. This implies a lower VOC concentration in the biofilm and thus, a reduction in the reaction rate. The process is controlled by kinetics and thus the value of k_a does not affect the time of response (upper plateau of curve 2). At low k_a values, the adsorption process is slow, but it leads to very fast kinetics due to the higher concentrations in the air and the biofilm. Hence, the process is again under kinetic limitation and this explains the lower plateau of curve 2. In the intermediate regime the process is under both mass transfer and kinetic control. It should be mentioned here that there is a lot of uncertainty in the values of k_a since there are not experimental data for them. Further studies are needed here.

As was mentioned earlier, in conjunction with Figure 5.1, there are cases in which the exit biofilter concentration overshoots the steady-state exit concentration. Computer simulations have shown that almost invariably, the exit concentration exhibits a peak. However, with the reference values this peak does not exceed the steady-state value. In

fact, a maximum (peak) is observed and then the concentration falls and subsequently rises and stabilizes at a level higher than the peak. In such cases, the existence of the concentration peak is unimportant from the practical point of view. This was the case for the studies performed with low inlet concentrations (2.0 g/m^3) and shown in the graphs of Figures B-6 through B-8. The only parameters found to affect the concentration at the peak are α and k_a (Figures B-7a and B-8). Observe that for very high α-values or very low values of k_a the peak concentration is a real overshoot; that is, it exceeds the steady-state value. This becomes an important factor (at realistic α and k_a values) when the inlet VOC concentration increases. The time at which the peak concentration occurs is only affected by the void fraction of the bed, as Figure B-9 indicates. A comparison of Figures B-1 and B-9 shows that as υ increases the extent of the transient period of the process increases from every point of view.

A small effort was made in examining whether the inlet concentration value of the VOC affects the extent of the transients of the process. An example is shown in Figure B-10. The effect of inlet concentration appears to be minimal. However these results should be considered with extreme caution as they were obtained based on correlations for δ and e (Table 5.1) which were originally derived for a range of inlet concentrations much narrower than the one used in Figure B-10.

The conclusion which can be drawn from the results of the studies presented in this chapter is that the transients of the biofiltration process are primarily affected by the mass transfer coefficient and the void fraction (porosity) of the biofilter bed.

CHAPTER 6

CONCEPTUAL DESIGN FOR REMEDIATION OF A CONTAMINATED AQUIFER THROUGH THE USE OF BIOFILTRATION

In this chapter results from some preliminary work performed on the conceptual design of an integrated soil venting/biofiltration process are presented. This integrated process is proposed as an alternative method for treating a contaminated aquifer. A schematic of this integrated process is shown in Figure 6.1.
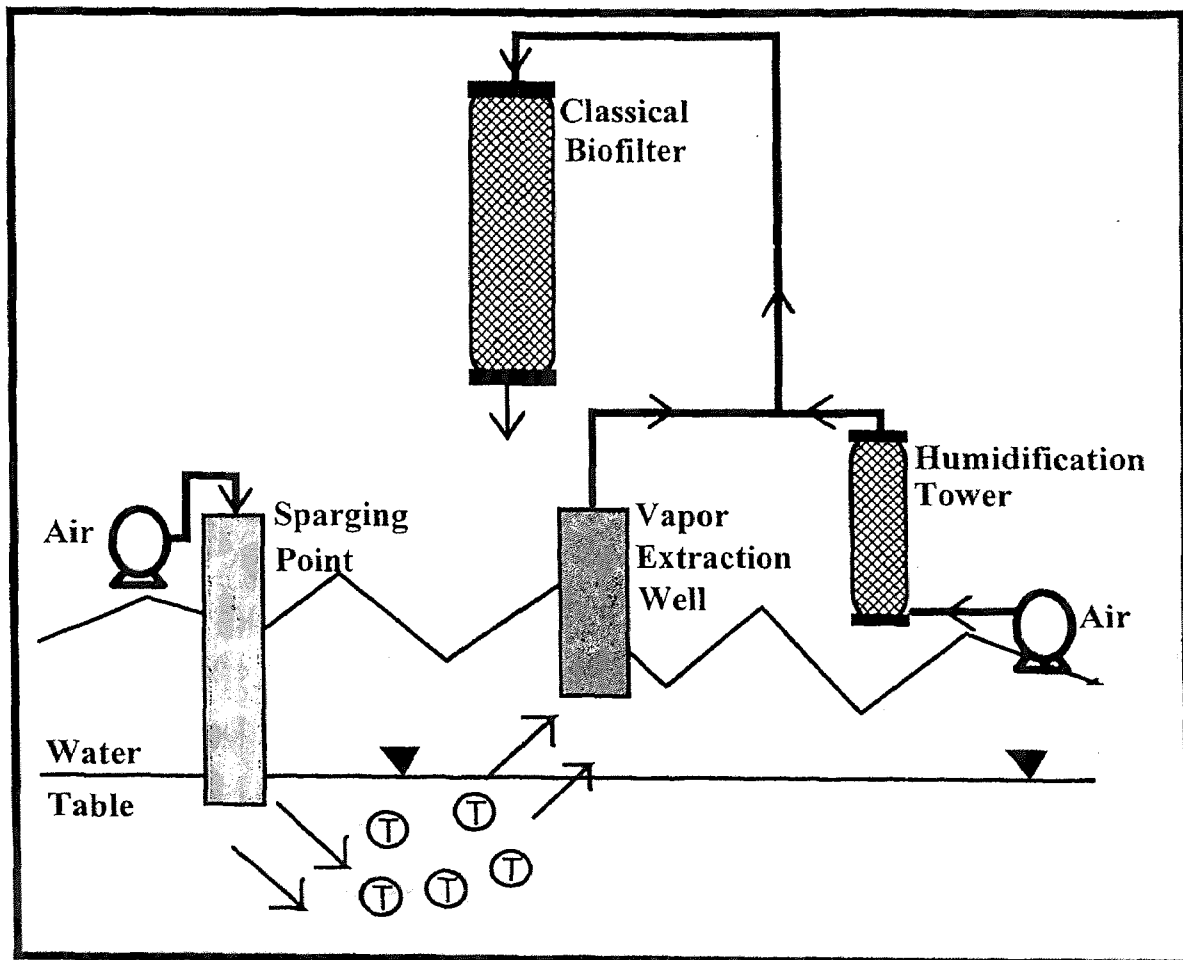
Figure 6-1 Schematic of the integrated soil venting/biofiltration process.

The ideas incorporated in the schematic of Figure 6.1 are the following. Air is passed through the aquifer via multiple sparging wells. The air forces the volatile contaminants into the gas phase (air) which is then collected through vapor extraction wells. If the air exiting the soil (aquifer) contains VOCs at high concentrations which are not appropriate to be supplied to the biofilter, it is diluted with clean air. The clean air is first humidified in order to ensure that the biofilter bed will not get dry. After mixing of the two airstreams, the combined stream is passed through a biofilter. The inlet to the biofilter is at the top of the biofilter bed so that if humidification of the airstream is not 100% and thus, there is the potential of drying part of the biofilter bed, this part is at the top and can be easily brought to the correct conditions of water content by supplying water at this location. The airstream exiting the biofilter bed meets the appropriate environmental standards. Looking at the process as a whole, the inlet consists of clean air (to the sparging wells and humidification tower) and the exit (from the biofilter) also consists of clean air.

6.1 Basic Assumptions

The work presented in this chapter is based on the following assumptions.

1. Contamination is uniform throughout the liquid (water) contained in the aquifer. The volume of the water is constant.

2. There are no contaminants adsorbed on the soil.

3. Air sparging is uniform throughout the aquifer.

4. All air supplied to the soil at sparging points is recovered at the extraction wells.

5. There is a single contaminant (toluene was assumed) in the aquifer.

6. The contaminant in the liquid phase (aquifer) and the gas phase (air) is in equilibrium at all times. The equilibrium distribution is dictated by Henry's law.

Based on the foregoing assumptions, one can write the following equations,

$$V_L \frac{dc_L}{dt} = -Q_G c_G \qquad (6.1)$$

$$c_G = m_j c_L \qquad (6.2)$$

Combining equations (6.1) and (6.2) and integrating the resulting equation leads to the following expression,

$$c_L = c_{L_o} e^{-\frac{Q_G m_j t}{V_L}} \qquad (6.3)$$

Combining expression (6.3) with (6.2) leads to the following expression,

$$c_G = m c_{L_o} e^{-\frac{Q_G m_j t}{V_L}} \qquad (6.4)$$

6.2 Objectives of the Design

The objectives set for the design of the integrated process were the following.

1. The concentration of toluene in the aquifer at the end of the remediation operation should be at or below the toluene Action Level in Groundwater as per existing regulations (see Table 6.1).

2. The concentration of toluene in the air exiting the extraction wells should be very close to the Threshold Limit Value (TLV) as per existing regulations (see Table 6.1).

Table 6.1 Regulations for control of toluene levels

Parameter	Value	Units
Threshold Limit Value (TLV) [a]	86.69	g/m^3
Acceptable Source Impact Level (ASIL) [b]	0.2817	g/m^3
Action Level in Groundwater [c]	1.0	g/m^3

[a] TLV established in the Federal Register (1993a).
[b] ASIL established by the Washington State Department of Ecology (1994).
[c] Action Level established in the Federal Register (1993b).

The TLV is the maximum concentration of a pollutant to which human exposure is allowed for short time periods. It usually relates to people in the immediate vicinity of the source (e.g. workers). For example, if there was a leak from or a rupture of the pipe carrying the air from the extraction wells people in the immediate vicinity of the accident would be exposed to the toluene concentration in the air exiting the aquifer.

3. The concentration of toluene at the exit of the biofilter should meet the Acceptable Source Impact Level (ASIL) as per existing regulations (see Table 6.1).

4. The biofilter should be exposed to relatively constant toluene concentration over the majority of the remediation operation. The maximum toluene concentration in air supplied to the biofilter should be at values for which there is experimental evidence that the process works (i.e., this concentration should not be too high). Finally the required volume of the biofilter bed should have a reasonable value.

6.3 Methodology for Calculations

The methodology followed in the calculations was as follows.

The values of various parameters for toluene (given in Chapter 4 as base-values) were used in calculations with the steady-state biofiltration model. In these calculations, the value of the toluene concentration in the air entering the biofilter was set at different levels and the required space (residence) time was calculated so that the exit toluene concentration met ASIL requirements. It was finally decided to use an inlet concentration of 9.2 g/m^3 as biofiltration experiments had been performed at values close to that and showed that the process works [Shareefdeen (1994)]. Clearly, this represents the "worst case scenario" since this is the maximum value of toluene concentration that the biofilter would be faced with.

Having the space time from the calculations above (a value of $\tau = 22$ min was determined) and since $\tau = V_p/F$, a value of $F = 0.85$ m^3/min (30 cfm) was selected so that the volume of the biofilter bed comes to a very reasonable value of 18.70 m^3.

The volumetric flowrate of the air supplied to the biofilter was kept constant at all times, and the next objective was to determine what should be the flowrate of the air sparged through the aquifer, so that the concentration profile of the toluene concentration at the inlet of the biofilter was maintained relatively constant over substantial time periods. In order to meet objective 4 (see preceding section) a trial and error approach, using expressions (6.3) and (6.4) and accounting for appropriate dilution with clean air was used. This led to the concentration profile shown in Figure 6.2. Based on this profile and taking into consideration the value of the action level for toluene in groundwater, it was determined that the remediation would take 51 days. It should be mentioned here that the minimum value for Q_G in formula (6.4) was 3 cfm, implying that
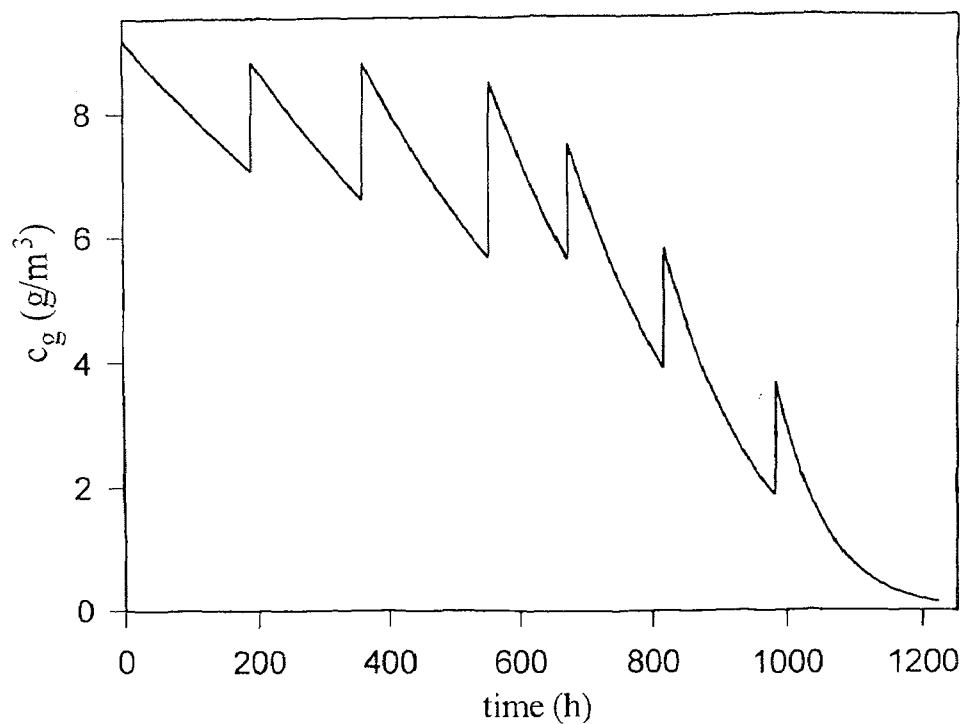
Figure 6.2 Toluene concentration profile in the air supplied to the biofilter.
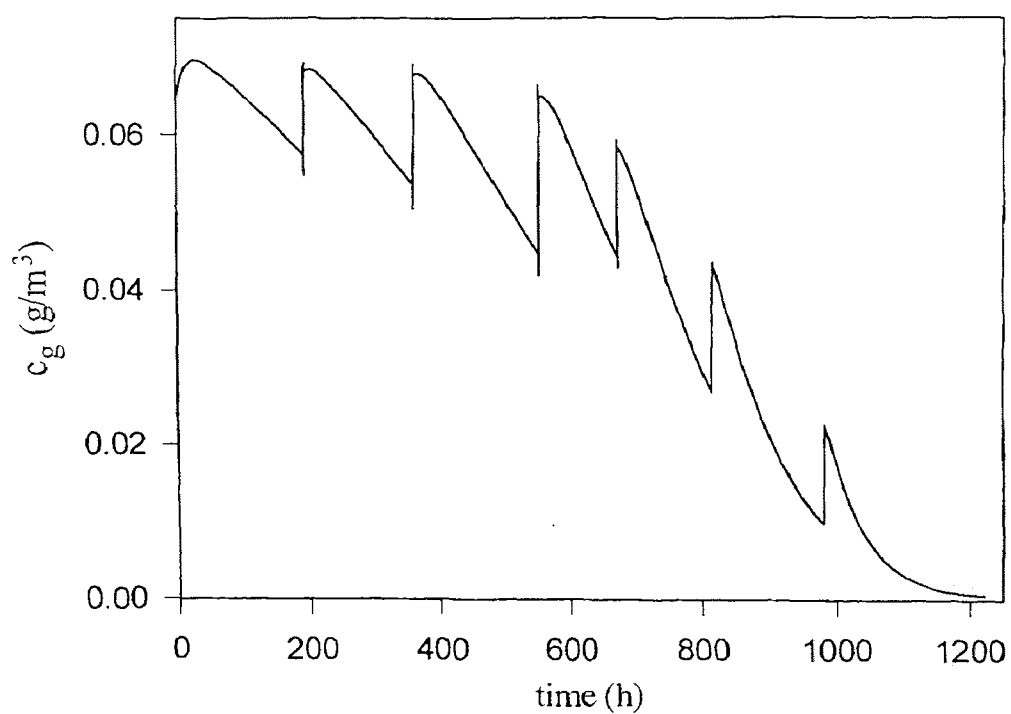
Figure 6.3 Toluene concentration profile in the air exiting the biofilter.

during the first period of time shown in Figure 6.2 the air from the extraction wells is diluted with clean air at a 1:10 ratio. The value of Q_G, and that of dilution, were varied in each segment of the profile of Figure 6.2.

Once the toluene concentration profile in the air supplied to the biofilter was determined, the transient biofiltration model (discussed in Chapter 5) was used in order to determine the profile of the toluene concentration at the exit of the biofilter. This required some modifications in the computer code given in Appendix D of the thesis. This code solves the transient biofiltration problem for a single value of the inlet pollutant concentration. However here the inlet toluene concentration varies continuously as per the profile of Figure 6.2. The modifications of the code are given in Appendix E of the thesis. The results are shown in the concentration profile of Figure 6.3. The reason for performing these calculations was the following. Although the space time was originally calculated for the maximum value of inlet toluene concentration in ways that ensure exit concentration below ASIL values, this was done under steady-state conditions. However, the biofilter here works always under transient conditions. The variations in the inlet concentration imply that adsorption and desorption phenomena are also occurring during the periods of inlet concentration increase and decrease, respectively. It was thus necessary to ensure that desorption phenomena during the operation will never lead to exit concentration values exceeding ASIL. This is in fact the case, as can be seen from Table 6.2 where all parameter values (other than those mentioned for toluene in Chapters 4 and 5) and results are listed. It is worth noticing that the maximum toluene concentration at the exit of the biofilter is in fact higher than the corresponding value

Table 6.2 Parameters for the design of the integrated soil venting/biofiltration process.

Parameter	Value	Units
Volume of Aquifer (V_L)	1000	m^3
Initial Toluene Concentration in Aquifer ($c_{L,0}$)	340	g/m^3
Volume of Biofilter (V_P)	18.70	m^3
Flowrate of Air Supplied to Biofilter (F)	0.85	m^3/min
Residence Time in Biofilter (τ)	22	min
Maximum Toluene Concentration at Biofilter Exit (actual)	0.0700	g/m^3
Maximum Toluene Concentration at Biofilter Exit (steady-state model prediction)	0.0585	g/m^3
Final Toluene Concentration in Aquifer	0.4967	g/m^3
Time for Remediation of Aquifer	51	days

determined from the steady-state biofiltration model. Thus, desorption effects need to be considered in the calculations.

6.4 Discussion of Results and Other Calculations

As the numbers in Table 6.2 indicate, with the design calculations discussed in the previous section the final toluene concentration in the aquifer is 50% of the value of the action level in groundwater while the exit concentrations from the biofilter are always substantially lower than the ASIL value. These levels were chosen as a "safety factor"

and do not add more than 4-5 days to the time needed for remediation based on values conforming exactly to the regulations. A possibly substantial reduction in remediation time could be obtained if the flowrate of the air supplied to the biofilter is increased at about the 25th day. As can be seen from Figure 6.3, after the 25th day the exit concentrations drop substantially. A decrease in space time (i.e., an increase in air flowrate) would keep the toluene concentration exiting the biofilter relatively constant. This should be investigated in the future.

Keeping most of the parameters the same as in the case of calculations discussed in the preceding section (i.e., V_p, τ, F) the values of the volume of the aquifer (V_L) and the initial concentration of toluene in the aquifer (c_{L0}) were varied and calculations were performed regarding the time needed for remediation. Essentially, these studies were based on inlet concentration profile determination (as in the case of Figure 6.2) and the intent was to get a profile with a maximum value of 9 g/m^3 (for toluene at the inlet of the biofilter) which remains practically constant over most of the remediation time.

For a constant value of c_{L0} equal to 100 g/m^3, Figure 6.4 shows the days required for remediation as a function of the volume of the aquifer. Similarly, for an aquifer of a constant volume of 1,000 m^3 the initial (maximum) toluene concentration in the aquifer was varied and the time required for remediation was determined; these results are shown in Figure 6.5. For both cases, six calculations were performed and the results were found to fall on the straight lines shown in Figures 6.5 and 6.6.

In another set of calculations both V_L and c_{L0} were varied so that their product remained constant. As the results shown in Figure 6.6 indicate the time drops as the
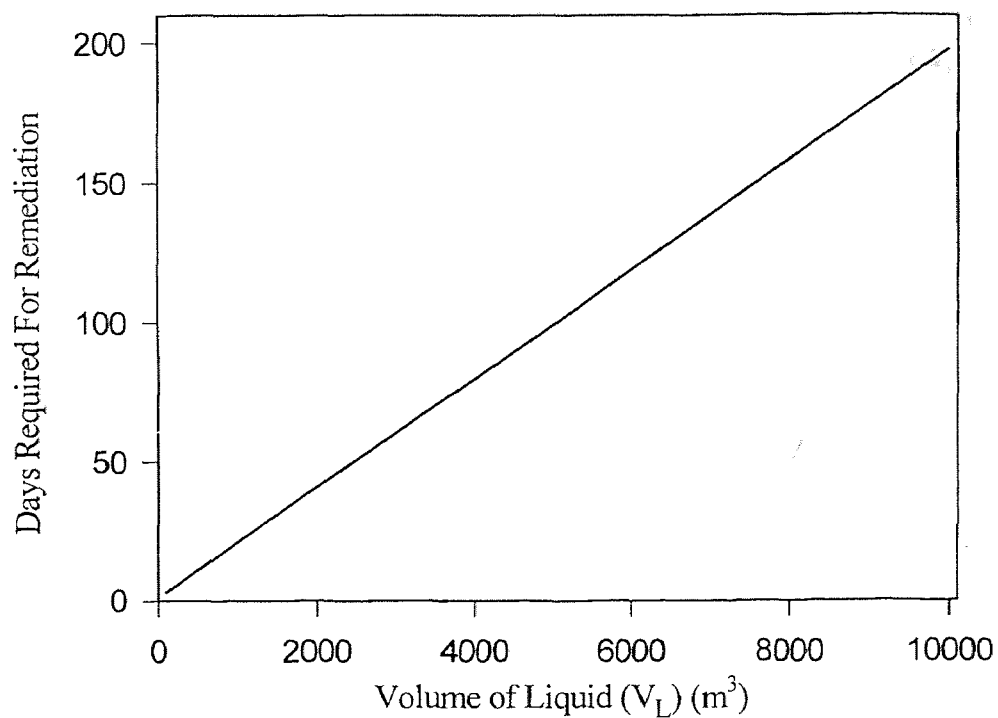
Figure 6.4 Remediation time as a function of the volume of the aquifer when $c_{L0} = 100$ g/m^3.
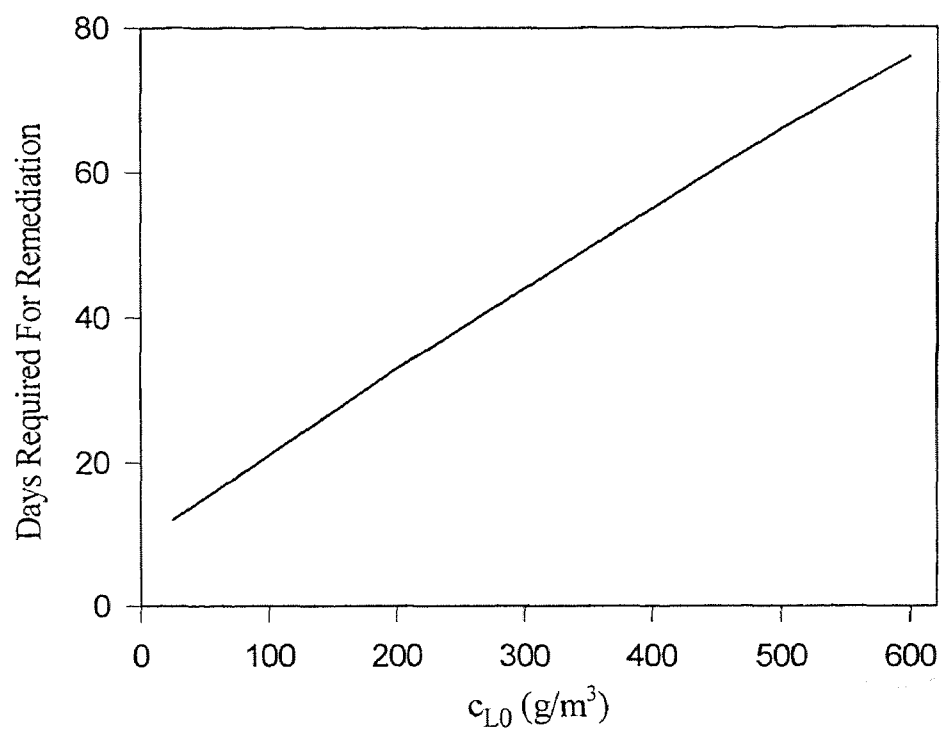
Figure 6.5 Remediation time as a function of the maximum toluene concentration in an aquifer of volume 1,000 m^3.
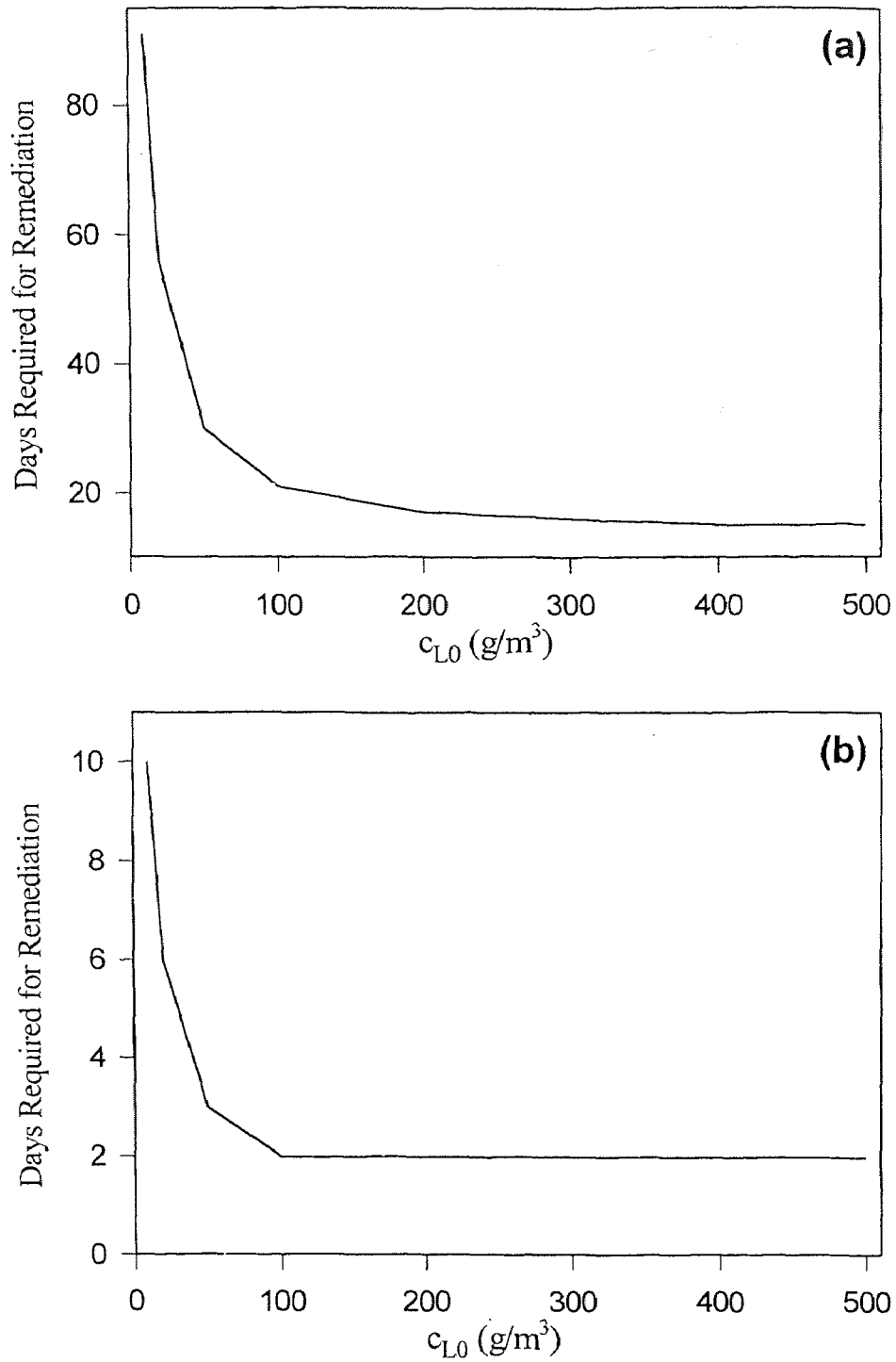
Figure 6.6 Remediation time as a function of c_{L0} for constant original amounts $(V_L c_{L0})$ of toluene in the aquifer. $V_L c_{L0} = 100$ kg in (a) and 10 kg in (b).

value of $c_{1,0}$ increases (which implies that the value of V_L decreases). It appears that some type of a minimum is reached. The results suggest that in a case where there is a spill, the faster it is decided to remediate it (before it spreads; spreading implies larger volume and lower concentration) the shorter is the time for that remediation. On the other hand, if the volume of the aquifer decreases due to seasonal variations in the water level and if the entire amount of the pollutant stays in the water, the results of Figure 6.6 suggest that it is better to select as the date of remediation that season which results in the lowest volume (V_L).

6.5 Conclusions

The results presented here are only from a preliminary effort to perform design calculations for an integrated soil venting/biofiltration process. The only conclusion that can be drawn is that these calculations (based on worst case scenaria in most instances) do indicate that remediation appears to be feasible in reasonable time-frames and with reasonable biofilter sizes.

Extensive studies need to be performed for this process in the future. The exit (from the biofilter) concentration profiles (e.g. Figure 6.3) need to be optimized so that concentration values do not fall far below ASIL values. This will require the use of a variable flowrate value (F). The assumption of equilibrium between aquifer and air is probably incorrect. More realistic expressions (discussed in Chapter 2 of this thesis) need to be used. It appears that future efforts need to concentrate on the air sparging portion of the process while this thesis primarily dealt with biofiltration.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

The main conclusion from the sensitivity studies performed during the course of this thesis is that despite the fact that biofiltration models contain a large number of parameters, few of them need to be accurately known for safely predicting the size of a biofilter.

The parameters which mostly affect the size of biofilters are: two kinetic constants, the mass transfer coefficient, and the specific area of biofilm. Although few in number, the important parameters fall in three different categories: kinetics, flow characteristics (mass transfer), and type of packing/microbiology (the latter affect/ determine the biofilm surface area). Hence, there is no single aspect of the biofiltration process which is more important than others when it comes to design.

The fact that two kinetic parameters are important implies that efforts to model the process with one kinetic parameter (zero- or first-order kinetics) are bound to lead to inaccurate designs for biofilter units. A detailed kinetics study appears to be needed for each application. However, these studies are not too difficult or expensive to perform. The high importance of kinetic constants also implies that the selection of microorganisms for the biofilter unit is very important as kinetics depend not only on the identity of the VOC, but also on that of the bacteria (catalysts of the reaction).

Determination of the mass transfer coefficient is also important. This relates to the selection of the packing material and knowledge of the air flow characteristics through the bed.

The biofilm surface area has been found to be very important. As could be anticipated, the larger the area of the biofilm the smaller is the required biofilter volume. This implies that optimal biofilter design depends on the selection of packing material which allows for complete (if possible) coverage of the surface with bacterial films. Unfortunately, experimental determination of the biofilm surface area is not easy, if at all possible. New techniques need to be developed before this parameter can be accurately determined.

Another interesting finding from this study is that in some cases the size of the biofilter has a complex dependence on the inlet concentration. The existence of a minimum volume under constant load for the case of butanol suggests (as originally reported by Baltzis (1994) for the case of ethanol) that one could substantially reduce the required biofilter volume by mere dilution of the contaminated airstream with clean air.

Following are some recommendations for further sensitivity studies. With steady-state biofiltration models studies are needed for cases where the airstreams carry mixtures of pollutants. Regarding transient behavior, studies are needed in a number of areas. In the present study characteristic values for a hydrophobic compound were used. For the case of hydrophilic compounds other parameters (such as adsorption isotherm constants) may be important. In addition, during the present study variations in the inlet concentration of VOC were not studied regarding changes from one steady-state to

another. Consequently, desorption effects which are known to occur upon a decrease in the inlet VOC concentration were not considered and need to be studied in the future. Furthermore, in this study it was assumed that the value of the mass transfer coefficient is constant at all values of air flowrates. Recent studies [Wojdyla (1996)] have developed correlations between the mass transfer coefficient and the air flowrate, and they need to be used in future sensitivity studies with the transient model. Finally, modeling and sensitivity studies are needed for the transient biofiltration of VOC mixtures.

The results from the design calculations for the integrated soil venting/ biofiltration process can only be viewed as preliminary. The air was assumed to be uniformly passing through the aquifer and always carrying toluene (the VOC) at concentrations in equilibrium with those in the aquifer. The presence of the VOC on the soil per se was not considered. Future studies need to relax the foregoing assumptions and also consider factors such as the radius of influence for the air forced into the soil and minimum (and/or maximum) air flowrates which are realistically permissible. However, the results of the first attempt to study (from the design perspective) this integrated process have shown that the process seems feasible with very reasonable biofilter volumes, the time of remediation is relatively short, and the concentration at the inlet of the biofilter can be kept relatively constant over considerable time periods. Two interesting findings were the following; in cases where there is a spill, the faster it is decided to remediate it the shorter is the time for that remediation, and that a non-spreading aquifer (or plume) can be treated faster during periods in which -due to weather changes- its volume is at a minimum level. It is believed that the work performed here

paved the path for future detailed studies on the design of the integrated soil venting/biofiltration process.

APPENDIX A

BIOFILTRATION OF SINGLE VOCs UNDER STEADY-STATE CONDITIONS: RESULTS (IN GRAPHICAL FORM) OF SENSITIVITY AND BIOFILTER SIZING STUDIES
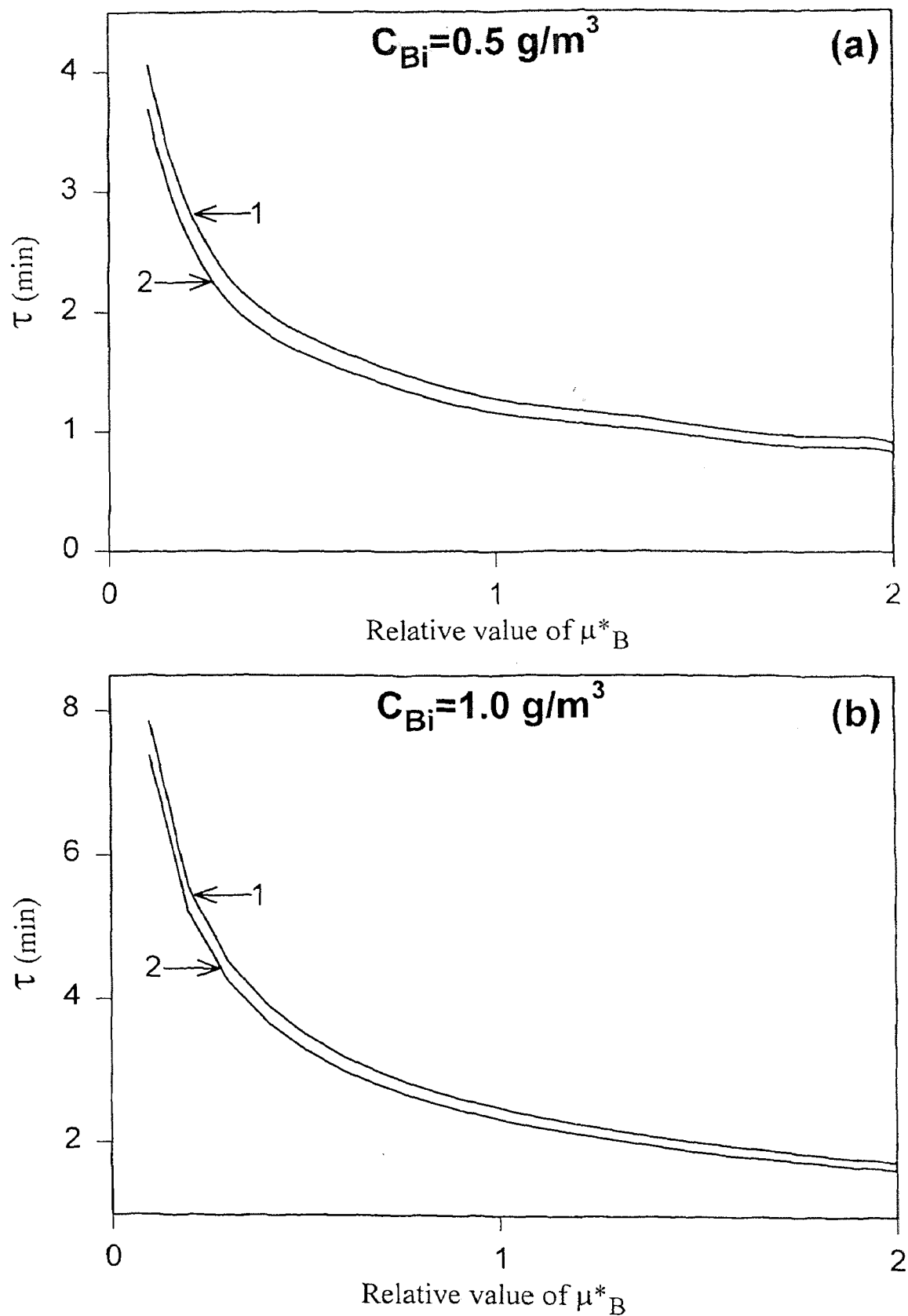
Figure A-1 Sensitivity of the model to the value of μ^*_j for the case of butanol. Curve 1: 99% conversion; Curve 2: 95% conversion
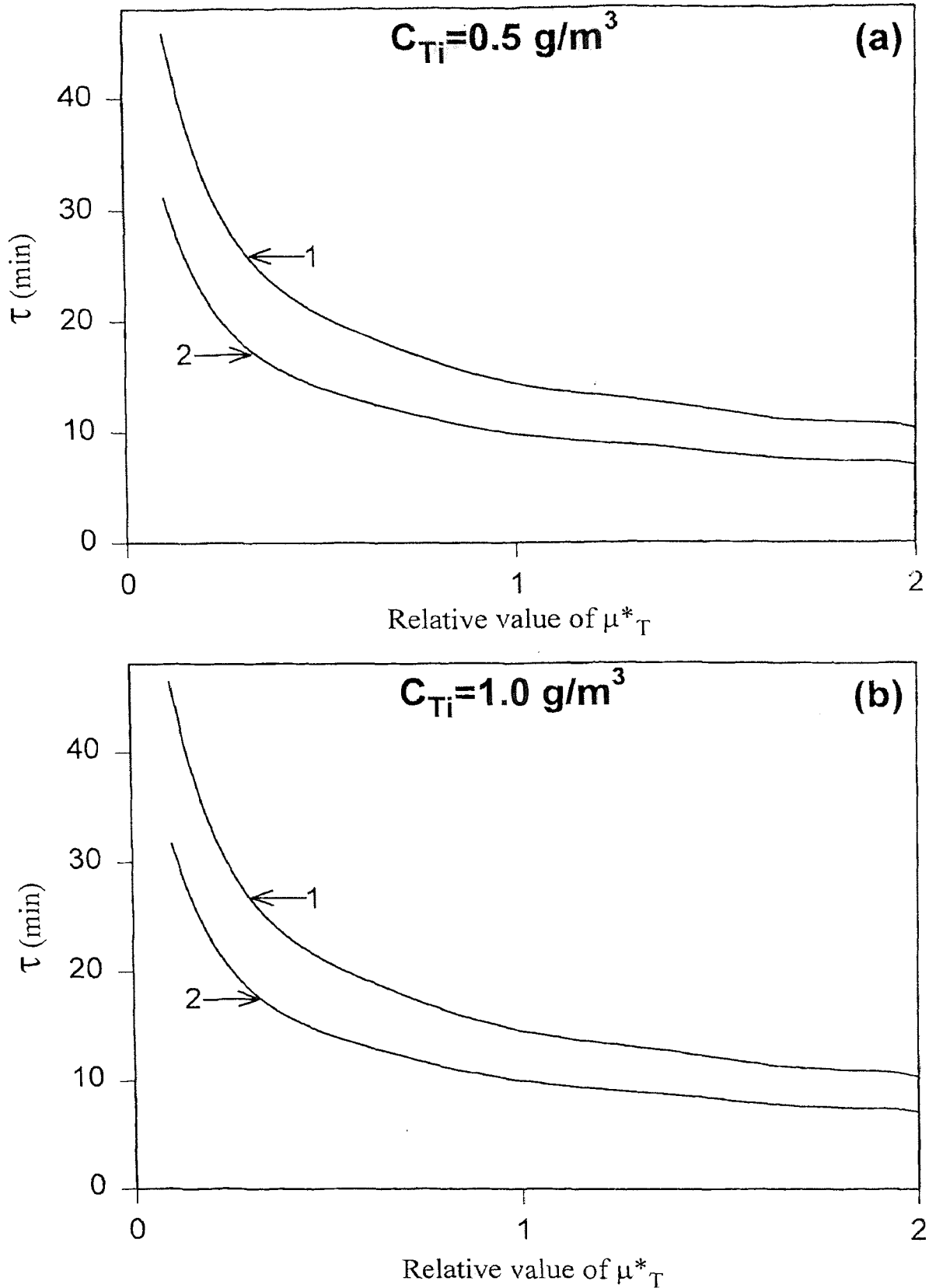
Figure A-2 Sensitivity of the model to the value of μ^*_j for the case of toluene. Curve 1: 99% conversion; Curve 2: 95% conversion
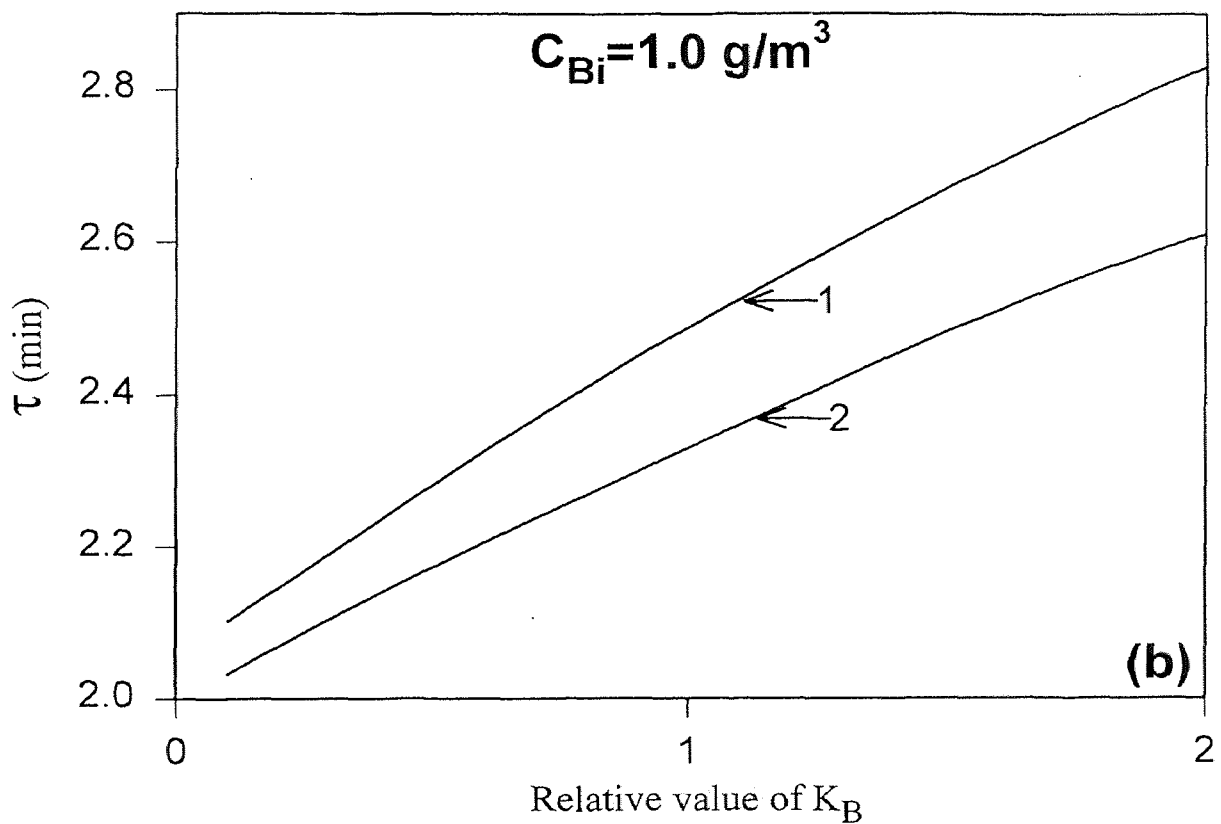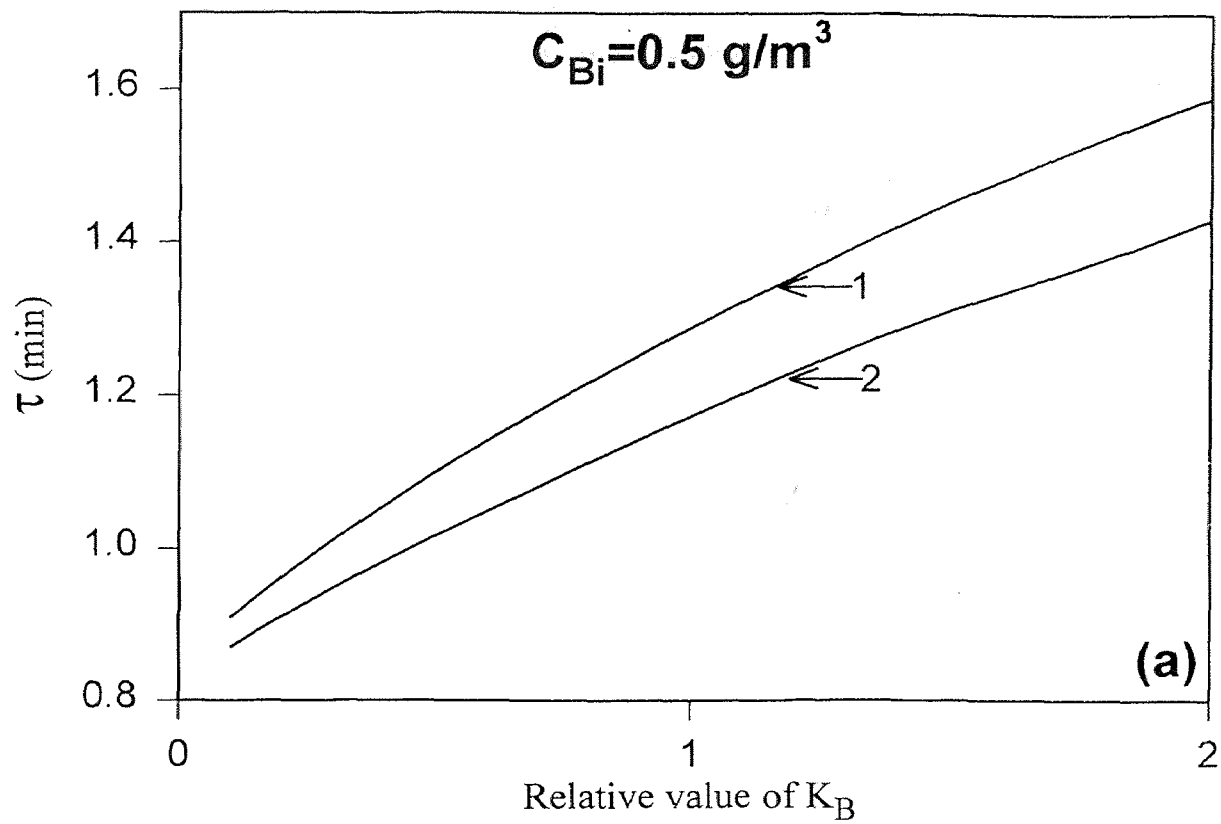
Figure A-3 Sensitivity of the model to the value of K_j for the case of butanol.
Curve 1: 99% conversion; Curve 2: 95% conversion
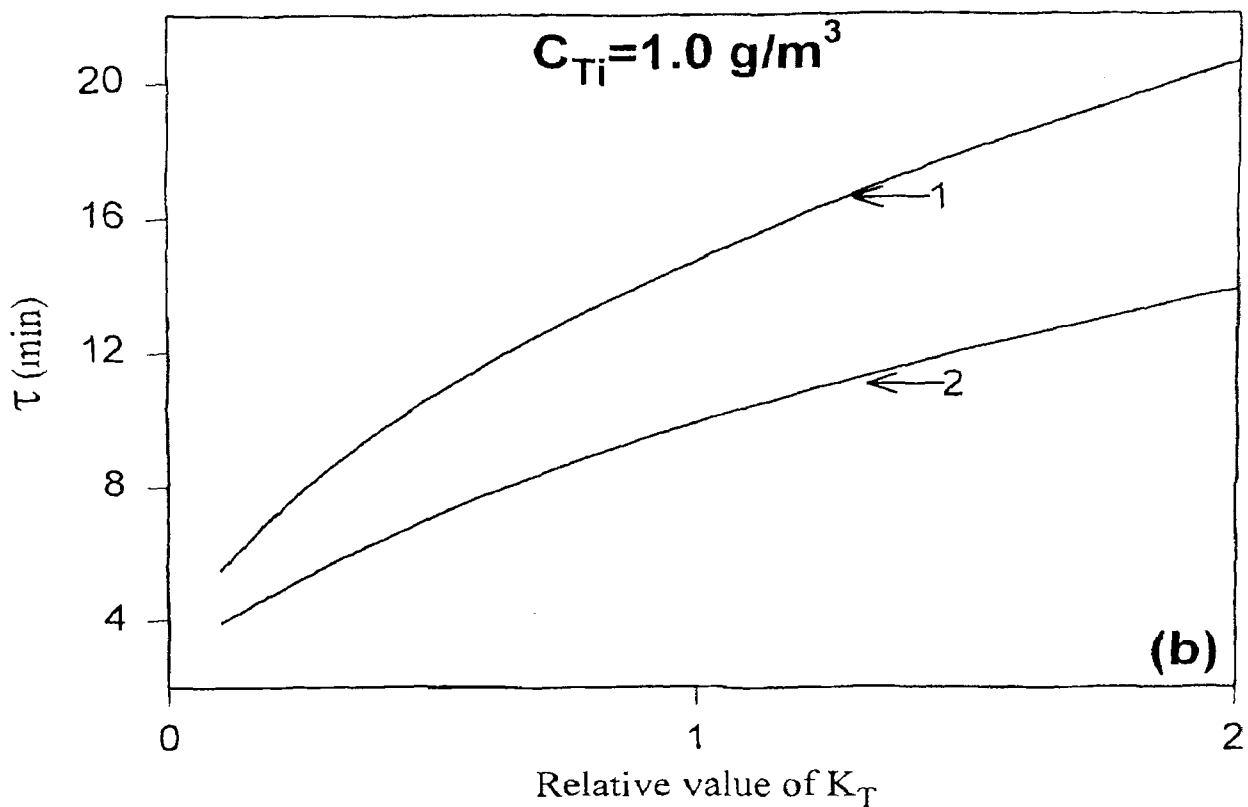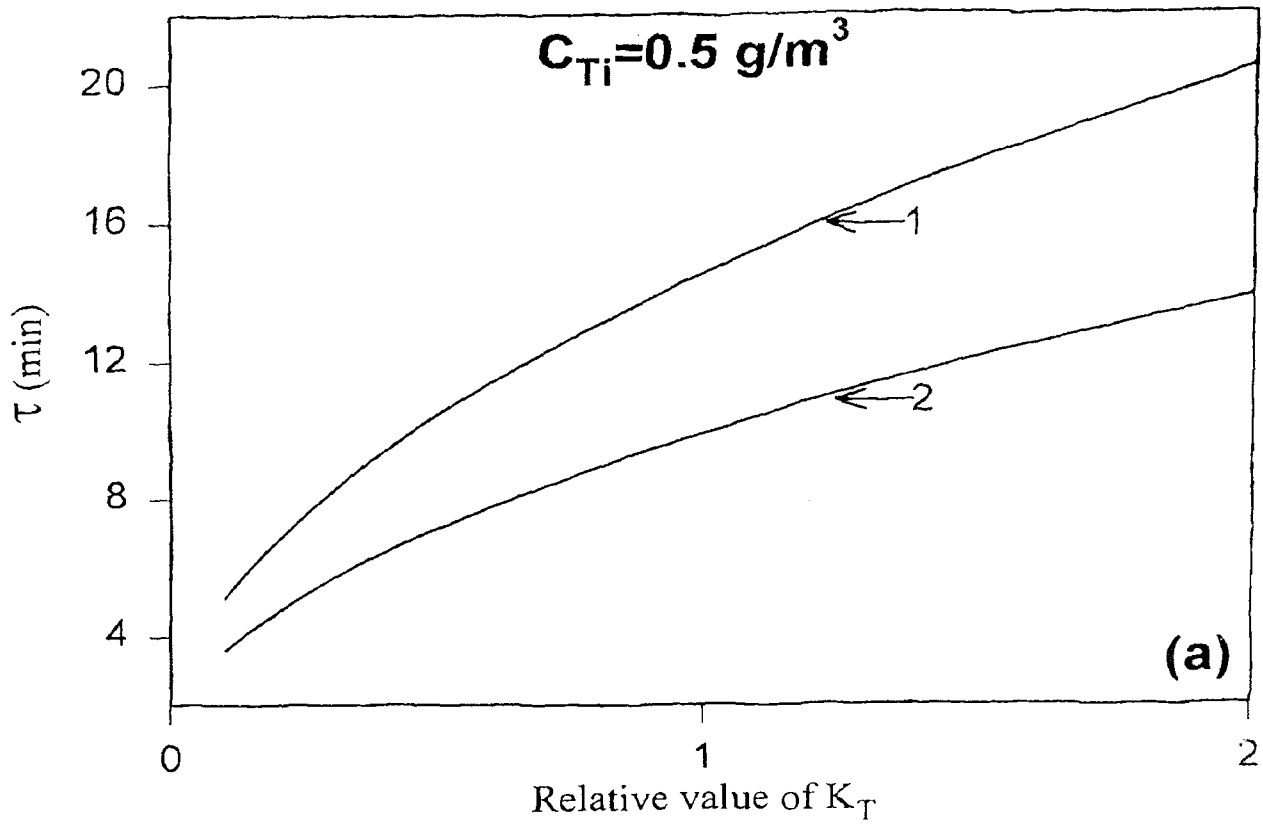
Figure A-4 Sensitivity of the model to the value of K_j for the case of toluene.
Curve 1: 99% conversion; Curve 2: 95% conversion
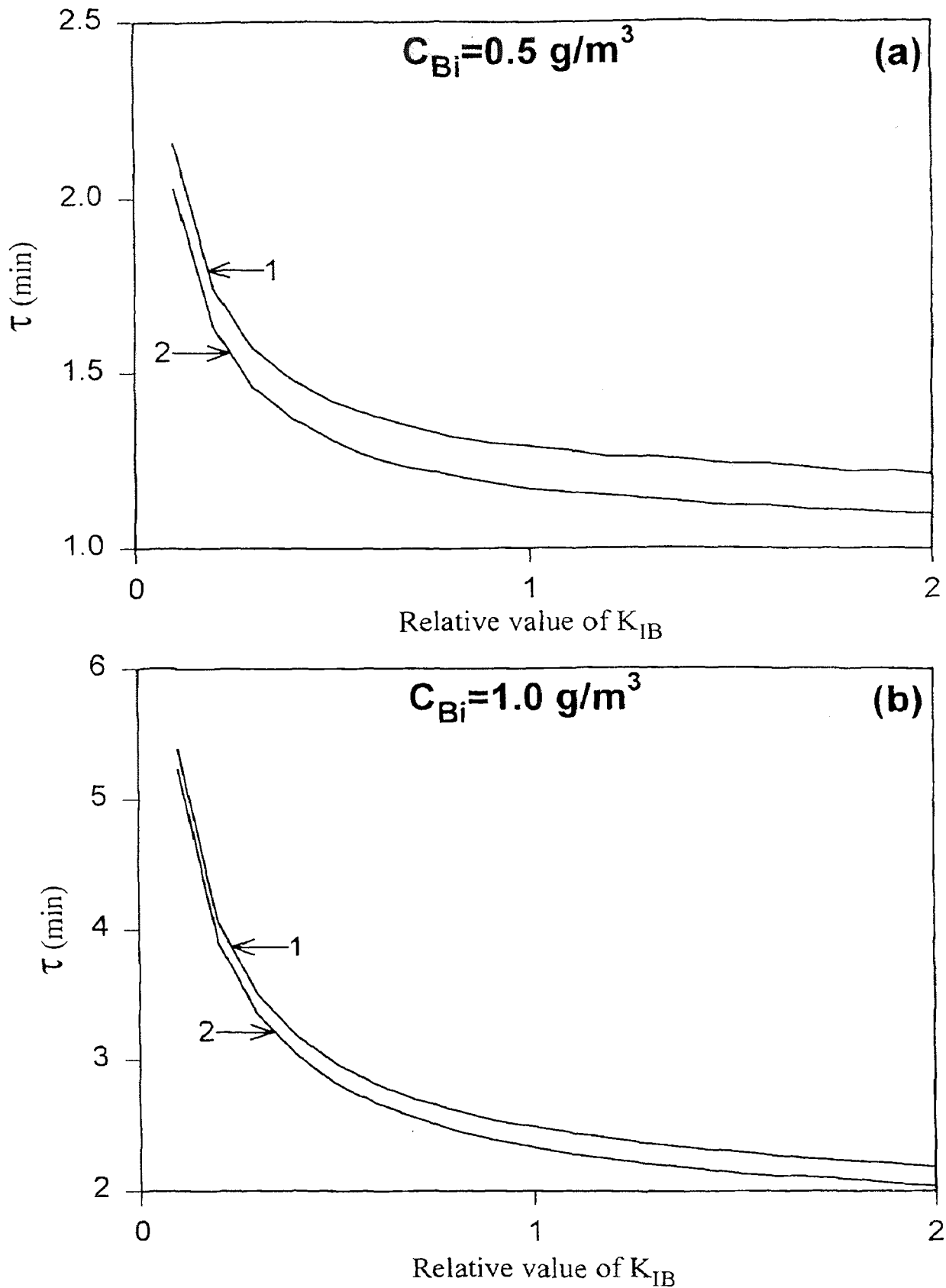
Figure A-5 Sensitivity of the model to the value of K_{Ij} for the case of butanol. Curve 1: 99% conversion; Curve 2: 95% conversion
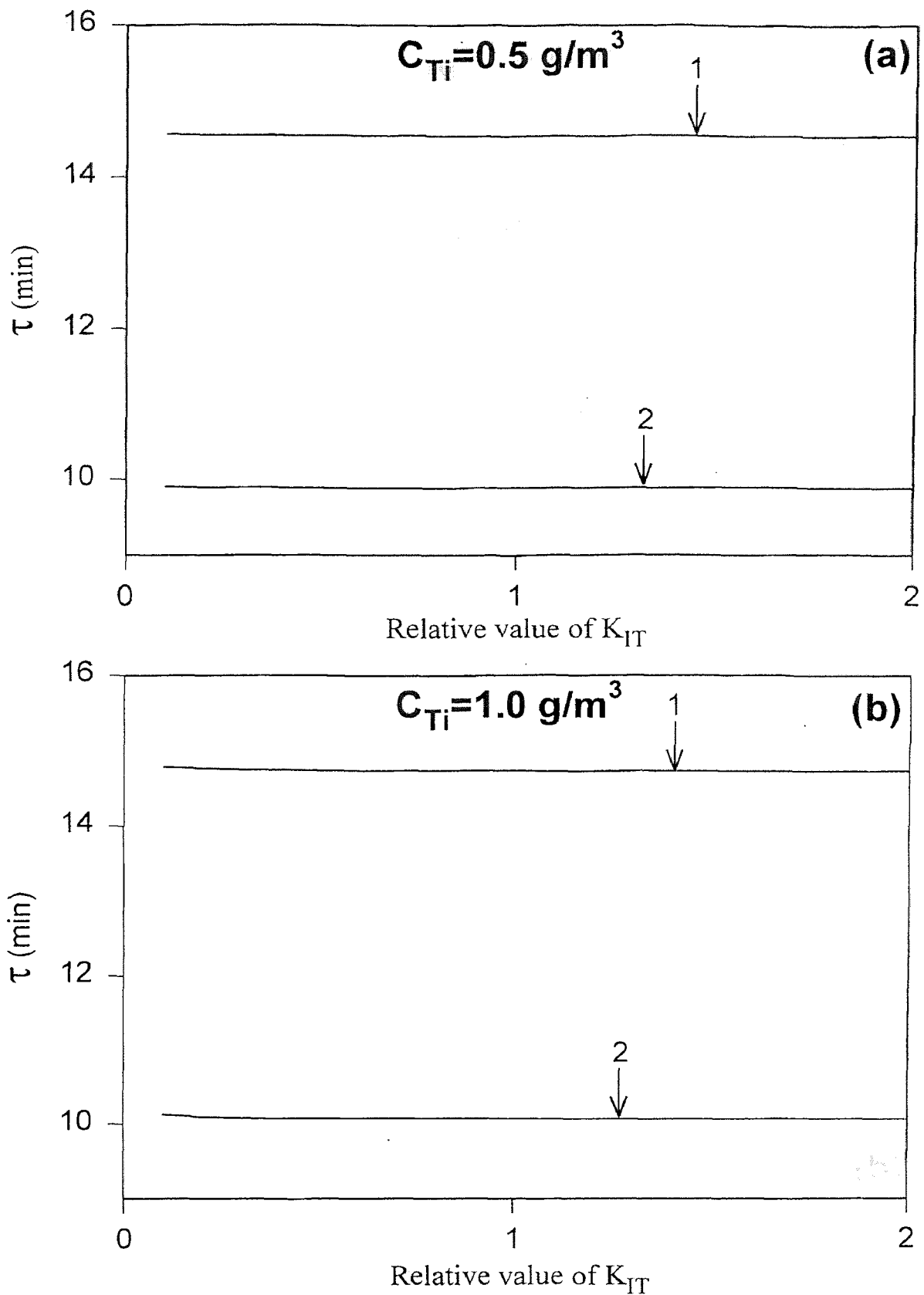
Figure A-6 Sensitivity of the model to the value of K_{Ij} for the case of toluene. Curve 1: 99% conversion; Curve 2: 95% conversion
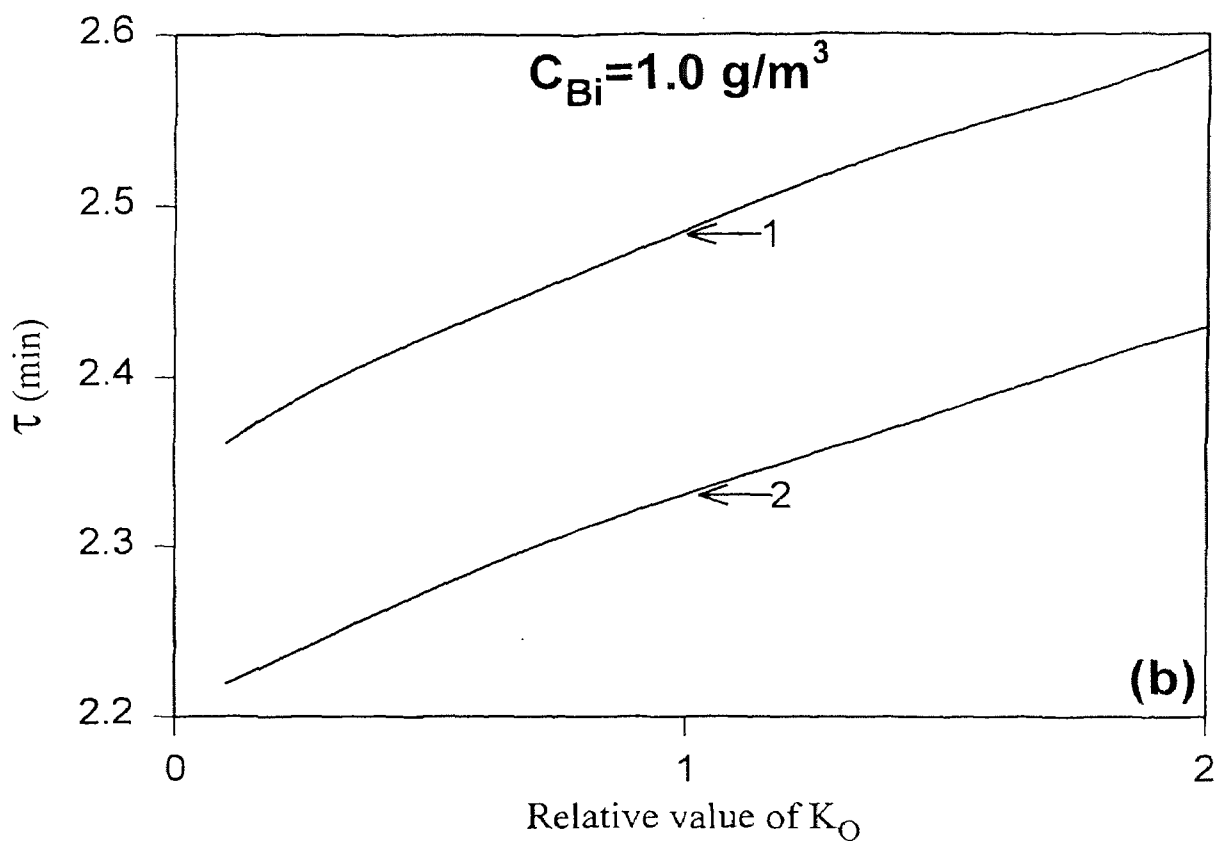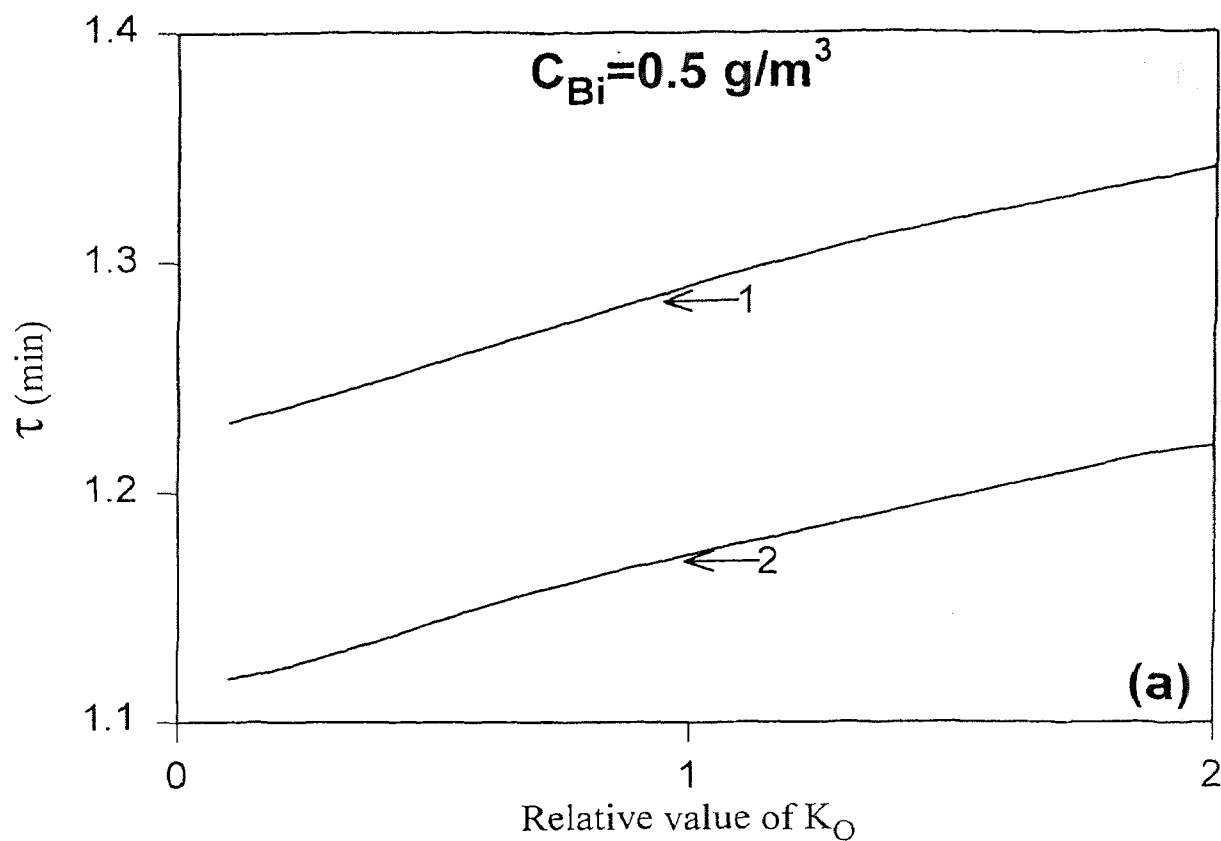
Figure A-7 Sensitivity of the model to the value of K_O for the case of butanol.
Curve 1: 99% conversion; Curve 2: 95% conversion
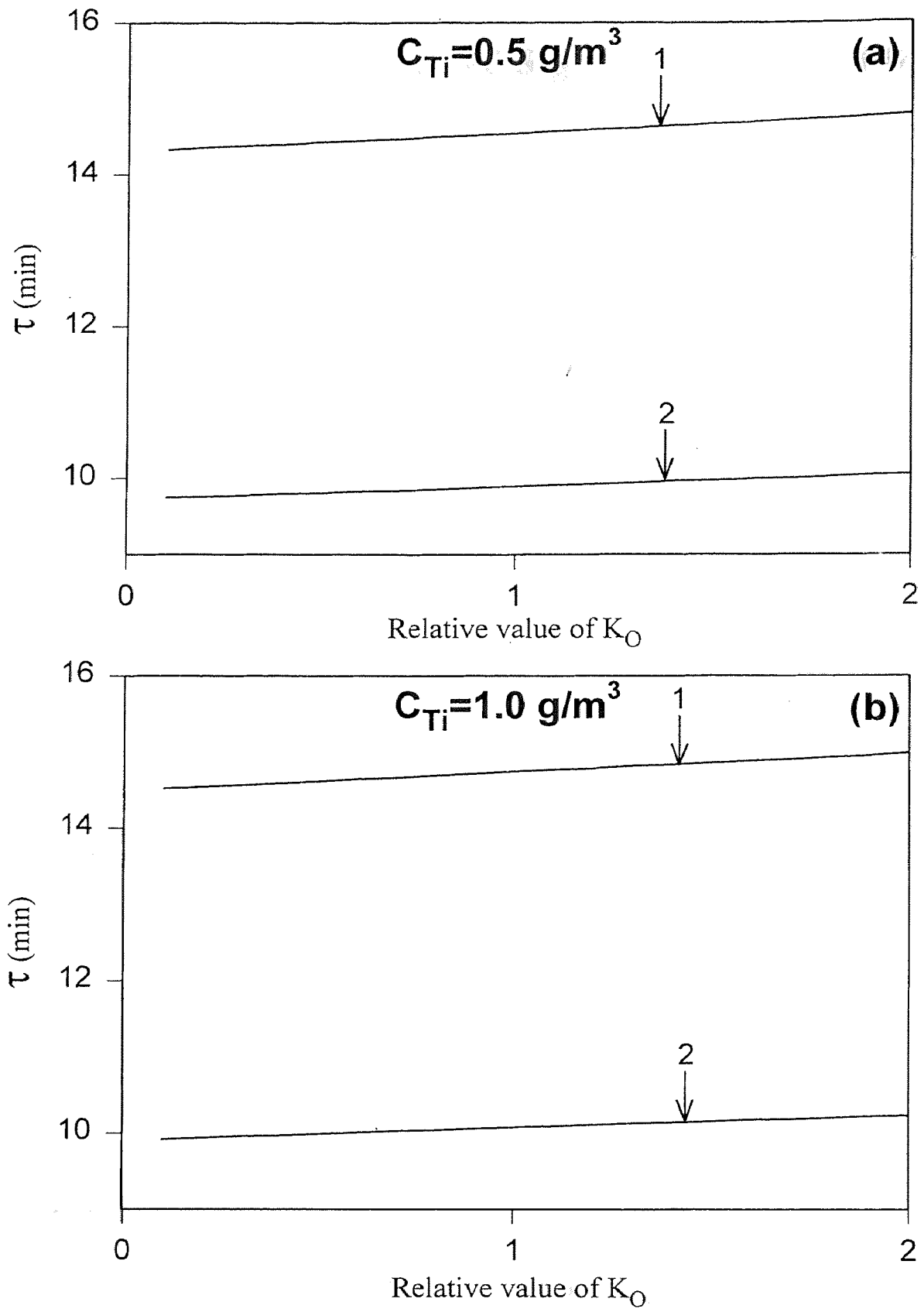
Figure A-8 Sensitivity of the model to the value of K_O for the case of toluene.
Curve 1: 99% conversion; Curve 2: 95% conversion
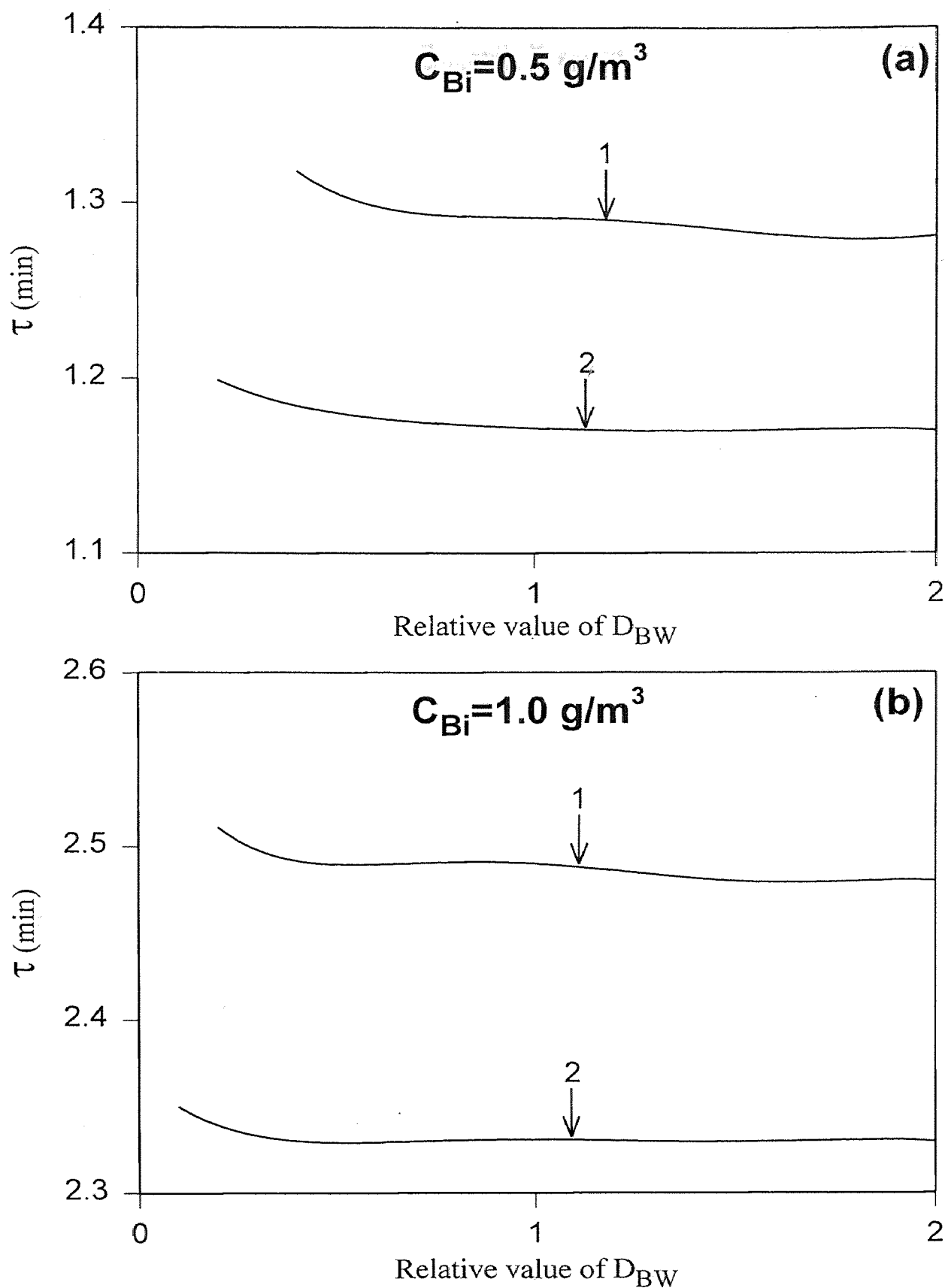
Figure A-9 Sensitivity of the model to the value of D_{jw} for the case of butanol. Curve 1: 99% conversion; Curve 2: 95% conversion
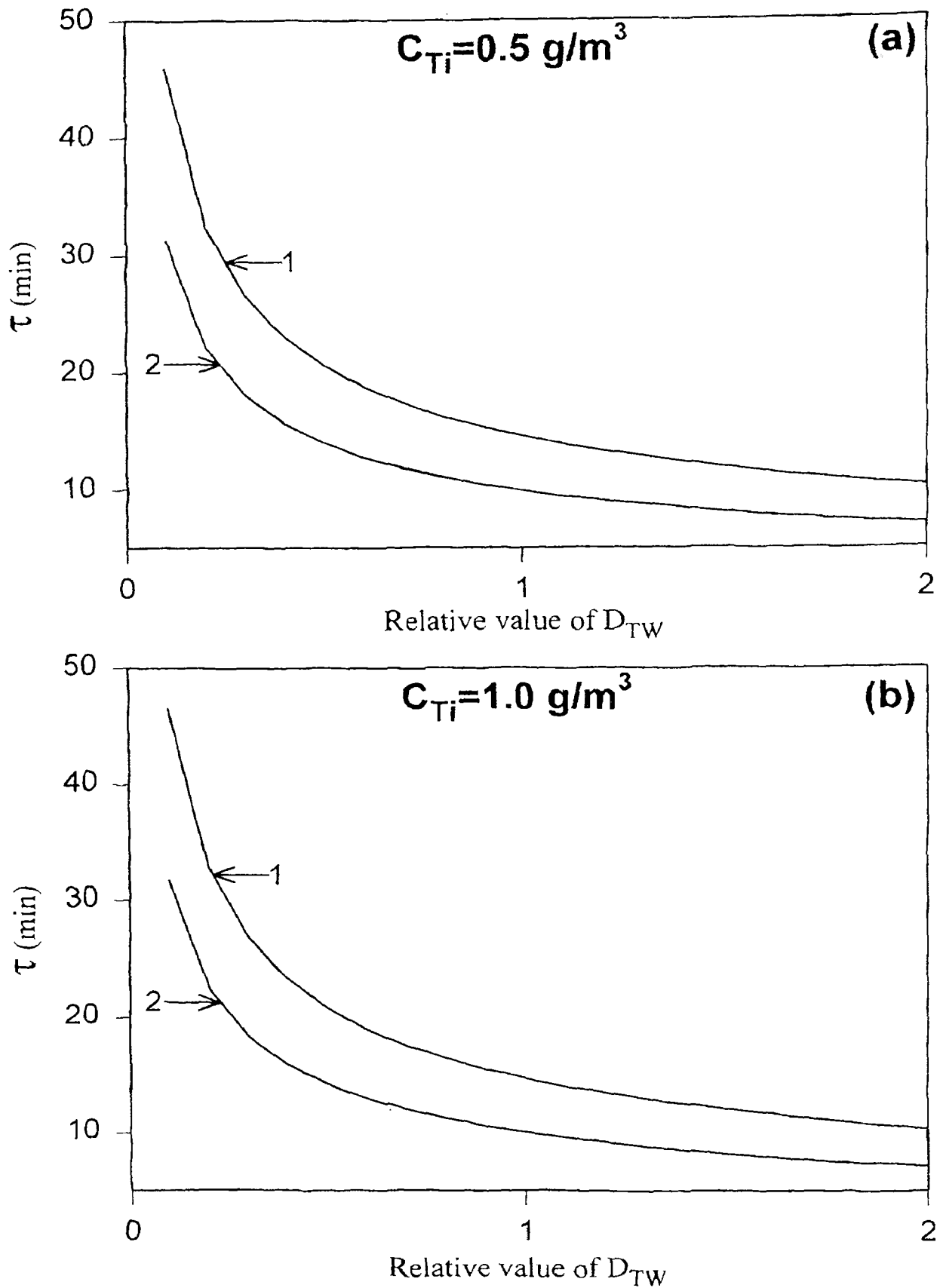
Figure A-10 Sensitivity of the model to the value of D_{jw} for the case of toluene. Curve 1: 99% conversion; Curve 2: 95% conversion
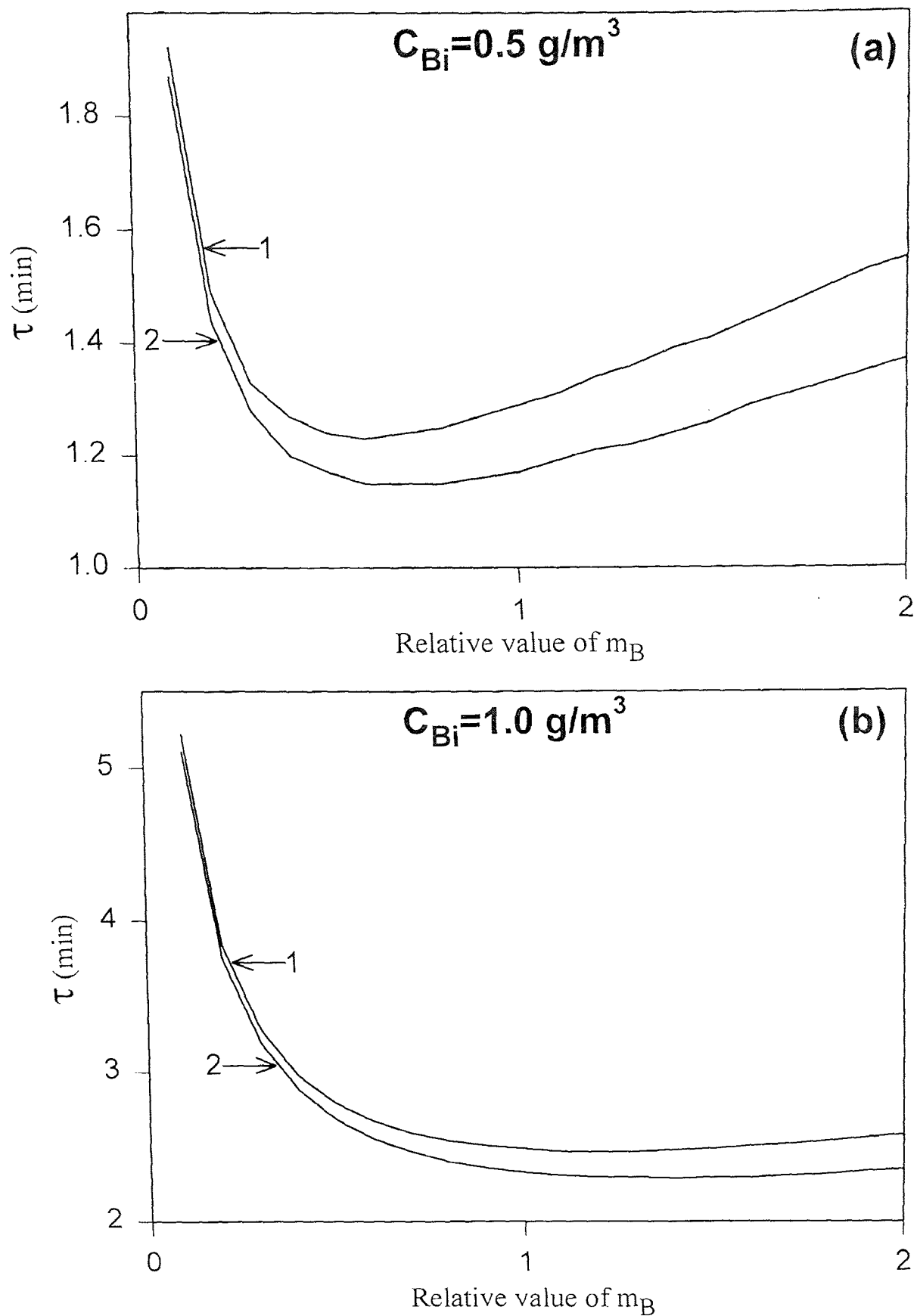
Figure A-11 Sensitivity of the model to the value of m_j for the case of butanol. Curve 1: 99% conversion; Curve 2: 95% conversion
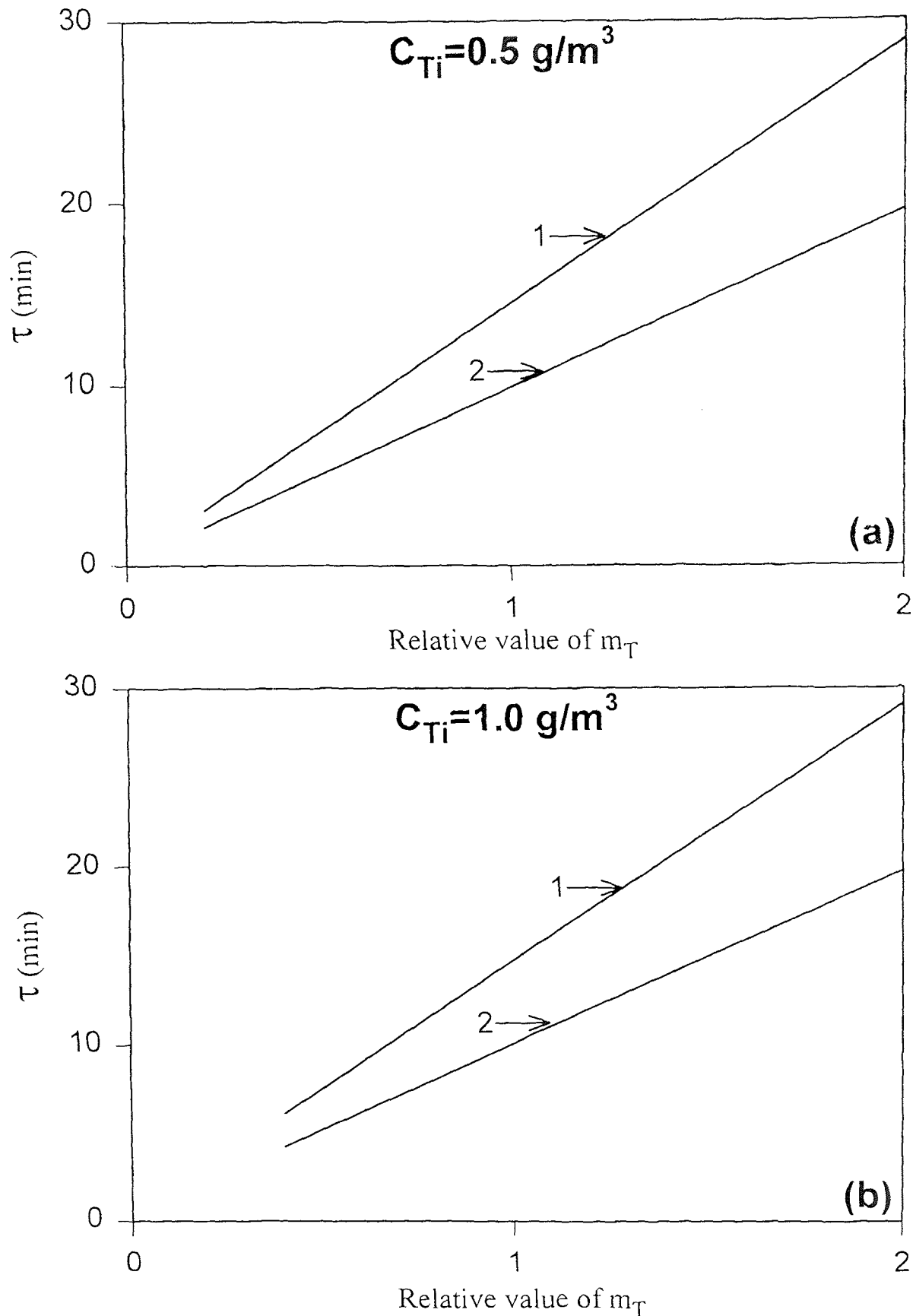
Figure A-12 Sensitivity of the model to the value of m_j for the case of toluene.
Curve 1: 99% conversion; Curve 2: 95% conversion
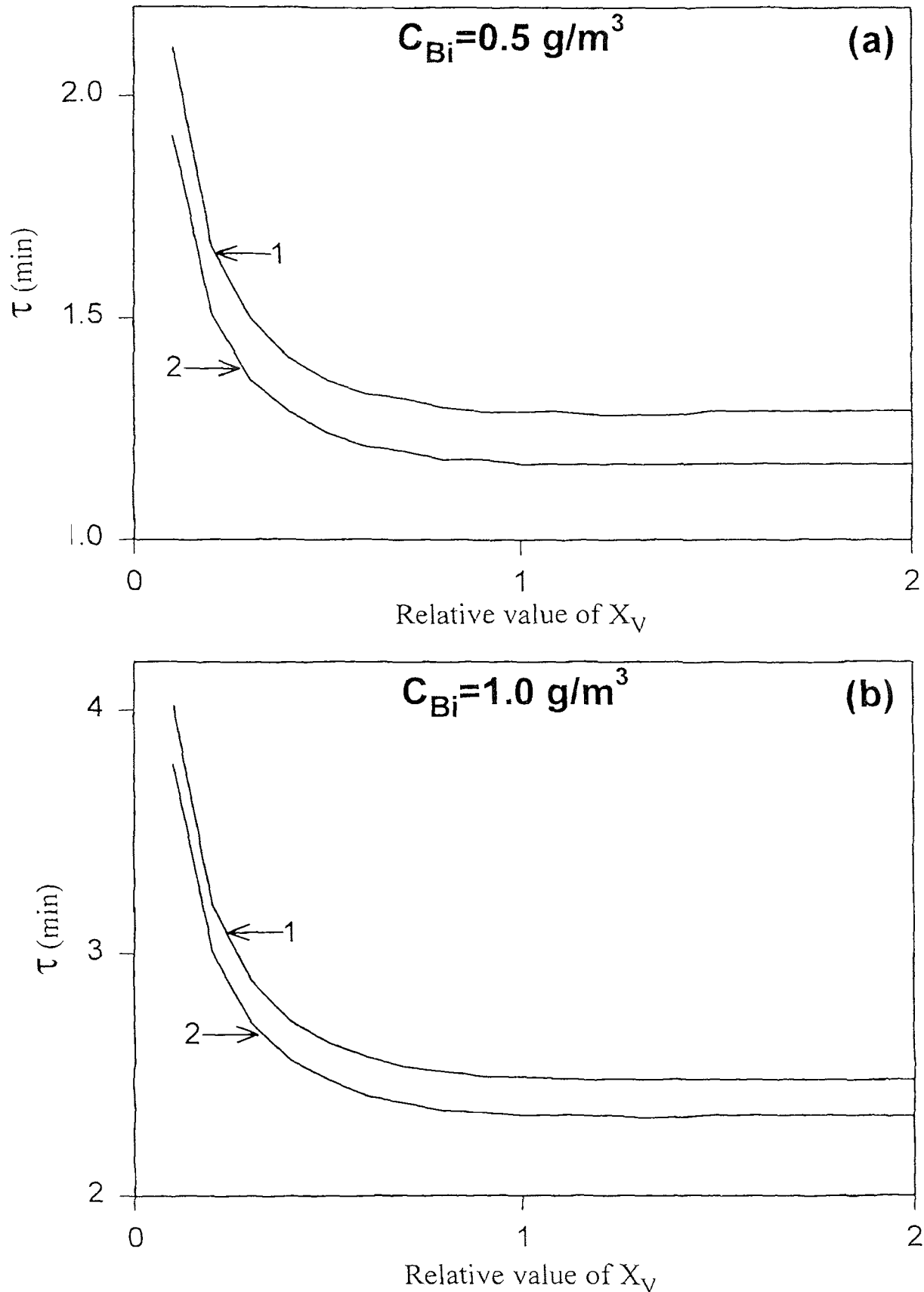
Figure A-13 Sensitivity of the model to the value of X_V for the case of butanol.
Curve 1: 99% conversion; Curve 2: 95% conversion
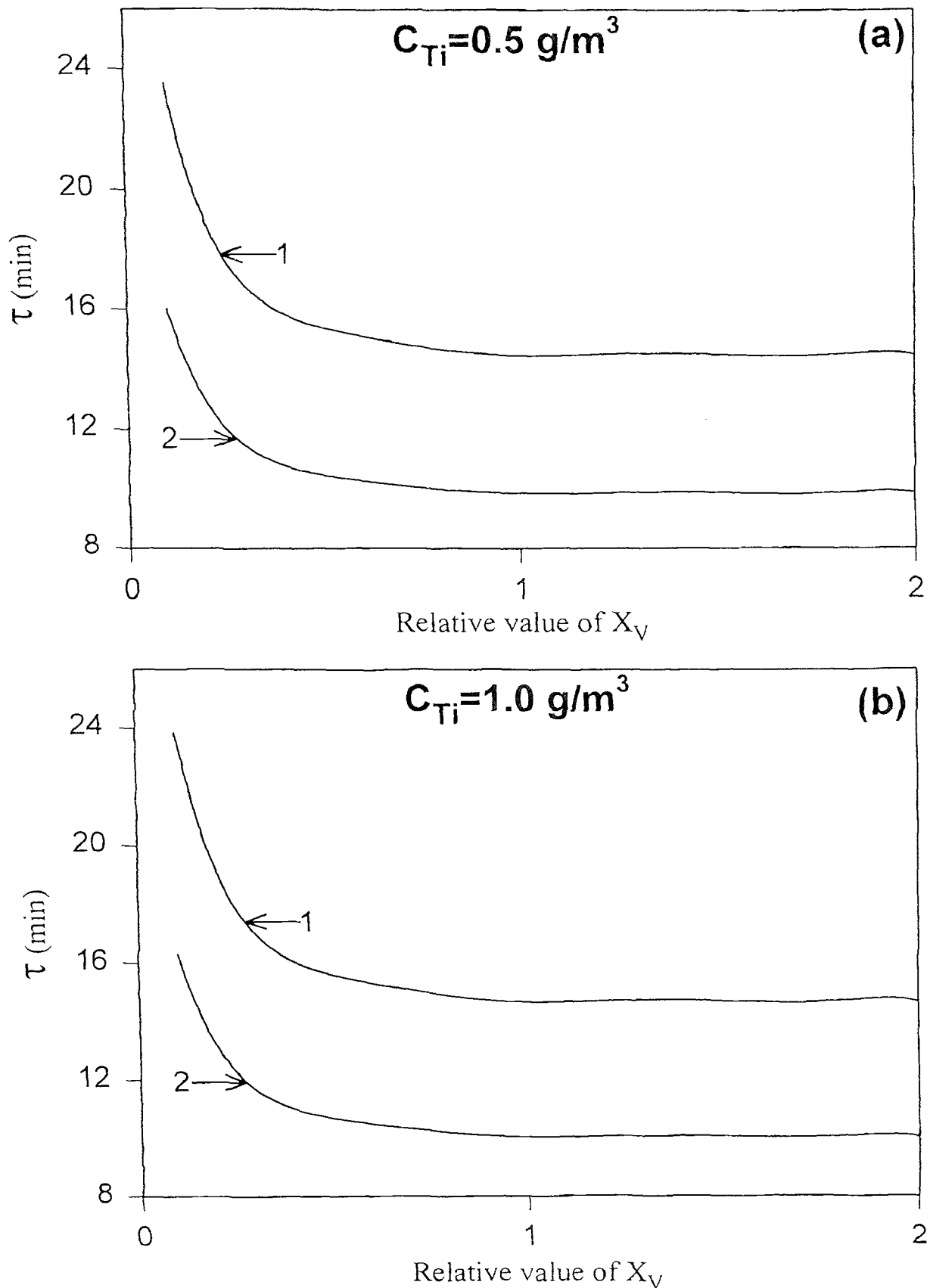
Figure A-14 Sensitivity of the model to the value of X_V for the case of toluene.
Curve 1: 99% conversion; Curve 2: 95% conversion
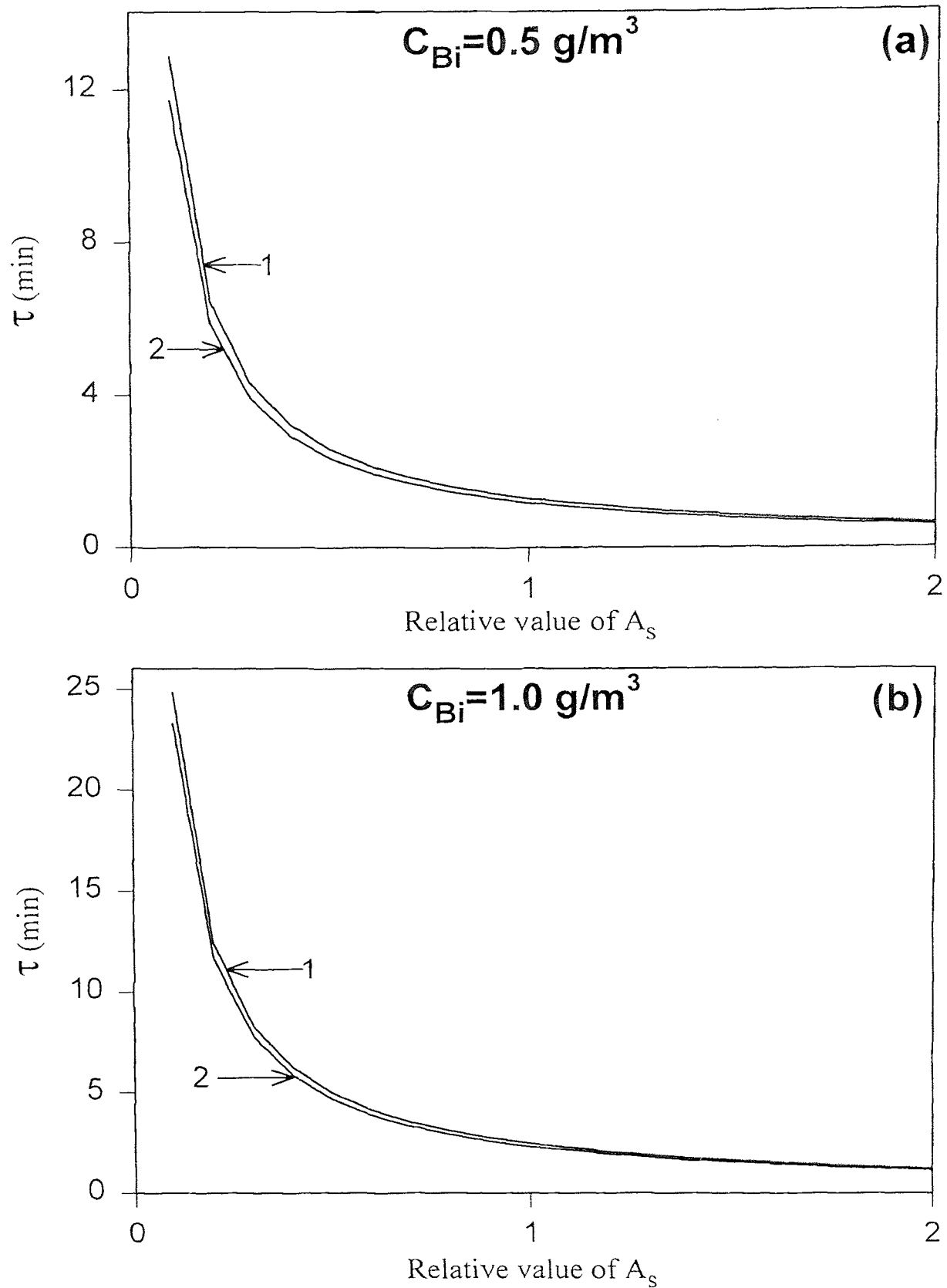
Figure A-15 Sensitivity of the model to the value of A_S for the case of butanol.
Curve 1: 99% conversion; Curve 2: 95% conversion
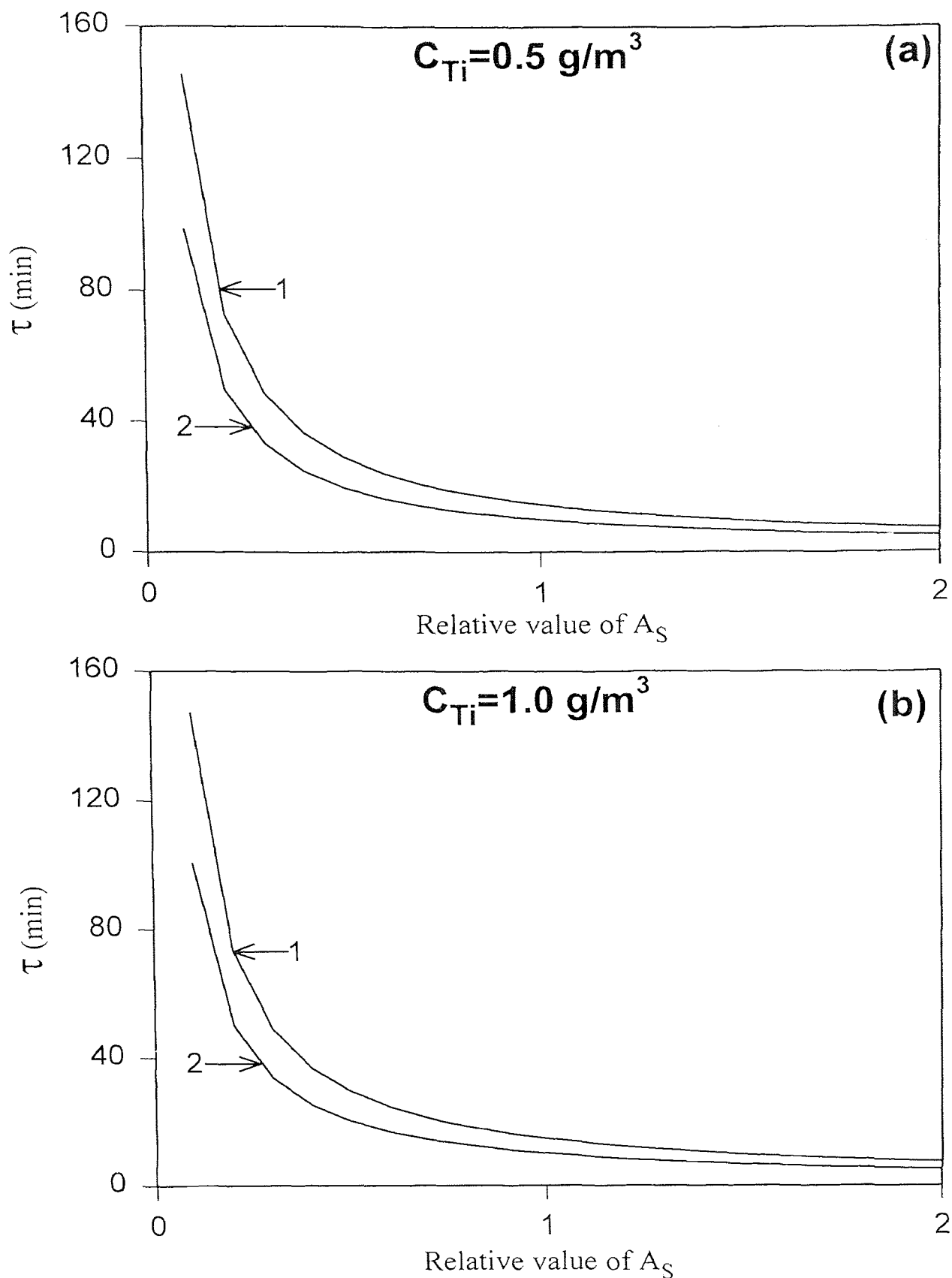
Figure A-16 Sensitivity of the model to the value of A_S for the case of toluene.
Curve 1: 99% conversion; Curve 2: 95% conversion
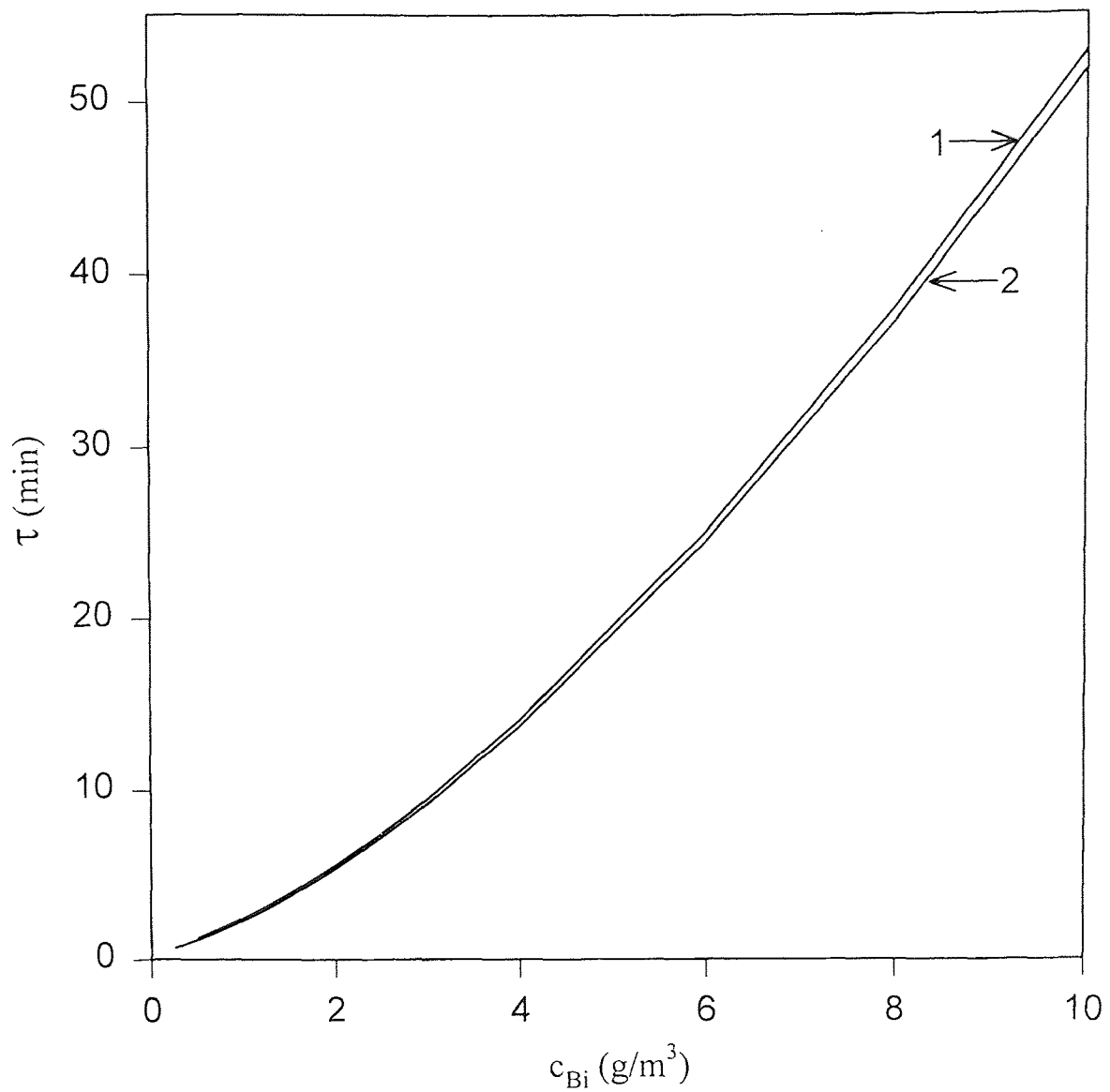
Figure A-17 Required residence time in a biofilter removing butanol as a function of inlet butanol concentration.
Curve 1: 99% conversion; Curve 2: 95% conversion
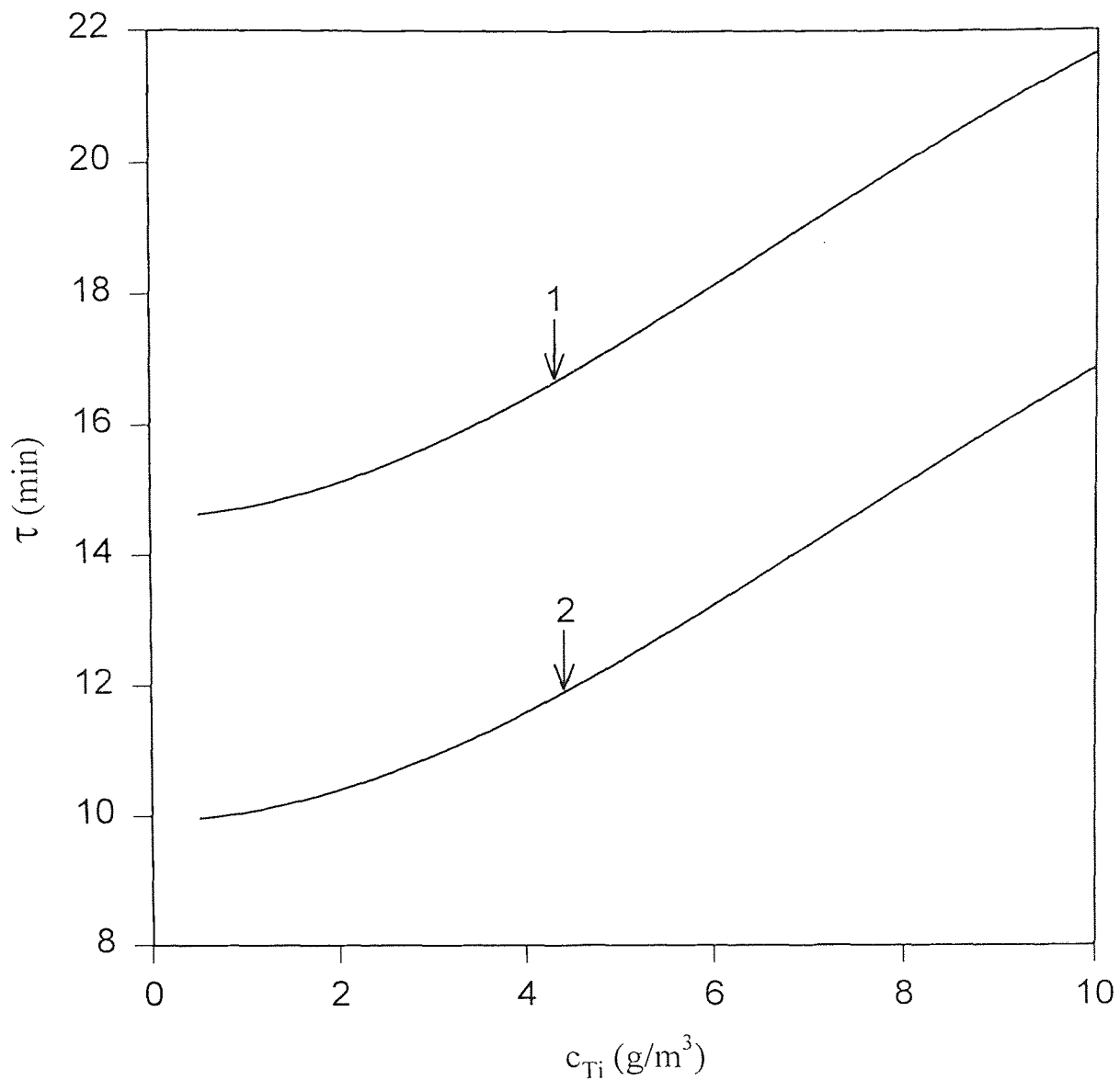
Figure A-18 Required residence time in a biofilter removing toluene as a function of inlet toluene concentration.
Curve 1: 99% conversion; Curve 2: 95% conversion
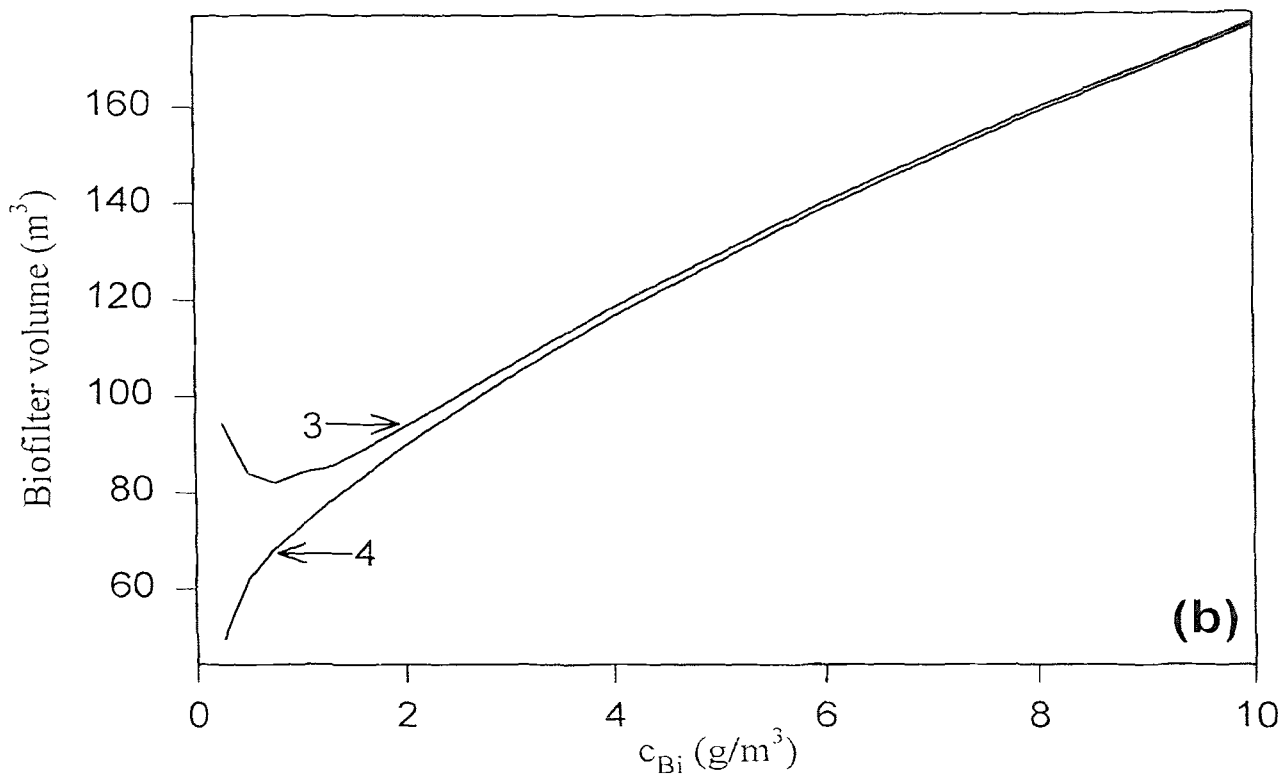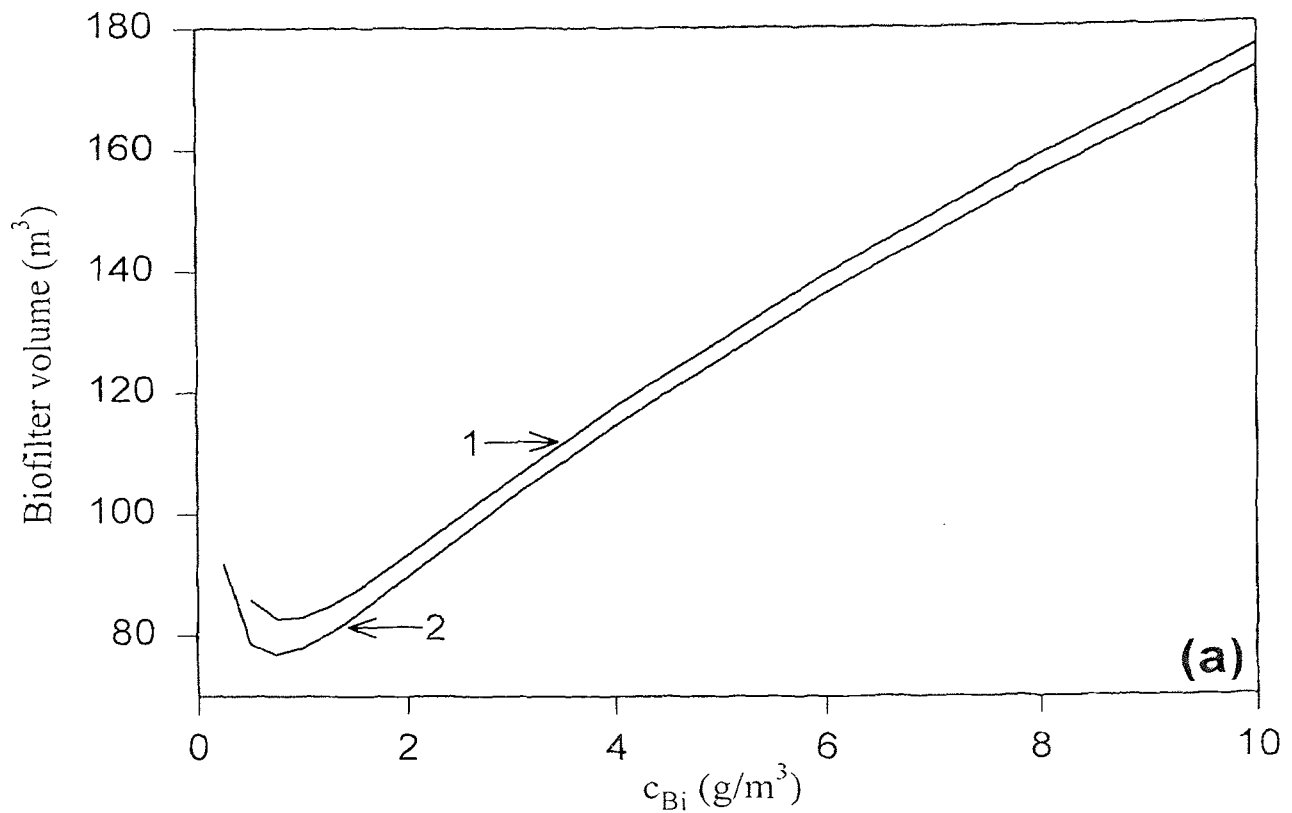
Figure A-19 Required biofilter volume as a function of inlet concentration for a butanol load of 2.0 kg/h.
Curve 1: 99% conversion; Curve 2: 95% conversion;
Curve 3: C_{ex}=0.01 g/m^3; Curve 4: Cex=0.1 g/m^3
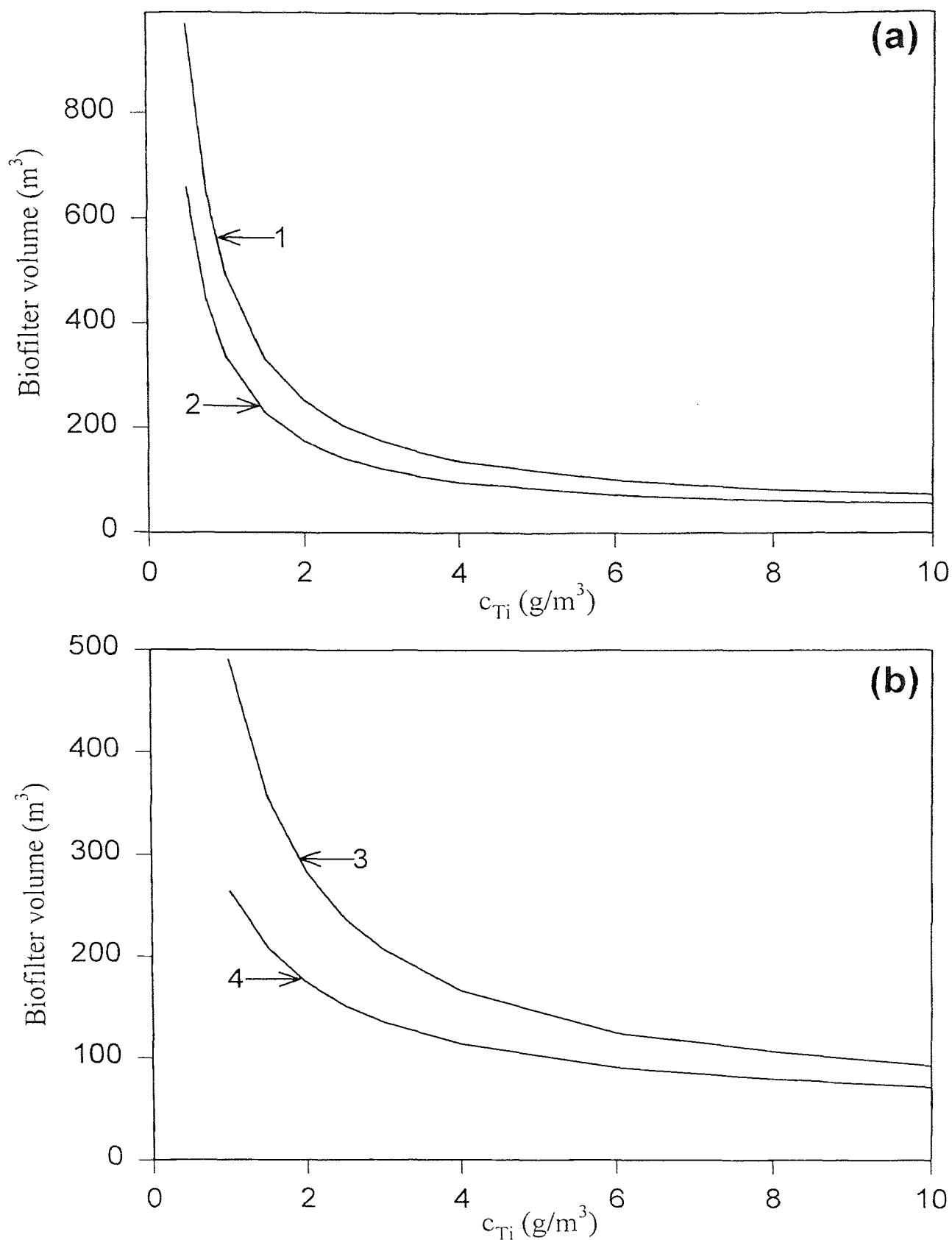
Figure A-20 Required biofilter volume as a function of inlet concentration for a toluene load of 2.0 kg/h.
Curve 1: 99% conversion; Curve 2: 95% Conversion
Curve 3: C_{ex}=0.01 g/m³; Curve 4: C_{ex}=0.1 g/m³
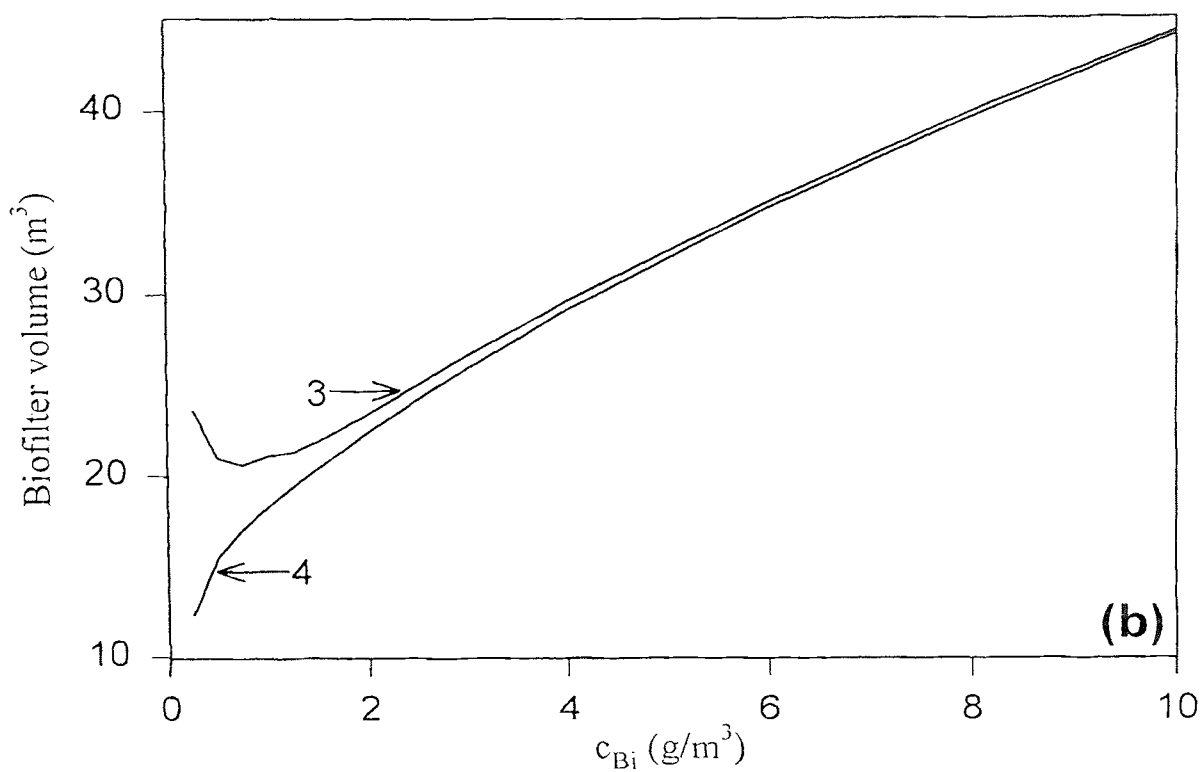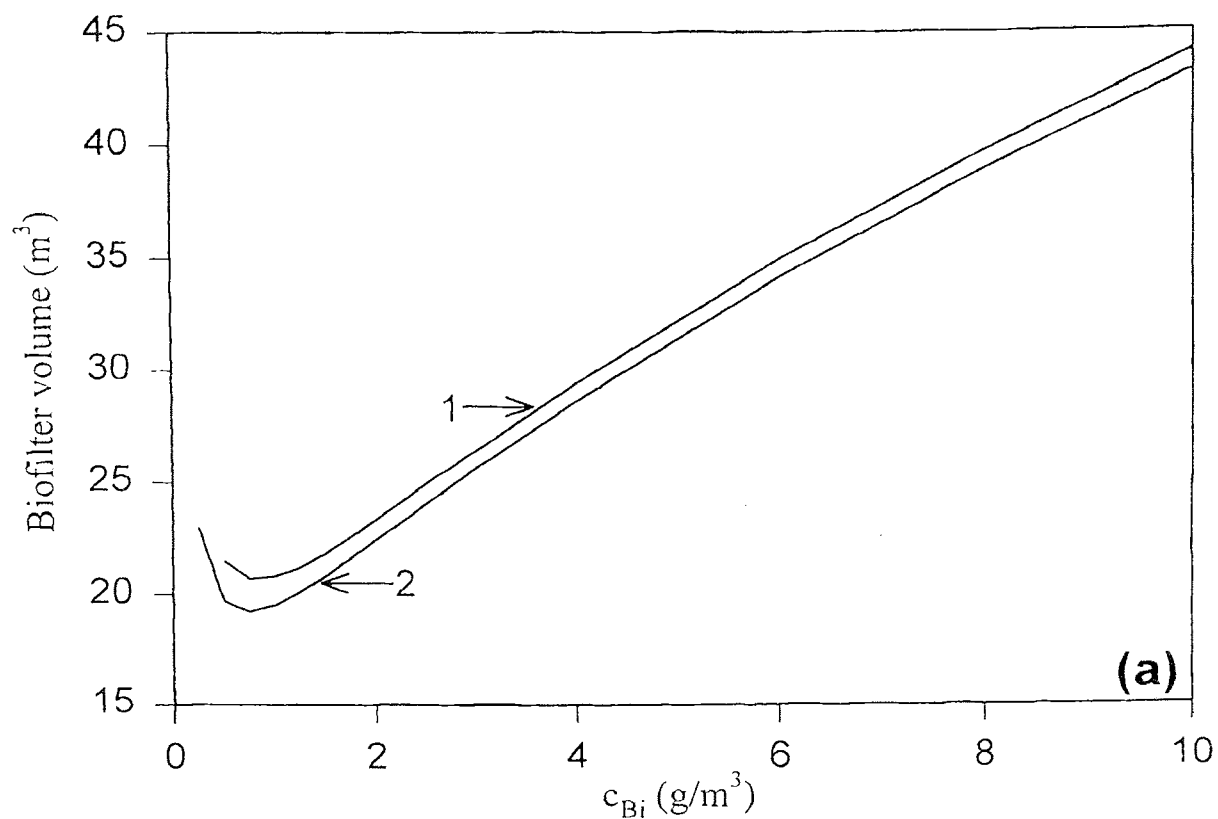
Figure A-21 Required biofilter volume as a function of inlet concentration for a butanol load of 0.5 kg/h.
Curve 1: 99% conversion; Curve 2: 95% conversion
Curve 3: C_{ex}=0.01 g/m^3; Curve 4: C_{ex}=0.1 g/m^3
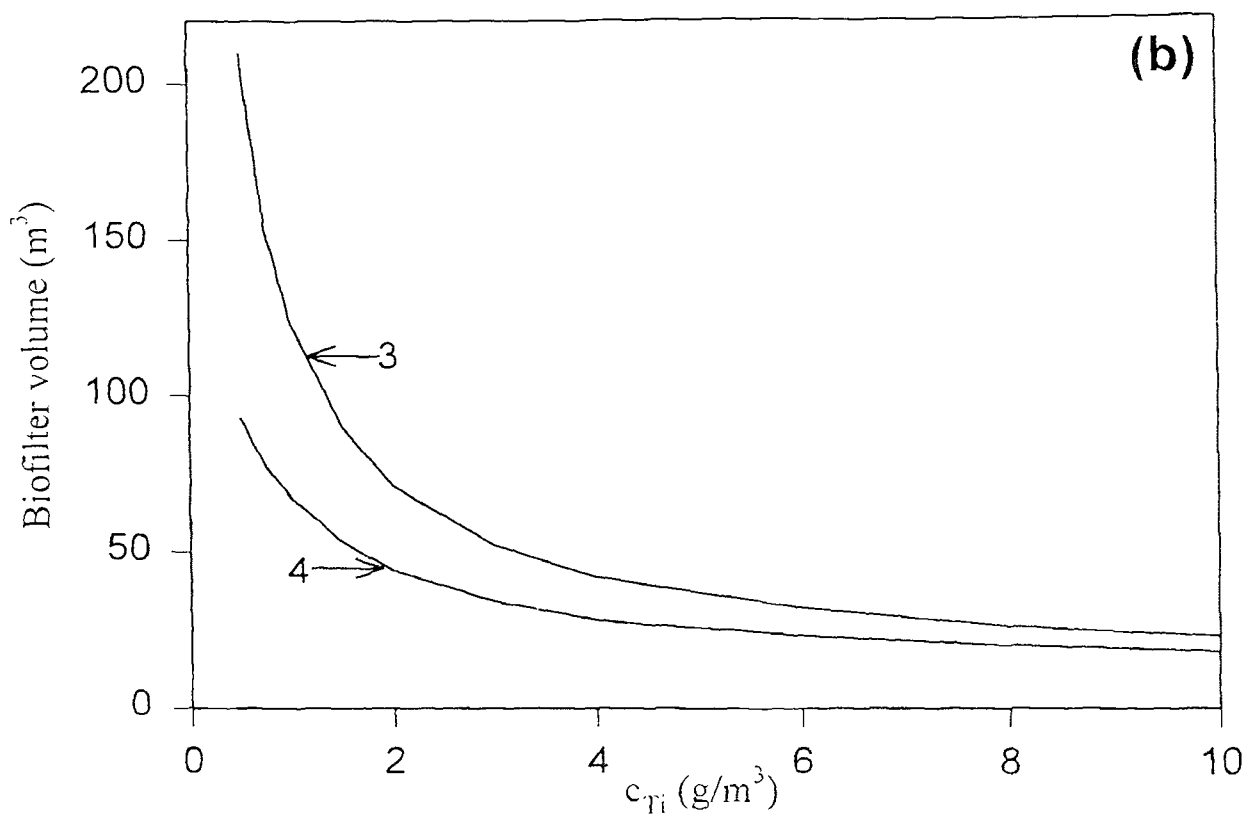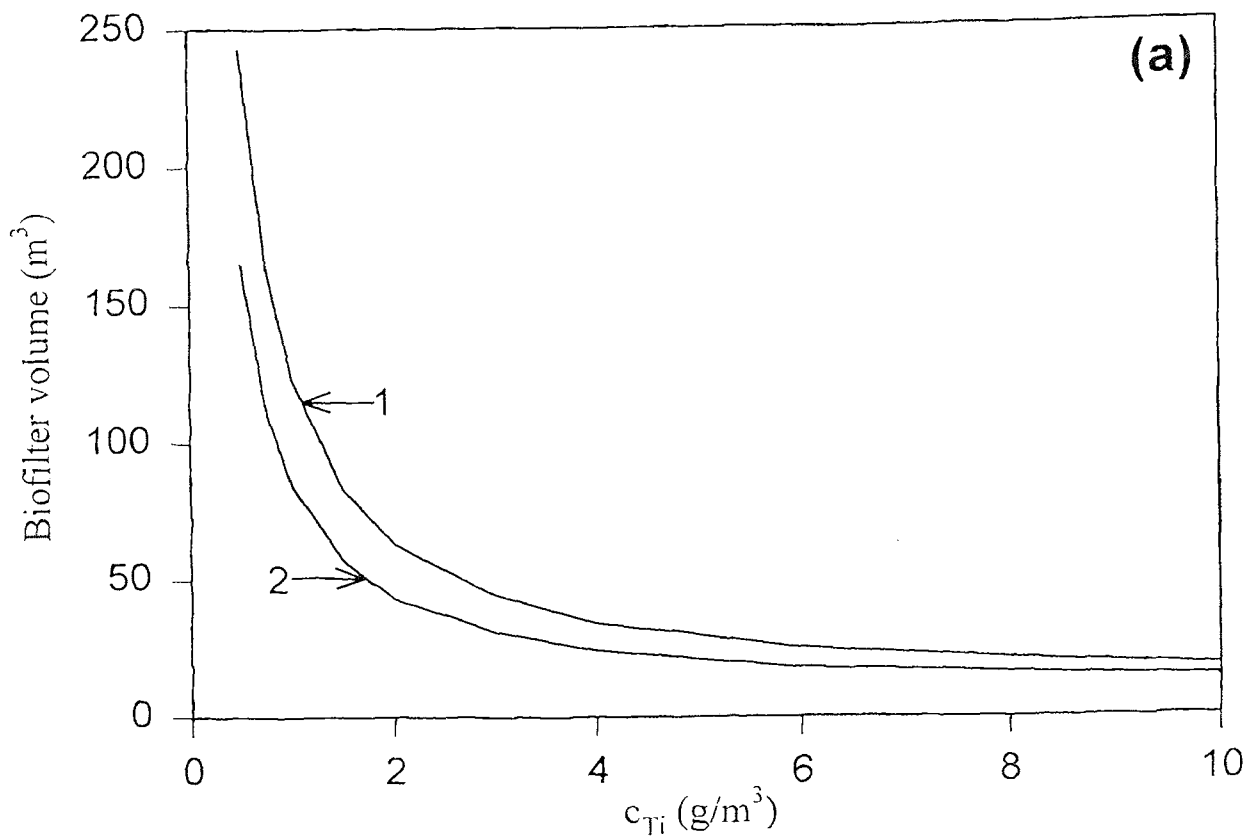
Figure A-22 Required biofilter volume as a function of inlet concentration for a toluene load of 0.5 kg/h.
Curve 1: 99% conversion; Curve 2: 95% conversion
Curve 3: C_{ex}=0.01 g/m^3; Curve 4: C_{ex}=0.1 g/m^3

APPENDIX B

RESULTS FROM STUDIES WITH THE TRANSIENT BIOFILTRATION MODEL
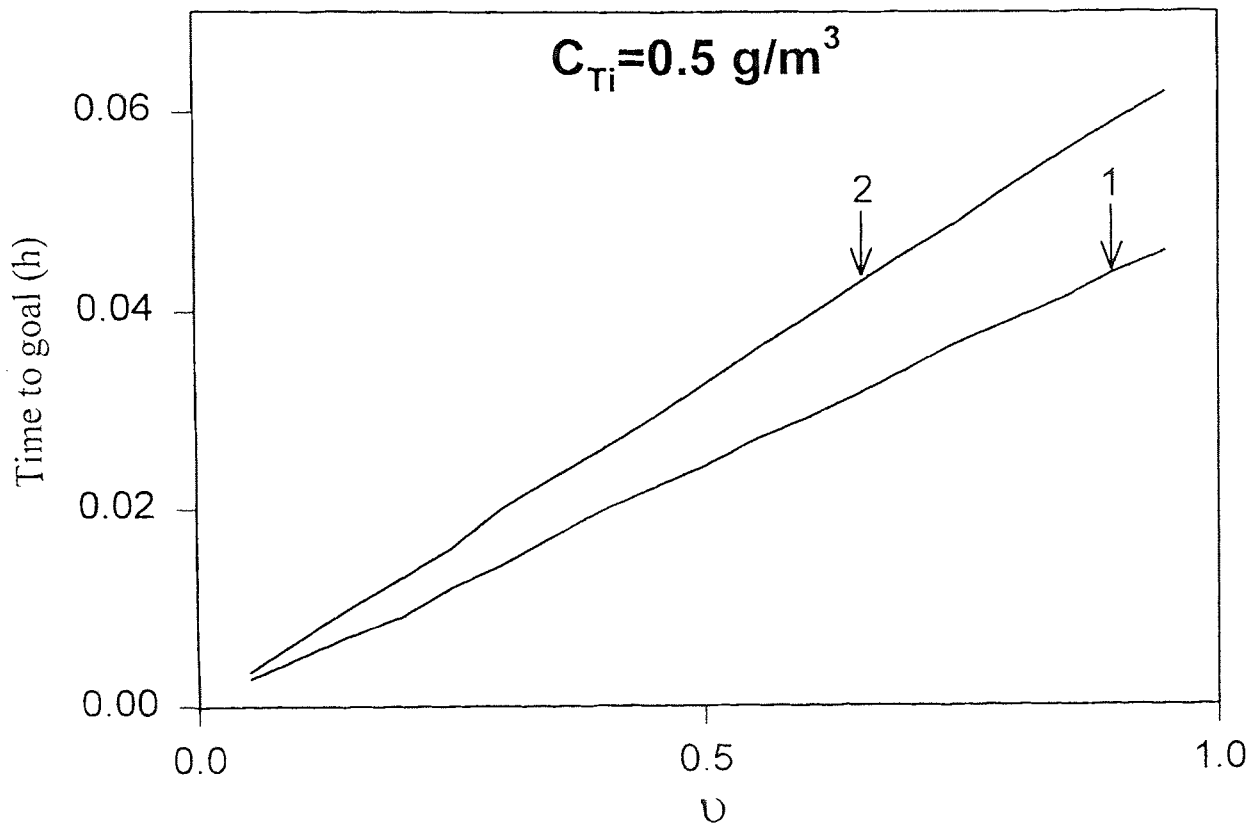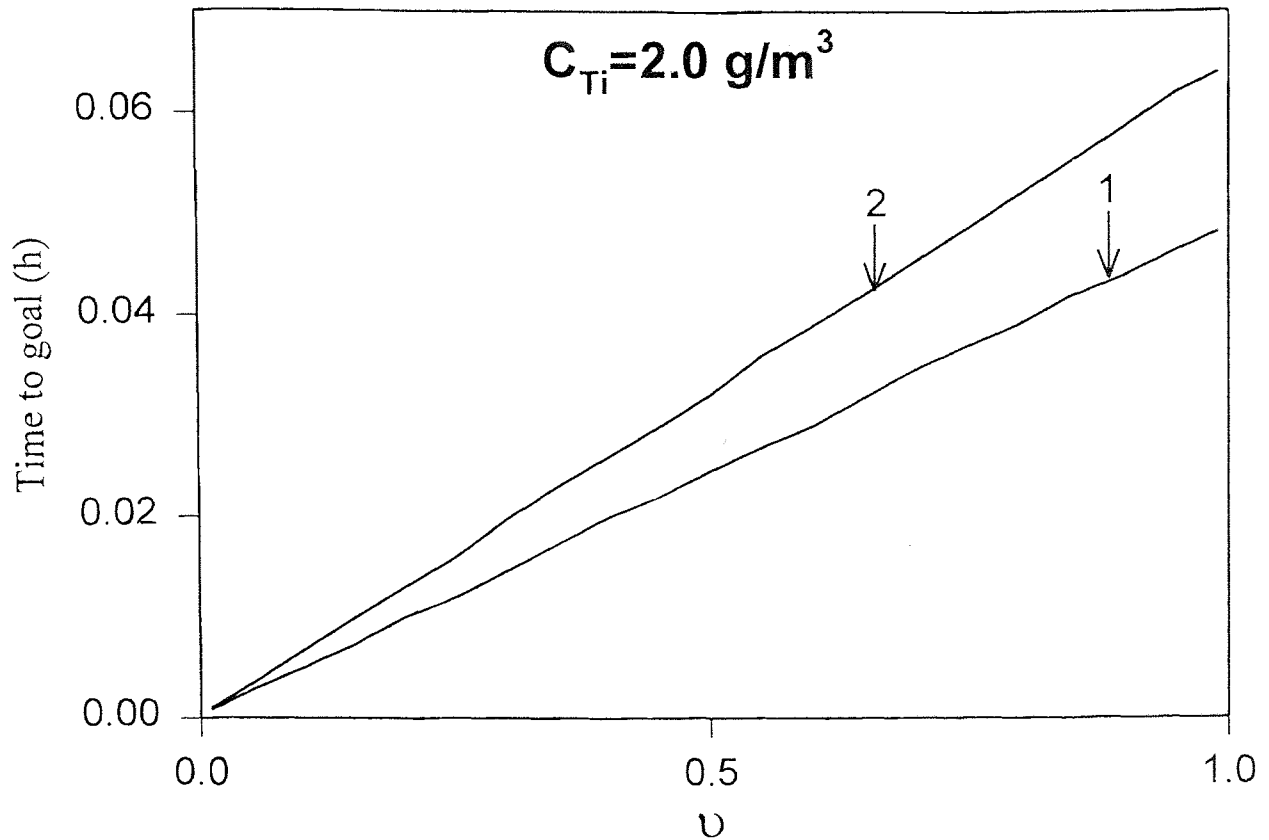
Figure B-1 Time for reaching 50% (curve 1) and 90% (curve 2) of the steady state level as a function of void fraction (υ).
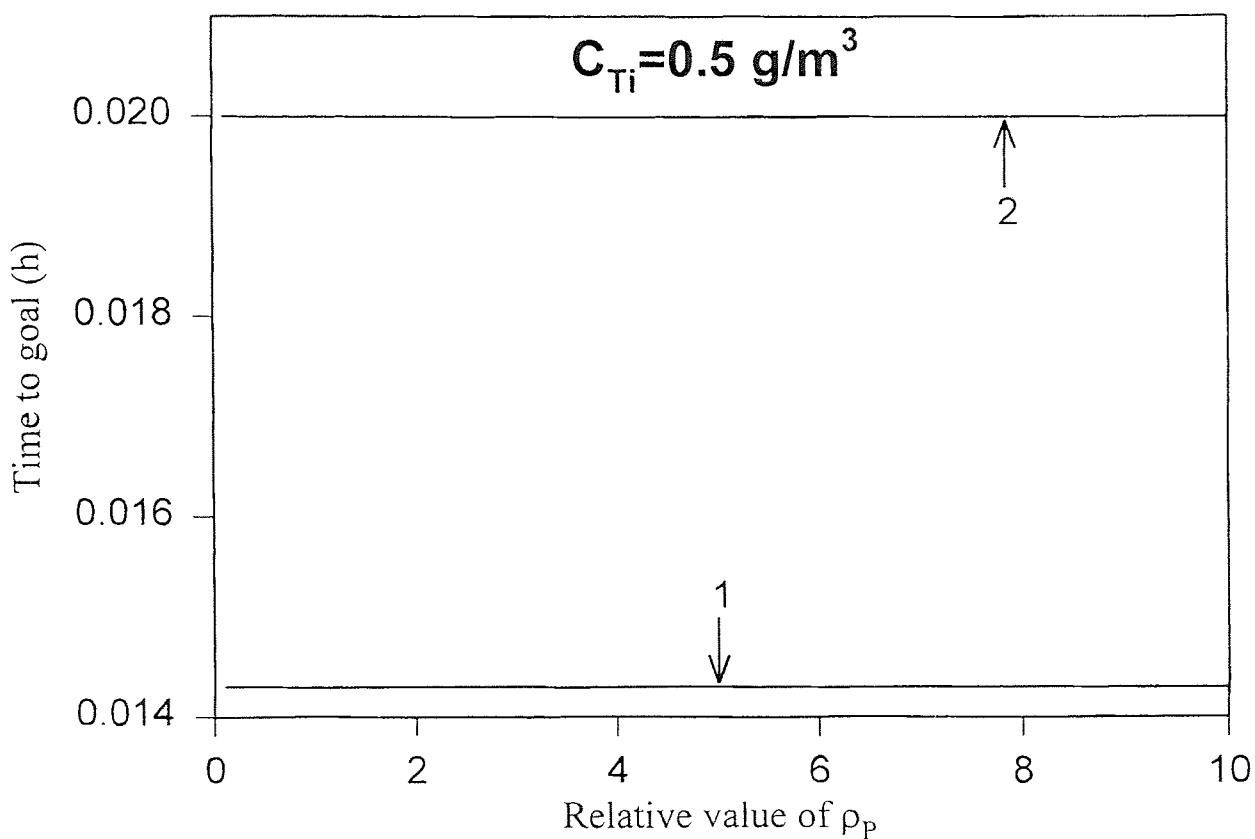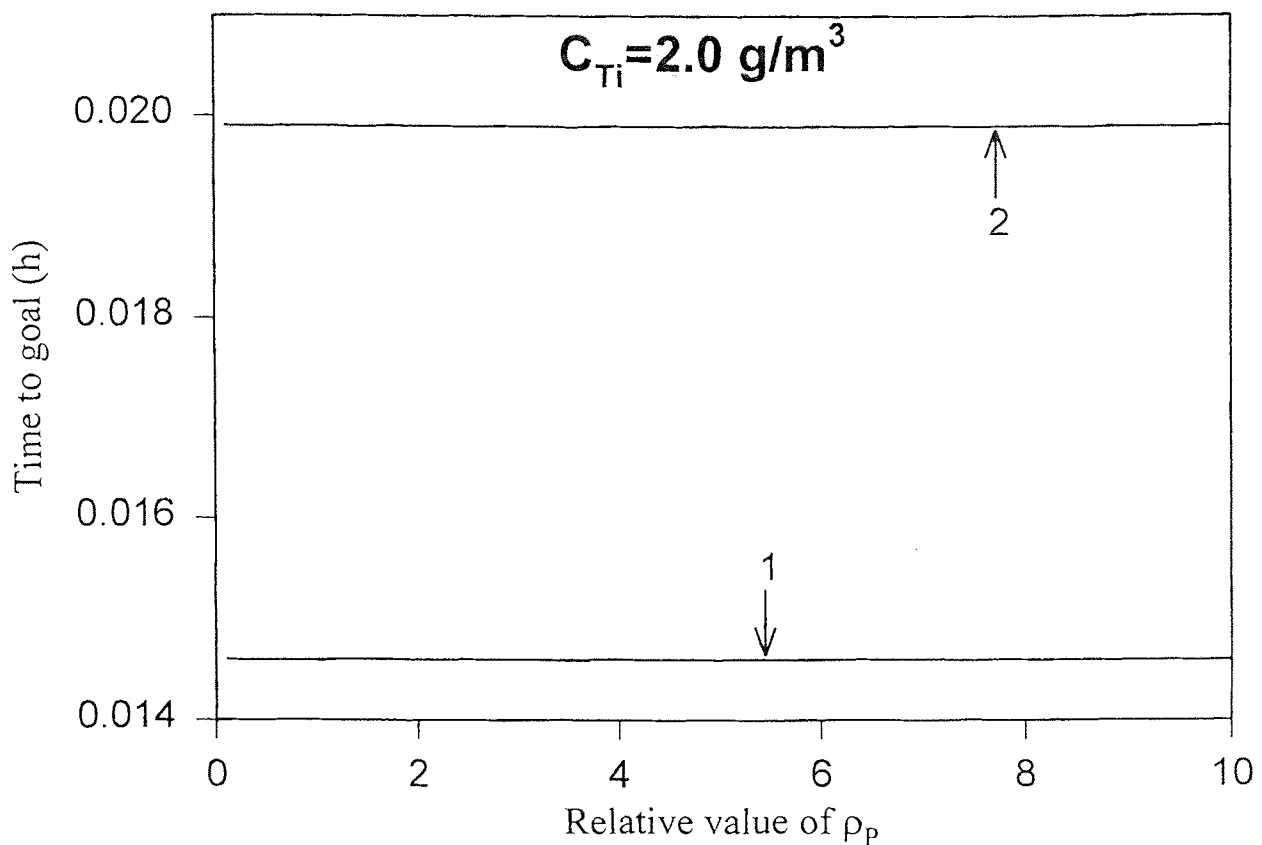
Figure B-2 Time for reaching 50% (curve 1) and 90% (curve 2) of the steady state level as a function of the density of packing (ρ_P).
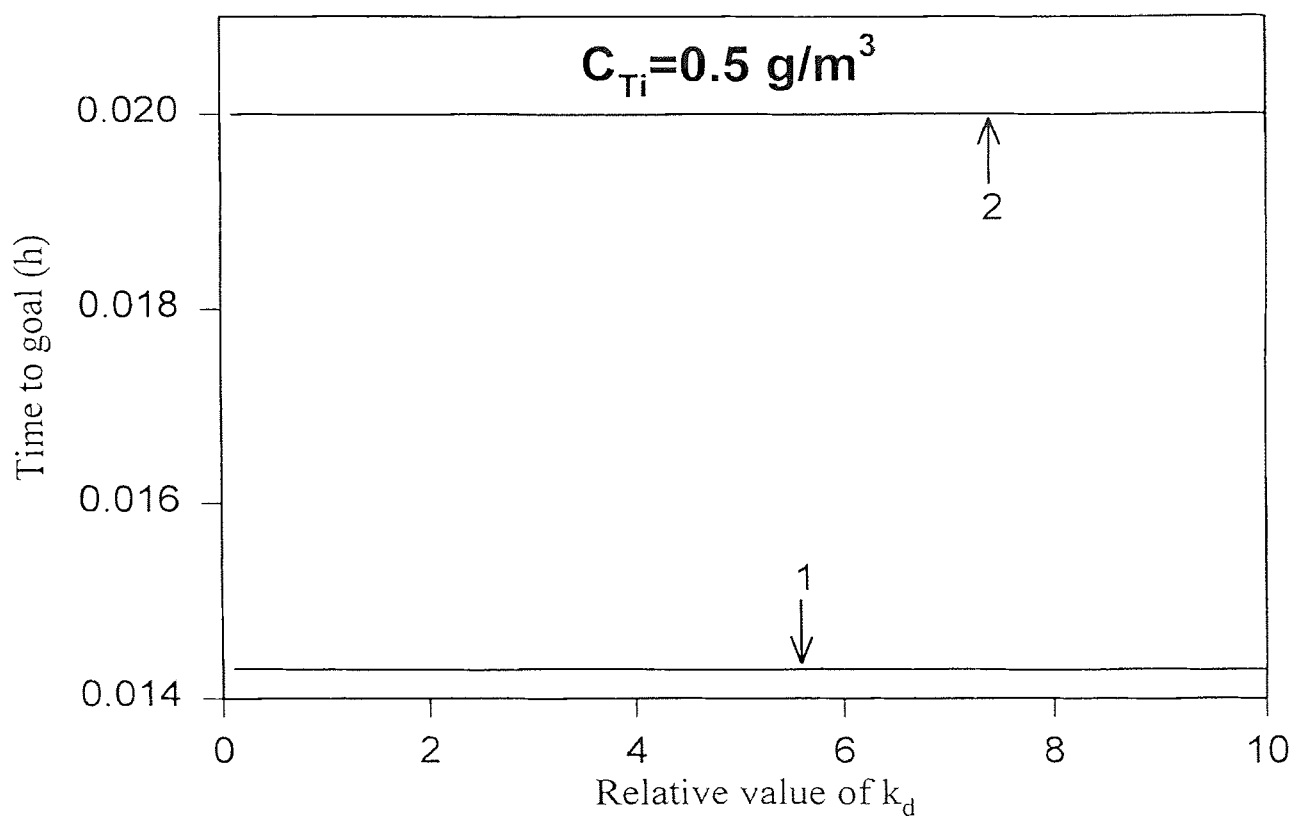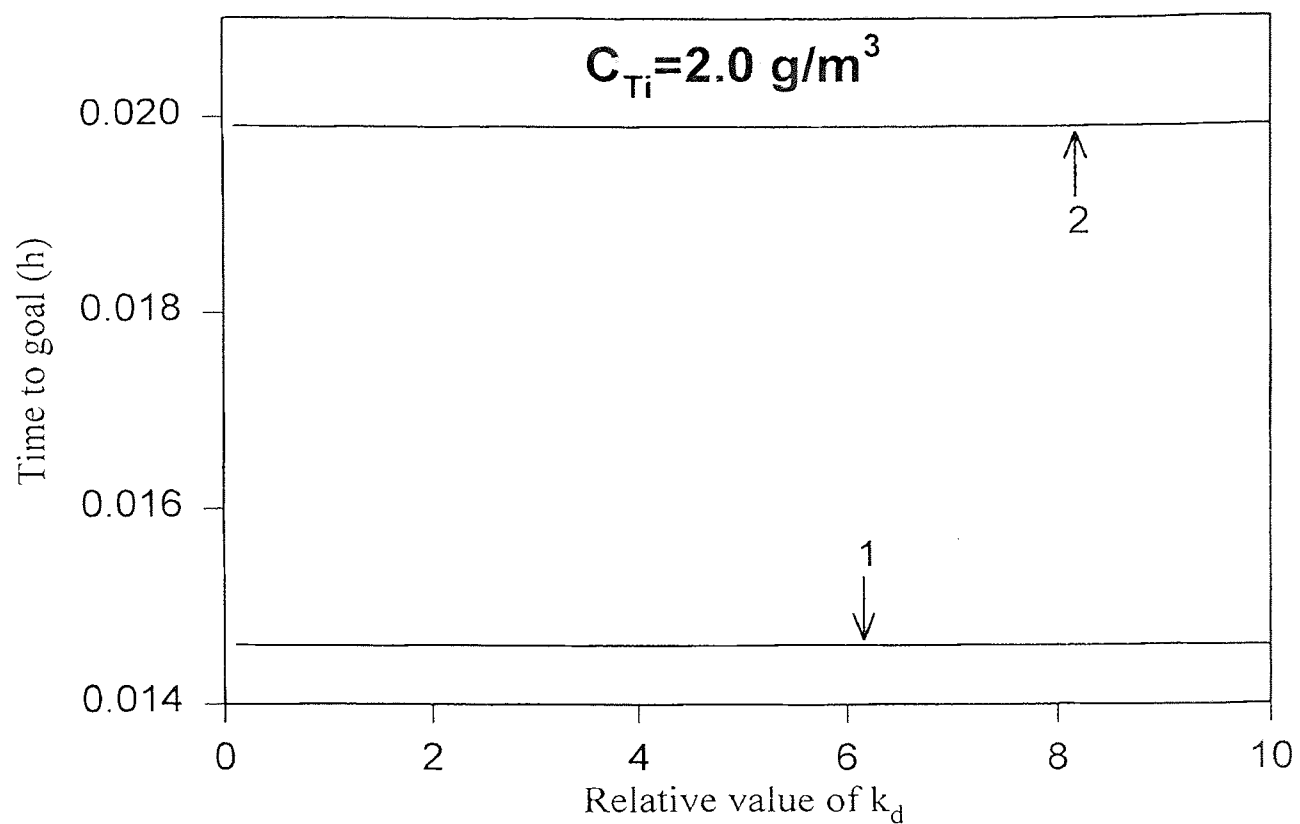
Figure B-3 Time for reaching 50% (curve 1) and 90% (curve 2) of the steady state level as a function of the Freundlich adsorption constant (k_d).
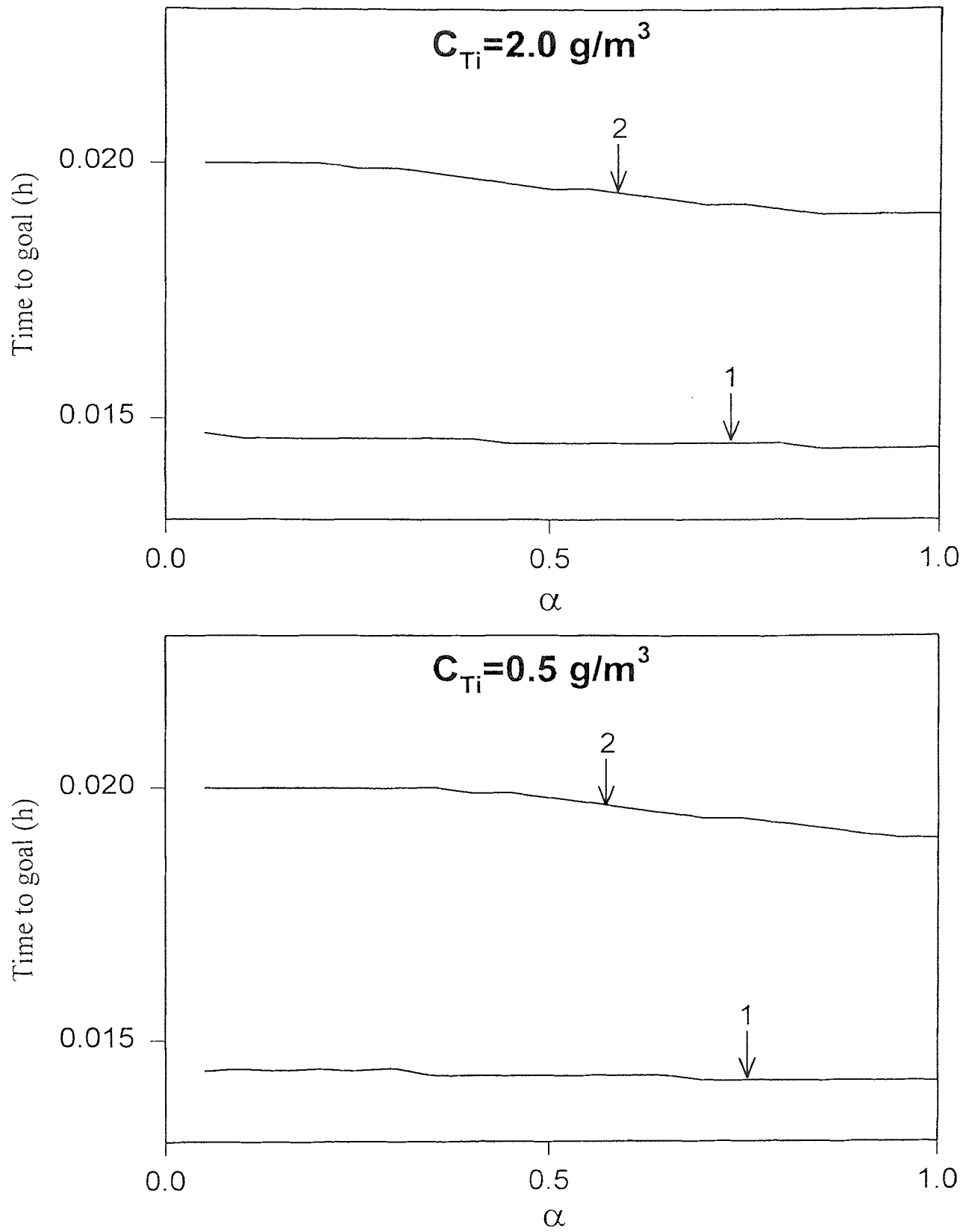
Figure B-4 Time for reaching 50% (curve 1) and 90% (curve 2) of the steady state level as a function of surface biofilm coverage (α).
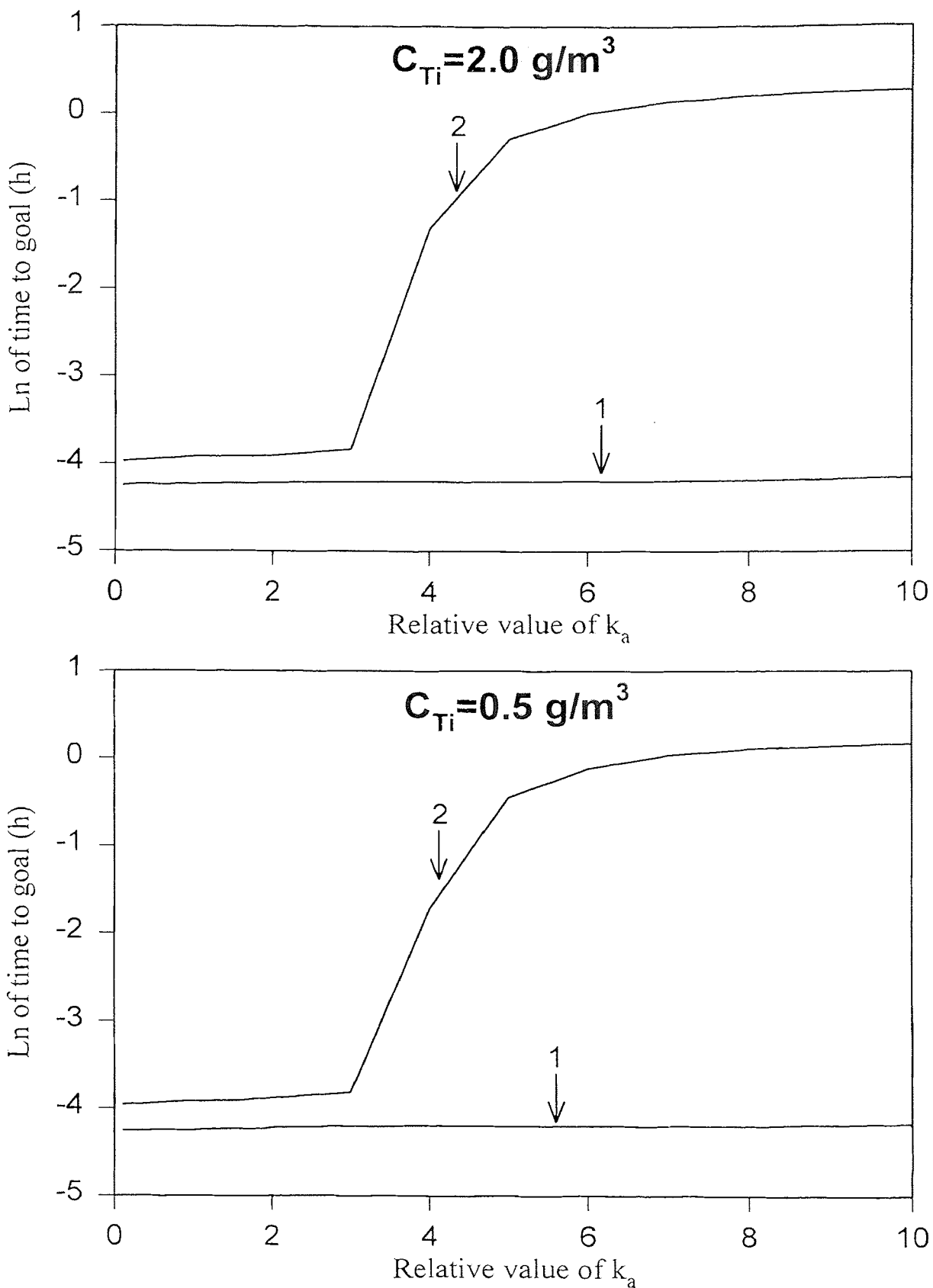
Figure B-5 Time for reaching 50% (curve 1) and 90% (curve 2) of the steady state level as a function of the mass transfer coefficient (k_a).
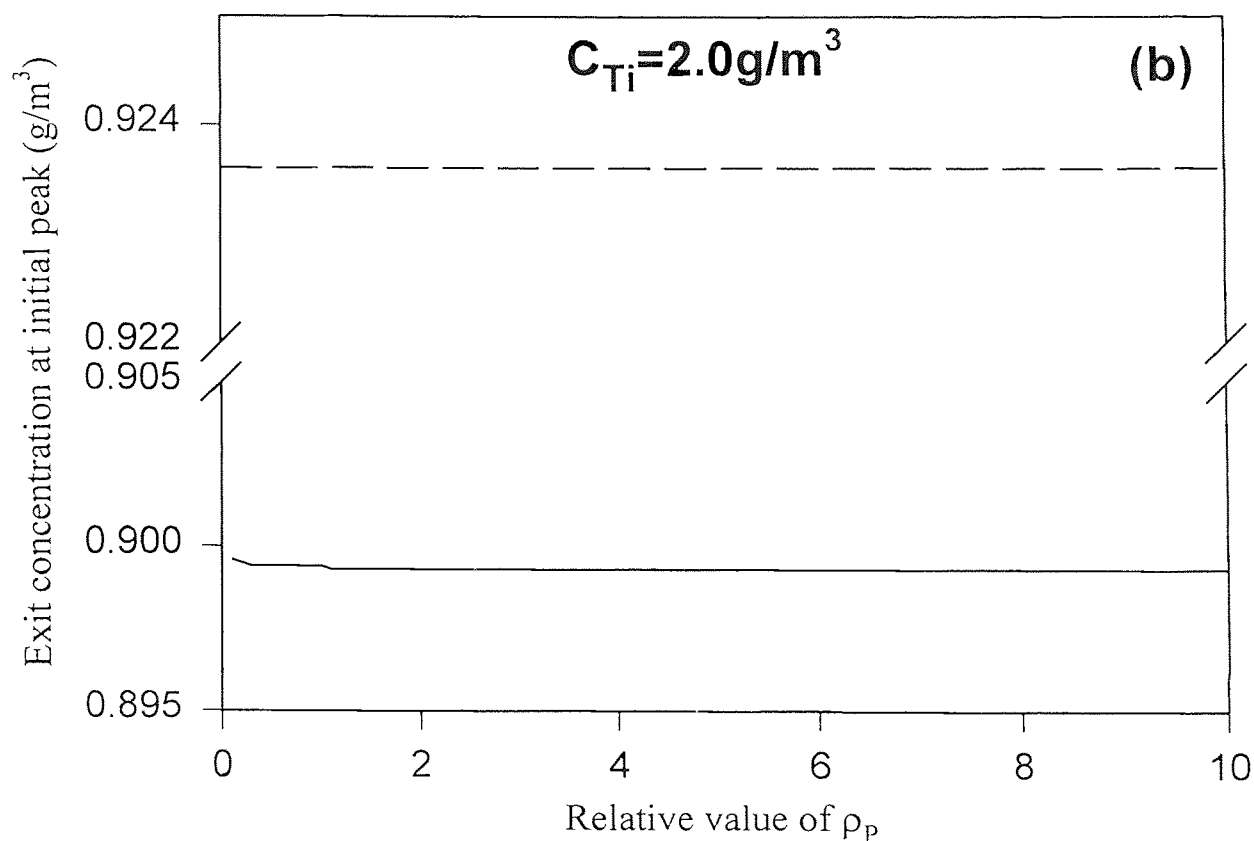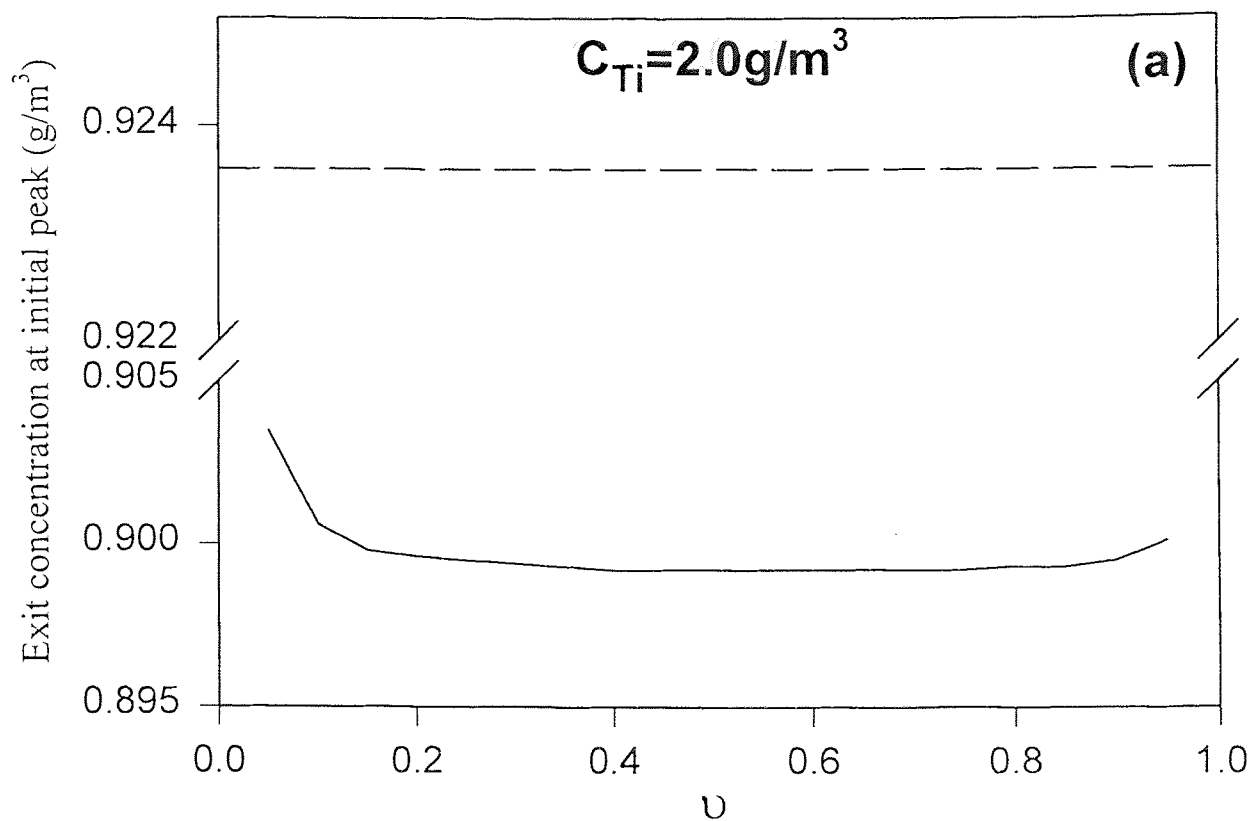
Figure B-6 Peak-VOC concentration in exiting exiting air stream as a function of υ (a) and ρ_p (b). Dashed lines represent steady state exit concentrations.
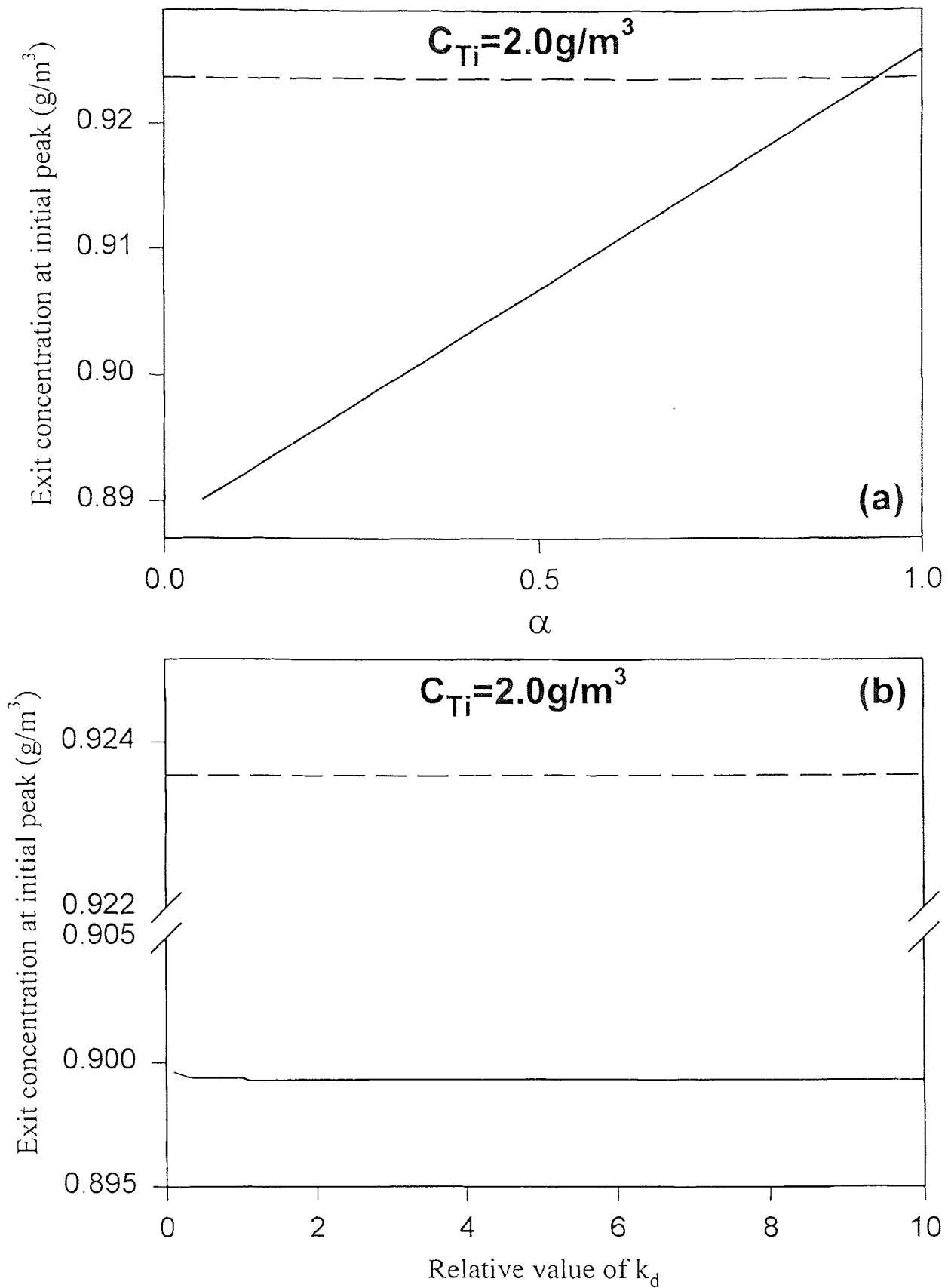
Figure B-7 Peak-VOC concentration in exiting exiting air stream as a function of α (a) and k_d (b). Dashed lines represent steady state exit concentrations.
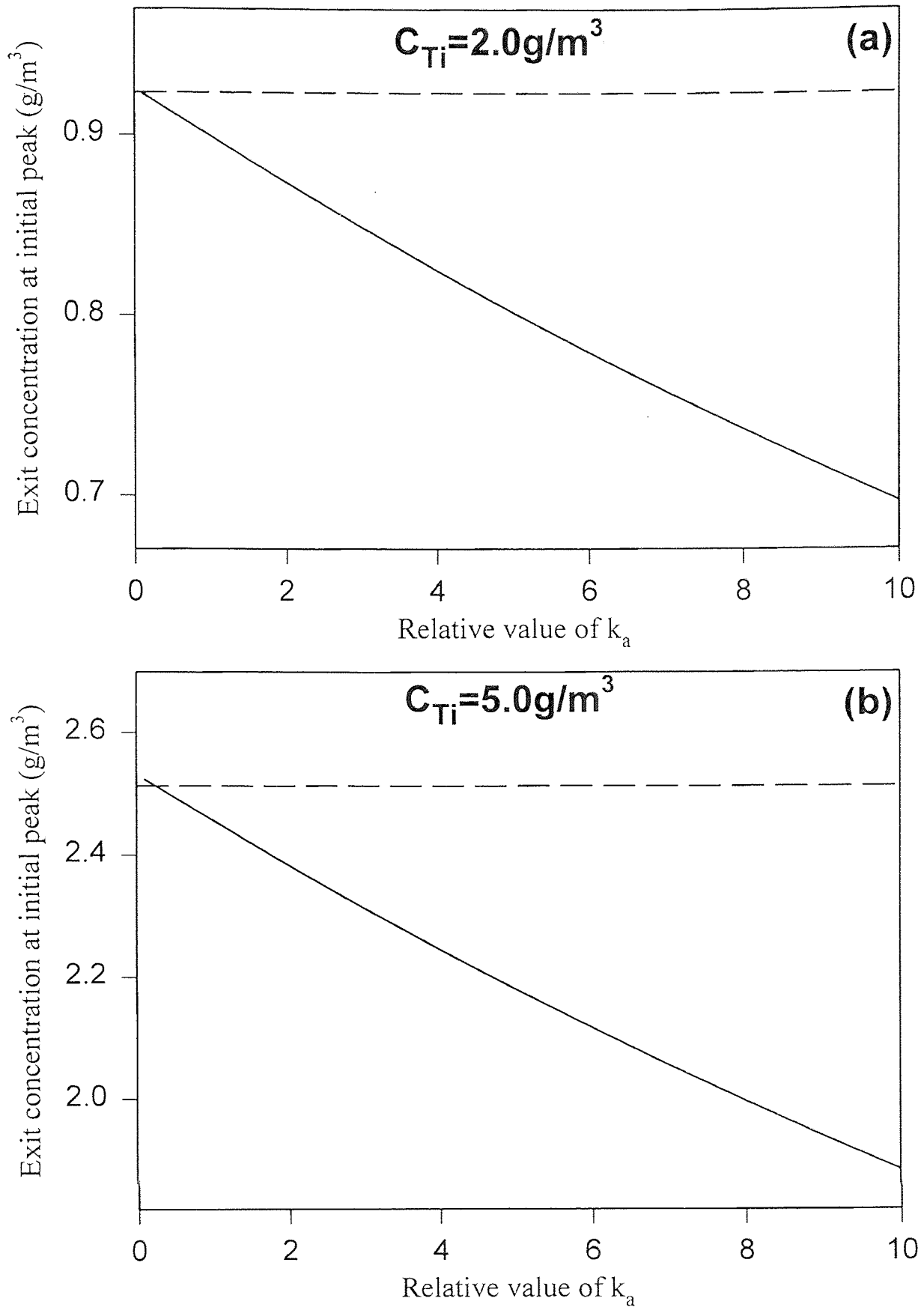
Figure B-8 Peak-VOC concentration in exiting exiting air stream as a function of k_a. Dashed lines represent steady state exit concentrations.
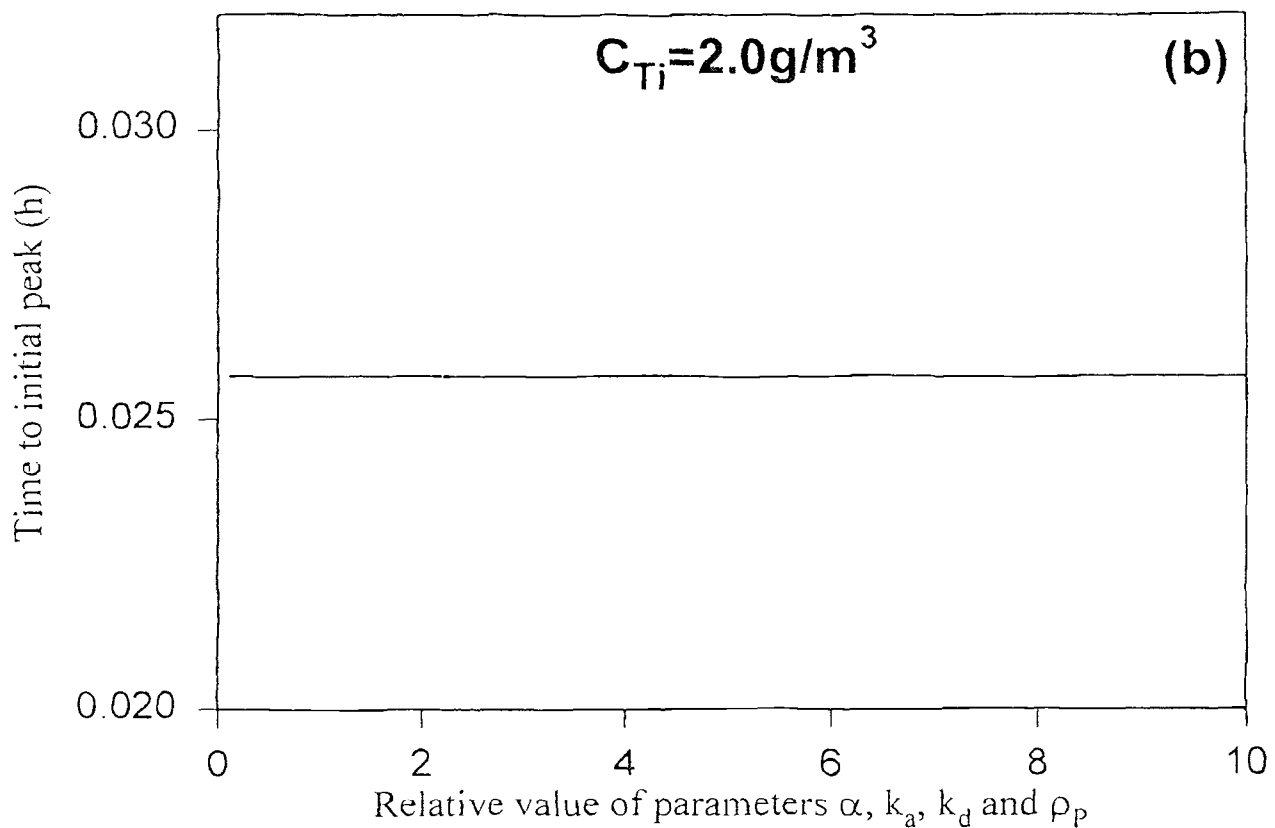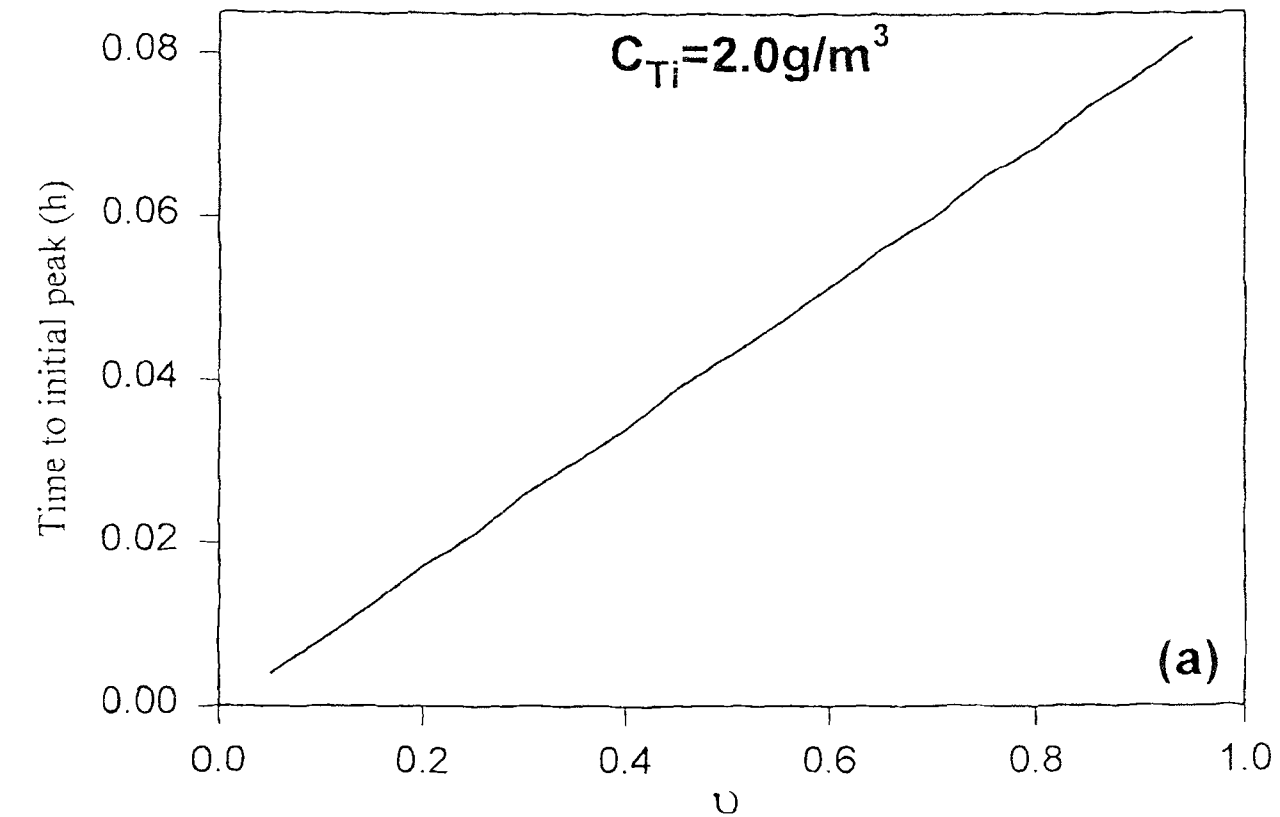
Figure B-9 Time at which the VOC-peak concentration occurs as a function of υ (a) and α, k_a, k_d and $ρ_p$ (b).
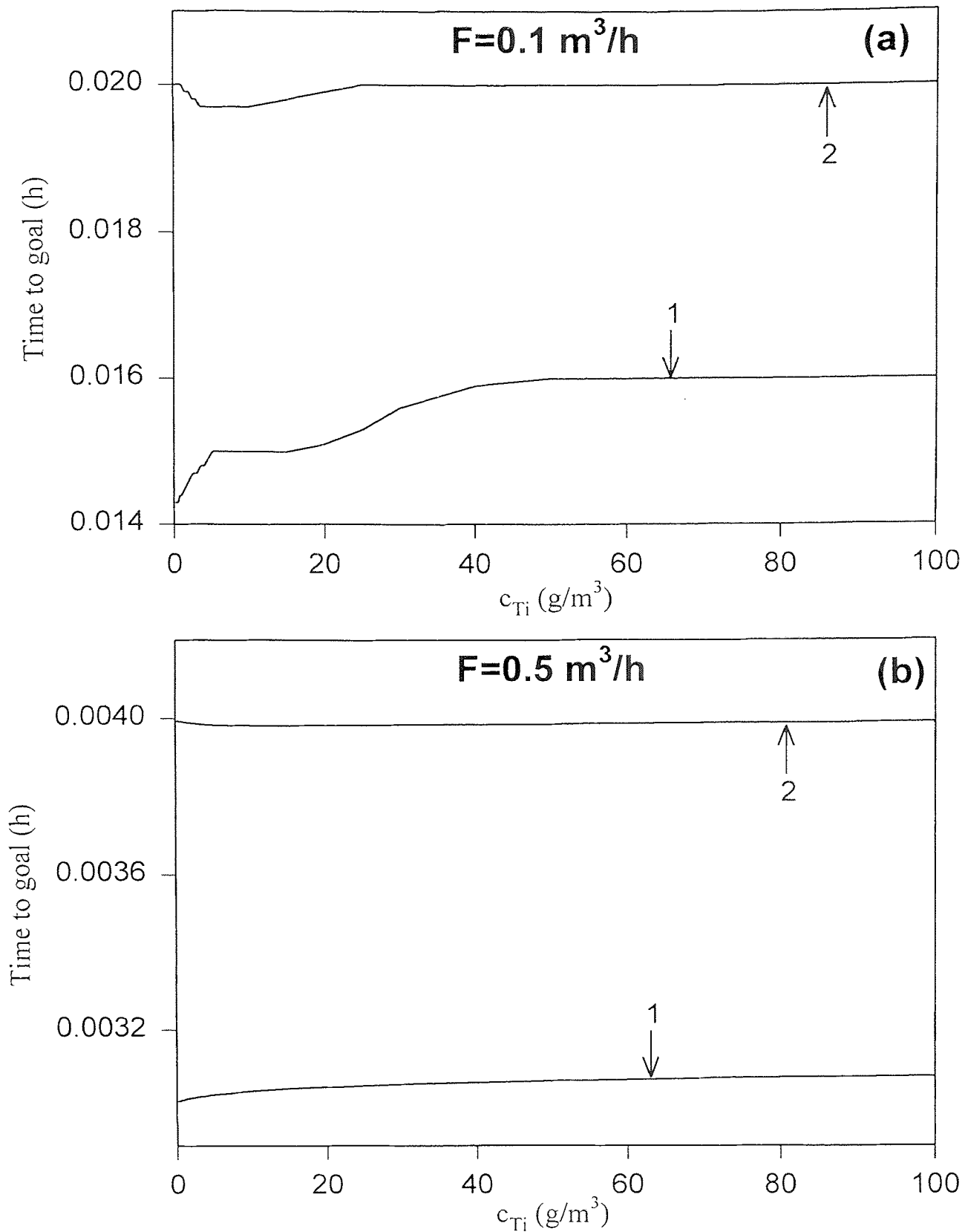
Figure B-10 Time for reaching 50% (curve 1) and 90% (curve 2) of the steady state level of operation as a function of the VOC concentration in the inlet airstream.

APPENDIX C

COMPUTER CODE FOR SOLVING THE STEADY-STATE BIOFILTRATION MODEL FOR A SINGLE VOC

main.f

```
c*****************************************************************
c
c  Purpose :      "Solution Of The Steady-State Biofiltration
c                  Model For Single VOCs"
c
c  Method :       Orthogonal Collocation
c
c  Language :     FORTRAN
c
c  Written By :   Dimitrios Tsangaris, Newark, NJ on January 26,1995
c
c  Updated By :   Michael Cohen, Newark, NJ in May 1995
c
c*****************************************************************
        implicit none
        include "Include/parameters.h"
        real*8 height(ng+1),gasB(ng+1), gasO(ng+1)
        real*8 solcol(2*n),xdat(n+2),Bdat(n+2),Odat(n+2)
        real*8 delz,z
        real*8 deriB,deriO,one,deltainit
        real*8 resinit,foo,step,relative,mBin
        character outfile*80
        integer outrep,rep,maxrep
        integer igas,k
        integer status,iconv
        include "Include/operating.h"
        include "Include/collocation.h"
        include "Include/liquid.h"
        include "Include/gas.h"
        include "Include/interface.h"
        include "Include/system.h"
        include "Include/volumetric.h"
        include "Include/kinetic.h"
        real*8 fcNew
        external fcNew

*       open(6,file='btcolw.out',status='new')
*       Read the program parameters
        one = 1
*       This controls which area is examined during the
*       sensitivity analysis
        step = 0.02
        open (unit=4,file='in', status='old')
        read (4,*) foo
        read (4,*) resinit
        close (4)
*       This gives the initial value of the parameter
*       to be studied, it will be then multiplied by
*       different relative values (usually 0.1-2.0)
        mBin=0.27
        maxrep = 2000
        do outrep=2,20
```

```
              relative=outrep/10.0
              rep = 1
              do while (rep.lt.maxrep)
                      restime=resinit+step*(rep-1)
              mB=mBin*relative
                      if (restime.lt.10) then
                              write(outfile,'(f4.2,1x,f3.1)') restime,relative
                      else
                              write(outfile,'(f5.2,1x,f3.1)') restime,relative
                      endif
                      open (unit=6,file=outfile,status='unknown')
                      call input()
                      call Update()
                      call output()
                      call PrintDim()
                      deltainit = delta
c         Initialize the orthogonal collocation routines
          call InitCollocation()
          status=TRUE
          status=DEBUG
          CALL today
          WRITE (6,101) n
101       Format(' Solution of the Model using Orthogonal Collocation '
    &        ,/,'          with [',i3,'] col. points',/)
c gas
          delz      = 1./float(ng)
          z         = 0.0
          height(1)= z
          gasB(1)  = cgasB
          gasO(1)  = cgasO
*START THE LOOP OVER Z AXIS
          do 100 igas=2,ng+1
                  iconv = FALSE
          write(6,123)
          write(6,102)z+delz
102               format(' Height =',5x, f14.3)
                  delta = deltainit
6         call Update()
                  if (status.eq.DEBUG) call PrintDim()
                  call InitProfile(solcol)
C                 CALCULATE LIQUID PHASE CONCENTRATION
                  call newton(status,solcol)
                  if (status.eq.DEBUG) call PrintSolution(solcol,root)
                  call interpolate(status,solcol,nt,root,dif1,xdat,Bdat,Odat)
                  call CheckConvergance(iconv,Bdat,Odat)
                  if (iconv.eq.TRUE) then
                  call interpolate(THETACONV,solcol,nt,root, dif1,
    &                             xdat,Bdat,Odat)
c                 CALCULATE GAS PHASE CONCENTRATION
          call deri (solcol, deriB, deriO)
                  CALL RK4(eta,one,deriB,delz,cgasB)
                  z = z + delz
                  height(igas) = z
                  cgasO = fcNew(cgasB)
```

```
                              gasB(igas) = cgasB
                              gasO(igas) = cgasO
                              write(6,*)
                              write(6,'(a)')'    Gas phase concentration so far'
                              do k=1,igas
                                      write(6,'(4(f12.6,3x))')
      &                               height(k),gasB(k),gasO(k)
                              enddo
                   else
                              goto 6
                   endif
100      continue
*END OF LOOP OVER Z AXIS
         call Results(height,gasB,gasO)
         close(6)
123      FORMAT(/,'==================================================================',/)
         if (gasB(ng+1).gt.(0.30)) then
                   rep=rep+19
         else
                   if (gasB(ng+1).ge.(0.10)) rep=rep+9
                   if (gasB(ng+1).ge.(0.070).and.gasB(ng+1).lt.(0.10)) rep=rep+4
         endif
         if (gasB(ng+1).ge.(0.020).and.gasB(ng+1).lt.(0.049)) rep=rep+9
         if (gasB(ng+1).ge.(0.015).and.gasB(ng+1).lt.(0.020)) rep=rep+4
         rep=rep+1
         if (gasB(ng+1).le.(.01)) rep=2001
                   end do
         end do
         stop
         end




                                      io.f

*****************************************************************************
*
*       This subroutine reads the Kinetic constants and Operating
*       parameters only
*
*****************************************************************************

         subroutine input()

         implicit none
         include "Include/parameters.h"
         include "Include/liquid.h"
         include "Include/interface.h"
         include "Include/gas.h"
         include "Include/collocation.h"
         include "Include/operating.h"
         include "Include/volumetric.h"
         include "Include/kinetic.h"
         include "Include/system.h"
```

```
        character filename*80
        real*8 foo
        integer status,unfn
        unfn=7
*ResTime must be entered in Minutes
*cgB00 must be entered in g/m3
        open (unit=5,file='in',status='old')
        read(5,*) cgb00,foo
        read(5,'(a)')filename
        close (5)
        open (unit=unfn,file=filename,status='old')
        read(unfn,*)
        read(unfn,*) foo
        read(unfn,*)
        read(unfn,*) foo
        read(unfn,*)
        read(unfn,*) delta
        read(unfn,*) foo
        read(unfn,*)
*       Now read from the standard input the operating conditions
*       System parameters
        itmax  = 100
        iprnewton = TRUE
        iprnewton = FALSE
        eps1  = 1.e-9
        eps2  = 1.e-9
*       Collocation parameters
*       n0 :1 if 0 is included in the collocation interval
*       n1 :1 if 1 is included in the collocation interval
*       alpha and beta are the parameters for the Gauss Trial functions
        n0 = 1
        n1 = 1
        nt = n0+n1+n
        alpha = 0.
        beta  = 0.
*       Biofilm parameter ( in kg/m3)
        read(unfn,*)b0
*       Kinetic constants for Ethanol and Butanol
        read(unfn,*) miouB
        miouB =  miouB/3600
        read(unfn,*) KB
        read(unfn,*) KBI
        read(unfn,*) KO
*       Diffucivities
        read(unfn,*) DBW
        read(unfn,*) DOW
*       Yield coefficients
        read(unfn,*) YB
        read(unfn,*) YOB
*       Henry's constants
*       This parameter (mB) is not read in
*       here, as it is the one being tested
c       read(unfn,*) mB
        read(unfn,*) foo
```

```
          read(unfn,*) mO
*         Entrance concentrations
          cgB00   = cgB00 * 1.e-3
          read(unfn,*) cgo00
          cgo00   = cgO00 * 1.e-3
*         Volumetric properties
*         Delta belongs here but it is more convenient to put it at the top
*         of the input file since it is Inlet conditions dependent
          read(unfn,*) As
          read(unfn,*) Volume
          read(unfn,*) Surface
          restime = restime*60
* Initialize  the concentrations of B,E,O at zero theta
          cgasB =  1.0
          cgasO =  1.0
* Now calculate the delta dependent values: eta,phiB2,phiE2
          rewind(unfn)
c         This Update is now called in the main program
c         call Update()
          return
          end


**************************************************************************
*         This subroutine calculates some dimensionless units that depend on
*         the parameter 'delta'. Delta, is the depth of the biofilm
**************************************************************************

          subroutine Update()
          implicit none
          include "Include/parameters.h"
          include "Include/volumetric.h"
          include "Include/kinetic.h"
          include "Include/operating.h"
          include "Include/interface.h"
          include "Include/gas.h"
          include "Include/liquid.h"
          real*8 xv,fd
          real*8 deltaMt

* Now calculate some Dimensionless quantities using the above values
* This section (10 lines) was in Input, but has been moved to Update
          lamdaB = DBW*KB*YB/(DOW*Ko*YOB)
          gamaB  = KB/KBI
          omegaB = (DOW*KO*cgB00)/(DBW*KB*cgO00)
          epsilnB = cgB00/(mB*KB)
          epsilnO = cgO00/(mO*KO)
          deltaMt = delta*1.e-6
          xv = b0
          fd = 1-0.43*xv**0.92/(11.19+0.27*xv**0.99)
          eta = As*DBW*fd*restime*KB/(deltaMt*cgB00)
          phiB2 = xv*deltaMt*deltaMt*miouB/(fd*DBW*KB*YB)
          return
          end
```

```
*************************************************************************
*          This subroutine Prints out the Updated values of the dimensionless units
*          that depend on delta
*************************************************************************
          subroutine PrintDim()
          implicit none
          include "Include/liquid.h"
          include "Include/interface.h"
          include "Include/gas.h"
          include "Include/operating.h"

          WRITE(6,123)
          WRITE(6,1)
   1      FORMAT (' ',' Dimensionless Parameters :', /)
          write(6,52)delta,eta
   52     format (3x,'delta    = ',f12.3,
     &    3x,'eta      = ',f12.3)
          WRITE(6,2) phiB2,lamdaB
   2      FORMAT (3x,'phiB^2   = ',e12.6,
     &    3x,'lamda B  = ',e12.6)
          write(6,*)
          WRITE(6,3) epsilnB,epsilnO
   3      FORMAT (3x,'Epsilon B = ',f12.6,
     &             3x,'Epsilon O = ',f12.6)
          write(6,*)
          WRITE(6,5) omegaB,gamaB
   5      FORMAT (3x,'omega B  = ',e12.6,
     &    3x,'gamma B  = ',e12.6)
          write(6,*)
          write(6,*)
  123     format('_____',/)
          return
          end


c************************************************************************
*          Print the variables
c************************************************************************
          subroutine output ()

          implicit none
          include "Include/parameters.h"
          include "Include/liquid.h"
          include "Include/gas.h"
          include "Include/interface.h"
          include "Include/operating.h"
          include "Include/kinetic.h"
          include "Include/volumetric.h"
c
          write(6,123)
          WRITE(6,1)
   1      FORMAT (' ',//,' VARIABLES IN THE MODEL',//)
          WRITE(6,2)
   2      FORMAT (3x,'1 - Butanol',/,3x,'2 - Ethanol',/,3x,'3 - Oxygen',/)
```

```
        WRITE(6,19) restime/60
19      format (' ', 'Residence Time (min)      = ', f12.3)
        WRITE(6,3) volume*1e6
3       FORMAT (' ', 'Volume of the column(cm3)   = ', f12.3)
        WRITE(6,4) As
4       FORMAT (' ', 'Biolayer Sur.Area( m2/m3)   = ', f12.3)
        write(6,44) b0
44      format (' ', 'Biomass Conc. (kg/m3)      = ', f12.3)
        WRITE(6,5) delta*1e-3
5       FORMAT (' ', 'Film thickness (mm)        = ', f12.3)
        WRITE(6,18) CGB00*1000.
        WRITE(6,22) CGO00*1000.
18      FORMAT (' ', 'Inlet conc. (g/m3 of air)(B) = ', f12.3)
22      FORMAT (' ', 'Inlet conc. (g/m3 of air)(O) = ', f12.3)
        write(6,31) YB
31      format (' ', 'Yield Coefficient (B)      = ', f12.3)
        write(6,34) YOB
34      format (' ', 'Yield Coefficient (OB)     = ', f12.3)
        WRITE(6,51) DBW*1.e+9
        WRITE(6,55) DOW*1.e+9
51      format (' ', 'Diff. Coeff. (B)*1e9(m2/s)   = ', f12.3)
55      format (' ', 'Diff. Coeff. (O)*1e9(m2/s)   = ', f12.3)
        WRITE(6,565) mB
565     FORMAT (' ', 'Dist. Coeff.    (B)    = ', e12.3)
        WRITE(6,567) mO
567     FORMAT (' ', 'Dist. Coeff.    (O)    = ', e12.3)
        write(6,123)
        write(6,*) '        Andrews and other  Parameters'
        WRITE(6,6)
     &           miouB*3600,KB*1000,KBI*1000,Ko*1000
6       format (' ',/,
     &           ' miou B(1/hr) = ',f12.3,/,
     &              ' KB   (g/m3) = ',f12.3,/,
     &           ' KBI  (g/m3) = ',f12.3,/,
     &           ' KO   (g/m3) = ',f12.3 )
123     FORMAT('_____',/)
        return
        end
**********************************************************************
*
        subroutine today
*       EXTERNAL   TDATE
*       CALL TDATE (IDAY, MONTH, IYEAR)
        write(6,123)
        WRITE (6,66) month,iday,iyear
66      Format( 3x, ' Date :   ',i2,'/',i2,'/',I4,//,
     &   '     Model Predictions for Ethanol-Butanol Mixture',/,
     &   '        by Orthogonal Collocation Method      ',/,
     &   '        written by Dimitrios Tsangaris        ',/,
     &   '   ===========================================' //)
123     FORMAT('_____',/)
        return
        end
```

```fortran
      subroutine Results(height,gasB,gasO)
      implicit none
      include "Include/parameters.h"
      include "Include/liquid.h"
      include "Include/operating.h"
      include "Include/volumetric.h"
      real*8 height(1),gasB(1),gasE(1),gasO(1)
      real*8 removal(2)
      integer igas

      write(6,123)
      WRITE(6,22)
22    format(//,5x,'      Gas Phase Concentration Profile',//)
      WRITE(6,13)
13    FORMAT (' ',12x, 'Height',10x,'Cg(B)',10x,'Cg(O)',10x,'1-Cg(B)'/)
      do 44 igas=1,ng+1
              write(6,33) height(igas), gasB(igas), gasO(igas) ,1.- gasB(igas)
44    continue
33    format(4x,F14.6,2x,F14.6,1x,f14.6,1x,f14.6,1x,f14.6)
      WRITE(6,25)
25    format(//,5x,'      Gas Phase Concentration Profile[g/m3]',//)
      WRITE(6,15)
15    FORMAT (' ',12x, 'Height',10x,'Cg(B)',10x,'Cg(O)',10x,'Cg(-)'/)
      do 45 igas=1,ng+1
              write(6,34) height(igas),
     +                 gasB(igas)*cgB00*1000.,
     +                 gasO(igas)*cgO00*1000.
45    continue
      removal(1) = (3600/restime) * (gasB(1) - gasB(ng+1))*(cgB00*1000.)
      removal(2) = (3600/restime) * (gasO(1) - gasO(ng+1))*(cgB00*1000.)
      write(6,99) removal(1),removal(2)
99    format(/,' Removal rates for Butanol[g/m3*hr] ',f12.4,/,
     &        '                Oxygen [g/m3*hr] ',f12.4)
34    format(4x,F14.6,2x,F14.6,1x,f14.6,1x,f14.6)
      write(6,123)
123   FORMAT(' _____  _____',/)
      return
      end
```

init.f

****

```fortran
      subroutine InitCollocation()

      implicit none
      include "Include/parameters.h"
      include "Include/system.h"
      include "Include/collocation.h"
      integer i,j

c     ----calculate the collocation point----
```

```
            call jcobi(nt,n,n0,n1,alpha,beta,dif1,dif2,dif3,root)
c   ----calculate the discretization matrices a & b----
            do i=1,nt
                    call dfopr(nt,n,n0,n1,i,1,dif1,dif2,dif3,root,v)
                    do j=1,nt
                            a(i-1,j-1)=v(j)
                    enddo
                    call dfopr(nt,n,n0,n1,i,2,dif1,dif2,dif3,root,v)
                    do j=1,nt
                            b(i-1,j-1)=v(j)
                    enddo
            enddo
            return
            end
*************************************************************************
*
            subroutine InitProfile(xold)
            implicit none
            include "Include/parameters.h"
            real*8 xold(1)
            integer i,nlocal
            nlocal = n
            do i=1,nlocal
                    xold(i) = .1
            enddo
            return
            end
```

newton.f

```
C************************************************************************
c   purpose :       NEWTON RAPHSON to solve the system of non-linear algebraic equations
C                   There are nsize equations, where nsize is 2*n
C                   because every concentration has n unknowns-equations
c
c************************************************************************

            subroutine newton(status,xold)
            implicit none
            include "Include/parameters.h"
            real*8 xold(1)
            include "Include/collocation.h"
            include "Include/system.h"
            include "Include/operating.h"
*
*           Define new nmatrices for the Newton method
            real*8 xinc(2*n),jac(2*n,2*n+1)
            integer iter,n2,indic,i,itcon,status
            real*8 deter
            real*8 simul1
            external simul1
            n2 = n
```

```
            if ((status.eq.DEBUG).or.(status.eq.TRUE)) then
            write(6,123)
123         FORMAT(' --------------NEWTON ITERATION-------------------',/)
            endif
            if(iprnewton.eq.TRUE) then
                    write(6,202) iter,deter,n2,(xold(i),i=1,n2)
            endif
            do 9 iter=1,itmax
            call Model(xold,jac,n2)
c                   call PrintA(jac,n)
c                   Simul computes the Jacobian and the Correction DX in xinc
                    indic = 1
                    deter = simul1 (n2,jac,xinc,eps1,indic,n2+1)
                    if (deter.eq.0.0) then
                            write(6,201)
                            return
                    endif
c                   check for convergence and update xold value
                    itcon=TRUE
                    do 5 i=1,n2
                            if(dabs(xinc(i)).gt.eps2) itcon=FALSE
                            xold(i)=xold(i)+xinc(i)
5                   continue
                    if(iprnewton.eq.TRUE) then
                            write(6,202) iter,deter,n2,(xold(i),i=1,n2)
                    endif
                    if (itcon.ne.FALSE) then
                            if ((status.eq.TRUE).or.(status.eq.DEBUG)) then
                                    write(6,203)iter,delta
                            endif
                            return
                    endif
9           continue
            write(6,204)
            return
C           formats for input and output statements
200         format(' itmax  =',i8,/' iprint =',i8/' n     =',i8/
    &' eps1   =',1pe14.1/' eps2   =',1pe14.1/10x,'xold(1)...xold(',
    & i2,')'///(1h ,1p4e16.6))
201         format(38h0matrix is ill-conditioned or singular)
202         format(' iter  =',i8/ 10h deter  =,e18.5/
    $  26h       xold(1)...xold(,i2,1h) / (1h ,1p4e16.6) )
203         format(' Successful convergence: Iteration=',i5,
    &  ' Delta=',f8.2,/)
204         format(' no convergence' )
            end
```

interpolate.f

```
*************************************************************
*  purpose :     interpolating the results that you get from
*                newton raphson subroutine
*                The xold contains the solution of sB,sO at positions
*                between 1-->n-->2n.
*                Those values are unpacked to Bsol,Osol and later
*                used to obtain the values at xdat->Bdat,Odat
*
*************************************************************
        subroutine interpolate(iflag,xold,nt,root, dif1,xdat,Bdat,Odat)

        implicit none
        integer iflag
        include "Include/parameters.h"
        real*8 xold(1),root(1),dif1(1)
        real*8 xdat(1),Bdat(1),Odat(1)
        real*8 xintp(n+2),Bsol(n+2),Osol(n+2)
        real*8 so,sb,dist
        integer i,j,nt
        include "Include/operating.h"
        include "Include/interface.h"
        real*8 sb0,so0
        real*8 fsNew
        external fsNew

        if ((iflag.eq.DEBUG).or.(iflag.eq.THETACONV)) then
        write(6,123)
                WRITE(6,12)
12      FORMAT (
   &         ' ',10x, 'Concentration Profiles in the Biofilm ', /)
        WRITE(6,13)
13      FORMAT (
   &             5x,' x ',6x,'s(B)',10x,'s(O)',10x,'s(E)'/)
        endif
        sb0 = epsilnB*cgasB
        Bsol(1)=sb0
        do i=1,n
                Bsol(i+1)=xold(i)
        enddo
        Bsol(n+2)=Bsol(n+1)
        do 20 i=1,n+1
                dist = float(i-1)/n
                call intrp(nt,nt,dist,root,dif1,xintp)
                sb=0.0
                do 30 j=1,n+2
                        sb = sb+xintp(j)*Bsol(j)
30              continue
                so = fsNew(sb)
                if ((iflag.eq.DEBUG).or.(iflag.eq.THETACONV)) then
                        write(6,40) dist,sb,so
                endif
                xdat(i)=dist
```

```
               Bdat(i)=sb
               Odat(i)=so
20       continue
40       format(2x,f7.2,2x,
   &     f12.6,2x,f12.6,2x,f12.6,2x,
   &     f14.6,2x,f12.6)
123      FORMAT(' ++++++++++++++++++++++++++++++++++++++++++++++++++++++++',/)
         return
         end
c**************************************************************
c
c    Subroutine for evaluating the derivative
c    necessary for gas phase profiles
c
c**************************************************************
         subroutine deri (xold, deriB, deriO)

         implicit none
         include "Include/parameters.h"
         include "Include/liquid.h"
         include "Include/interface.h"
         include "Include/gas.h"
         include "Include/collocation.h"
         include "Include/operating.h"
         real*8 deriB,deriO,sb0,so0,sb,so
         real*8 xold(1)
         real*8 sum1,sum2
         integer j
         sum1 = 0.0
         sum2 = 0.0
         do 10 j =1,n
               sb = xold(j)
               sum1 = sum1+(a(0,j)-a(0,n+1)*a(n+1,j)/a(n+1,n+1))*sb
10       continue
         sb0 = epsilnB*cgasB
         deriB = sum1+(a(0,0)-a(0,n+1)*a(n+1,0)/a(n+1,n+1))*sb0
         return
         end
c**************************************************************
c        for gas phase
c        using the fourth order runge kutta method
cc*************************************************************
         SUBROUTINE RK4(eta,omega,deri,H,cg)

         implicit none
         real*8 eta,omega,deri,cg,H,FUN
         external FUN
         cg= cg+H*FUN(eta,omega,deri)
         RETURN
         END
c**************************************************************
c  purpose :     give the function for RK method, in the gas phase
c                balance
c**************************************************************
```

```fortran
      real*8 FUNCTION Fun(eta,omega,deri)
      implicit none
      real*8 eta,omega,deri
      Fun = eta*omega*deri
      RETURN
      END
```

check.f

```fortran
      subroutine CheckConvergance(status,Bdat,Odat)

      implicit none
      include "Include/parameters.h"
      include "Include/interface.h"
      include "Include/operating.h"
      integer status
      real*8 Bdat(1),Odat(1)
      real*8 uplmB,uplmO
      real*8 sbf,sof
      real*8 fsnew
      external fsnew

* *** calculate the concentrations at the end of the biofilm (theta=1)
      sBf  = Bdat(n+1)
      sOf  = Odat(n+1)
      sof  = fsNew(sbf)
      if (sof.ne.Odat(n+1)) then
      print*,'Warning: Odat contains different value for SB then SO predicts'
      print*,sof,odat(n+1)
      endif
      uplmB = epsilnB*cgasB*PERCENTAGE
      uplmO = epsilnO*cgasO*PERCENTAGE
      if (sof.gt.0.0.and.sof.le.uplmO)then
              status = TRUE
      elseif ((sbf.gt.0.0.and.sbf.le. uplmB).or.
     &    (sof.gt.0.0.and.sof.le. uplmO)) then
              status = TRUE
      elseif(delta.lt.300)then
              delta = delta + 2.0
              status = FALSE
      elseif(delta.ge.300)then
      delta = 300
              status = FALSE
      endif
      return
      end
```

printsol.f

```
subroutine PrintSolution(xold,root)

implicit none
include "Include/parameters.h"
real*8 xold(1),root(1)
integer i
write(6,'(3a12)')' theta',' S (B)',' S (O)'
do i=1,n+1
          write(6,'(3f14.8)') root(i+1),xold(i)
enddo
return
end
subroutine PrintA(a,m)
implicit none
real*8 a(20,1)
integer m,i,j
do i=1,m
do j=1,m
          if (a(i,j).ne.0.0) then
          write(*,'(2i5,f14.8)') i,j,a(i,j)
          endif
enddo
enddo
do i=1,m
          write(*,'(2i5,f14.8)') i,0,a(i,m+1)
enddo
return
end
```

model.f

```
c************************************************************
c   purpose :     construct the jacobian matrix and on the last
c                 column vector -f
c                 df contains the jacobian  at (nsize,nsize)
c                 and the -F vector at (nsize,nsize+1)
C                 notice that nsize = n
c************************************************************
        subroutine Model(x,df,nsize)
        implicit none
        include "Include/parameters.h"
        real*8 x(nsize),df(nsize,n+1)
        integer nsize
        include "Include/liquid.h"
        include "Include/interface.h"
        include "Include/collocation.h"
        include "Include/operating.h"
        real*8 sB,sO,sB0,so0
        real*8 kinetic10,kinetic11
        real*8 fk1
```

```
      real*8 sum,q1,q2,q3
      real*8 fsNew

      external fsNew
      integer i,j,m
*Reset The jacobian matrix
      do i=1,nsize
            do j=1,nsize+1
                  df(i,j)=0.0
            enddo
      enddo
*Those are the concentrations at theta=0
      sb0= epsilnB*cgasB
      so0= epsilnO*cgasO
      m = nsize
      do 10 i=1,m
            sum = 0.
            do 20 j=1,m
                  sB = x(j)
                  sO = fsNew(sb)
                  df(i,j) = b(i,j)-b(i,m+1)/a(m+1,m+1)*a(m+1,j)
                  if (i.eq.j) then
                        q1 = 1. + sB + gamaB*sB*sB
                        q2 = 1. + sO
                        q3 = 1. - gamaB*sB*sB
                        kinetic10 = phiB2*sO/q2*sB/q1
                        kinetic11 = phiB2*((1/q2/q2)*(sb/q1)*lamdaB+(sO/q2)*(q3/q1/q1))
                        df(i,j)  = df(i,j) - kinetic11
                  endif
                  sum = sum + (b(i,j)-b(i,m+1)/a(m+1,m+1)*a(m+1,j))*sb
20          continue
*           This is the value of F{k}
            fk1=(b(i,0) - b(i,m+1)*a(m+1,0)/a(m+1,m+1))*sb0
            df(i,(nsize+1))=-(fk1 + sum - kinetic10)
10    continue
      return
      end
c**************************************************************
c     Those subroutines calculate the concentration of fsE,fcE
c     given the concentrations of the other components
c**************************************************************
      real*8 function fsNew(sb)
      implicit none
      include "Include/parameters.h"
      include "Include/liquid.h"
      include "Include/gas.h"
      include "Include/interface.h"
      include "Include/operating.h"
      real*8 sb
      real*8 cO,cB

      cO = cgasO
      cB = cgasB
*Now solve  for SB given CO,CB and SO
```

```
        fsNew =  epsilnO*Co + lamdaB*(sb-epsilnB*Cb)
        return
        end
        real*8 function fcNew(cb)
        implicit none
        include "Include/parameters.h"
        include "Include/liquid.h"
        include "Include/gas.h"
        include "Include/interface.h"
        real*8 sb,so
        real*8 cb
*Now solve  for SB given CO,CB and SO
        fcNew = (cb-1.)*lamdaB*omegaB + 1.
        return
        end
```

```
SRC=main.f Init.f Check.f Model.f PrintSol.:
        interpolate.f
OBJ=main.o Init.o Check.o Model.o PrintSol.o io.o newton.o\
        interpolate.o
LIB=~mike/lib/orthcol.o
OPT= -extend_source
.SUFFIXES: .o .f
.f.o: ; f77 -c $(OPT) $*.f -o $*.o
single: $(OBJ)
        f77 $(OBJ) $(LIB) -o ssicol
clean:
        rm -f *.o
```

in

1.65
7.7
Toluene.in

Toluene.in

```
*****************************************
0.00    Toluene concentration (g/m3)
*****************************************
0.00 0  Residence time (in min) or Flowrate (m3/h)
*****************************************
30      This is used by the old version
1       Initial guess for delta (in micrometers)
*****************************************
100     B0      [Kg]
1.50    Miou i  [1/h]
11.03e-3 Ki     [Kg/m3]
78.94e-3 Kil    [Kg/m3]
0.26e-3  KO     [Kg/m3]
```

1.03e-9 DiW   [m2/s]
2.41e-9 DOW   [m2/s]
0.708   Yi   [Kg/Kg]
0.341   YOi   [Kg/Kg]
0.27   mi   [-]
34.4   mO   [-]
275   C[o]   [g/m3]
40   As   [m-1]
15291e-6 Volume [m3]
1.82e-2 Surface [m^2]

Include

parameters.h

```
integer n,ng,ndata
parameter (n=20,ng=20,ndata=20)
real*8 PERCENTAGE
parameter (PERCENTAGE=0.01)
integer TRUE,FALSE
parameter (TRUE=1,FALSE=0)
integer RESITENCE
parameter (RESITENCE = 10)
integer LASTZ,MIDDLEZ
parameter (LASTZ = 20,MIDDLEZ=30)
integer THETACONV
parameter (THETACONV = 10)
integer DEBUG
parameter (DEBUG = 100)
```

operating.h

```
real*8 cgB00,cgo00
common /cgas/cgB00,cgo00
real*8 delta
common/del/delta
real*8 cgasB,cgasO
common /cgas/cgasB,cgasO
```

collocation.h

```
real*8 a(0:n+1,0:n+1),b(0:n+1,0:n+1)
common /colloc/ a,b
real*8 root(n+2)
real*8 dif1(n+2),dif2(n+2),dif3(n+2)
real*8 v(n+2)
common /colloc1/ root,dif1,dif2,dif3,v
```

liquid.h

```
real*8 lamdaB
real*8 gamaB
common /lamda/lamdaB
```

```
common /gama/gamaB
real*8 phiB2
common /phi/phiB2
real*8 sigmaB
common /sigma/sigmaB
```

gas.h

```
real*8 omegaB,eta
common /omega/eta,omegaB
```

interface.h

```
real*8 epsilnB,epsilnO
common /epsiln/epsilnB,epsilnO
```

system.h

```
real*8 eps1,eps2
common/sysm/eps1,eps2
integer itmax,iprnewton
common/flowcontrol/itmax,iprnewton
integer n0,n1,nt
real*8 alpha,beta
common/initll/n0,n1,nt
common/expon/alpha,beta
```

volumetric.h

```
real*8 volume,surface,volrate,restime
real*8 as
common/volumetric/volume,surface,volrate,restime
common/volumetric1/as
```

kinetic.h

```
real*8 KB,KBl,KO,miouB
real*8 DOW,DBW
real*8 YB,YOB
real*8 b0
common/kinetic1/KB,KBl,KO,miouB
common/kinetic2/DOW,DBW
common/kinetic3/YB,YOB
common/kinetic4/b0
real*8 mB,mO
common/henry/mB,mO
```

APPENDIX D

COMPUTER CODE FOR SOLVING THE TRANSIENT BIOFILTRATION MODEL FOR A SINGLE VOC

main.f

```
**********************************************************
c
c  Purpose :     "Solution Of The Transient Biofiltration
c                 Model For A Single VOC"
c
c
c  Method :      ODESSA-Ordinary Differential Equation
c                Solver With Explicit Sensitivity Analysis;
c                Stiff Mode When User Supplied Jacobian
c                Option Is Used
c
c
c  Language :    FORTRAN
c
c
c  Written By :  Dimitrios Tsangaris, Newark, NJ on March 28,1995
c
c
c  Updated By :  Michael Cohen, Newark, NJ in August 1995
c
c**********************************************************
          implicit none
          include "Include/parameters.h"
          external fun,dfun,jfun
          include "Include/odessa.h"
          real*8 cg(0:nhmax,0:ntmax)
          real*8 co(0:nhmax,0:ntmax)
          real*8 cp(0:nhmax,0:ntmax)
          real*8 time(0:ntmax), ht(0:nhmax)
          include "Include/operation.h"
          include "Include/dimensional.h"
          include "Include/dimensionless.h"
          real*8 dt,t,tout,err,tau,avcgb
          integer istate,istatus,i
          integer ndim,npar,nt,nh,it,ih,tlast
          integer sens
*
*         Read the system parameters, and initialize the concentrations
          do sens=1,10
          call Reset(rwork,iwork,itask,istate,iopt,mf)
          call today()
*         This gives the original value of the parameter to be
*         examined, and then checks its relative values
*         (usually 0.1 to 10.0)
          kapaA=6.04e-3
          kapaA=kapaA*sens
          call ReadParam(istatus,ndim,npar,nh,nt,dt,err,tau)
          call InitConditions(istatus,nh,nt,cg,co,cp,ht,dt,time,y)
          call InitOdessa(neq,ndim,npar,
     +              iopt,itask,lrw,liw,mf,itol,rtol,atol,err)
          call PrintDimensional()
          call PrintDimensionless()
          call PrintOne(cg,co,cp,time,ht,nh,zero)
          call C2Y(cg,co,cp,y,zero,nh)
          do it = 1,ntmax
```

```
                    T = time(it-1)
                    tout = time(it)
*          Find the average concentration for this time instance
                    avcgb = cg(nh/2,it-1)*cgB00
                    if (it.eq.1) avcgb=0.0
                    call Update(avcgB)
                    if (it.gt.(ntmax-5)) call PrintDimensionless()
                    call Pack(par)
                    istate  = TRUE
                    CALL ODESSA(fun,dfun,NEQ,Y,PAR,T,TOUT,ITOL,RTOL,ATOL,
     &                  ITASK,ISTATE, IOPT,RWORK,LRW,IWORK,LIW,jfun,MF)
                    if(istate.LT.0) then
                            write(6,*)' istate= ',istate
                            stop
                    endif
                    call Y2C(cg,co,cp,y,it,nh)
                    if (it.gt.(ntmax-5)) call PrintOne(cg,co,cp,time,ht,nh,it)
                    call CheckSteadyState(istatus,cg,co,cp,nh,it,nt,tau,tout)
                    if (istatus.eq.TRUE) then
                            tlast = it
                            goto 10
                    endif
            enddo
c           Output your results
10          continue
            call PrintSum(cg,co,cp,time,ht,nh,tlast,tau)
            enddo
            stop
            end




                              init.f

      subroutine InitConditions(istatus,nh,nt,cg,co,cp,ht,dt,time,y)

      implicit none
      include "Include/parameters.h"
      include "Include/dimensionless.h"
      include "Include/operation.h"
      integer istatus,nh,nt
      real*8 cg(0:nhmax,0:ntmax)
      real*8 co(0:nhmax,0:ntmax)
      real*8 cp(0:nhmax,0:ntmax)
      real*8 time(0:ntmax),ht(0:nhmax)
      real*8 y(neqmax,nparmax+1)
      real*8 dt
      character*80 filename
      integer ih,it
*
* Initial profile along the z axis (t=0)
          if (colstatus.eq.OLD) then
                  open(unit=9,file=fileprev,status='old')
                  read(9,*)
```

```
            do ih = 0,nh
                    read (9,*) ht(ih),cg(ih,0),co(ih,0),cp(ih,0)
                    cg(ih,0) = cg(ih,0)/cgB00/1000
                    co(ih,0) = co(ih,0)/cgO00/1000
            enddo
            cg(0,0) = 1.0
            co(0,0) = 1.0
            cp(0,0) = (cg(0,0)/psi)**(1./an)
            close(9)
      elseif(colstatus.eq.FRESH) then
      do ih = 0,nh
                    cg(ih,0) = 1.0e-4
                    co(ih,0) = 1.0e-4
      enddo
      cg(0,0) = 1.0
      co(0,0) = 1.0
      cp(0,0)= (cg(0,0)/psi)**(1./an)
      else
      write(6,'(a)')'Error in Initial Status selection'
      stop
      endif
      do ih = 0,nh
      cp(ih,0) = cg(ih,0)/psi
      enddo
Initial conditions (t=0) at the entrance of the column
      do it =1,nt
                    cg(0,it) = cg(0,0)
                    co(0,it) = co(0,0)
                    cp(0,it) = cp(0,0)
      enddo
      do it =0,nt
                    time(it) = float(it)*dt
      enddo
      if (istatus.ne.OLD) then
                    do ih=0,nh
                           ht(ih) = float(ih)*dz
                    enddo
      endif
      return
      end
c*****************************************************************
      subroutine InitOdessa(neq,ndim,npar,iopt,itask,lrw,liw,mf,itol,rtol,atol,err)

      implicit none
      include "Include/parameters.h"
      real*8 atol(neqmax,nparmax+1),rtol(neqmax,nparmax+1)
      integer mf,itask,lrw,liw,itol
      integer neq(2),iopt(3)
      integer ndim,npar
      integer nsv
      real*8 err
      integer i,j
      neq(1)=ndim
      neq(2)=npar
```

```
nsv=nparmax+1
itol=4
do i=1,ndim
        do j=1,nsv
                rtol(i,j)=err
                atol(i,j)=err
        enddo
enddo
itask=1
iopt(1)=0
iopt(2)=0
iopt(3)=0
lrw=lrwmax
liw=liwmax
mf=22
return
end


subroutine Reset(rwork,iwork,itask,istate,iopt,mf)
implicit none
include "Include/parameters.h"
real*8 rwork(lrwmax)
integer iwork(liwmax)
integer itask,istate,iopt(3),mf,i

istate= 0
itask = 0
mf    = 0
iopt(1) = 0
iopt(2) = 0
iopt(3) = 0
do i=1,lrwmax
        rwork(i) = 0.
enddo
do i=1,liwmax
        iwork(i) = 0
enddo
return
end
```

print.f

```
C********************************************************************
C print concentration changes along the column time
C
C********************************************************************
        subroutine  PrintAll(cg,co,cp,time,nt,ht,nh)
        implicit none
        include "Include/parameters.h"
        real*8 cg(0:nhmax,0:ntmax)
        real*8 co(0:nhmax,0:ntmax)
        real*8 cp(0:nhmax,0:ntmax)
```

```
          real*8 time(0:ntmax),ht(0:nhmax)
          integer nt,nh,it,ih

          write (6,84)
84        format(//,5x,'Solution of the Transient Model',//)
          do it = 0, nt
                  write (6,86) time(it)
86                format (/, 10x, 'At Time  = ', f14.3,/)
                  write (6,89)
89                format(//,8x,'h/H',9x,'cg',13x,'co',13x,'cp',//)
                  do ih = 0, nh
                          write (6,96) ht(ih), cg(ih,it), co(ih,it), cp(ih,it)
96                        format (5x, f7.3,3x,f10.4,5x,f10.4,5x,f10.4)
                  enddo
          enddo
          return
          end
*************************************************************************
*         The subroutine Prints out the Updated values of the dimensionless units
*         that depend on delta
*************************************************************************

          subroutine PrintDimensionless()
          implicit none
          include "Include/dimensionless.h"

          WRITE(6,123)
          WRITE(6,1)
1         FORMAT (' ','    Dimensionless Parameters :', /)
          write(6,*)
          WRITE(6,5) betaB,betaO
5         FORMAT (2x,'Beta  [B]= ',f14.8,
     &    2x,'Beta  [O]= ',f14.8)
          WRITE(6,6) effectB,effectO
6         FORMAT (2x,'Effectv[B]= ',f14.8,
     &    2x,'Effectv[O]= ',f14.8)
          WRITE(6,3) epsilnB,epsilnO
3         FORMAT (2x,'Epsilon[B]= ',f14.8,
     &    2x,'Epsilon[O]= ',f14.8)
          write(6,*)
          WRITE(6,8) gamaB
8         FORMAT (2x,'gamaB    = ',f14.8)
          write(6,*)
          write(6,52) psi,beta
52        format (2x,'psi     = ',f14.8,
     &    2x,'beta    = ',f14.8)
          WRITE(6,2) porosity,delta
2         FORMAT (2x,'porosity = ',f14.8,
     &    2x,'delta    = ',f14.8)
          WRITE(6,7) 1./an,dz
7         FORMAT (2x,'n       = ',f14.8,
     &    2x,'dz      = ',f14.8)
          write(6,*)
          write(6,*)
```

```fortran
123       format('_____',/)
          return
          end
c*********************************************************************
*         Print the variables
c*********************************************************************
          subroutine Printdimensional ()
          implicit none
          include "Include/parameters.h"
          include "Include/dimensional.h"
          include "Include/operation.h"
          include "Include/dimensionless.h"
          write(6,123)
          WRITE(6,1)
1         FORMAT (' ',//,' VARIABLES IN THE MODEL',//)
          WRITE(6,11)
11        FORMAT (5x,/,' Input data for Transient Biofilter Model',/)
          WRITE(6,2)
2         FORMAT (2x,'1 - Toluene',/,2x,'2 - Oxygen',/)
          WRITE(6,19) restime/60
19        format (' ', 'Resitance Time (min)     = ', f12.5)
          WRITE(6,3) volume*1e6
3         FORMAT (' ', 'Volume of the column(cm3)   = ', f12.5)
          WRITE(6,4) As
4         FORMAT (' ', 'Biolayer Sur.Area( m2/m3)   = ', f12.5)
          WRITE(6,41) alpha
41        FORMAT (' ', '% area covered by biomass   = ', f12.5)
          write(6,44) b0
44        format (' ', 'Biomass Conc. (kg/m3)     = ', f12.5)
          WRITE(6,5) delta*1e-3
5         FORMAT (' ', 'Film thickness (mm)       = ', f12.5)
          WRITE(6,59) porosity
59        FORMAT (' ', 'Porosity               = ', f12.5)
          WRITE(6,18) CGB00*1000.
          WRITE(6,22) CGO00*1000.
18        FORMAT (' ', 'Inlet conc. (g/m3 of air)(B) = ', f12.5)
22        FORMAT (' ', 'Inlet conc. (g/m3 of air)(O) = ', f12.5)
          write(6,31) YB
31        format (' ', 'Yield Coefficient (B)     = ', f12.5)
          write(6,34) YOB
34        format (' ', 'Yield Coefficient (OB)     = ', f12.5)
          WRITE(6,51) DBW*1.e+9
          WRITE(6,55) DOW*1.e+9
51        format (' ', 'Diff. Coeff. (B)*1e9(m2/s)  = ', f12.5)
55        format (' ', 'Diff. Coeff. (O)*1e9(m2/s)  = ', f12.5)
          WRITE(6,565) mB
565       FORMAT (' ', 'Dist. Coeff.    (B)   = ', e12.5)
          WRITE(6,567) mO
567       FORMAT (' ', 'Dist. Coeff.    (O)   = ', e12.5)
          WRITE(6,566) Kapaa*3600.
566       FORMAT (' ', 'Mass Trans. Coef. Ka   [m/h]= ', e12.5)
          WRITE(6,568) Kapad
568       FORMAT (' ', 'Adsorption Parameter Kd [g/g]= ', e12.5)
          WRITE(6,569) rho
```

```fortran
569     FORMAT (' ', 'Particle Density    [kg/m3]= ', e12.5)
        write(6,123)
        write(6,*) '        Andrews and other  Parameters'
        WRITE(6,6)
     &            miouB*3600,KB*1000,KBI*1000,Ko*1000
6       format (' ',/,
     &  ' miou B(1/hr) = ',f12.3,/,
     &                  ' KB  (g/m3)  = ',f12.3,/,
     &  ' KBI  (g/m3) = ',f12.3,/,
     &  ' KO  (g/m3)  = ',f12.3 )
        write(6,123)
123     FORMAT('_____',/)
        return
        end
*****************************************************************
*
        subroutine today()
*    EXTERNAL  TDATE
*    CALL TDATE (IDAY, MONTH, IYEAR)
        write(6,123)
        WRITE (6,66) month,iday,iyear
66      Format( 2x, ' Date :   ',i2,'/',i2,'/',I4,//,
     &  '    Model Predictions for Toluene System      ',/,
     &  '       written by Dimitrios Tsangaris       ',/,
     &  '   ================================================' //)
123  FORMAT('_____',/)
        return
        end
C****************************************************************
C print concentration changes along the column time
C****************************************************************
        subroutine PrintOne(cg,co,cp,time,ht,nh,it)
        implicit none
        include "Include/parameters.h"
        include "Include/operation.h"
        real*8 cg(0:nhmax,0:ntmax),co(0:nhmax,0:ntmax)
        real*8 cp(0:nhmax,0:ntmax)
        real*8 time(0:ntmax),ht(0:nhmax)
        integer nh,it,ih

        write (6,84)
84      format(//,5x,'Solution of the Transient Model',//)
        write (6,86) time(it)
86      format (/, 10x, 'At Time  = ', f14.3,/)
        write (6,89)
89      format(//,8x,'h/H',9x,'cg',13x,'co',13x,'cp',//)
        do ih = 0, nh
c       write (6,96) ht(ih), cg(ih,it), co(ih,it), cp(ih,it)
        write (6,96) ht(ih), cg(ih,it)*cgB00*1000,
     &  co(ih,it)*cgo00*1000, cp(ih,it)
96      format (5x, f7.3,3x,f10.4,5x,f10.4,5x,f10.4)
        enddo
        return
        end
```

```
C****************************************************************
C print concentration changes along the column time
C****************************************************************
      subroutine PrintSum(cg,co,cp,time,ht,nh,nt,tau)
      implicit none
      include "Include/parameters.h"
      include "Include/dimensional.h"
      include "Include/operation.h"
      real*8 cg(0:nhmax,0:ntmax),co(0:nhmax,0:ntmax)
      real*8 cp(0:nhmax,0:ntmax)
      real*8 time(0:ntmax),ht(0:nhmax),tau
      integer nh,it,ih,nt

      write (6,84)
84    format(//,5x,'Summary Results for the Transient Model',//)
      write (6,89)
89    format(//,8x,'time(h)',4x,'cg (1/3)',7x,'cg (2/3)',7x,'cg',//)
      do it = 0, nt
              write (6,96) time(it)*tau*24,cg(nh/3,it)*cgB00*1000,
     &        cg(nh*2/3,it)*cgB00*1000,cg(nh,it)*cgB00*1000
96            format (5x, f8.3,3x,f10.4,5x,f10.4,5x,f10.4)
      enddo
      write(6,*)
      return
      end
```

readparam.f

```
      subroutine  ReadParam(istatus,ndim,npar,nh,nt,dt,err,tau)
      implicit none
      include "Include/parameters.h"
      include "Include/operation.h"
      include "Include/dimensional.h"
      include "Include/dimensionless.h"
      include "Include/system.h"
      integer istatus,nh,nt,ndim,npar,column
      real*8 dt,err,tau,flowrate
      real*8 foo
      character filename*80

      integer unfn
      unfn=7
      filename = 'Toluene.in'
*     ResTime must be entered in Minutes
*     cgB00 must be entered in g/m3
*     read(5,*) cgb00,restime
*     read(5,'(a)')filename
      open (unit=unfn,file=filename,status='old')
      read(unfn,*) flowrate
      read(unfn,*) cgb00
      read(unfn,*)
```

```
*          Now read from the standard input the operating conditions
*          System parameters
           iprnewton =  TRUE
           iprnewton =  FALSE
*          ODESSA parameters
           ndim = neqmax
           npar= nparmax
           nt  = ntmax
           nh  = nhmax
*          Biofilm parameter ( in kg/m3)
           read(unfn,*)b0
*          Kinetic constants for Ethanol and Butanol
           read(unfn,*) miouB
           miouB =  miouB/3600
           read(unfn,*) KB
           read(unfn,*) KBI
           read(unfn,*) KO
*          Diffucivities
           read(unfn,*) DBW
           read(unfn,*) DOW
*          Yield coefficients
           read(unfn,*) YB
           read(unfn,*) YOB
*          Henry's constants
           read(unfn,*) mB
           read(unfn,*) mO
*          Entrance concentrations
           cgB00  = cgB00 * 1.e-3
           read(unfn,*) cgo00
           cgo00  = cgO00 * 1.e-3
*          Volumetric properties
           read(unfn,*) Volume
           read(unfn,*) Surface
           flowrate = flowrate/3600
           restime  = volume/flowrate
           velocity = 1./restime
           tau      = restime/24.0/3600
*          Adsorption parameters
           read(unfn,*)
           read(unfn,*) As
           read(unfn,*) alpha
           read(unfn,*) foo
c          read(unfn,*) kapaA
           read(unfn,*) kapad
           read(unfn,*) porosity
           read(unfn,*) rho
           read(unfn,*) an
           kapaa= kapaa/3600.
*          Numerical parameters
           colstatus= FRESH
           read(unfn,*) column
           if (column.eq.(20.0)) colstatus = OLD
           read(unfn,'(a)') fileprev
           read(unfn,*) err
```

```
* Now calculate some Dimensionless quantities using the above values
        gamaB   = KB/KBl
        epsilnB = cgB00/(mB*KB)
        epsilnO = cgO00/(mO*KO)
* Now calculate some Dimensionless quantities using the above values
        istatus= FRESH
c       dt = 0.01
        read(unfn,*) dt
        dz = 1.0/float(nh)
        call Update(zero)
        close (unfn)
        return
        end
***********************************************************************
*       The subroutine calculates some dimensionless units that depend on
*       the parameter 'delta'. Delta, is the depth of the biofilm
***********************************************************************

        subroutine Update(cgasB)
        implicit none
        include "Include/parameters.h"
        include "Include/operation.h"
        include "Include/dimensional.h"
        include "Include/dimensionless.h"
        real*8 xv,cgasB
        real*8 deltaMt,const

        call FindDelta(cgasB,delta,effectB,effectO)
*A correction is needed because the empirical formula Cjp = Kd (Cj*)^n holds
*       only when cj is in [g_j / m3 air]. Then, const= g/m3->Kgr/m3.
*       After this correction, cstar_reduced = psi*Cjp_reduced
        const  = 1.e-3
        deltaMt = delta*1.e-6
        xv = b0
        betaB=effectB*(alpha*As)*deltaMt*xv*restime*miouB/(YB *cgB00*porosity)
        betaO=effectO*(alpha*As)*deltaMt*xv*restime*miouB/(YOB*cgO00*porosity)
        beta = kapaa*(1-alpha)*As*restime/porosity
        psi  = (const/cgB00)*(porosity*cgB00/((1-porosity)*rho*Kapad))**an
        return
        end



                                util.f

        subroutine C2Y(cg,co,cp,y,it,nh)
        implicit none
        include "Include/parameters.h"
        real*8  y(neqmax,nparmax+1)
        real*8  cg(0:nhmax,0:ntmax), co(0:nhmax,0:ntmax),cp(0:nhmax,0:ntmax)
        integer ih,it,nh

        do ih = 1, nh
                y(ih,1)    = cg(ih,it)
```

```fortran
            y(ih+nh,1)  = co(ih,it)
            y(ih+2*nh,1)= cp(ih,it)
      enddo
      return
      end

      subroutine Y2C(cg,co,cp,y,it,nh)
      implicit none
      include "Include/parameters.h"
      real*8  y(neqmax,nparmax+1)
      real*8  cg(0:nhmax,0:ntmax), co(0:nhmax,0:ntmax),cp(0:nhmax,0:ntmax)
      integer ih,it,nh

      do ih = 1, nh
            cg (ih,it) = y(ih,1)
            co (ih,it) = y(ih+nh,1)
            cp (ih,it) = y(ih+2*nh,1)
      enddo
      return
      end
***********************************************************************

      subroutine pack(par)
      implicit none
      include "Include/parameters.h"
      include "Include/dimensionless.h"
      real*8 par(1)

      par(1) = epsilnB
      par(2) = epsilnO
      par(3) = gamaB
      par(4) = betaB
      par(5) = betaO
      par(6) = beta
      par(7) = psi
      par(8) = porosity
      par(9) = dz
      par(10)= an
      return
      end
***********************************************************************

      subroutine CheckSteadyState(istatus,cg,co,cp,nh,it,nt,tau,tout)
      implicit none
      include "Include/parameters.h"
      real*8 cg(0:nhmax,0:ntmax), co(0:nhmax,0:ntmax),cp(0:nhmax,0:ntmax)
      real*8 d1,d2,d3,tau,tout
      integer it,nh,istatus,nt

      d1 = abs (cg(nh,it) - cg(nh,it-1))
      d2 = abs (co(nh,it) - co(nh,it-1))
      d3 = abs (cp(nh,it) - cp(nh,it-1))
c     if(d1.le.TOLERR.and.d2.le.TOLERR.and.d3.le.TOLERR) then
*     As this program is checking sensitivity it goes until done,
```

```
*          and does not use the steady state checker
           if(it.ge.ntmax-5) then
                   istatus = TRUE
                   write(6,47) tout*tau, it, nt
           else
                   istatus = FALSE
           endif
47         format(//,5x,'Steady state was reached in',f10.3,' days',
      &           /,5x,'Iterations       = ',i10,
      &              /,5x,'Maximum Iterations = ',i10,//)
           return
           end
```

model.f

```
C*************************************************************************
c  this subroutine computes the vectorfield
c*************************************************************************
           subroutine fun(ndim,t,y,par,ydot)
           implicit none
           include "Include/parameters.h"
           real*8 y(neqmax),ydot(neqmax),par(nparmax),t
           integer ndim
           real*8 y1,y2,y3,fun1,fun2,y4
           real*8 cbpr,copr,cb,co,cp
           real*8 der1,der2
           real*8 cstar
           integer i,offset,nh
           include "Include/UNFOLD.h"

*          write(6,'(5f10.4)')(par(i),i=1,10)
*          write(6,'(10f8.4)')(y(i),i=1,ndim)
           nh = ndim/3
*          print*,nh
           do i = 1,nh
                   cb   = y(i)
                   cbpr = 0.
                   if (i.ge.2) cbpr = y(i-1)
                   co   = y(i+nh)
                   cp   = y(i+2*nh)
                   cstar= psi*(cp**an)
                   y1   = epsilnB*cb
                   y2   = epsilnO*co
                   y3   = 1. + y1 + gamaB*y1*y1
                   y4   = 1. + y2
                   fun1 = (y1/y3)*(y2/y4)
                   fun2 = cb-cstar
                   if (i.eq.1)then
                           der1  = (cb-1.)/dz
                   else
                           der1  = (cb-cbpr)/dz
                   endif
```

```
                ydot(i)= -der1/porosity-betaB*fun1-beta*fun2
        enddo
        offset = nh
        do i = 1,nh
                cb   = y(i)
                co   = y(i+nh)
                copr = 0.
                if (i.ge.2) copr = y(i+nh-1)
                y1   = epsilnB*cb
                y2   = epsilnO*co
                y3   = 1.+y1 + gamaB*y1*y1
                y4   = 1.+y2
                fun1 = (y1/y3)*(y2/y4)
                if (i.eq.1)then
                        der2  = (co-1)/dz
                else
                        der2  = (co-copr)/dz
                endif
                ydot(i+offset)= -der2/porosity-betaO*fun1
        enddo
*       Equations for Solid adsorption
        offset = 2*nh
        do i = 1,nh
                cb   = y(i)
                cp   = y(i+2*nh)
                cstar= psi*(cp**an)
                fun2 = cb-cstar
                ydot(i+offset) = beta*fun2
        enddo
*       write(6,'(10(f8.4))')(ydot(i),i=1,ndim)
        return
        end
C*********************************************************************
        subroutine dfun(ndim,t,y,par,dfdp,jpar)
c*******************************************
c       partial derivatives wrt. parameters of interest
        implicit none
        real*8 y(ndim),par(1),dfdp(1),t
        integer ndim,jpar
        return
        end
C***********************************************************************
c  this subroutine computes the jacobian
c  of the vectorfield
c************************************************************************

        subroutine jfun(ndim,t,y,par,ml,mu,pd,nrpd)
        implicit none
        include "Include/parameters.h"
        integer ndim,ml,mu,nrpd
        real*8 y(neqmax),pd(neqmax,neqmax),par(nparmax),t
        real*8 y1,y2,y3,y4,y5
        real*8 co,cb,cp,dfyi,dfyn
        integer offset,nh,i,j
```

```
      include "Include/UNFOLD.h"

      nh = ndim/3
c jacobian of the vectorfield
      do i=1,ndim
            do j=1,ndim
                  pd(i,j)=0.
            enddo
      enddo
*     dCB partial derivatives
      do i = 1, nh
            cb  = y(i)
            co  = y(i+nh)
            cp  = y(i+2*nh)
            y1  = epsilnB*cb
            y2  = epsilnO*co
            y3  = 1. + y1 + gamaB*y1*y1
            y4  = 1. + y2
            y5  = epsilnB*(1.-gamaB*y1*y1)
            dfyi = (y5/y3/y3)*(y2/y4)
            dfyn = (y1/y3)*(epsilnO/y4/y4)
            if (i.gt.1) pd(i,i-1) = 1/porosity/dz
            pd(i,i) = -1/porosity/dz-betaB*dfyi-beta
            pd(i, nh+i) = -betaB*dfyn
            pd(i,2*nh+i)=  beta *psi*an*(cp**(an-1))
      enddo
*     dCO partial derivatives
      offset = nh
      do i = 1, nh
            cb  = y(i)
            co  = y(i+nh)
            cp  = y(i+2*nh)
            y1  = epsilnB*cb
            y2  = epsilnO*co
            y3  = 1. + y1 + gamaB*y1*y1
            y4  = 1. + y2
            y5  = epsilnB*(1.-gamaB*y1*y1)
            dfyi = (y5/y3/y3)*(y2/y4)
            dfyn = (y1/y3)*(epsilnO/y4/y4)
            pd(offset+i,i) = -betaO*dfyi
            pd(offset+i, offset+i) = -1/porosity/dz-betaO*dfyn
            if (i.gt.1) pd(offset+i,offset+i-1) = 1/porosity/dz
      enddo
*     dCp partial derivatives
      offset = 2*nh
      do i = 1, nh
            cb = y(i)
            pd (offset+i,i)      = beta
            pd (offset+i,offset+i)       = -beta*psi*an*(cb**(an-1))
      enddo
      return
      end
```

finddelta.f

```
subroutine FindDelta(avcg,delta,effectB,effectO)
implicit none

real*8 avcg,delta,effectB,effectO

delta  = 1.513*avcg*1e3+33.35
effectB= 0.031*avcg*1e3+ 0.19

*       if (avcg.eq.0) then
*               delta  = 20
*               effectB = 0.2
*       else
*               delta    = 23.3061*(avcg**0.436968)
*               effectB = 0.43163*(avcg**(-1.87141))
*       endif
        effectO = effectB
        return
        end
```

Makefile.f

```
SRC=main.f Model.f Print.f Init.f ReadParam.f Util.f FindDelta.f
OBJ=main.o Model.o Print.o Init.o ReadParam.o Util.o FindDelta.o
LIB=$(HOME)/lib/odessa.o
#OPT= -extend_source  -check_bounds -trapuv -g
OPT= -extend_source -O2
.SUFFIXES: .o .f
.f.o: ; f77 -c $(OPT) $*.f -o $*.o
ssmix: $(OBJ)
        f77 $(OBJ) $(LIB) -o toluene
clean:
        rm -f *.o
```

Toluene.in

0.1	Flowrate	
5.0	cg	
***********Toluene*****************		
100	B0	[Kg]
1.50	Miou i	[1/h]
11.03e-3	Ki	[Kg/m3]
78.94e-3	Kil	[Kg/m3]
0.26e-3	KO	[Kg/m3]
1.03e-9	DiW	[m2/s]
2.41e-9	DOW	[m2/s]
0.708	Yi	[Kg/Kg]
0.341	YOi	[Kg/Kg]
0.27	mi	[-]

```
34.4           mO          [-]
275            C[o]        [g/m3]
5.15e-3        Volume      [m3]
1.82e-2        Surface     [m^2]
-----------------------------------------------------
133.33         As          [m-1]
0.3            Alpha       [-]
6.04e-3        Ka          [m/h]
2.254e-5       Kd          [g/g]
0.3            Porosity [-]
428            RhoP        [kg/m3]
0.96           1/n         [This is 1/n]
10         if 20 then use OLD initial profile below
0.625.last
1e-6       Error for ODESSA
0.01       dt
```

Include

parameters.h

```
integer nhmax,ntmax
parameter(ntmax=100)
parameter(nhmax=20)
integer neqmax,nparmax,liwmax,lrwmax,colstatus
character *80 fileprev
parameter(neqmax=3*nhmax,nparmax=20,lrwmax=5000,liwmax=100)
integer FRESH,OLD
parameter (FRESH=10,OLD=20)
real*8 TOLERR
parameter (TOLERR=1.e-4)
integer TRUE,FALSE
parameter (TRUE=1,FALSE=0)
real*8 zero
parameter (zero=0.0)
common/colstatus/colstatus
common/fileprev/fileprev
```

odessa.h

```
real*8 par(nparmax),y(neqmax,nparmax+1)
real*8 atol(neqmax,nparmax+1),rtol(neqmax,nparmax+1)
real*8 rwork(lrwmax)
integer iwork(liwmax)
integer neq(2),iopt(3)
integer mf,itask,lrw,liw,itol
```

operation.h

```
* Concentrations
        real*8 cgB00,cgo00
        common/concen/cgB00,cgo00
```

dimensional.h

```
* Kinetic Parameters
        real*8 KB,KBI,KO,miouB
        real*8 DOW,DBW
        real*8 YB,YOB
        real*8 b0
        common/kinetic1/KB,KBI,KO,miouB
        common/kinetic2/DOW,DBW
        common/kinetic3/YB,YOB
        common/kinetic4/b0
* Henry's Parameters
        real*8 mB,mO
        common/henry/mB,mO
* Volumetric Parameters
        real*8 volume,surface,volrate,restime,As,velocity
        common/volumetric/volume,surface,volrate,restime,As,velocity
*Adsorption
        real*8 kapaa,kapad,alpha,rho
        common/adsorp/kapaa,kapad,alpha,rho
```

dimensionless.h

```
        real*8 epsilnB,epsilnO
        real*8 betaB,betaO
        real*8 psi,gamaB,porosity,beta
        real*8 dz,an
        common/dim1/epsilnB,epsilnO,betaB,betaO
        common/dim2/beta,gamaB,psi,porosity,dz,an
        real*8 effectB,effectO,delta
        common/aux/effectB,effectO,delta
```

system.h

```
        integer iprnewton
        common/flowcontrol/iprnewton
```

unfold.h

```
real*8 epsilnB,epsilnO,gamaB,betaB,betaO,beta,psi,porosity,dz,an
        epsilnB  = par(1)
        epsilnO  = par(2)
        gamaB    = par(3)
        betaB    = par(4)
        betaO    = par(5)
        beta     = par(6)
        psi      = par(7)
        porosity = par(8)
        dz       = par(9)
        an       = par(10)
```

APPENDIX E

COMPUTER CODE FOR SOLVING THE TRANSIENT BIOFILTRATION MODEL FOR A SINGLE VOC WITH A VARYING INLET CONCENTRATION
(only portions differing from Appendix D shown)

main.f

```
************************************************************

c  Purpose :      "Solution Of The Transient Biofiltration
c                  Model For A Single VOC,
c                  With A Varying Inlet Concentration."
c
c  Method :       ODESSA-Ordinary Differential Equation
c                  Solver With Explicit Sensitivity Analysis;
c                  Stiff Mode When User Supplied Jacobian
c                  Option Is Used
c
c  Language :     FORTRAN
c
c  Written By :   Dimitrios Tsangaris, Newark, NJ on March 28,1995
c
c  Updated By :   Michael Cohen, Newark, NJ in August 1995


c************************************************************

        implicit none
        include "Include/parameters.h"
        external fun,dfun,jfun
        include "Include/odessa.h" ·
        real*8 cg(0:nhmax,0:ntmax)
        real*8 co(0:nhmax,0:ntmax)
        real*8 cp(0:nhmax,0:ntmax)
        real*8 time(0:ntmax), ht(0:nhmax)
        include "Include/operation.h"
        include "Include/dimensional.h"
        real*8 dt,t,tout,err,tau,avcgb
        integer istate,istatus,i
        integer ndim,npar,nt,nh,it,ih,tlast
*
*       Read the system parameters, and initialize the concentrations
        call Reset(rwork,iwork,itask,istate,iopt,mf)
        call today()
        call ReadParam(istatus,ndim,npar,nh,nt,dt,err,tau)
        call InitConditions(istatus,nh,nt,cg,co,cp,ht,dt,time,y)
        call InitOdessa(neq,ndim,npar,
     +           iopt,itask,lrw,liw,mf,itol,rtol,atol,err)
        call PrintDimensional()
        call PrintDimensionless()
        call PrintOne(cg,co,cp,time,ht,nh,zero)
        call C2Y(cg,co,cp,y,zero,nh)
        do it = 1,ntmax
                T    = time(it-1)
                tout = time(it)
*       Find the average concentration for this time instance
c       The formula used is cgB00=mB*Clo*exp(-Qg*mB*time)/VL)
c       to get time from time(it) it is necessary to divide
c       by restime, so replace Qg with volume (and divide
c       by dilution, as only Qg, not flowrate is leaving soil)
```

```
            cgB00   = mB*Clo*exp(-(volume/dilution)
&                     *mB*time(it)/Vplume)
            cgB00   = cgB00/dilution
            inlet(it) = cgB00
            cgB00   = cgB00*1.e-3
            avcgb   = cg(nh/2,it-1)*cgB00
            if (it.eq.1) avcgb=0.0
            call Update(avcgB)
            if (it.lt.(ntmax-20)) goto 888
            call PrintDimensionless()
888         continue
            call Pack(par)
            istate   = TRUE
            CALL ODESSA(fun,dfun,NEQ,Y,PAR,T,TOUT,ITOL,RTOL,ATOL,
&                       ITASK,ISTATE, IOPT,RWORK,LRW,IWORK,LIW,jfun,MF)
            if(istate.LT.0) then
                    write(6,*)' istate= ',istate
                    stop
            endif
            call Y2C(cg,co,cp,y,it,nh)
            if (it.lt.(ntmax-20)) goto 999
      call PrintOne(cg,co,cp,time,ht,nh,it)
999         continue
            call CheckSteadyState(istatus,cg,co,cp,nh,it,nt,tau,tout)
            if (istatus.eq.TRUE) then
                    tlast = it
                    goto 10
            endif
      enddo
c     Output your results
10    continue
      call PrintSum(cg,co,cp,time,ht,nh,tlast,tau)
      stop
      end
```

print.f (partial)

```
C************************************************************
C print concentration changes along the column time
C************************************************************
      subroutine PrintOne(cg,co,cp,time,ht,nh,it)
      implicit none
      include "Include/parameters.h"
      include "Include/operation.h"
      real*8 cg(0:nhmax,0:ntmax),co(0:nhmax,0:ntmax)
      real*8 cp(0:nhmax,0:ntmax)
      real*8 time(0:ntmax),ht(0:nhmax)
      integer nh,it,ih

      write (6,84)
84    format(//,5x,'Solution of the Transient Model',//)
      write (6,86) time(it),cgB00*1000
```

```
86      format (/, 10x, 'At Time = ', f14.3,3x,'Cg = ' f10.3,/)
        write (6,89)
89      format(//,8x,'h/H',9x,'cg',13x,'co',13x,'cp',//)
        do ih = 0, nh
                write (6,96) ht(ih), cg(ih,it)*cgB00*1000,
     &          co(ih,it)*cgo00*1000, cp(ih,it)
96              format (5x, f7.3,3x,f10.4,5x,f10.4,5x,f10.4)
        enddo
        return
        end
C********************************************************
C print concentration changes along the column time
C********************************************************

        subroutine PrintSum(cg,co,cp,time,ht,nh,nt,tau)
        implicit none
        include "Include/parameters.h"
        include "Include/operation.h"
        real*8 cg(0:nhmax,0:ntmax),co(0:nhmax,0:ntmax)
        real*8 cp(0:nhmax,0:ntmax)
        real*8 time(0:ntmax),ht(0:nhmax),tau
        integer nh,it,ih,nt

        write (6,84)
84      format(//,5x,'Summary Results for the Transient Model',//)
        write (6,89)
89      format(//,8x,'time(h)',6x,'cgin',9x,'cg (1/3)',9x,'cg',//)
        do it = 0, ntmax
                write (6,96) time(it)*tau*24,inlet(it),
     &                  cg(nh/3,it)*inlet(it),cg(nh,it)*inlet(it)
96              format (5x,f8.3,3x,f10.4,5x,f10.4,5x,f10.4)
        enddo
        write(6,*)
        return
        end
```

readparam.f

```
        subroutine  ReadParam(istatus,ndim,npar,nh,nt,dt,err,tau)
        implicit none
        include "Include/parameters.h"
        include "Include/operation.h"
        include "Include/dimensional.h"
        include "Include/dimensionless.h"
        include "Include/system.h"
        integer istatus,nh,nt,ndim,npar,column
        real*8 dt,err,tau,flowrate
        character *80 filename
        integer unfn
        unfn=7
        filename = 'Toluene.in'
*       ResTime must be entered in Minutes
```

```
*         cgB00 must be entered in g/m3
*         read(5,*) cgb00,restime
*         read(5,'(a)')filename
          open (unit=unfn,file=filename,status='old')
          read(unfn,*) flowrate
          read(unfn,*) Clo
          read(unfn,*) Vplume
          read(unfn,*) dilution
          read(unfn,*)
*
*         Now read from the standard input the operating conditions
*         System parameters
          iprnewton = TRUE
          iprnewton = FALSE
*         ODESSA parameters
          ndim = neqmax
          npar= nparmax
          nt  = ntmax
          nh  = nhmax
*         Biofilm parameter ( in kg/m3)
          read(unfn,*)b0
*         Kinetic constants for Ethanol and Butanol
          read(unfn,*) miouB
          miouB = miouB/3600
          read(unfn,*) KB
          read(unfn,*) KBI
          read(unfn,*) KO
*         Diffucivities
          read(unfn,*) DBW
          read(unfn,*) DOW
*         Yield coefficients
          read(unfn,*) YB
          read(unfn,*) YOB
*         Henry's constants
          read(unfn,*) mB
          read(unfn,*) mO
*         Entrance concentrations
          cgB00   = mB*Clo
          cgB00   = cgB00/dilution
          inlet(0) = cgB00
          cgB00   = cgB00 * 1.e-3
          read(unfn,*) cgo00
          cgo00   = cgO00 * 1.e-3
*         Volumetric properties
          read(unfn,*) Volume
          read(unfn,*) Surface
          flowrate = flowrate/3600
          restime  = volume/flowrate
          velocity = 1./restime
          tau = restime/24.0/3600
*         Adsorption parameters
          read(unfn,*)
          read(unfn,*) As
          read(unfn,*) alpha
```

```
            read(unfn,*) kapaA
            read(unfn,*) kapad
            read(unfn,*) porosity
            read(unfn,*) rho
            read(unfn,*) an
            kapaa= kapaa/3600.
*       Numerical parameters
            colstatus= FRESH
            read(unfn,*) column
            if (column.eq.20) colstatus = OLD
            read(unfn,'(a)') fileprev
            read(unfn,*) err
* Now calculate some Dimensionless quantities using the above values
            gamaB   = KB/KBl
            epsilnB = cgB00/(mB*KB)
            epsilnO = cgO00/(mO*KO)
* Now calculate some Dimensionless quantities using the above values
            istatus= FRESH
            dt = 0.01
            dz = 1.0/float(nh)
            call Update(zero)
            close (unfn)
            return
            end
**********************************************************************
*       The subroutine calculates some dimensionless units that depend on
*       the parameter 'delta'. Delta, is the depth of the biofilm
**********************************************************************
            subroutine Update(cgasB)
            implicit none
            include "Include/parameters.h"
            include "Include/operation.h"
            include "Include/dimensional.h"
            include "Include/dimensionless.h"
            real*8 xv,cgasB
            real*8 deltaMt,const

* Now recalculate some Dimensionless quantities using the new values
* of cgB00
            epsilnB = cgB00/(mB*KB)
            epsilnO = cgO00/(mO*KO)
            call FindDelta(cgasB,delta,effectB,effectO)
*A correction is needed because the empirical formula Cjp = Kd (Cj*)^n holds
*       only when cj is in [g_j / m3 air]. Then, const= g/m3->Kgr/m3.
*       After this correction, cstar_reduced = psi*Cjp_reduced
            const  = 1.e-3
            deltaMt = delta*1.e-6
            xv = b0
            betaB=effectB*(alpha*As)*deltaMt*xv*restime*miouB/(YB *cgB00*porosity)
            betaO=effectO*(alpha*As)*deltaMt*xv*restime*miouB/(YOB*cgO00*porosity)
            beta = kapaa*(1-alpha)*As*restime/porosity
            psi  = (const/cgB00)*(porosity*cgB00/((1-porosity)*rho*Kapad))**an
            return
            end
```

Toluene.in

51	Flowrate	
100	Clo	
1000	Volume of Plume	
3	dilution	

***********Toluene*************

100	B0	[Kg]
1.50	Miou i	[1/h]
11.03e-3	Ki	[Kg/m3]
78.94e-3	Kil	[Kg/m3]
0.26e-3	KO	[Kg/m3]
1.03e-9	DiW	[m2/s]
2.41e-9	DOW	[m2/s]
0.708	Yi	[Kg/Kg]
0.341	YOi	[Kg/Kg]
0.27	mi	[-]
34.4	mO	[-]
275	C[o]	[g/m3]
18.70	Volume	[m3]
1.82e-2	Surface	[m^2]

133.33	As	[m-1]
0.3	Alpha	[-]
6.04e-3	Ka	[m/h]
2.254e-5	Kd	[g/g]
0.3	Porosity	[-]
428	RhoP	[kg/m3]
0.96	1/n	[This is 1/n]
10	If 20 use OLD initial profile below	
42.last		
1e-6	Error for ODESSA	

include (partial)

operation.h

```
* Concentrations
      real*8 cgB00,cgo00,Clo,Vplume,dilution
      real*8 inlet(0:ntmax)
      common/concen/cgB00,cgo00,Clo,Vplume,dilution,inlet
```

REFERENCES

Androutsopoulou, H. 1994. *A Study of the Biofiltration Process Under Shock-Loading Conditions*, MS Thesis, New Jersey Institute of Technology, Newark NJ.

Angell, K.G. 1991. "Air Sparging Case Studies: Proof That an Innovative Technique Works," *Proceedings of the Eastern Region FOCUS Conference*, National Water Well Association, Portland, ME: 739-760.

Baltzis, B.C. and S.M. Wojdyla. 1995a. "Characteristics of Biofiltration of VOC Mixtures," *Proceedings of the 4th Conference on Environmental Science and Technology*, Th. Lekkas (ed.), University of the Aegean Press, Lesvos, Greece: 322-331 (vol. B).

Baltzis, B.C. and S.M. Wojdyla. 1995b. "Towards a Better Understanding of Biofiltration of VOC Mixtures," *Proceedings of the 1995 Conference on Biofiltration (an Air Pollution Control Technology)*, D.S. Hodge and F.E. Reynolds, Jr. (eds.), The Reynolds Group, Tustin CA: 131-138.

Baltzis, B.C. 1994. *Engineering Analysis and Design of Biological Filters for Removing Volatile Organic Compound (VOC) Emissions*. Final Report for Project A-7, Emission Reduction Research Center, New Jersey Institute of Technology, Newark, NJ.

Baltzis, B.C. and H. Androutsopoulou. 1994. "A Study on the Response of Biofilters to Shock Loading," *Proceedings of the 87th A&WMA Meeting*, paper no. 94-RP-115B.02, Air and Waste Management Association, Cincinnati, OH: 1-14.

Baltzis, B.C. and Z. Shareefdeen. 1994. "Biofiltration of VOC Mixtures: Modeling and Pilot Scale Experimental Verification," *Proceedings of the 87th A&WMA Meeting*, paper no. 94-TA260.10P, Air and Waste Management Association, Cincinnati, OH: 1-16.

Deshusses, M.A., G. Hamer and I.J. Dunn. 1995. "Behavior of Biofilters for Waste Air Biotreatment. 1. Dynamic Model Development," *Environ. Sci. Technol.*, 29(4): 1048-1068.

Diks, R.M.M. and S.P.P. Ottengraf. 1991. "Verification Studies of a Simplified Model for Removal of Dichloromethane from Waste Gases Using a Biological Trickling Filter. Part 1," *Bioprocess Eng.*, 6: 93-99.

Fan, L.S., R. Leyva-Ramos, K.D. Wisecarver and B.J. Zehner. 1990. "Diffusion of Phenol Through a Biofilm Grown on Activated Carbon Particles in a Draft-Tube Three-Phase Fluidized-Bed Bioreactor," *Biotechnol. Bioeng.*, 35: 279-286.

135

Federal Register. 1993a. "Table Z-1-A-Limits for Air Contaminants," 29CFR17, Part 1910.1000: 28.

Federal Register. 1993b.. "Subpart D-Reporting, Public Notification & Recordkeeping (49) Toluene," 40CFR1, Part 141.32: 674.

Fouhy, K. 1992. "Cleaning Waste Gas, Naturally," *Chem. Eng.*, **99**: 41-46.

Fouhy, K. and W. Grinthal. 1994. "Biology Boosts Waste Treatment," *Chem. Eng.*, **101**: 30-31.

Gierke, S.M., N.J. Hutzler and D.B. McKenzie. 1992. "Vapor Transport in Unsaturated Soil Columns: Implications for Vapor Extraction," *Water Resour. Res.*, **28**: 323-335.

Hodge, D.S. and J.S. Devinny. 1994. "Biofilter Treatment of Ethanol Vapors," *Environ. Progress*, **13**: 167-173.

Hutzler, N.J., B.E. Murphy and J.S. Gierke. 1991. "State of Technology Review: Soil Vapor Extraction Systems," *J. Hazard. Mater.*, **26**: 225-230.

Moretti, E.C. and N. Mukhopadhyay. 1993. "VOC Control: Current Practices and Future Trends," *Chem. Eng. Progress*, **89**(7): 20-26.

Ottengraf, S.P.P. and A.H.C. van den Oever. 1983. "Kinetics of Organic Compound Removal from Waste Gases with a Biological Filter," *Biotechnol. Bioeng.*, **25**: 3089-3102.

Shareefdeen, Z.M. 1994. *Engineering Analysis of a Packed-Bed Biofilter for Removal of Volatile Organic Compound (VOC) Emissions.* PhD thesis, New Jersey Institute of Technology, Newark, NJ.

Shareefdeen, Z.M. and B.C. Baltzis. 1994. "Biofiltration of Toluene Vapor Under Steady State and Transient Conditions: Theory and Experimental Results," *Chem. Eng. Sci.,* **49** (24A): 4347-4360.

Shareefdeen, Z.M., B.C. Baltzis, Y. S. Oh, and R. Bartha. 1993. "Biofiltration of Methanol Vapor," *Biotechnol. Bioeng.*, **41**: 512-524.

Silka, L.R., H.D. Cirpili and D.L. Jordan. 1991. "Modeling Applications to Vapor Extraction Systems," *Soil Vapor Extraction Technology: Reference Handbook*, Risk Reduction Engineering Laboratory, Cincinnati, Ohio, EPA/540/2-91/003: 252-259.

Togna, A.P. and M. Singh. 1994. "Biological Vapor-Phase Treatment Using Biofilter and Biotrickling Filter Reactors: Practical Operating Regimes," *Environ. Progress*, **13**: 94-97.

Washington State Department of Ecology. 1994. "Class B Toxic Air Pollutants," WAC 173-460-160.

Wilson, D.E., R.E. Montgomery and M.R. Sheller. 1987. "A Mathematical Model for Removing Volatile Subsurface Hydrocarbons by Miscible Displacement," *Water, Air, and Soil Pollution*, **33**: 231-255.

Wojdyla, S.M. 1996. *Determination of Physical Parameters Affecting Biofiltration of VOCs*. MS Thesis, New Jersey Institute of Technology, Newark NJ.

Zilli, M., A. Converti, A. Lodi, M. Del Borghi and G. Ferraiolo. 1993. "Phenol Removal from Waste Gases with a Biological Filter by *Pseudomonas putida*," *Biotechnol. Bioeng.*, **41**: 693-698.