

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

A METHODOLOGY FOR THE DESIGN OF AN INTEGRATED AIR STRIPPING/BIOFILTRATION PROCESS TO CLEAN CONTAMINATED AQUIFERS

**by
Pothitos Ioannis Stamatiadis**

This thesis deals with the conceptual design of an integrated air stripping/biofiltration process for cleaning aquifers contaminated with volatile organic pollutants. In this technology, the pollutant is transferred from the aquifer to an air stream which is subsequently treated in a biofilter. The work presented here is an effort to develop a design methodology for the integrated process and suggests a general, quantitative approach which can be used in preliminary technology evaluation. This methodology involves relatively simple equations for the stripping process and detailed (oftentimes complex) models for biofiltration under conditions of continuous variation of the properties of the polluted airstream which is fed to the biofilter. In this thesis, conventional biofilters were considered. These units have a porous organic support for the biofilms and do not involve supply of nutrients to the organisms through a liquid phase. The air stripping process was considered to be either under equilibrium distribution conditions for the pollutant between the aquifer and the air sparged through it, or under conditions deviating from equilibrium. This deviation was expressed through the use of a fraction of the Henry's constant.

The proposed methodology was applied to eight different situations (case studies). These involved differences in the equilibrium distribution of the pollutant, and the number and extent of the time periods in which the total remediation time is divided. This division is proposed for maintaining relatively constant pollutant concentrations to the inlet of the biofilter over an extended period of time.

Sample calculations have been performed assuming that a constant volume aquifer is contaminated with toluene. The results show that the integrated process can be designed based on predictive engineering models, and that the size of the required biofilter bed and the time frame for decontamination of the aquifer can be very reasonable.

**A METHODOLOGY FOR THE DESIGN OF AN INTEGRATED
AIR STRIPPING/BIOFILTRATION PROCESS TO CLEAN
CONTAMINATED AQUIFERS**

**by
Pothitos Ioannis Stamatiadis**

**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 1997

Blank Page

APPROVAL PAGE

A METHODOLOGY FOR THE DESIGN OF AN INTEGRATED
AIR STRIPPING/BIOFILTRATION PROCESS TO CLEAN
CONTAMINATED AQUIFERS

Pothitos Ioannis Stamatiadis

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: Pothitos Ioannis Stamatiadis

Degree: Master of Science

Date: January 1997

Undergraduate and Graduate Education:

- Master of Science in Environmental Engineering,
New Jersey Institute of Technology, Newark, NJ, 1997
- Bachelor of Science in Civil Engineering,
Aristotle University of Thessaloniki, Thessaloniki, Greece, 1995

Major: Environmental Engineering

This thesis is dedicated to
Drs. Manolis Tsangaris and Dimitris Tsangaris
without whom I would not have gotten this far.

ACKNOWLEDGMENT

The author would like to express his sincere gratitude to his thesis advisor, Dr. Basil C. Baltzis, for his encouragement and expert guidance throughout this research. The author would also like to express thanks to Dr. Hsin-Neng Hsieh and Dr. Gordon A. Lewandowski for serving as his committee members.

The author appreciates the help, support and friendship of his fellow students and friends including: Jay Best, Changgen Luo, Michael Cohen, Dr. Socratis Ioannidis, Dr. Dionisios Karvelas, Dilip Mandal, Armando Mora, Christos Mpanias, Evangelos Tsimis, Dr. Kung-Wei Wang and Naihong Zhu.

The author would like to sincerely express his appreciation to his father Ioannis, mother Fotini, sister Panayiota, aunt Marika, grandparents, and the Pahountis and Tsangaris families for their love and support.

Finally, the author would also like to express special thanks to Dr. Dimitrios Tsangaris, and Dr. Manolis Tsangaris -to whom this thesis is dedicated- for their excellent guidance, support and training.

TABLE OF CONTENTS

| Chapter | Page |
|--|------|
| LIST OF TABLES | ix |
| LIST OF FIGURES | xi |
| NOMENCLATURE | xiii |
| 1 INTRODUCTION | 1 |
| 2 LITERATURE REVIEW | 4 |
| 2.1 Soil Venting | 4 |
| 2.2 Vapor Extraction | 6 |
| 2.3 Biofiltration | 8 |
| 3 OBJECTIVES | 15 |
| 4 BASIC MODEL EQUATIONS | 19 |
| 4.1 Equations for a Contaminated Aquifer | 19 |
| 4.2 Equations for Steady-State Biofiltration | 22 |
| 4.3 Equations for Transient Biofiltration | 27 |
| 5 METHODOLOGY AND PRELIMINARY CALCULATIONS | 31 |
| 6 RESULTS AND DISCUSSION | 38 |
| 7 CONCLUSIONS | 46 |
| APPENDIX A: RESULTS FROM THE STUDIES ON THE INTEGRATED AIR STRIPPING/BIOFILTRATION PROCESS | 48 |
| APPENDIX B: COMPUTER CODE FOR SOLVING THE STEADY-STATE BIOFILTRATION MODEL FOR A SINGLE VOC | 67 |

TABLE OF CONTENTS
(Continued)

| Chapter | Page |
|---|-------------|
| APPENDIX C: COMPUTER CODE FOR SOLVING THE TRANSIENT BIOFILTRATION MODEL FOR A SINGLE VOC UNDER TEMPORALLY VARYING INLET CONDITIONS..... | 89 |
| REFERENCES..... | 117 |

LIST OF TABLES

| Table | Page |
|---|------|
| 3.1 Regulations for control of toluene levels | 17 |
| 5.1 Model parameter values for toluene..... | 33 |
| 5.2 Conditions under which toluene gets depleted before oxygen in the biolayers present in the biofilter..... | 34 |
| 5.3 Expressions for parameter δe as a function of toluene concentration in the air (c_T , in g/m^3)..... | 36 |
| 6.1 Number of dilutions of the airstream exiting the aquifer and values of relative volatility, σ , for the cases considered in the design of the integrated process..... | 39 |
| 6.2 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 1 of Table 6.1..... | 39 |
| 6.3 Results from the design studies on the integrated air stripping/biofiltration process for Cases 1-8 of Table 6.1..... | 42 |
| A.1 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 2 of Table 6.1..... | 49 |
| A.2 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 3 of Table 6.1..... | 50 |
| A.3 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 4 of Table 6.1..... | 51 |
| A.4 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 5 of Table 6.1..... | 52 |
| A.5 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 6 of Table 6.1..... | 53 |
| A.6 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 7 of Table 6.1..... | 54 |
| A.7 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 8 of Table 6.1..... | 55 |

LIST OF TABLES
(Continued)

| Table | Page |
|--|-------------|
| A.8 Times required for the breakthrough toluene concentration values to reach the ASIL value ($c_T = 0.28 \text{ g/m}^3$) and 95% of the inlet concentration value. The biomass in the biofilter is assumed to be completely inactive. Cases 1-8 correspond to those of Table 6.3..... | 56 |

LIST OF FIGURES

| Figure | Page |
|--|------|
| 3-1 Schematic of the integrated air stripping/biofiltration process | 15 |
| 6-1 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter for Case 1 (see Table 6.1)..... | 41 |
| 6-2 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter unit when the biomass is completely inactive (dead). The profiles correspond to Case 1 of Table 6.1 | 42 |
| A-1 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter for Case 2 (see Table 6.1)..... | 57 |
| A-2 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter for Case 3 (see Table 6.1)..... | 58 |
| A-3 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter for Case 4 (see Table 6.1)..... | 59 |
| A-4 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter for Case 5 (see Table 6.1)..... | 60 |
| A-5 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter for Case 6 (see Table 6.1)..... | 61 |
| A-6 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter for Case 7 (see Table 6.1)..... | 62 |
| A-7 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter for Case 8 (see Table 6.1)..... | 63 |
| A-8 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter unit when the biomass is completely inactive (dead). These profiles correspond to Case 3 of Table 6.1..... | 64 |
| A-9 Toluene breakthrough concentration profiles at the exit of biofilters when the biomass is completely inactive (curves 1) and completely active (curves 2). The biofilters are assumed to operate under constant inlet toluene concentrations of (a):9.18 and (b): 6.37 g/m ³ . Other conditions are, (a): $\tau = 17.2$ min, $\sigma = 1$; (b): $\tau = 9.7$ min, $\sigma = 0.5$ | 65 |

LIST OF FIGURES
(Continued)

| Figure | Page |
|---|-------------|
| A-10 Toluene breakthrough concentration profiles from biofilters with completely inactive biomass. Profiles also show the time needed for the exit concentration to reach 95% of the corresponding inlet concentration value. All operating conditions are correspondingly those of Figures A-9(a) and (b)..... | 66 |

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 toluene in air entering the extraction well (g/m^3)
- c_L : concentration of toluene in the contaminated aquifer (g/m^3)
- c_{L0} : concentration of toluene 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_{O_i} : oxygen concentration in the air at the inlet of the biofilter (g/m^3)
- c_T : concentration of toluene in the air at a position h along the biofilter (g/m^3)
- c_T^* : equilibrium concentration of toluene at the gas/solid interface (g/m^3)
- c_{T_i} : value of c_T at $h = 0$ (g/m^3)
- c_{TP} : concentration of toluene on the solid particles (g of toluene-absorbed/g particle)
- \bar{c}_O : dimensionless concentration of oxygen in the air ($\bar{c}_O = c_O/c_{O_i}$)
- $\bar{c}_{O,0}$: value of \bar{c}_O at $t = 0$
- \bar{c}_T : dimensionless concentration of toluene in the air ($\bar{c}_T = c_T/c_{T_i}$)
- $\bar{c}_{T,0}$: value of \bar{c}_T at $t = 0$
- \bar{c}_{TP} : dimensionless concentration of toluene on the solid particles defined as $(1-\nu)\rho_P c_{TP}/\nu c_{T_i}$
- $\bar{c}_{TP,0}$: value of \bar{c}_{TP} at $t = 0$
- D_{Ow} : diffusion coefficient of oxygen in water (m^2/h)
- D_{Tw} : diffusion coefficient of toluene in water (m^2/h)

NOMENCLATURE (Continued)

- e_O : effectiveness factor based on oxygen (dimensionless)
- e_T : effectiveness factor based on toluene (dimensionless)
- F : flow rate of airstream (m^3/h)
- $f(X_V)$: ratio of diffusivity of a compound in the biofilm to that in water
- $g(\bar{c}_T, \bar{c}_O)$: dimensionless version of $\mu_T(s_T, s_O)$
- 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_{IT} : inhibition constant in the specific growth rate expression of a culture growing on toluene ($g\text{-toluene}/m^3$)
- K_O : constant in the specific growth rate expression of a culture, expressing the effect of oxygen ($g\text{-oxygen}/m^3$)
- K_T : constant in the specific growth rate expression of a culture growing on toluene ($g\text{-toluene}/m^3$)
- m_O : distribution coefficient for the oxygen-in-air/water system
- m_T : distribution coefficient for the toluene/water system
- n : Freundlich isotherm parameter
- Q_G : flowrate of air entering the sparging point (m^3/h)
- Q_{Gk} : flowrate of air entering the sparging point during the k-th period (m^3/h)
- s_O : oxygen concentration at a position x in the biolayer, at a point h along the column (g/m^3)

NOMENCLATURE (Continued)

- s_T : concentration of toluene at a position x in the biolayer at a point h along the column (g/m^3)
- \bar{s}_O : dimensionless oxygen concentration in the biolayer ($\bar{s}_O = s_O/K_O$)
- \bar{s}_T : dimensionless concentration of toluene in the biolayer ($\bar{s}_T = s_T/K_T$)
- 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_{OT} : yield coefficient of a culture on oxygen (g-biomass/g-oxygen), when toluene is the carbon source
- Y_T : yield coefficient of a culture on toluene (g-biomass/g-toluene)
- 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 $e_T \alpha \delta A_S^* X_v H \mu_T^* / Y_T u_g c_{Ti} v$
- β_2 : dimensionless parameter defined as $e_O \alpha \delta A_S^* X_v H \mu_T^* / Y_{OT} u_g c_{Oi} v$
- β_3 : dimensionless parameter defined as $k_a (1 - \alpha) A_S^* H / u_g v$
- γ : inverse dimensionless inhibition constant ($\gamma = K_T / K_{IT}$)
- δ : effective biolayer thickness (m)

NOMENCLATURE
(Continued)

- ε_1 : dimensionless quantity defined as $c_{Ti}/(K_T m_T)$
- ε_2 : dimensionless quantity defined as $c_{O_i}/(K_O m_O)$
- ζ : dimensionless time defined as t/τ
- η : dimensionless quantity defined as $A_S f(X_V) D_{Tw} K_T \tau / (\delta c_{Ti})$
- θ : dimensionless position in the biolayer ($\theta = x/\delta$)
- λ : dimensionless quantity defined as $D_{Tw} K_T Y_T / (D_{Ow} K_O Y_{OT})$
- $\mu_T(s_T, s_O)$: specific growth rate (h^{-1})
- μ_T^* : constant in the specific growth rate expression for toluene (h^{-1})
- ρ_p : density of the solid particles (g/m^3)
- σ : fraction indicating deviation from equilibrium for the distribution of toluene between water and air.
- τ : space time defined as V/F
- ν : void fraction of the biofilter bed
- ϕ^2 : square of Thiele modulus based on toluene, defined as $\mu_T^* \delta^2 X_V / (f(X_V) D_{Tw} K_T Y_T)$
- ψ : dimensionless quantity defined as $1/c_{Ti} [\nu c_{Ti} / (1-\nu) \rho_p k_d]^{1/n}$
- ω : dimensionless quantity defined as $K_O D_{Ow} c_{Ti} / (K_T D_{Tw} c_{O_i})$

CHAPTER 1

INTRODUCTION

Clean water resources, clean soil, and air are essential for life on our planet. Past practices, which have now been banned, as well as accidental releases of pollutants have generated a large number of contaminated sites. Existing laws and regulations require cleaning of sites, something which is expensive and time consuming. For years, the effort of industry and researchers has been to develop technologies which are economically attractive and efficient in cleaning contaminated sites.

The initial approach for groundwater treatment was to “pump-and-treat”. This means that contaminated water was brought from the earth subsurface to facilities (lagoons, reactors) for treatment and was subsequently reinjected to the aquifer. Contaminated soil was excavated and treated. Treatment was based on chemical (e.g., oxidation), biochemical (e.g., biodegradation), and incineration techniques. This approach is very expensive.

Subsequent efforts concentrated on *in situ* treatment through air supply, nutrients addition, bioaugmentation etc. These methods, although they have led to some interesting results, are usually slow and oftentimes do not ensure the required treatment level.

In search of new techniques, a number of new ideas and technologies have been conceived and tested for feasibility in the recent years. Some of these technologies try to take advantage of aspects of earlier successful technologies and try to combine them in a new ensemble which can be called a hybrid. An example of such a hybrid technology involves combination of air stripping and biofiltration and is the topic of the present thesis.

An integrated air stripping/biofiltration technology can be used in cleaning aquifers which have been contaminated with volatile organic compounds susceptible to biological degradation to innocuous final products. The idea is relatively simple and is based on the use of air as an intermediary for temporarily transferring pollutants from the aquifer (water phase) to air. Subsequently, the intentionally contaminated air is treated in a biofilter. Air stripping is a simple and effective process. Biofiltration is a much more complex process which, when properly designed, can be economical for air decontamination. There are some experimental studies on this integrated process, but no methodology of design based on predictive engineering models was developed. This was the objective of this study.

Specifically, in this study the objective was to use existing models for the air stripping process and models for the biofiltration process and combine them in such a way that a protocol develops and can be used in relatively easy calculations when technology options are discussed before pilot (experimental) studies are undertaken.

Soil venting as well as soil vapor extraction are two other terms which in many ways are similar to air stripping. Usually, they refer to treatment of soil but a main feature (especially of soil vapor extraction) of them is the transfer of the pollution from the subsurface to the air. For this reason, the methodology discussed in this thesis can be also applied for soil cleaning in combination with biofiltration. Thus, the term air stripping could be interchanged with soil vapor extraction or even soil venting in this thesis.

In the integrated process considered here, the key technology used for the destruction of the pollutants is biofiltration. This technology has received a lot of attention in recent years as a means to clean air from the presence of volatile organic compound (VOC) emissions. VOCs are organic chemicals which, according to the definition of the US

Environmental Protection Agency, have vapor pressures of at least 0.1 mm Hg under standard conditions (20°C and 760 mm Hg atmospheric pressure). This technology is reviewed in the next chapter after a review on soil venting and soil vapor extraction is presented.

This thesis has developed a detailed methodology for the design of an integrated air stripping/biofiltration process and has tested this methodology in eight cases assuming a constant volume aquifer which has been contaminated with toluene.

CHAPTER 2

LITERATURE REVIEW

2.1 Soil Venting

The inefficiencies of traditional site remediation efforts have prompted development of innovative techniques such as soil venting to reduce the cost and time of site remediation. Soil venting is an emerging *in situ* technology that enhances desorption and bioremediation of saturated soils by forcing air under pressure into the saturated zone. Since remediation of adsorbed and dissolved phase contamination oftentimes is the longest and most costly part of site clean up, the application of soil venting in multi-phase cleanup program promotes time and cost effective remediation by addressing these difficult phases of contamination.

Removal of VOCs below the water table can be accomplished by sparging air under pressure through soils below the water table. This approach effectively creates a crude air stripper in the subsurface. The saturated soil acts as the packing. Injected air flows through the water over the packing. Air bubbles contact dissolved/adsorbed phase contaminants in the aquifer causing the VOCs to volatilize. The entrained organics are carried by the air bubbles into the vadose zone where they can be captured by a vapor extraction system or, where permissible, allowed to escape from the ground surface. As a bonus, the sparged air maintains high dissolved oxygen, which enhances natural biodegradation. The stripability of the VOC contaminants by the air sparger system is roughly indicated by their Henry's Law constant, K_H . A K_H of at least 10^{-5} atm m³/mole indicates a strippable volatile constituent [Hutzler et al. (1991)].

The key to successful soil venting operation is attaining good contact between the injected air and contaminated soil and groundwater. Below the water table, the air bubbles need to travel vertically through the aquifer in order to strip the VOC contaminant(s). A permeability differential (i.e. clay barrier) above the zone of air injection may severely reduce the effectiveness of soil venting, so such barriers should be identified during site investigation. Studies such as those of Angell (1991) and Hutzler et al. (1991) have shown that soil venting can be an effective process.

Soil venting is a difficult process to model. Existing models are drastic simplifications of actual field conditions. Models that have been proposed usually assume equilibrium between the flowing gas and the surrounding aqueous and sorbed concentrations. There are however, some models which take into account other effects. For example, the model of Wilson et al. (1987) takes into account diffusion of the pollutant through the soil.

Sellers et al. (1993) proposed an air sparging diffusive-flux-limited model. This model states that the rate of contaminants diffusing into the sparging bubbles is balanced by loss of dissolved phase contaminant from the groundwater. The model assumes that there is a concentration gradient around the sparging bubbles, and an even distribution of bubbles within the "volume of influence" around the air sparger. The volume of influence is defined by the extent to which sparging bubbles penetrate the contaminated aquifer.

Other models have investigated the importance of advection, liquid diffusion, gaseous dispersion, and the mass transfer resistance at the air/water interface [Hein et al. (1993)].

2.2 Vapor Extraction

Extracting vapor from soil is a cost-effective technique for the removal of volatile organic compounds. Among the advantages of soil vapor extraction process are that it minimally disturbs the contaminated soil, it can be constructed from standard equipment, it has been demonstrated at pilot -and field- scale, it can be used to treat larger volume of soil which is not practical to be excavated, and it has potential for product recovery. Soils may become contaminated in a number of ways with VOCs (such as industrial solvents and gasoline components). The sources of contamination at or near the surface of the earth include intentional disposal, leaking underground storage tanks, and accidental spills. Contamination of groundwater from these sources can continue even after discharge has stopped because the unsaturated zone above a groundwater aquifer can retain a portion or all of the contaminant discharge.

During remediation, the blower is turned on and the air flow through the soil reaches an equilibrium. The flows that are finally established are a function of the equipment, the flow control devices, the geometry of well layout, the site characteristics, and the air permeability of the soil. At the end of operation, the final distribution of VOCs in the soil can be measured to ensure decontamination of the site. Extraction wells usually consist of slotted elastic pipe placed in permeable packing. They may be aligned vertically or horizontally. Vertical alignment is typical for deeper contamination zones and for residue in radial flow patterns. If the depth of the contaminated soil or the depth to the groundwater table is less than 10 to 15 ft, it may be more practical to dig a trench across the area of contamination and install horizontal perforated piping in the trench bottom rather than to install vertical extraction wells. Usually several wells are installed at a site. Well spacing is

usually based on some estimate of the radius of influence of an individual extraction well. This radius decreases as soil bulk density increases or the porosity of the soil increases. Also air/water separators must be installed [Hutzler et al. (1991)].

Existing models provide simulation of the process in two zones: an unsaturated high porosity region, which is advection dominated, and a saturated, low porosity zone, which is diffusion dominated.

Other approaches are based on a three-dimensional gas flow model which takes into account the non-homogeneity and anisotropy of the soil [Sepehr et al. (1993)]. Two important factors affecting the gas movement in the soil are the pore size and the soil moisture condition. The rate of movement increases as pore size increases and moisture content decreases. However, adsorption of chemicals onto completely dry soils will slow the extraction process. It has also been observed that extraction rates are generally faster in cohesionless material than in aggregated material [McKenzie, (1993)].

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 non-equilibrium 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.

Among the various factors affecting vapor extraction, i.e., gas advection, gas diffusion, gas-water mass transfer, gas-water partitioning, sorption, and intraaggregate diffusion on subsurface movement of organic vapors, modeling has shown that the most important ones are gas advection and diffusion [Hutzler et al. (1991)]. It has been also reported that there are important differences between chemical vapor transport in, and therefore removal from, granular and aggregated soils. The breakthrough and removal of VOCs from the aggregated soils takes more time in terms of gas volume extracted than from sandy soils.

2.3 Biofiltration

Biofiltration is a technology for treatment of VOC-laden airstreams. It is based on the ability of bacterial and fungal species to biodegrade substances such as VOCs into innocuous products (carbon dioxide, water, and mineral salts). Biofiltration has received a lot of attention from industrial and academic researchers in response to the Clean Air Act Amendments (CAAA) of 1990. These regulations impose strict control on the release of VOCs to the atmosphere since VOCs have been implicated as a major contributor to photochemical smog, which can cause haze, damage to plant and animal life, and eye irritation, and respiratory problems for humans. In some individuals, exposure to VOCs may also contribute to an increased risk of developing cancer.

Biofiltration is based on biological destruction of VOC vapors by microorganisms

immobilized on a solid support material. These solids are placed in open or closed structures known as biofilters.

There are two types of biofilters: classical or conventional biofilters and biotrickling filters. Classical biofilters utilize porous solid particles of an organic base (e.g., peat moss, compost, bark, etc.) as a substratum for the formation of layers of microorganisms. The pores of the solids are partially filled with water, thus providing the necessary moisture for microbial activity. They do not involve a continuous liquid (water) phase and require complete humidification of the polluted airstream before it enters the biofilter bed. Classical biofilters are open or closed structures containing the solids. Closed structures are easier to control albeit their higher capital cost. Classical biofilters are packed-bed vapor phase biological reactors. Their operation is relatively simple, requires no engineering attendance, and its cost appears to be low.

Biotrickling filters are always closed structures containing non-porous particles of an inorganic base (plastics, ceramics) as a substratum for the formation of biofilms. They employ a continuous liquid phase which trickles through the bed of solids. The liquid phase is primarily water which also contains various nutrients other than carbon/energy sources for the microorganisms (e.g., sources of nitrogen, phosphorous, vitamins, etc.). They lead to formation of substantial amounts of biomass which needs to periodically be removed from the filter-bed. Biotrickling filters also allow for good pH-control and seem to be ideal in cases of treatment of chlorinated VOCs. They seem to have an operating cost which is substantially higher than that of classical biofilters and require engineering attendance. On the other hand, it appears that biotrickling filters have a lower capital cost due to smaller size requirements -than classical biofilters- for treatment of a given load.

Although, recently, most researchers in the United States have shifted their interests towards the development of biotrickling filters, it is not yet clear whether they have an unconditional advantage over classical biofilters. It is most probable that the answer to the question as to what type of biofilter should be used, is application specific.

The present study deals with classical biofilters and thus, biotrickling filters are not considered in this thesis.

When a contaminated airstream is passed through a biofilter, the VOCs are transferred to the biofilms formed on the surface of the solids where they undergo biological oxidation. Thus, the airstream exiting a biofilter contains amounts of VOCs less than the stream entering the unit. Clearly, the ultimate objective is to design biofilters in ways which ensure that the exiting airstreams are pollutant-free.

The interest in biofiltration stems from the fact that it is a non-energy-intensive technology which leads to VOC destruction without the potential of producing pollutants more hazardous than the original VOCs. Of course it requires proper selection of microorganisms and process conditions. The low energy requirements of biofiltration is probably the reason for which this technology has been more popular in Europe than in the United States. The fuel costs in the U.S. are only 20% of those in Western Europe [Reynolds and Hodge, (1995)], something which makes alternatives to biofiltration viable options in the U.S. markets.

Some factors affecting biofiltration are the void fraction of the biofilter bed, the density and thickness of the biofilms formed around the solids, the equilibrium distribution (Henry's constant) of VOCs and oxygen between the air and biofilm, the diffusivity of pollutants and oxygen in the biofilm, the biofilm/air interfacial area, the mass transfer

coefficient of VOCs onto (during adsorption) and from (during desorption) the solid packing, etc.

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 complicated) 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 the limiting factor 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. Same conclusions regarding oxygen and VOC limitation were obtained. 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 Devanny (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.

In most cases, polluted air streams contain more than one pollutant. Recent experimental studies have shown that when the same type of biomass can biodegrade more than one pollutant, the pollutants are involved in kinetic interactions. If different types of biomass biodegrade each one of the pollutants, kinetic interactions are absent although a competitive interference for oxygen is always present. Based on the foregoing discussion, models that have been proposed for steady state biofiltration of VOC mixtures fall in two categories depending on whether the same or different biomass is involved in biofiltration of the various VOCs. Baltzis and Shareefdeen (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 (1995) 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 complicated 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. Hodge and Devinny (1995) have also proposed a model for transient biofiltration of a single VOC. They have assumed zero-order biodegradation kinetics, neglected oxygen effects, but introduced as a new feature the effects of dispersion on the process. Dispersion implies deviation from plug flow conditions regarding the passage of the polluted stream through the biofilter bed. Their results suggest that non-ideal flow (i.e., dispersion) effects are not significant and thus, all other models which consider plug flow of the air seem to be well justified.

The work performed in the present thesis relating to VOC biofiltration is based on the models of Shareefdeen et al. (1993) and Shareefdeen and Baltzis (1994) for steady state and transient operation, respectively.

As it has been mentioned in the Introduction, this thesis deals with an integrated air stripping/biofiltration process. The idea of using biofiltration for treatment of volatile pollutants removed from soils and aquifers is not novel. Saberiyan et al. (1994) have performed biofiltration studies on the removal of gasoline constituents in biofilters under steady state conditions and flows up to 8.5 ft³/min. This was a feasibility study targeting the design of a field-scale unit at a site involving a service station where soils and

groundwater were contaminated with gasoline. No data on the integrated process were reported.

Li (1995) has reported data from a field study with hydrocarbon vapors from a site occupied in the past by a gasoline service station. In his case, the biofilter was essentially the soil above the aquifer. Li worked with air flowrates of about 8 ft³/min (in the injection wells) and showed about 40% removal of total hydrocarbons and a 90% removal of BTEX constituents which constituted about 45% of the total hydrocarbons.

Chang and Yoon (1995) have also reported data from an actual integrated soil vapor extraction/biofiltration study at a gasoline contaminated site. Working with a 4.7 ft³ biofilter and flowrates of about 90 ft³/min they achieved hydrocarbon removal of 43% on the average.

Leson and Smith (1995) have performed studies funded by the Petroleum Environmental Research Forum (PERF) on the applicability of biofiltration for hydrocarbon vapor removal. Two of their studies involved integrated soil vapor extraction/biofiltration at two gasoline service stations. Working with flowrates of 5-40 ft³/min in one case and 18 ft³/min in another, and employing two different, commercially available, biofilters these investigators have concluded that the process is efficient for aromatic hydrocarbons but yields relatively poor results for aliphatic compounds. Finally, they used their data to estimate treatment cost on a preliminary basis.

No attempt was made to model the integrated process in the aforementioned studies.

CHAPTER 3

OBJECTIVES

The present study deals with an integrated air stripping/biofiltration process which is schematically shown in Figure 3-1.

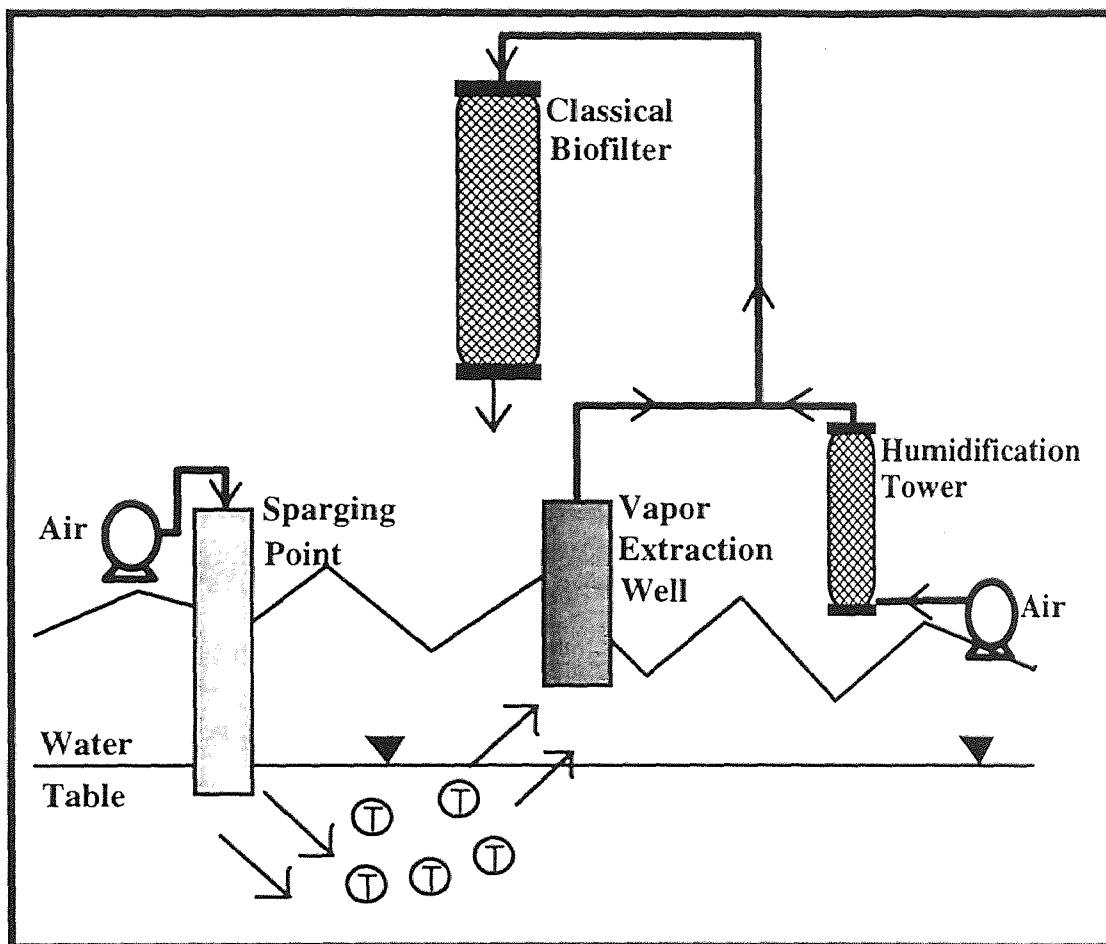


Figure 3-1 Schematic of the integrated air stripping/biofiltration process.

The ideas incorporated in the schematic of Figure 3-1 are the following. Air is passed through an 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 be 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.

The main objective of this study was to suggest a detailed methodology for the design of an integrated air stripping/biofiltration process to clean contaminated aquifers. The methodology should be such that it allows for calculations of the required biofilter volume and the time frame within which the contaminated aquifer could be remediated.

Based on a single pollutant, the specific objectives set for the design of the integrated process were the following.

1. The concentration of the pollutant in the aquifer at the end of the remediation operation should be at or below the Action Level in Groundwater for the said pollutant as per existing regulations.
2. The concentration of the pollutant in the air exiting the extraction wells should be, at

most, very close to the Threshold Limit Value (TLV) as per existing regulations.

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 pollutant concentration in the air exiting the aquifer, and this should be less than its TLV value.

3. The concentration of the pollutant at the exit of the biofilter should meet the Acceptable Source Impact Level (ASIL) as per existing.
4. The biofilter should be exposed to a relatively constant pollutant concentration over most of the time of the operation to avoid shock-loading effects.

The methodology was developed and tested using toluene as the model compound. The TLV and ASIL for toluene as per existing regulations are given in Table 3.1. These values were used in satisfying objectives 1 to 3.

Table 3.1 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 reason for selecting toluene as the model compound was that there is a lot of experimental information on biofiltration of this compound. Furthermore, the models used [Shareefdeen et al.(1993); Shareefdeen and Baltzis (1994)] in this study have been tested against toluene biofiltration data from both steady-state and transient biofilter operation. In an effort to further ensure that the proposed methodology leads to acceptable results for the case of toluene, the maximum toluene concentration in the air supplied to the biofilter was never allowed to exceed values for which there is experimental evidence that the biofiltration process actually works.

CHAPTER 4

BASIC MODEL EQUATIONS

The process considered in the present thesis consists in reality two separate processes. The first process is air stripping of the pollutant (toluene) from the aquifer. The assumptions and the model equations used in describing the air stripping process are presented in section 4.1.

Regarding biofiltration, the methodology discussed in Chapter 5 requires that calculations are performed under both steady state and transient biofilter operation. These equations are those of Shareefdeen et al. (1993) and Shareefdeen and Baltzis (1994), and are presented [for the sake of completeness] in sections 4.2 and 4.3.

4.1 Equations for the Contaminated Aquifer

Air stripping of the pollutant (toluene) from the aquifer has been described with equations 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) in the aquifer.

6. The contaminant is distributed between the liquid phase (aquifer) and the gas phase (air) in a way which assumes either equilibrium as dictated by Henry's law or a fraction of equilibrium, σ .

Based on the foregoing assumptions, one can write the following equations, which constitute a mass balance on the pollutant in the aquifer:

$$V_L \frac{dc_L}{dt} = -Q_G c_G \quad (4.1)$$

$$c_G = \sigma m_T c_L \quad (4.2)$$

Equations (4.1) and (4.2) lead to the following expressions, under the initial condition $c_L = c_{L_0}$ at $t = 0$,

$$c_L = c_{L_0} \exp\left(\frac{-Q_G \sigma m_T t}{V_L}\right) \quad (4.3)$$

$$c_G = \sigma m_T c_{L_0} \exp\left(\frac{-Q_G \sigma m_T t}{V_L}\right) \quad (4.4)$$

If the air from the extraction well is directly supplied to the biofilter, then $c_G = c_{Ti}$ and $Q_G = F$. If, on the other hand, the air from the extraction well is diluted with pure air before it is supplied to the biofilter then the following equation is valid:

$$c_{Ti} = \frac{\sigma m_T c_{L_0} Q_G}{F} \exp\left(\frac{-Q_G \sigma m_T t}{V_L}\right) \quad (4.5)$$

Equation (4.5) gives the time concentration profile of the pollutant at the entrance of the biofilter when a single flowrate, Q_G , is used for the air supplied to the sparging points. It should be noted that Q_G is a fraction of F .

Supplying air to the sparging points at a constant flowrate over the entire period of remediation of the aquifer is not really practical. One should try to maintain a relatively

constant pollutant concentration at the entrance of the biofilter over as long a period of time as possible. This can be achieved as follows. Initially, when the pollutant concentration in the aquifer is high, one should use a small (relative to F) value for Q_G . As the time progresses the value of Q_G should be increased.

Assume that the total remediation time is divided in n periods. The first period lasts from $t = 0$ to $t = t_1$ and the flowrate of the air supplied to the sparging points is Q_{G1} . The second period lasts from $t = t_1$ to $t = t_2$ and the flowrate of the air supplied to the aquifer is Q_{G2} . In general, the k -th period lasts from $t = t_{k-1}$ to $t = t_k$ and during it, the flowrate of the air supplied to the aquifer is Q_{Gk} . During the n -th and last period, no dilution of the air coming out of the extraction wells is needed and thus, $Q_{Gn} = F$.

According to the protocol described above, the inlet concentration of the pollutant to the biofilter during the first period is given by equation (4.5) provided that one substitutes Q_{G1} for Q_G . Thus,

$$0 < t \leq t_1 : \quad c_{Ti} = \frac{\sigma m_T c_{L0} Q_{G1}}{F} \exp\left(\frac{-Q_{G1} \sigma m_T t}{V_L}\right) \quad (4.6)$$

During the second phase, equation (4.1) has to be modified to

$$V_L \frac{dc_L}{dt} = -Q_{G2} c_G \quad (4.7)$$

and has to be integrated subject to the following initial condition :

$$\text{at } t = t_1, \quad c_L = c_{L0} \exp\left(\frac{-Q_{G1} \sigma m_T t_1}{V_L}\right) \quad (4.8)$$

Equations (4.7) and (4.8) along with (4.2) lead to the following :

$$t_1 < t \leq t_2: \quad c_{Ti} = \frac{\sigma m_T c_{Lo} Q_{G2}}{F} \exp\left(\frac{-Q_{G1} \sigma m_T t_1}{V_L}\right) \exp\left[\frac{-Q_{G2} \sigma m_T (t - t_1)}{V_L}\right] \quad (4.9)$$

In general, the time concentration profile at the inlet of the biofilter during the k-th period of treatment is given by the following expression.

For $t_{k-1} < t \leq t_k$:

$$c_{Ti} = \frac{\sigma m_T c_{Lo} Q_{Gk}}{F} \exp\left[\frac{\sigma m_T \sum_{r=0}^{k-1} t_r (Q_{Gr+1} - Q_{Gr})}{V_L}\right] \exp\left(\frac{-Q_{Gk} \sigma m_T t}{V_L}\right) \quad (4.10)$$

with $t_0 = 0$ and $Q_{G0} = 0$.

4.2 Equations for Steady-State Biofiltration

The equations describing steady state biofiltration of a single compound (toluene) are practically those of Shareefdeen et al. (1993). There is a minor modification which has been introduced here. The aforementioned model assumes that the pollutant at the air/biofilm interface is in equilibrium distribution as dictated by Henry's law. The modification allows for non-equilibrium distribution described by a fraction σ . The modified model equations are as follows.

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

$$f(X_v) D_{Tw} \frac{d^2 s_T}{dx^2} = \frac{X_v}{Y_T} \mu_T(s_T, s_O) \quad (4.11)$$

with boundary conditions,

$$s_T = \frac{c_T}{\sigma m_T} \quad \text{at } x = 0 \quad (4.12)$$

$$\frac{ds_T}{dx} = 0 \quad \text{at } x = \delta \quad (4.13)$$

Equation (4.11) implies that the rate of diffusion of toluene 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:

$$f(X_v)D_{ow} \frac{d^2s_O}{dx^2} = \frac{X_v}{Y_{OT}} \mu_T(s_T, s_O) \quad (4.14)$$

with boundary conditions,

$$s_O = \frac{c_O}{m_O} \quad \text{at } x = 0 \quad (4.15)$$

$$\frac{ds_O}{dx} = 0 \quad \text{at } x = \delta \quad (4.16)$$

Equation (4.14) 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 toluene in the airstream at a position h along the biofilter bed:

$$\frac{H}{\tau} \frac{dc_T}{dh} = A_s f(X_v) D_{Tw} \left[\frac{ds_T}{dx} \right]_{x=0} \quad (4.17)$$

with boundary condition,

$$c_T = c_{Ti} \quad \text{at } h = 0 \quad (4.18)$$

Equation (4.17) implies that the rate of loss of toluene from the air along the biofilter is equal to the rate at which toluene 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:

$$\frac{H}{\tau} \frac{dc_O}{dh} = A_s f(X_v) D_{ow} \left[\frac{ds_O}{dx} \right]_{x=0} \quad (4.19)$$

with boundary condition,

$$c_O = c_{Oi} \quad \text{at } h=0 \quad (4.20)$$

The terms in equation (4.19) have the same meaning for oxygen as those in equation (4.17) have for toluene.

Function $\mu_T(s_T, s_O)$ which appears in equations (4.11) and (4.14) is given by,

$$\mu_T(s_T, s_O) = \frac{\mu_T^* s_T s_O}{(K_T + s_T + s_T^2 / K_{\pi})(K_O + s_O)} \quad (4.21)$$

Equation (4.21) implies that the degradation kinetics of toluene follow an Andrews expression with respect to the availability of the carbon source (toluene), and a Monod expression with respect to the availability of oxygen. This has been experimentally confirmed [Shareefdeen, (1994)].

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,

$$\begin{aligned} \bar{s}_T &= \frac{s_T}{K_T}, & \bar{s}_O &= \frac{s_O}{K_O}, & \bar{c}_T &= \frac{c_T}{c_{Ti}}, & \bar{c}_O &= \frac{c_O}{c_{Oi}}, & \gamma &= \frac{K_T}{K_{\pi}}, \\ \lambda &= \frac{D_{Tw} K_T Y_T}{D_{Ow} K_O Y_{OT}}, & \phi^2 &= \frac{\mu_T^* \delta X_v}{f(X_v) D_{Tw} K_T Y_T}, & \eta &= \frac{A_s f(X_v) D_{Tw} K_T \tau}{\delta c_{Ti}}, \\ \omega &= \frac{K_O D_{Ow} c_{Ti}}{K_T D_{Tw} c_{Oi}}, & \epsilon_1 &= \frac{c_{Ti}}{K_T \sigma m_T}, & \epsilon_2 &= \frac{c_{Oi}}{K_O m_O}, & \theta &= \frac{x}{\delta}, & z &= \frac{h}{H} \end{aligned}$$

Equations (4.11)-(4.20), when expression (4.21) is also taken into account, take correspondingly the following form.

$$\frac{d^2 \bar{s}_T}{d\theta^2} = \phi^2 \frac{\bar{s}_T \bar{s}_O}{(1 + \bar{s}_T + \gamma \bar{s}_T^2)(1 + \bar{s}_O)} \quad (4.22)$$

$$\bar{s}_T = \varepsilon_1 \bar{c}_T \quad \text{at } \theta = 0 \quad (4.23)$$

$$\frac{d\bar{s}_T}{d\theta} = 0 \quad \text{at } \theta = 1 \quad (4.24)$$

$$\frac{d^2 \bar{s}_O}{d\theta^2} = \phi^2 \lambda \frac{\bar{s}_T \bar{s}_O}{(1 + \bar{s}_T + \gamma \bar{s}_T^2)(1 + \bar{s}_O)} \quad (4.25)$$

$$\bar{s}_O = \varepsilon_2 \bar{c}_O \quad \text{at } \theta = 0 \quad (4.26)$$

$$\frac{d\bar{s}_O}{d\theta} = 0 \quad \text{at } \theta = 1 \quad (4.27)$$

$$\frac{d\bar{c}_T}{dz} = \eta \left[\frac{d\bar{s}_T}{d\theta} \right]_{\theta=0} \quad (4.28)$$

$$\bar{c}_T = 1 \quad \text{at } z = 0 \quad (4.29)$$

$$\frac{d\bar{c}_O}{dz} = \eta \omega \left[\frac{d\bar{s}_O}{d\theta} \right]_{\theta=0} \quad (4.30)$$

$$\bar{c}_O = 1 \quad \text{at } z = 0 \quad (4.31)$$

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:

$$\bar{c}_T = \frac{\bar{c}_O - 1}{\lambda \omega} + 1 \quad (4.32)$$

$$\bar{s}_T = \frac{\bar{s}_O}{\lambda} - \varepsilon_1 \left(\frac{1}{\lambda\omega} - 1 \right) - \frac{1}{\lambda} \left(\varepsilon_2 - \frac{\varepsilon_1}{\omega} \right) \bar{c}_O \quad (4.33)$$

It is easy to show that equations (4.32) and (4.33) can be equivalently expressed as follows:

$$\bar{c}_O = \lambda\omega(\bar{c}_T - 1) + 1 \quad (4.34)$$

$$\bar{s}_O = \lambda\bar{s}_T - \lambda(\varepsilon_1 - \omega\varepsilon_2)\bar{c}_T - \varepsilon_2(\lambda\omega - 1) \quad (4.35)$$

Because of relations (4.32)-(4.35) one needs to solve two rather than four differential equations. There are two possible sets; set 1: equations (4.22)-(4.24), (4.28), (4.29), (4.34), and (4.35); set 2: equations (4.25)-(4.27), and (4.30)-(4.33). From the numerical point of view, one needs to keep differential equations referring to variables which exhibit the largest gradient. Hence, in cases where toluene 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 toluene one needs to work with the equations of set 2.

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

$$V = \tau F \quad (4.36)$$

The model equations were solved through the use of computer codes which are based on the use of the orthogonal collocation method for solving equation (4.22) [or (4.25)], and the Runge-Kutta method for solving equation (4.28) [or (4.30)]. 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 B of this thesis.

4.3 Equations for Transient Biofiltration

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), and modified for accounting possible deviations from equilibrium are as follows:

I. Mass balance for toluene and for oxygen in the biofilm:

$$\frac{\partial s_T}{\partial t} = f(X_v) D_{Tw} \frac{\partial^2 s_T}{\partial x^2} - \frac{X_v}{Y_T} \mu_T(s_T, s_O) \quad (4.37)$$

$$\frac{\partial s_O}{\partial t} = f(X_v) D_{Ow} \frac{\partial^2 s_O}{\partial x^2} - \frac{X_v}{Y_O} \mu_T(s_T, s_O) \quad (4.38)$$

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

$$v \frac{\partial c_T}{\partial t} = -u_g \frac{\partial c_T}{\partial h} + D_{Tw} f(X_v) \alpha A_s^* \left(\frac{\partial s_T}{\partial x} \right)_{x=0} - k_a (1 - \alpha) A_s^* (c_T - c_T^*) \quad (4.39)$$

$$v \frac{\partial c_O}{\partial t} = -u_g \frac{\partial c_O}{\partial h} + D_{Ow} f(X_v) \alpha A_s^* \left(\frac{\partial s_O}{\partial x} \right)_{x=0} \quad (4.40)$$

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

$$(1 - v) \rho_p \frac{\partial c_{Tp}}{\partial t} = k_a (1 - \alpha) A_s^* (c_T - c_T^*) \quad (4.41)$$

Equations (4.37)-(4.40) are the unsteady state versions of equations (4.11), (4.14), (4.17), and (4.19), respectively. The sole difference is the last term in equation (4.39) which stands for the rate of mass transfer of toluene to the solid particles (excluding biolayer). Equation (4.41) expresses that the rate of accumulation of toluene in the solid particles is

equal to the rate of mass transfer of toluene 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 $\mu_T(s_T, s_O)$ appearing in equations (4.37) and (4.38) expresses the kinetics of biodegradation and -as was also the case in Section 4.2- it is given by the following expression:

$$\mu_T(s_T, s_O) = \frac{\mu_T^* s_T}{K_T + s_T + \frac{s_T^2}{K_{IT}}} \frac{s_O}{K_{OT} + s_O} \quad (4.42)$$

The driving force for the mass transfer of toluene from the air (gas phase) to the particles (solid phase) is $c_T - c_T^*$, as indicated in equations (4.39) and (4.41). Concentration c_T^* is related to the concentration of toluene in the solid phase through an adsorption isotherm, which has been found [Shareefdeen (1994)] to follow the Freundlich equation. Hence, one can write

$$c_{Tp} = k_d (c_T^*)^n \quad (4.43)$$

Equations (4.37)-(4.41) have not been exactly solved to date. They have been solved through an approximation which introduces the use of effectiveness factors defined as,

$$e_T = \frac{f(X_v) D_{Tw} \left(\frac{\partial s_T}{\partial x} \right)_{x=0}}{\delta \frac{X_v}{Y_T} [\mu_T(s_T, s_O)]_{x=0}} \quad (4.44)$$

$$e_o = \frac{f(X_v) D_{ow} \left(\frac{\partial s_o}{\partial x} \right)_{x=0}}{\delta \frac{X_v}{Y_{OT}} [\mu_T(s_T, s_o)]_{x=0}} \quad (4.45)$$

Actually it can be easily shown that $e_T = e_o$. The use of the effectiveness factor allows for omission of equations (4.37) and (4.38). 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

$$\frac{\partial \bar{c}_T}{\partial \zeta} = -\frac{1}{v} \frac{\partial \bar{c}_T}{\partial z} - \beta_1 g(\bar{c}_T, \bar{c}_o) - \beta_3 (\bar{c}_T - \bar{c}_T^*) \quad (4.46)$$

$$\frac{\partial \bar{c}_o}{\partial \zeta} = -\frac{1}{v} \frac{\partial \bar{c}_o}{\partial z} - \beta_2 g(\bar{c}_T, \bar{c}_o) \quad (4.47)$$

$$\frac{\partial \bar{c}_{TP}}{\partial \zeta} = \beta_3 (\bar{c}_T - \bar{c}_T^*) \quad (4.48)$$

where

$$g(\bar{c}_T, \bar{c}_o) = \frac{\varepsilon_1 \bar{c}_T}{(1 + \varepsilon_1 \bar{c}_T + \varepsilon_1^2 \gamma \bar{c}_T^2)} \frac{\varepsilon_2 \bar{c}_o}{(1 + \varepsilon_2 \bar{c}_o)} \quad (4.49)$$

and

$$\bar{c}_T^* = \psi(\bar{c}_{TP})^{\frac{1}{n}} \quad (4.50)$$

The initial and boundary conditions for equations (4.46)-(4.50) are,

$$\bar{c}_T = 1, \quad \bar{c}_o = 1, \quad \bar{c}_{TP} = \bar{c}_{TP,0}(0) \quad \text{at } \zeta = 0 \text{ and } z = 0 \quad (4.51)$$

$$\bar{c}_T = \bar{c}_{T,0}(z), \quad \bar{c}_o = \bar{c}_{o,0}(z), \quad \bar{c}_{TP} = \bar{c}_{TP,0}(z) \quad \text{at } \zeta = 0 \text{ and } 0 < z \leq 1 \quad (4.52)$$

$$\bar{c}_T = 1, \quad \bar{c}_o = 1 \quad \text{at } \zeta \geq 0 \text{ and } z = 0 \quad (4.53)$$

Equations (4.46)-(4.48) are in dimensionless form, and the dimensionless quantities appearing in them are related to the dimensional quantities of the original equations through the following,

$$\begin{aligned} \bar{c}_T &= \frac{c_T}{c_{Ti}}, & \bar{c}_T^* &= \frac{c_T^*}{c_{Ti}}, & \bar{c}_O &= \frac{c_O}{c_{Oi}}, & \bar{c}_{TP} &= \frac{(1-\nu)\rho_p c_{TP}}{\nu c_{Ti}}, \\ z &= \frac{h}{H}, & \zeta &= \frac{t}{\tau}, & \varepsilon_1 &= \frac{c_{Ti}}{\sigma m_T K_T}, \varepsilon_2 &= \frac{c_{Oi}}{m_O K_O}, & \gamma &= \frac{K_T}{K_{IT}}, \\ \beta_1 &= \frac{e_T \alpha \delta A_S^* X_v \mu_T^* \tau}{Y_T c_{Ti} \nu}, & \beta_2 &= \frac{e_O \alpha \delta A_S^* X_v \mu_T^* \tau}{Y_O c_{Oi} \nu}, \\ \beta_3 &= \frac{k_a (1-\alpha) A_S^* \tau}{\nu}, & \psi &= \frac{1}{c_{Ti}} \left[\frac{\nu c_{Ti}}{(1-\nu)\rho_p k_d} \right]^{\frac{1}{n}} \end{aligned}$$

Solution of equations (4.47)-(4.50) when c_{Ti} is constant was obtained in the past through 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. This basic code of Shareefdeen (1994), was substantially modified during the course of this thesis in order to account for a temporally varying c_{Ti} . This new (modified) code is given in Appendix C of this thesis.

CHAPTER 5

METHODOLOGY AND PRELIMINARY CALCULATIONS

The methodology proposed consists of two major steps. The first relates to the selection of the profile of the pollutant at the inlet of the biofilter. It is proposed that for any given pollutant, its maximum concentration at the inlet of the biofilter never exceeds a value for which there is experimental evidence that the biofiltration process actually works. Selection of the inlet profile is related to the selection of air flow rates used in the air stripping process. The second major step relates to calculations for determining the required size of the biofilter bed.

Since toluene was selected as the model compound for implementing the proposed methodology, in the following the steps of the methodology itself are described in relation to toluene.

It was first decided that the maximum toluene concentration in the air supplied to the biofilter should never exceed the value of 9.2 g/m^3 . This choice was based on the fact that experiments with inlet concentrations of up to 9.2 g/m^3 have been performed [Shareefdeen (1994), Wojdyla and Baltzis (unpublished)] and they worked indeed with such concentrations. The value of 9.2 g/m^3 at the inlet of the biofilter represents essentially the “worst case scenario” since this is the maximum toluene concentration value to which any part of the biofilter would be exposed at any instant of time.

The methodology proposes that a constant value for the flowrate (F) of the air supplied to the biofilter is used. A value of $F = 51 \text{ m}^3/\text{h}$ (or $30 \text{ ft}^3/\text{min}$, i.e., 30 cfm) was

The methodology proposes that a constant value for the flowrate (F) of the air supplied to the biofilter is used. A value of $F = 51 \text{ m}^3/\text{h}$ (or $30 \text{ ft}^3/\text{min}$, i.e., 30 cfm) was selected for the calculations performed here. This value was set almost arbitrarily, but it also falls in the range of values used in experimental pilot-scale studies (see Chapter 2) involving soil vapor extraction integrated with biofiltration.

With the value of F set, the toluene concentration profile at the inlet of the biofilter (c_{Ti}) was determined through the use of equation (4.10). The objectives in determining c_{Ti} were threefold: the profile to be uniform during most of the time over which cleaning of the aquifer occurs (objective 4 in Chapter 3), the value of c_{Ti} never to exceed 9.2 g/m^3 , and the value of c_{Ti} at the end of remediation to correspond to a toluene concentration in the aquifer not higher than the action level of toluene in groundwater (Table 3.1). Determination of c_{Ti} required a trial and error approach. In this approach one can vary the number of time intervals into which the total remediation time is divided (see also section 4.1), the value of the air flowrate in the extraction well during each time interval (i.e., vary the value of Q_{Gk} , $k = 1, \dots, n$), and the extent of each time interval. It was decided to use as values for Q_{Gk} simple fractions of the values of F . The number of time intervals is also referred to as number of dilutions, implying dilution of the air coming out of the extraction well with pure (non-contaminated) air.

Once the toluene concentration profile at the inlet of the biofilter (c_{Ti}) was determined, its maximum value was recorded and used in the following procedure for determining the required volume of the biofilter packing material (V_p).

The case where a biofilter operates with time invariant inputs and reaches steady state was considered. This case is described by the equations presented in section 4.2 of

this thesis. As inlet concentration the maximum value of c_{Ti} (see above) was used. The equations were solved via the computer code given in Appendix B.

In order to solve the model equations one needs to know the values of the various parameters appearing in the model. For toluene, which is the compound considered here, the model parameter values were taken from Shareefdeen (1994) and are shown in Table 5.1. The value of σ is not given in the table because it was varied. The values used were 1, 0.8, 0.5, and 0.3.

Table 5.1 Model parameter values for toluene*

| 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 | — |
| v | 0.3 | — |
| ρ_P | 4.28×10^5 | g/m^3 |
| μ_T^* | 1.50 | h^{-1} |

*From Shareefdeen (1994)

As discussed in Chapter 4, there are two different sets of equations which need to be used under steady state operation depending on whether oxygen or toluene gets depleted first in the biolayer. Which compound (toluene or oxygen) gets depleted first in the biofilm depends on the concentration of toluene (or the pollutant in general) in the air. It also depends on the value of parameter σ . The maximum toluene concentration in the air which leads to depletion of toluene before oxygen in the biofilm was determined and the limits for various σ values are reported in Table 5.2. Determination of these limits requires a trial and error approach. After these limits were determined, the equations shown in section 4.2 were solved as follows.

Table 5.2 Conditions under which toluene gets depleted before oxygen in the biolayers present in the biofilter.

| Relative volatility (σ) | Maximum toluene concentration in the air (g/m^3) |
|----------------------------------|---|
| 1 | 2.80 |
| 0.8 | 2.25 |
| 0.5 | 1.40 |
| 0.3 | 0.80 |

Equations of set 2 in section 4.2 (oxygen depleted first) were used for calculating the space time (τ_1) needed for the toluene concentration to drop from the maximum c_{T1} value at the inlet to the value shown in Table 5.2. This required a trial and error approach as the code runs when the value for the space time is given. Hence, various values of τ_1 were tried till the one giving the corresponding value in Table 5.2 as an “exit” concentration was determined. Subsequently, equations of set 1 in section 4.2 (toluene

depleted first) were used for determining the space time (τ_2) needed for the toluene concentration to drop from the inlet value taken from Table 5.2 to an exit value of about 0.28 g/m^3 which is the ASIL value for toluene (Table 3.1). As for τ_1 , determination of τ_2 required a trial and error approach.

Having the values of τ_1 and τ_2 , the space time (τ) in a single biofilter achieving reduction of the toluene concentration from maximum c_{Ti} at its inlet to the ASIL value at its exit was determined as $\tau = \tau_1 + \tau_2$. Clearly, this biofilter is comprised of two zones. In the zone close to the entrance oxygen is depleted in the biolayers whereas in the zone close to the exit of the filter bed, toluene is depleted in the biolayers.

Having the value for τ and since the value of F was set at $51 \text{ m}^3/\text{h}$ the volume of the biofilter packing material (V_p) was determined via the equation $\tau = V_p/F$.

Determination of the value of V_p according to the methodology described above is not the final step in the design of the integrated process. Since the biofilter is operating under continuously varying [as dictated by equation (4.10)] rather than constant concentration, one needs to ensure that under the real conditions of operation the concentration profile of toluene at the exit of the biofilter is such that the ASIL toluene value is never exceeded. The following methodology was thus employed.

The model equations shown in section 4.3 of Chapter 4 were used and solved via a modification of the computer code originally written by Shareefdeen. Modification was needed because the original code (Shareefdeen, 1994) works only under constant concentration values at the inlet of the biofilter. The modifications of the code, reflecting an inlet toluene concentration which varies according to equation (4.10), are given in Appendix C of this thesis.

In order to solve the transient equations the values of the model parameters are again needed. In addition to the parameters involved in the steady-state model, the transient equations involve extra parameters such as adsorption constants for toluene on the packing, porosity of the bed, etc. These parameter values are shown in Table 5.1 and were taken from Shareefdeen and Baltzis (1994). For V_p , the value determined from the calculations with the steady-state model described in the preceding paragraphs was used. In addition to the parameter values shown in Table 5.1 the transient model cannot be solved unless the values of δ and e_T or e_o are known. In fact, a closer inspection of the equations in section 4.3 showed that the values of δ and e_T do not need to be individually known. What is really needed is their product δe_T (or, δe). Expressions for δe were determined by solving the steady state equations (section 4.2), determining values for δe and then regressing them to a polynomial expression of c_T (toluene concentration in the air). The expressions for δe are shown in Table 5.3.

Table 5.3 Expressions for parameter δe as a function of toluene concentration in the air (c_T , in g/m^3).

| Value of σ | Expression for δe | r^2 -value for regression |
|-------------------|--|-----------------------------|
| 1 | $0.014c_T^3 - 0.281c_T^2 + 1.837c_T + 6.097$ | 0.99 |
| 0.8 | $0.016c_T^3 - 0.309c_T^2 + 1.905c_T + 6.330$ | 0.97 |
| 0.5 | $0.015c_T^3 - 0.276c_T^2 + 1.658c_T + 7.085$ | 0.93 |
| 0.3 | $0.008c_T^3 - 0.145c_T^2 + 1.110c_T + 7.854$ | 0.95 |

If the toluene concentration profile as predicted by the transient model is such that the ASIL value is never exceeded, the design of the process has been completed. If the exit profile shows values above ASIL at any instant of time during the process, the value of V_p (obtained from the steady state calculations, as discussed previously) needs to be increased up to the point that ASIL requirements are met.

CHAPTER 6

RESULTS AND DISCUSSION

Eight cases were considered for the integrated air stripping/biofiltration process, and are shown in Table 6.1. These cases differ in the number of dilutions (intervals into which the aquifer remediation time is divided) and the value of σ . As has been mentioned earlier, $\sigma = 1$ implies that toluene is in equilibrium distribution between air and water. A value of σ less than 1 implies that equilibrium has not been reached.

For Case 1 of Table 6.1 the extent of time of each time interval and the value of the air flowrate in the air sparging and extraction wells for each time interval are given in Table 6.2. Values of the same quantities for Cases 2-8 of Table 6.1 are given in Tables A.1 to A.7 of Appendix A.

For each of the eight cases of Table 6.1 the methodology described in Chapter 5 was followed. Regarding the final step, it was found that the value of V_p determined from the steady state equations always led to a toluene concentration at the exit of the biofilter which met ASIL requirements. This is probably due to the fact that toluene does not adsorb strongly onto the packing material. Thus, during periods of decrease of the toluene concentration at the inlet of the biofilter, the amount of toluene desorbed from the packing material is small and thus, it does not lead to concentrations above ASIL at the exit of the biofilter. One could anticipate that for compounds less volatile than toluene ASIL values might be exceeded if the value of V_p from steady state calculations is used.

Table 6.1 Number of dilutions of the airstream exiting the aquifer and values of relative volatility, σ , for the cases considered in the design of the integrated air stripping/biofiltration process.

| Case | No. of dilutions | σ |
|------|------------------|----------|
| 1 | 8 | 1 |
| 2 | 8 | 0.8 |
| 3 | 8 | 0.5 |
| 4 | 8 | 0.3 |
| 5 | 7 | 1 |
| 6 | 7 | 0.8 |
| 7 | 7 | 0.5 |
| 8 | 7 | 0.3 |

Table 6.2 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 1 of Table 6.1

| Period (value of k) | Air flowrate (Q_{GK}) (m^3/h) | Extent of period (h) |
|------------------------|--|-------------------------|
| 1 | 5.1 | 160 |
| 2 | 6.375 | 170 |
| 3 | 8.5 | 120 |
| 4 | 10.2 | 120 |
| 5 | 12.75 | 120 |
| 6 | 17.0 | 120 |
| 7 | 25.5 | 120 |
| 8 | 51.0 | 208 |

This is due to the fact that a less volatile compound is expected to adsorb more strongly onto the packing, since “adsorption” here primarily means dissolution in the water retained in the pores of the packing. Thus, desorption effects would have a significant impact on the process [Baltzis (1994), Baltzis and Androutsopoulou (1994)].

Time concentration profiles of toluene at the entrance and exit of the biofilter for Case 1 of Table 6.1 are shown in Figure 6-1. For all other cases considered, the concentration profiles are shown in Figures A-1 through A-7 of Appendix A.

For each one of the eight cases considered, and for process design purposes, the two most important quantities are the time needed for cleaning the aquifer and the volume (size) of the required biofilter. The values of these quantities are shown in Table 6.3, along with the maximum values of the toluene concentration at the inlet and outlet of the biofilter. The air residence time in the biofilter (space time, τ) is also given in Table 6.3.

From Table 6.3 one can observe that for a given value of time intervals (dilutions) as the value of σ decreases the required biofilter volume decreases and the time required for cleaning the aquifer increases. However, this trend should not be taken as a generally valid conclusion. In fact, what was done here was to first consider Case 1, optimize the inlet concentration profile (c_{Ti}) and then use the values of Q_{Gk} of this profile for optimizing the inlet profiles of Cases 2-4 which involve the same number of dilutions as Case 1. Hence, optimization here was done under a constraint (values of Q_{Gk} set) and it involved determination of the extent of time of each time interval. This has led to c_{Ti} values [see graphs (a) in Figures 6-1 and A-1 through A-4] which decrease with the value of σ . If the values of Q_{Gk} were not preset in Cases 2-4, in fact if higher Q_{G1} values were

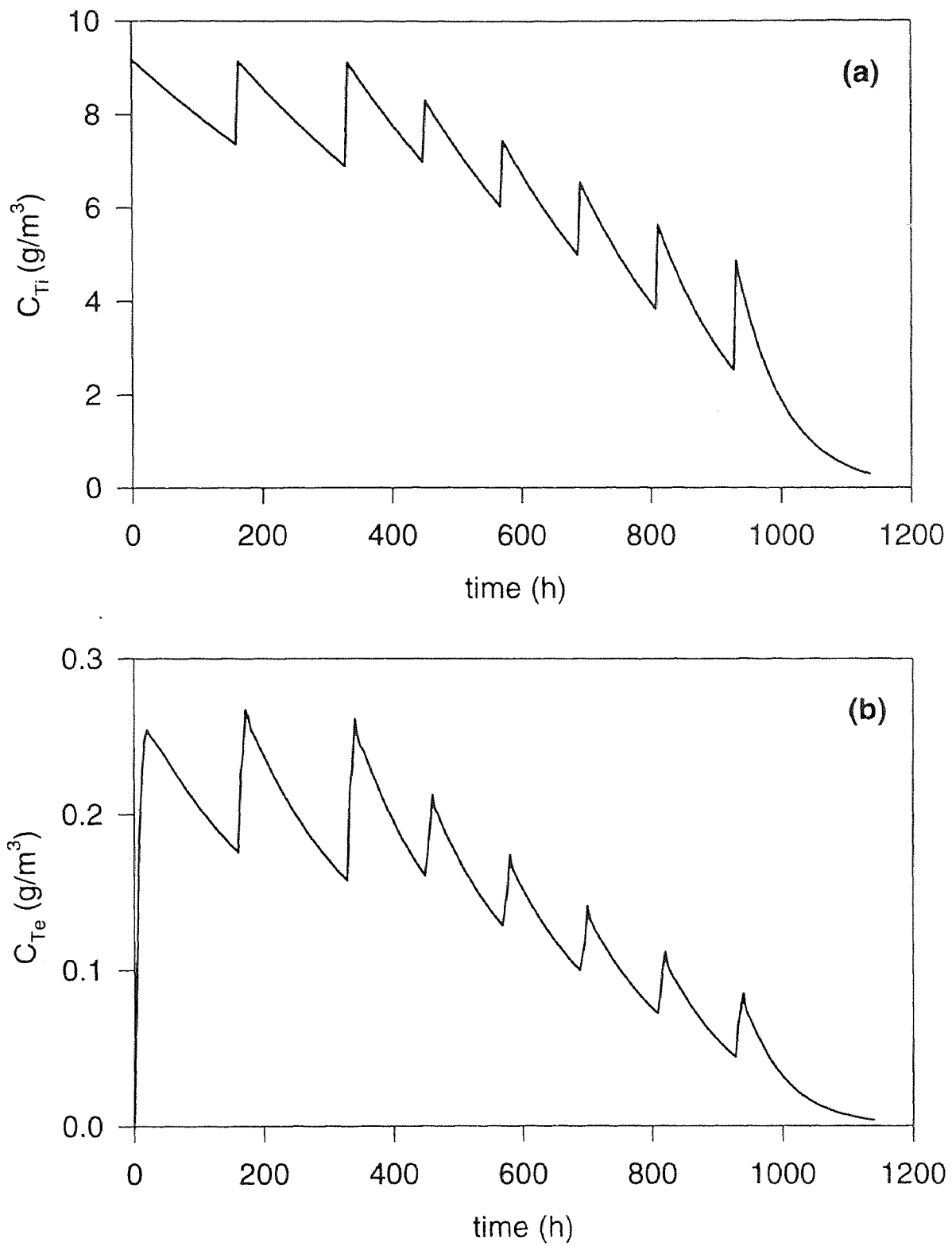


Figure 6-1 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter for Case 1 (see Table 6.1).

Table 6.3 Results from the design studies on the integrated air stripping/biofiltration process for Cases 1-8 of Table 6.1.

| C A S E | Maximum Inlet Toluene Concentration (g/m³) | Maximum Outlet Toluene Concentration (g/m³) | Required Volume of Biofilter (m³) | Air residence time in biofilter (min) | Time needed to clean the aquifer (h) |
|----------------------------|--|---|---|--|---|
| 1 | 9.18 | 0.267 | 14.62 | 17.2 | 1,138 |
| 2 | 8.41 | 0.236 | 12.50 | 14.7 | 1,280 |
| 3 | 6.37 | 0.216 | 8.25 | 9.7 | 1,752 |
| 4 | 6.66 | 0.207 | 7.82 | 9.2 | 2,382 |
| 5 | 9.18 | 0.279 | 14.62 | 17.2 | 1,170 |
| 6 | 8.29 | 0.238 | 12.33 | 14.5 | 1,250 |
| 7 | 7.53 | 0.209 | 9.61 | 11.3 | 1,750 |
| 8 | 4.95 | 0.261 | 5.78 | 6.8 | 2,720 |

used, the maximum c_{Ti} values would be close to the 9.2 g/m^3 value of Case 1 and thus, the value of σ would have no impact on the size of the biofilter. It should be mentioned though that with higher Q_{G1} values it becomes much more difficult to maintain a somewhat constant c_{Ti} profile at the beginning of the process.

Regarding objective 2 (Chapter 3), the assumptions made in this study imply that the maximum concentration of toluene in the extraction well is the one predicted by equation (4.4) when one sets $t = 0$ and thus, it is independent of the Q_G values. When $\sigma = 1$ the maximum c_G value exceeds the TLV value. For the case of $\sigma = 1$ the value of Q_{G1} affects the extent of time over which the TLV regulation is not met. It should be mentioned however, that in reality the Q_G values do affect the maximum concentration in the extraction well. In fact, the higher is Q_G , the smaller is the amount of time the air spends in the aquifer and thus, the lower is the probability that equilibrium distribution of toluene is achieved. Hence, a high Q_G value should imply a low σ value and thus, the c_G value is affected [eqn. (4.4)] by Q_G even at $t = 0$.

Because of the fact that toluene can be adsorbed onto the packing material, a question which arises is the following: If for whatever reason the biomass in the biofilter is inactive, how long would it take for the toluene concentration exiting the biofilter to reach the ASIL value, and/or to become identical with that of the concentration at the inlet?

Figures 6-2 and A-8 show toluene concentration profiles at the inlet and outlet of the biofilters assuming completely inactive biomass. The inlet profiles of Figures 6-2 and A-8 are identical, respectively, to those for Case 1 and Case 3 in Table 6.1. As can be seen from these graphs, a few hours after process initiation the exit profiles become

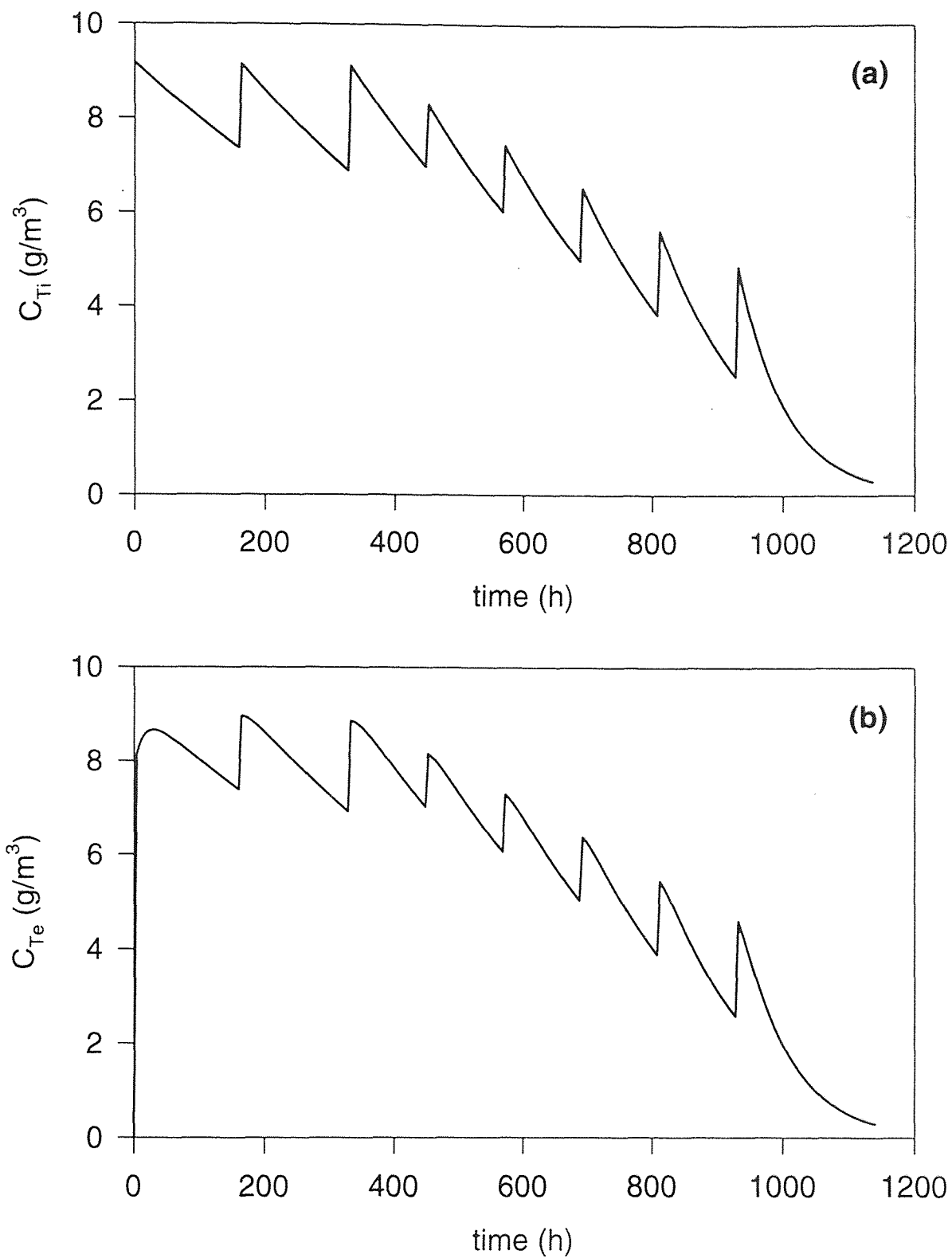


Figure 6-2 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter unit when the biomass is completely inactive (dead). These profiles correspond to Case 1 of Table 6.1.

identical to those at the inlet of the biofilter. The exit concentration profiles were obtained by using the code given in Appendix C and setting the kinetic parameter μ_T^* (see Table 5.1) equal to zero.

As a further investigation into the question of inactive biomass, the following calculations were performed. For each one of the cases in Table 6.1 the corresponding maximum inlet toluene concentration was taken (Table 6.3). It was then assumed that the biofilter operates under this concentration as a constant input. The original transient code of Shareefdeen (1994) was used and the times needed for the exit concentration to reach the ASIL value (time denoted by t_{ASIL}) and 95% of the inlet concentration value (time denoted by $t_{95\%c_{Ti}}$) were calculated. These values are given in Table A.8 of Appendix A. Two examples of exit concentration profiles are also shown in Figures A-9 and A-10 of Appendix A. The profiles shown in Figure A-9 are blow-ups of the profiles of Figure A-10 for low values of time. For comparison purposes, in Figure A-9 the corresponding profiles assuming completely active biomass are also shown. It is easy to see from Table A.8 that an inactive biofilter reaches (and then exceeds) the ASIL value at its exit at very small values of time. This implies that looking at an exit concentration profile (e.g. graphs (b) in Figures 6-1 and A-1 through A-7) which over extended time periods stays below the ASIL value should be a clear indication that biofiltration, not just adsorption, is occurring in the biofilter bed.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

This thesis has dealt with the design, based on quantitative and predictive engineering models, of an integrated air stripping/biofiltration process for cleaning a contaminated aquifer. Its major contribution is the development of a methodology for the design. This methodology is a refinement of the one first discussed by Cohen (1996). The methodology was implemented for eight case studies during which deviations from equilibrium were considered for the distribution of the pollutant between the aquifer water and the air used in the stripping process. Such deviations have never been considered in past studies with the integrated process. Another major contribution of the work presented in this thesis is the development of a modified transient biofiltration code which can run under continuously varying inputs (pollutant concentration).

The results of this study show that aquifer treatment can be achieved with biofilters which are not excessively large and within reasonable time frames. The results also show that when the clean-up time is divided into a given number of intervals and when the flow rate of air through the aquifer during each time interval is set, deviation from equilibrium (i.e., decreasing σ -values) leads to increased times for clean-up of the aquifer although clean-up is achieved with the use of smaller biofilters.

The methodology proposed here can be relatively easily used in preliminary decision making during the phase of technology options evaluation. It can be also used for guiding pilot testing and design of an actual process. The component of the

methodology referring to biofiltration is detailed, well-developed, and can be used with confidence. However, the component dealing with air stripping has been based on equations which were derived by making some assumptions that may prove drastic. Namely, it was assumed that the pollutant is uniformly distributed within the aquifer at all times, the fractional equilibrium (σ) achieved is independent of the air flowrate used, and the pollutant does not interact with the soil (i.e., sorption is not significant). It is recommended that future studies relax the foregoing assumptions, and also consider factors such as the radius of influence for the air forced into the soil as well as the possible (and partial) treatment of the pollutants by the organisms present in the aquifer water and on the soil. Relaxation of the assumptions made in the present study is not expected to change the methodology per se, but the results may be quantitatively different from the ones obtained here.

Future studies should also be performed with aquifers contaminated with mixtures of pollutants. In such studies the effects of kinetic interactions could be investigated but, most importantly, one could test the potential effects of differences in the Henry's constants for the pollutants. One could anticipate that the least volatile compound determines the clean-up time and the most volatile one determines the size of the biofilter. These intuitive predictions need to be checked with actual calculations.

APPENDIX A

**RESULTS FROM THE STUDIES ON THE INTEGRATED
AIR STRIPPING /BIOFILTRATION PROCESS**

Table A.1 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 2 of Table 6.1.

| Period (value of k) | Air flowrate (Q_{GR}) (m^3/h) | Extent of period (h) |
|------------------------|--|-------------------------|
| 1 | 5.1 | 150 |
| 2 | 6.375 | 150 |
| 3 | 8.5 | 150 |
| 4 | 10.2 | 150 |
| 5 | 12.75 | 150 |
| 6 | 17.0 | 150 |
| 7 | 25.5 | 150 |
| 8 | 51.0 | 230 |

Table A.2 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 3 of Table 6.1.

| Period (value of k) | Air flowrate (Q_{Gk}) (m^3/h) | Extent of period (h) |
|------------------------|--|-------------------------|
| 1 | 5.1 | 170 |
| 2 | 6.375 | 170 |
| 3 | 8.5 | 170 |
| 4 | 10.2 | 170 |
| 5 | 12.75 | 170 |
| 6 | 17.0 | 170 |
| 7 | 25.5 | 170 |
| 8 | 51.0 | 562 |

Table A.3 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 4 of Table 6.1.

| Period (value of k) | Air flowrate (Q_{Gk}) (m^3/h) | Extent of period (h) |
|------------------------|--|-------------------------|
| 1 | 5.1 | 170 |
| 2 | 6.375 | 190 |
| 3 | 8.5 | 170 |
| 4 | 10.2 | 170 |
| 5 | 12.75 | 180 |
| 6 | 17.0 | 170 |
| 7 | 25.5 | 270 |
| 8 | 51.0 | 1062 |

Table A.4 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 5 of Table 6.1.

| Period (value of k) | Air flowrate (Q_{Gk}) (m^3/h) | Extent of period (h) |
|------------------------|--|-------------------------|
| 1 | 5.1 | 200 |
| 2 | 6.375 | 160 |
| 3 | 8.5 | 190 |
| 4 | 12.75 | 125 |
| 5 | 17.0 | 125 |
| 6 | 25.5 | 185 |
| 7 | 51.0 | 185 |

Table A.5 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 6 of Table 6.1.

| Period (value of k) | Air flowrate (Q_{Gk}) (m^3/h) | Extent of period (h) |
|------------------------|--|-------------------------|
| 1 | 5.1 | 200 |
| 2 | 6.375 | 160 |
| 3 | 8.5 | 190 |
| 4 | 12.75 | 125 |
| 5 | 17.0 | 125 |
| 6 | 25.5 | 185 |
| 7 | 51.0 | 265 |

Table A.6 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 7 of Table 6.1.

| Period (value of k) | Air flowrate (Q_{Gk}) (m^3/h) | Extent of period (h) |
|------------------------|--|-------------------------|
| 1 | 5.1 | 240 |
| 2 | 6.375 | 210 |
| 3 | 8.5 | 240 |
| 4 | 12.75 | 180 |
| 5 | 17.0 | 130 |
| 6 | 25.5 | 160 |
| 7 | 51.0 | 590 |

Table A.7 Values of the air flowrate in the air sparging and extraction wells, and extent of each period for Case 8 of Table 6.1.

| Period (value of k) | Air flowrate (Q_{Gk}) (m^3/h) | Extent of period (h) |
|------------------------|--|-------------------------|
| 1 | 5.1 | 240 |
| 2 | 6.375 | 310 |
| 3 | 8.5 | 380 |
| 4 | 12.75 | 310 |
| 5 | 17.0 | 210 |
| 6 | 25.5 | 280 |
| 7 | 51.0 | 990 |

Table A.8 Times required for the breakthrough toluene concentration values to reach the ASIL value ($c_{Te} = 0.28 \text{ g/m}^3$) and 95% of the inlet concentration value. The biomass in the biofilter is assumed to be completely inactive. Cases 1-8 correspond to those of Table 6.3.

| Case | t_{ASIL} (h) | $t_{95\%C_{Ti}}$ (h) | C_{Ti} (g/m^3) |
|------|--------------------------|-------------------------|--------------------------------|
| 1 | 0.149 | 54.2 | 9.18 |
| 2 | 0.191 | 53.9 | 8.41 |
| 3 | 0.197 | 45.5 | 6.37 |
| 4 | 0.196 | 43.3 | 6.66 |
| 5 | 0.149 | 54.2 | 9.18 |
| 6 | 0.192 | 53.8 | 8.29 |
| 7 | 0.193 | 49.7 | 7.53 |
| 8 | 0.203 | 23.5 | 4.95 |

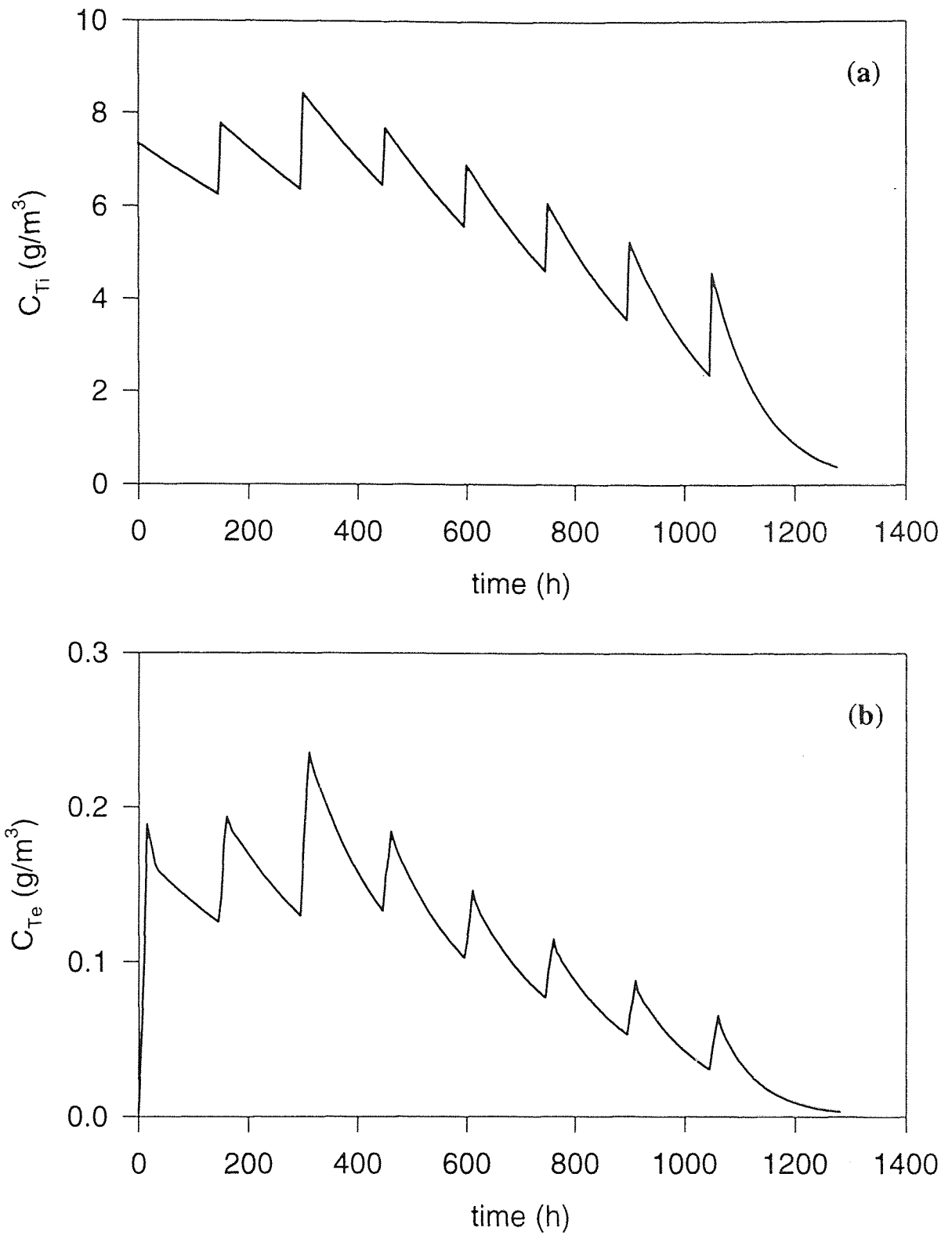


Figure A-1 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter for Case 2 (see Table 6.1).

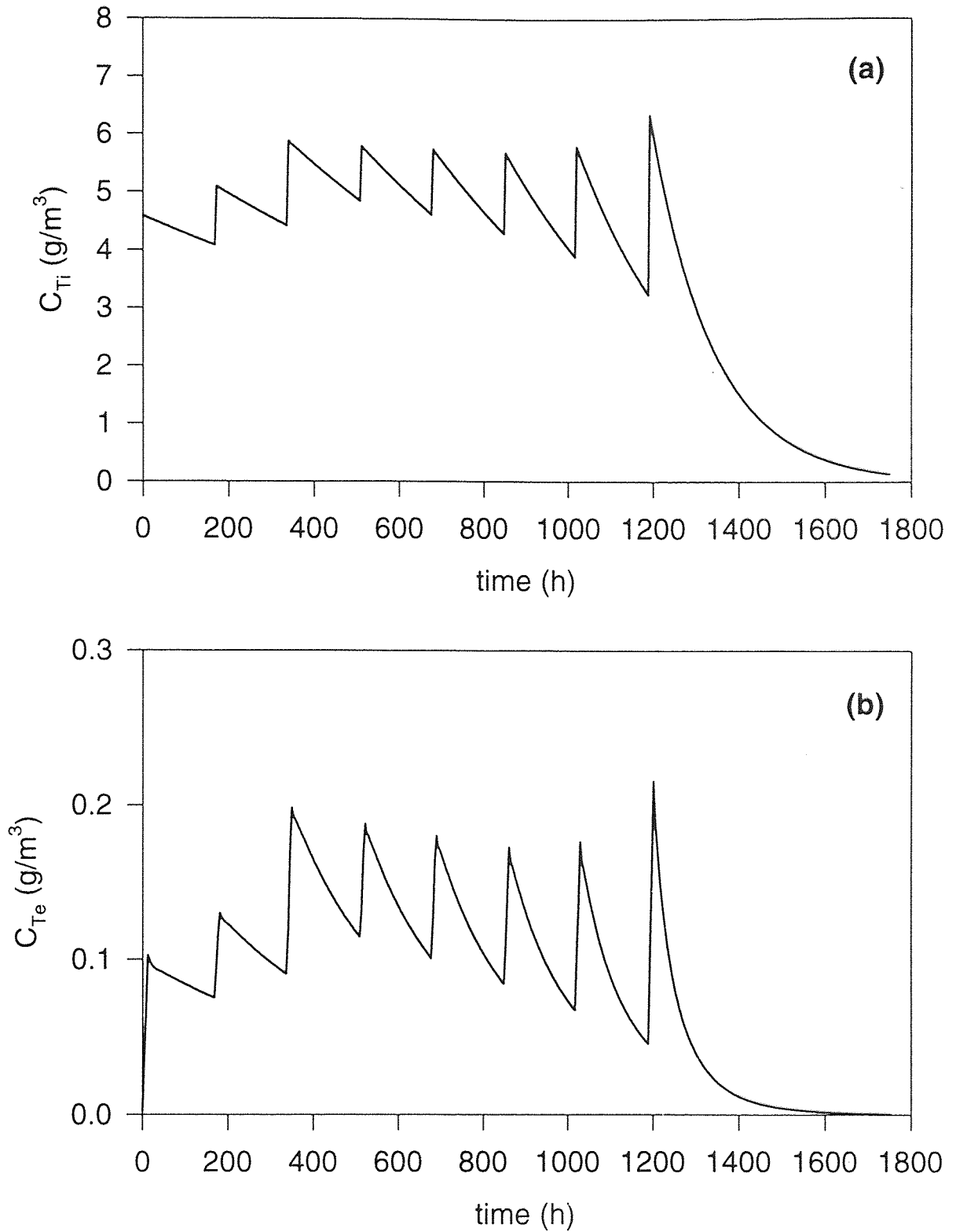


Figure A-2 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter for Case 3 (see Table 6.1).

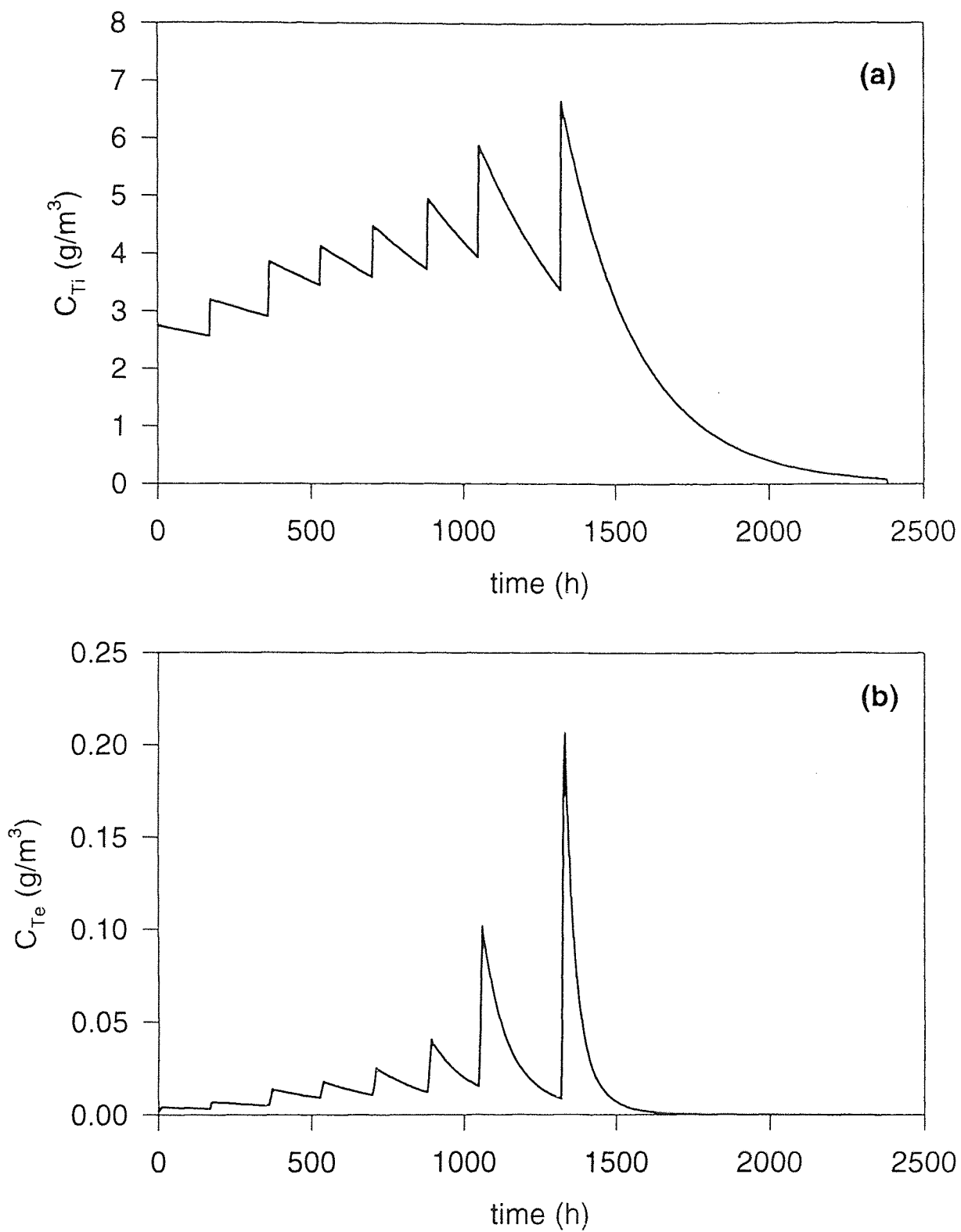


Figure A-3 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter for Case 4 (see Table 6.1).

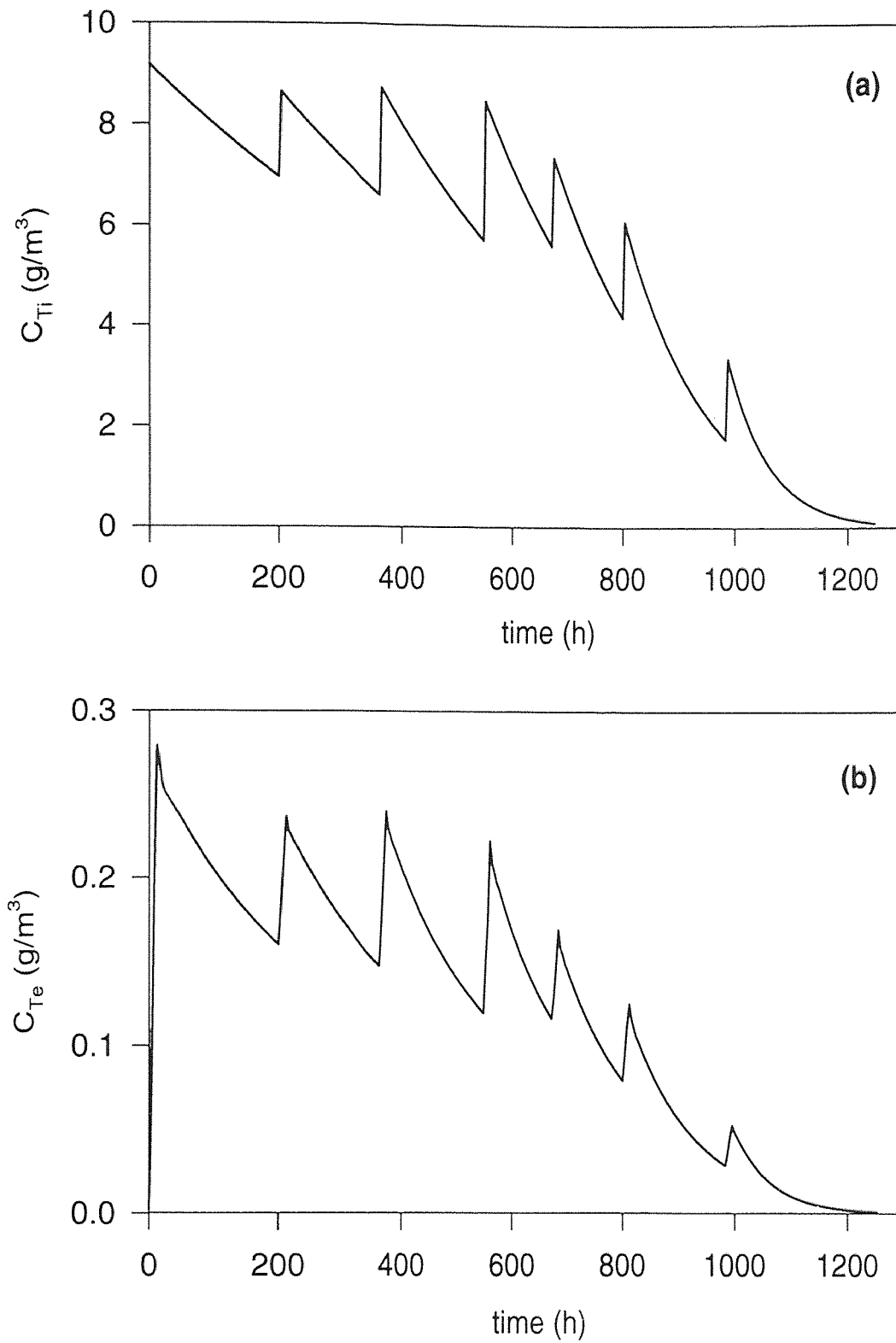


Figure A-4 Toluene concentration profile for the inlet (a) and outlet (b) of the biofilter for Case 5 (see Table 6.1).

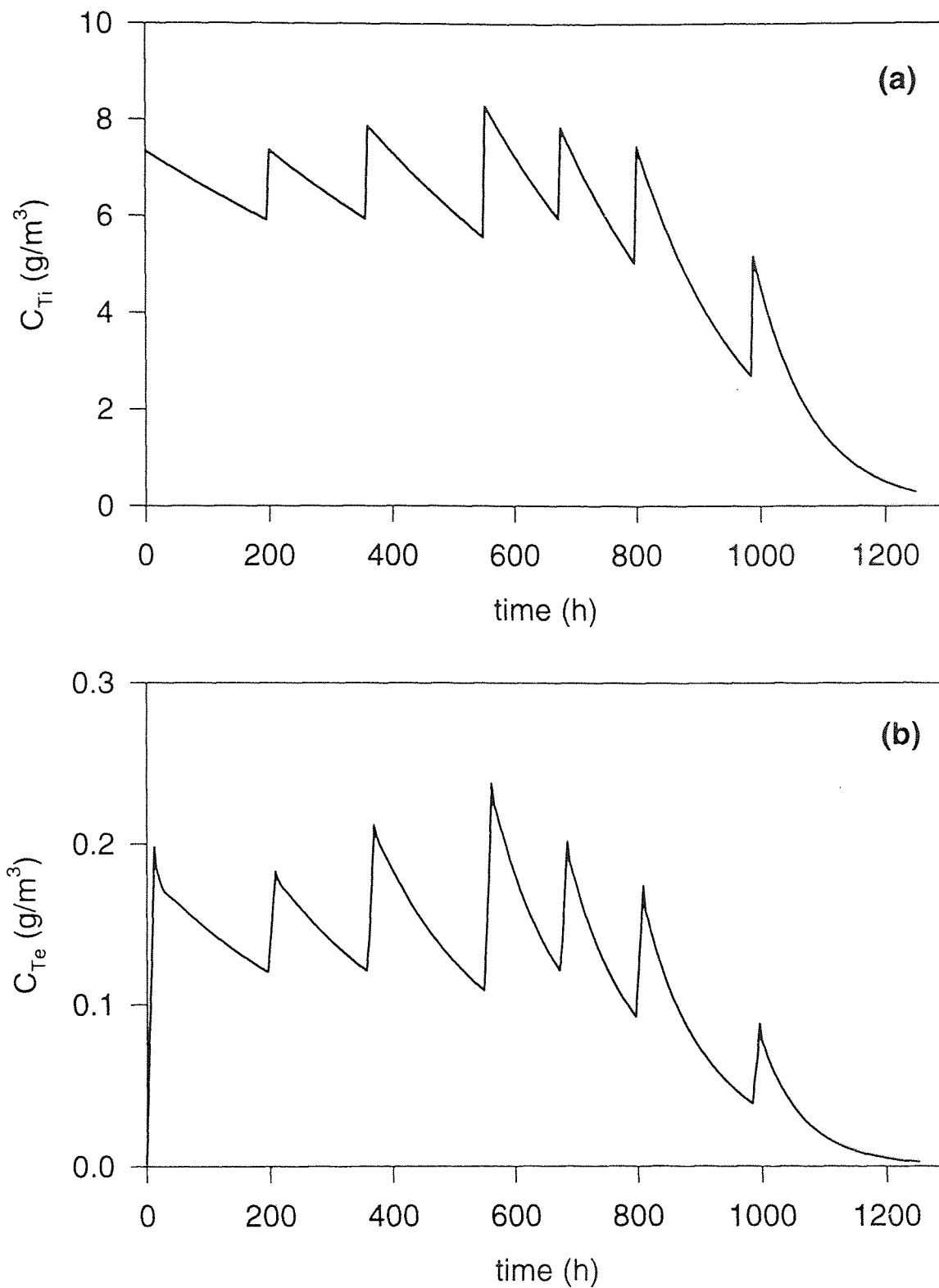


Figure A-5 Toluene concentration profile for the inlet (a) and outlet (b) of the biofilter for Case 6 (see Table 6.1).

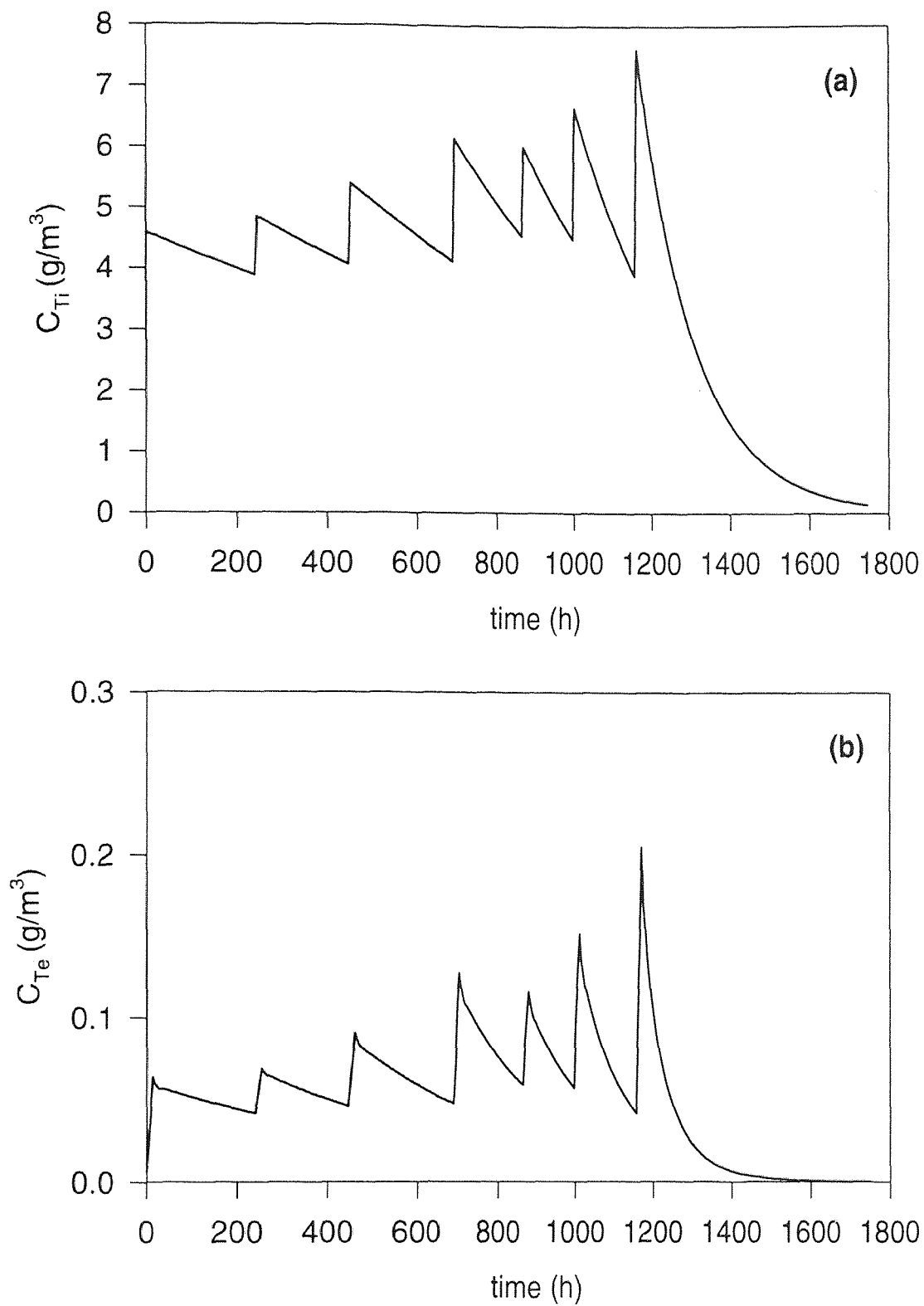


Figure A-6 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter for Case 7 (see Table 6.1).

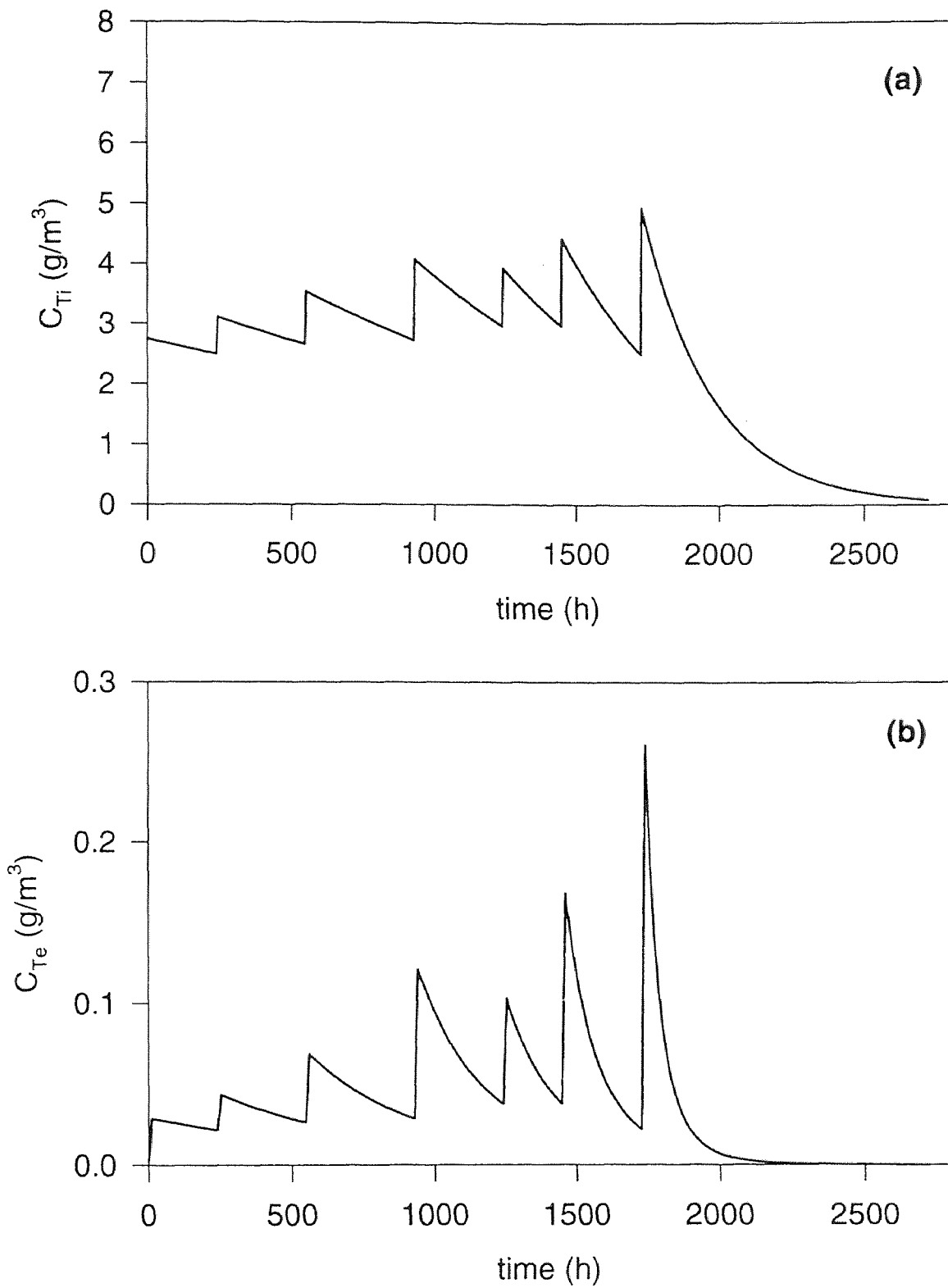


Figure A-7 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter for Case 8 (see Table 6.1).

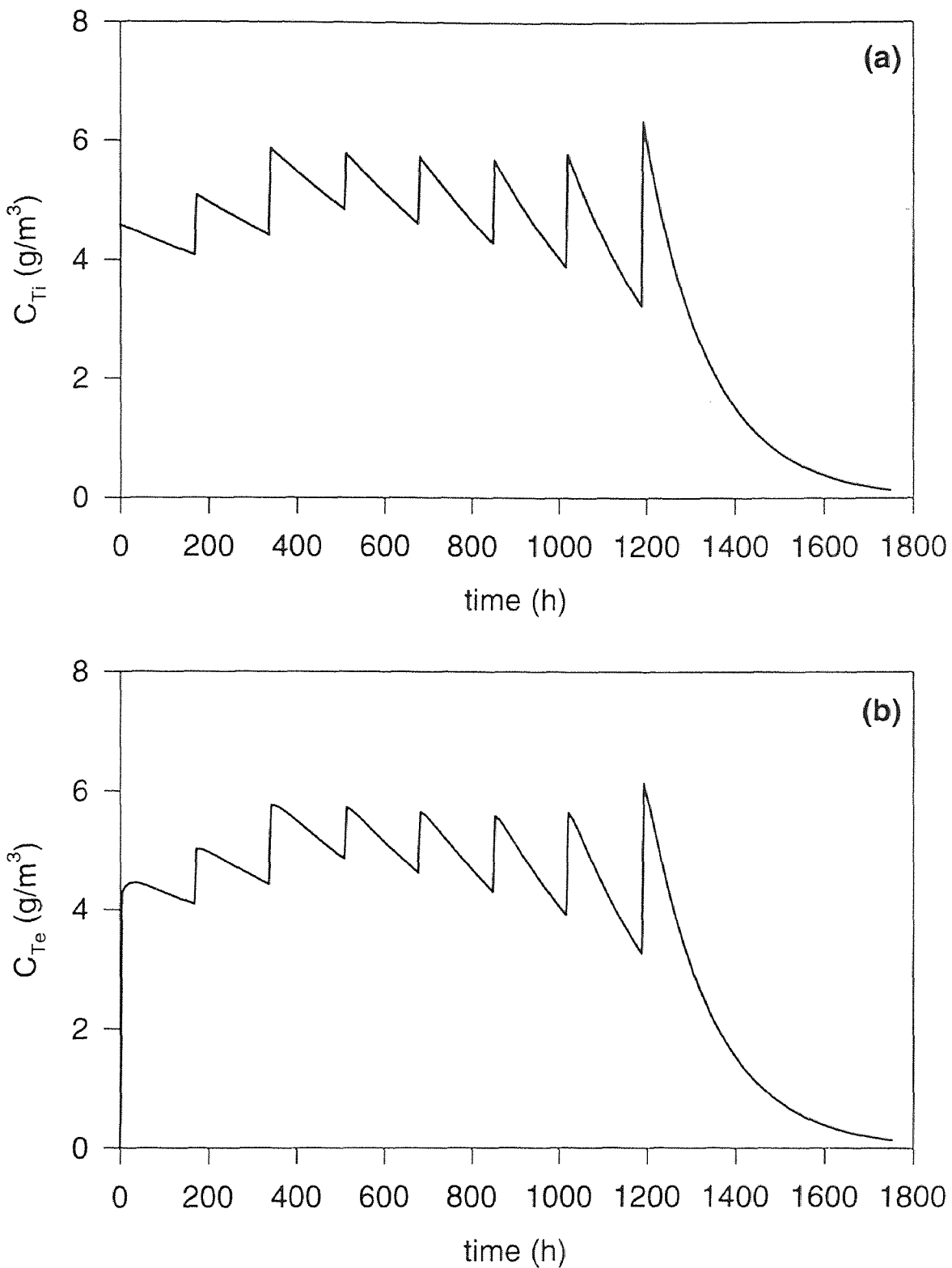


Figure A-8 Toluene concentration profile at the inlet (a) and outlet (b) of the biofilter unit when the biomass is completely inactive (dead). These profiles correspond to Case 3 of Table 6.1.

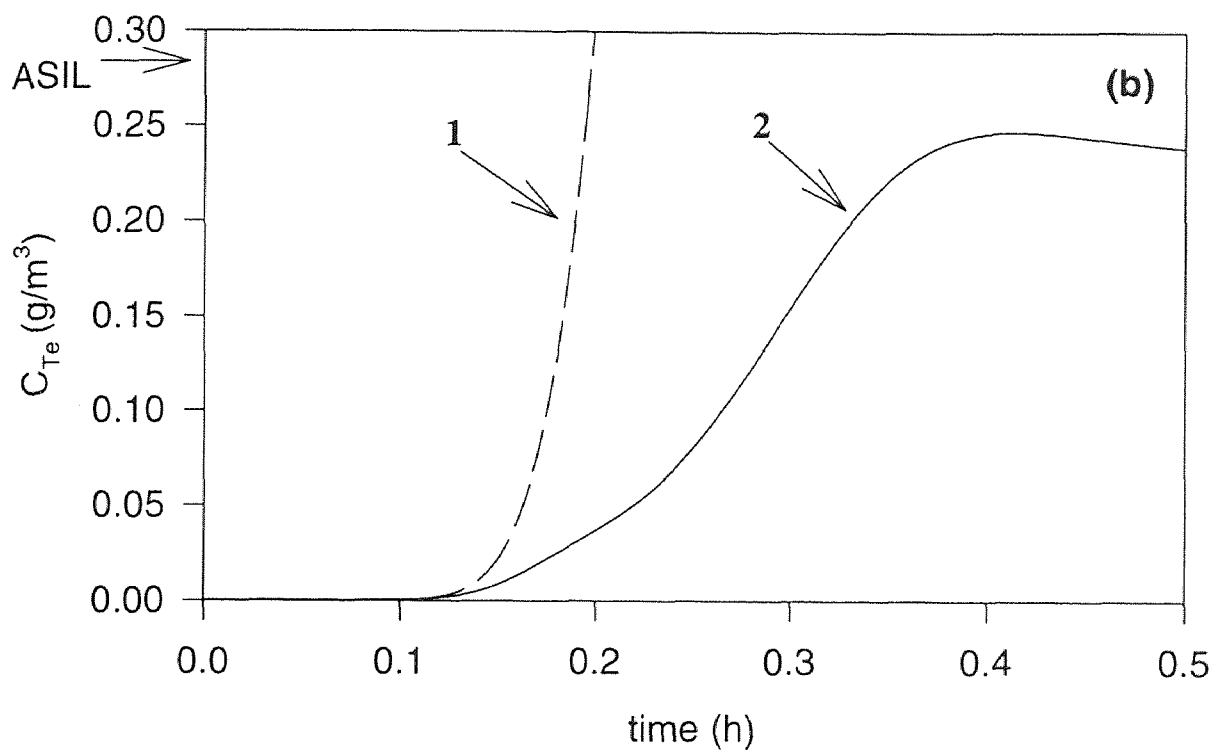
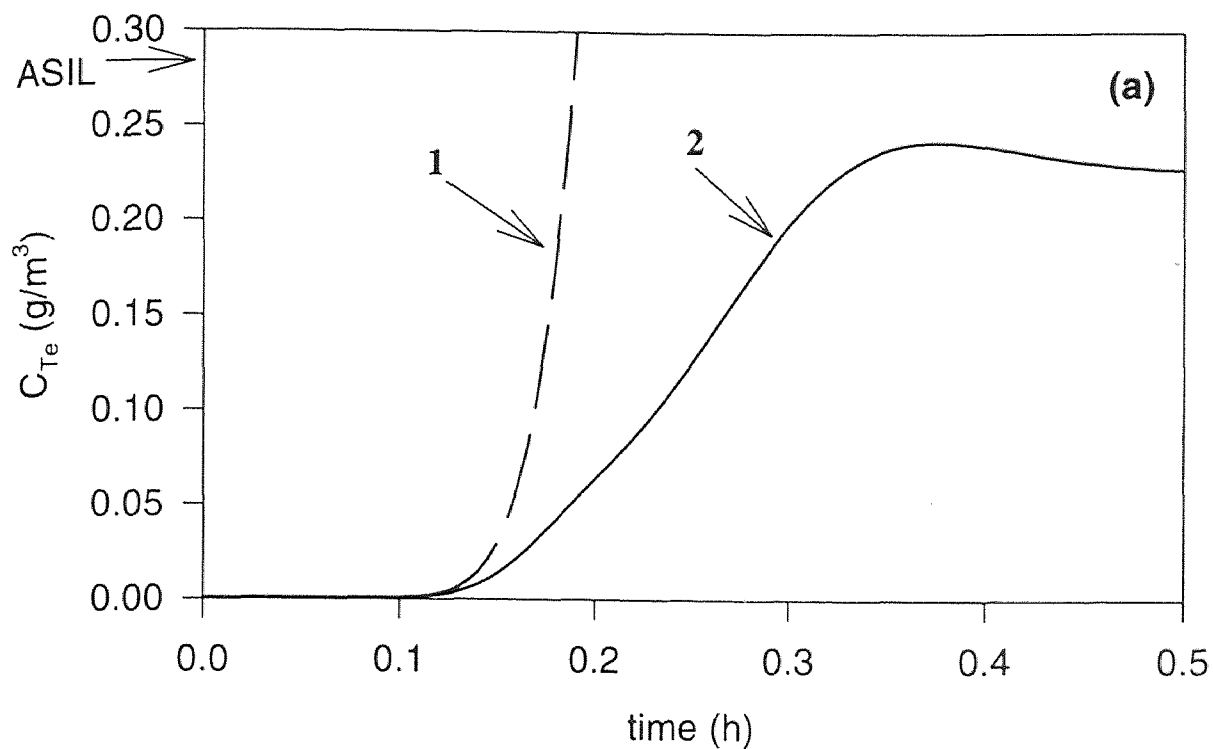


Figure A-9 Toluene breakthrough concentration profiles at the exit of biofilters when the biomass is completely inactive (curves 1) and completely active (curves 2). The biofilters are assumed to operate under constant inlet toluene concentrations of (a) : 9.18 and (b) : 6.37 g/m^3 . Other conditions are, (a): $\tau = 17.2 \text{ min}$, $\sigma = 1$; (b): $\tau = 9.7 \text{ min}$, $\sigma = 0.5$.

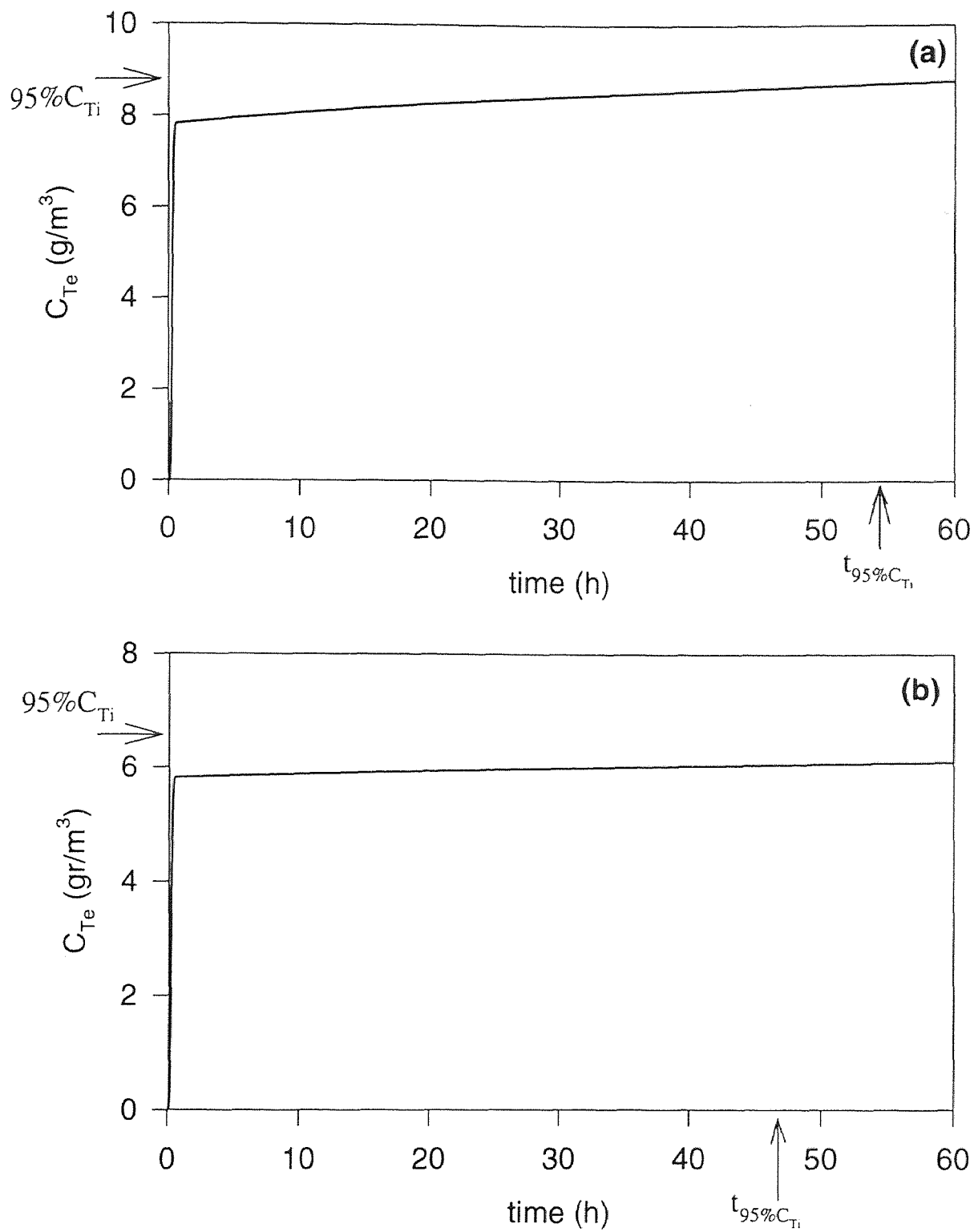


Figure A-10 Toluene breakthrough concentration profiles from biofilters with completely inactive biomass. Profiles also show the time needed for the exit concentration to reach 95% of the corresponding inlet concentration value. All operating conditions are correspondingly those of Figures A-9 (a) and (b).

APPENDIX B

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 By           : Dimitios Tsangaris, Newark, NJ on January 26,1995
c               Modification of the code of Shareefdeen (1994)
c
c Updated by   : Michael Cohen, Newark, NJ in May 1995
c*****

      implicit none
      include "Include/parameters.h"

      real*8 height(ng+1),gasB(ng+1), gasO(ng+1)
      real*8 effO(ng+1),effB(ng+1),del(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
      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"

      real*8 fcNew
      external fcNew

c
*   open(6,file='btcolw.out',status='new')
*
*   Read the program parameters
*
      one = 1.

      call input()
      call output()
      call PrintDim()
      deltainit = delta

c
c   Initialize the othogonal collocation routines
c
      call InitCollocation()
c

```



```

        status=TRUE
        status=DEBUG
    CALL today
c
    WRITE (6,101) n
101  Format( ' Solution of the Model using Orthogonal Collocation '
    &  ',/'      with ['i3,'] col. points',/)
c
c gas
c
    delz      = 1./float(ng)
    z        = 0.0
    height(1)= z
    gasB(1) = cgasB
    gasO(1) = cgasO
c
*START THE LOOP OVER Z AXIS
C
    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
C      CALCULATE LIQUID PHASE CONCENRATION
C
        call newton(status,solcol)

        if (status.eq.DEBUG) call PrintSolution(solcol,root)

        call interpolate(status,solcol,nt,root,dif1,xdat,Bdat,Odat)

        call CheckConvergence(iconv,Bdat,Odat)

        if (iconv.eq.TRUE) then
            call interpolate(THETACONV,solcol,nt,root, dif1,
&                xdat,Bdat,Odat)
c
c      CALCULATE GAS PHASE CONCENRATION
c
        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

        call FindEffect(
&   deriB,deriO,Bdat(1),Odat(1),effB(igas-1),effO(igas-1))
        del (igas-1) = delta

                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
                elseif (iconv.eq.FALSE) then
                    goto 6
                else
                    goto 5
                endif
100    continue
*
*END OF LOOP OVER Z AXIS
*
5      call Results(height,gasB,gasO,effB,effO,del)
        close(6)
123   FORMAT(/,
& '=====',/)
        stop
        end

```

io.f

```

*****
*
*   The 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

*ResTime must be entered in Minutes

*cgB00 must be entered in gr/m3

unfn=5

read(unfn,*)

read(unfn,*) cgb00

read(unfn,*)

read(unfn,*) restime

read(unfn,*)

read(unfn,*) delta

*

*

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 Kgr/m3)

*

read(unfn,*)

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,*) YOBI
*
*   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
*
*   Delta belongs here but it is more convient to put it at the top
*   of the input file since it is Inlet conditions dependant
  read(unfn,*) As
  read(unfn,*) Volume
  read(unfn,*) Surface

  restime = restime*60
*
* Now calculate some Dimensionless quantities using the above values
*
  lamdaB = DBW*KB*YB/(DOW*Ko*YOBI)

  gamaB = KB/KBI

  omegaB = (DOW*KO*cgB00)/(DBW*KB*cgO00)

  epsilnB = cgB00/(mB*KB)
  epsilnO = cgO00/(mO*KO)
*
* Initialize the concentrations of B,E,O at zero theta
*
  cgasB = 1.0
  cgasO = 1.0
*
* Now calculate the delta dependant values: eta,phiB2,phiE2
*
  call Update()

  return
end

```

```

*****
*       The 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/gas.h"
      include "Include/liquid.h"

      real*8 xv,fd
      real*8 deltaMt

      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

*****
* The 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 (' ', 'Resitance 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

subroutine Results(height,gasB,gasO,effB,effO,del)
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 effB(1),effO(1),del(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[gr/m3*hr] ',f12.4/,
&    '          Oxygen [gr/m3*hr] ',f12.4)
34  format(4x,F14.6,2x,F14.6,1x,f14.6,1x,f14.6)
    write(6,999)
999  format(/,' Effectivness results!!!! cg[gr/m3],d in [A]',/)
    do igas=1,ng+1
        write(6,333) height(igas), gasB(igas)*cgB00*1000.,
&    effB(igas),del(igas)*effB(igas),del(igas)
    enddo
333  format(4x,f12.6,2x,f12.6,1x,f12.6,1x,f12.6,1x,f12.6,1x,f12.6)
    write(6,123)
123  FORMAT(' _____',/)
    return
    end

```

init.f

```

*****
*
    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

```

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

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

c
        sb0 = epsilnB*cgasB
        Bsol(1)=sb0
        do i=1,n
            Bsol(i+1)=xold(i)
        enddo
        Bsol(n+2)=Bsol(n+1)

c
        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*****
c purpose : give the function for RK method, in the gas phase
c      balance ;
cc*****

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

```

check.f

```

      subroutine CheckConvergence(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.lt.0) then
    status=NILL
    return
endif

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.200)then
    delta = delta + 1.0
    status = FALSE
elseif(delta.ge.200)then
    delta = 200
    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
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
c           df contains the jacobian at (nsize,nsize)
c           and the -F vector at (nsize,nsize+1)
C
C           notice that nsize = n
c
cc*****
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= epsilonB*cgasB
    so0= epsilonO*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
c
    return
end

c*****
c Those subroutines calculate the concentration of fsE,fcE
c    given the concentrationsn 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

```

Makefile

```

SRC=main.f Init.f Check.f Model.f PrintSol.f io.f newton.f\
interpolate.f FindEff.f

OBJ=main.o Init.o Check.o Model.o PrintSol.o io.o newton.o\
interpolate.o FindEff.o

LIB=~dt/lib/orthcol.o

OPT= -extend_source -O3

.SUFFIXES: .o .f

.f.o: ; f77 -c $(OPT) $.f -o $.o

single: $(OBJ)
    f77 $(OBJ) $(LIB) -o ssicol

clean:
    rm -f *.o

```

Toluene.in

```

*****
0.730 Toluene concentration (gr/m3)
*****
3.9 Residence time (in min) or Flowrate (m3/h)
*****
1 Initial guess for delta (in mikroMeters)
*****

```

| | | |
|----------|---------|---------|
| 100 | B0 | [Kg] |
| 1.50 | Miou i | [1/h] |
| 11.03e-3 | Ki | [Kg/m3] |
| 78.94e-3 | KiI | [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] |
| 0.081 | mi | [-] |
| 34.4 | mO | [-] |
| 275 | C[o] | [gr/m3] |
| 40 | As | [m-1] |
| 3.32 | 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,NILL
parameter (TRUE=1,FALSE=0,NILL=-1)

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,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
real*8 mB,mO
common/henry/mB,mO
```

APPENDIX C

COMPUTER CODE FOR SOLVING THE TRANSIENT BIOFILTRATION MODEL FOR A SINGLE VOC UNDER TEMPORALLY VARYING INLET CONDITIONS

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             Based on Shareefdeen (1994)
c
c Updated By : Michael Cohen, Newark, NJ in August 1995
c             Pothitos Stamatiadis, Newark, NJ in October 1996

```

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
real*8 Qginst(0:ninstmax),tinst(0:ninstmax)
real*8 inlet(0:ntmax)
real*8 Clo,Vplume,Qg

```

```

integer istate,istatus,i
integer ndim,npar,nt,nh,it,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,tinst,
+             Qginst,Qg,Vplume,Clo,inlet)

```

```

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,izero,tau)

call C2Y(cg,co,cp,y,izero,nh)

do it = 1,ntmax
    T = time(it-1)
    tout = time(it)

    call Find_dilution(mB, Clo, Vplume, Qg,
+         Qginst,tinst,
+         time(it)*tau*24, cgB00)

    inlet(it)= CgB00*1000.

    cg(0,it) = cgB00/cgB00n
    co(0,it) = cgO00/cgO00n
*
*     Find the average concentration for this time instance
*
    avcgb = cg(nh/2,it-1)*cgB00n
    if (it.eq.1) avcgb=0.0

    call Update(avcgb)

c     if (it.ge.(ntmax-20)) call PrintDimensionless()
    if (mod(it,10) .eq.0) call PrintDimensionless()

call Pack(par,cg,co,mb,Vplume,Clo,Qg,Qginst,Tinst,tau,co(0,it),cgB00n)

    istrate = TRUE

&     CALL ODESSA(fun,dfun,NEQ,Y,PAR,T,TOUT,ITOL,RTOL,ATOL,
    ITASK,ISTATE, IOPT,RWORK,LRW,IWORK,LIW,jfun,MF)

    call Y2C(cg,co,cp,y,it,nh)

if(istrate.LT.0) then
    tlast = it
    goto 10
endif

c     if (it.ge.(ntmax-200)) call PrintOne(cg,co,cp,time,ht,nh,it,tau)
c     if (mod(it,10) .eq.0) call PrintOne(cg,co,cp,time,ht,nh,it,tau)
    call PrintOne(cg,co,cp,time,ht,nh,it,tau)

    call CheckSteadyState(istatus,cg,co,cp,nh,it,nt,tau,tout)

```

```

                if (istatus.eq.TRUE) then
                    tlast = it
                    goto 11
                endif
            enddo
c
c      Output your results
c
10      continue
11      call PrintSum(cg,co,cp,time,ht,nh,tlast,tau,inlet)

        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

    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)/cgB00n/1000
            co(ih,0) = co(ih,0)/cgO00n/1000
        enddo
        cg(0,0) = cgB00/cgB00n
        co(0,0) = cgO00/cgO00n
        cp(0,0) = (cg(0,0)/psi)**(1./an)
        close(9)

    elseif(colstatus.eq.FRESH) then
        print*,'Column starts empty'
        do ih = 1,nh

```



```

        cg(ih,0) = 1.0e-4
        co(ih,0) = 1.0e-4
    enddo
    cg(0,0) = cgB00/cgB00n
    co(0,0) = cgO00/cgO00n
    cp(0,0) = 0.0
else
write(6,(a))'Error in Initial Status selection'
    stop
endif
c
c The following loop converts the cp to the equilibria value
c (only at startup conditions)
c
c do ih = 0,nh
c     cp(ih,0) = (cg(ih,0)/psi)**(an)
c     cp(ih,0) = (cg(ih,0)/psi)**(an)
c enddo
c
* 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

*****

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

```

dilution.f

```
subroutine get_cb0(t,cbb,mb,Clo,Vplume,Qg,Qginst,tinst,cgB00n,tau)
```

```
implicit none
include "Include/parameters.h"
```

```
real*8 t,cbb
real*8 thr
real*8 mb,clo,Vplume,Qg,cgB00n
real*8 Qginst(0:ninstmax),tinst(0:ninstmax)
```

```
real*8 tau,cgB00_loc
```

```
c it and loc_inlet are dummy variables here
```

```
thr = t * tau * 24
```

```
call Find_dilution( mB, Clo, Vplume, Qg,
+                 Qginst,tinst,
+                 thr , cgB00_loc)
```

```
cbb = cgB00_loc/cgB00n
```

```
c print*,thr,cgB00_loc,cbb,cgB00n
```

```
return
end
```

```
c
c subroutine Find_dilution( mB, Clo, Vplume, Qg,
+                          Qginst, tinst,
+                          tint, cgB00)
```

```
c
c This subroutine calculates the inlet VOC concentration to the
c biofilter.
```

```
c
c Inlet variables: mB, Clo, Vplume, time, inlet,it
c Outlet variables: cgB00
```

```
c
c implicit none
c include "Include/parameters.h"
```

```
real*8 cgB00,mB, Clo,volume,dilution, Vplume , Qg,sum
real*8 time,tint
real*8 tinst(0:ninstmax), Qginst(0:ninstmax)
real*8 dx(0:ninstmax)
integer iter,ll,i
real*8 xt
```

```
99 continue
```

```
time = tint
```

```
sum = mB * Clo / Qg
```

```

iter = 0
1  continue
    if (time.gt. tinst(iter)) then
sum=sum*exp( mB*(Qginst(iter+1) - Qginst(iter))*tinst(iter)/Vplume)
        iter = iter + 1
        goto 1
    else
cgB00 = sum * Qginst(iter)* exp(-mB*Qginst(iter)*time/Vplume)
    endif
10  continue
c   write(*,'(8(f12.5,1x))')time, tinst(iter), Qginst(iter),Qg,cgB00

cgB00  = cgB00*1.e-3

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(6,66) effdelta,effdelta
66  FORMAT (2x,'EffB*delta= ',f14.8,
&         2x,'EffO*delta= ',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,*)

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 ( ' ', 'Residence Time (min)    = ', f12.5)
          WRITE(6,3) volume
          3  FORMAT ( ' ', 'Volume of the column( m3)  = ', 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
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
C print concentration changes along the column time
C
C*****

subroutine PrintOne(cg,co,cp,time,ht,nh,it,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

write (6,84)
84 format(//,5x,'Solution of the Transient Model',/)

write (6,86) time(it),cgB00n*1000
write (6,87) time(it)*tau*24.,cg(0,it)*cgB00n*1000
86 format (/, 10x, 'At Time = ', f14.3,3x,'Cg = ' f10.3,/)
87 format (/, 10x, 'At Time(hr)= ', 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)*cgB00n*1000,
& co(ih,it)*cgo00n*1000, cp(ih,it)
96 format (5x, f7.3,3x,f10.4,5x,f10.4,5x,f10.4)
enddo

return
end

C*****
C
C print concentration changes along the column time
C
C*****

subroutine PrintSum(cg,co,cp,time,ht,nh,nt,tau,inlet)
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
real*8 inlet(0:ntmax)
integer nh,it,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',/)

do it = 0, nt
c write (6,96) time(it),

```



```

        write (6,96) time(it)*tau*24,
&      ",",inlet(it),
c &      ",",cg(nh/3,it)*cgB00n*1000,
&      ",",cg(nh,it)*cgB00n*1000.,",",
96      format (5x,f8.3,a1,f10.4,a1,f10.4,a1,f10.4)
    enddo

```

```

write(6,*)

```

```

return
end

```

```

subroutine printcg(i,cg)
implicit none

```

```

include "Include/parameters.h"
real*8 cg(0:nhmax,0:ntmax)
integer l,i
do l=0,nhmax
    print*,l,cg(l,i)
enddo
return
end

```

readparam.f

```

subroutine ReadParam(istatus,ndim,npar,nh,nt,dt,err,tau
&      ,tinst, Qginst,Qg,Vplume,Clo,inlet)

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,ninst,i
real*8 dt,err,tau,flowrate
real*8 Qginst(0:ninstmax),tinst(0:ninstmax),Vplume,Clo,Qg
real*8 inlet(0:ntmax)
real*8 effdeltaMt

character *80 filename

integer unfn
unfn=5
filename = 'Toluene.in'

* ResTime must be entered in Minutes
* cgB00 must be entered in gr/m3
* read(5,*) cgb00,restime
* read(5,'(a)')filename

c open (unit=unfn,file=filename,status='old')

```

```

read(unfn,*) flowrate
read(unfn,*) Clo
read(unfn,*) Vplume
Qg      = flowrate

read(unfn,*) ninst
do i=1,ninst
    read(unfn,*) tinst(i),Qginst(i)
enddo
tinst (0) = 0.
Qginst(0)=Qginst(1)
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 Kgr/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,*) YO
*
*   Henry's constants
*
    read(unfn,*) mB
    read(unfn,*) mO
*

```

```

* Entrance concentrations
*
cgB00 = mB*Clo* Qginst(1)/Qg
inlet(0)= cgB00
cgB00 = cgB00 * 1.e-3

read(unfn,*) cgo00
cgo00 = cgO00 * 1.e-3

c The following are the renormalization factors
cgB00n = 10 * 2. * 1.e-3
cgo00n = 275 * 1.e-3
*
* Volumetric properties
*
read(unfn,*) Volume
read(unfn,*) Surface

flowrate = flowrate/3600
restime = volume/flowrate

velocity = 1./restime

c tau in days
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/KBI
epsilnB = cgB00n/(mB*KB)
epsilnO = cgO00n/(mO*KO)
*
* Now calculate some Dimensionless quantities using the above values

```

```

*
      istatus= FRESH

c      Read dt from the input file
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
      real*8 effdeltaMt

* Now recalculate some Dimensionless quantities using the new values
* of cgB00
      epsilnB = cgB00n/(mB*KB)
      epsilnO = cgO00n/(mO*KO)

      call FindDelta(cgasB,delta,effectB,effectO,mb,effdelta)

      delta=0.
      effectB=0.
      effectO=0.
      effdeltaMt=effdelta*1.e-6

* A correction is needed because the empirical formula  $C_{jp} = K_d (C_j^*)^n$  holds
* only when  $c_j$  is in  $[gr_j / m^3 \text{ air}]$ . Then,  $const = gr/m^3 \rightarrow K_{gr}/m^3$ .
* After this correction,  $cstar\_reduced = psi * C_{jp\_reduced}$ 

      const = 1.e-3
      deltaMt = delta*1.e-6

      xv = b0

      betaB=effdeltaMt*(alpha*As)*xv*restime*miouB/(YB *cgB00n*porosity)
      betaO=effdeltaMt*(alpha*As)*xv*restime*miouB/(YOB*cgO00n*porosity)

```

```

beta = kapaa*(1-alpha)*As*restime/porosity
psi = (const/cgB00n)*(porosity*cgB00n/((1-porosity)*rho*Kapad))**an

```

```

return
end

```

util.f

```

subroutine C2Y(cg,co,cp,y,it,nh)
implicit none
include "Include/parameters.h"
include "Include/dimensionless.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)
    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,cg,co,mb,Vplume,Clo,Qg,Qginst,Tinst,tau,cbo,cgB00n)
implicit none

```

```

include "Include/parameters.h"
include "Include/dimensionless.h"

```

```

real*8 cg(0:nhmax,0:ntmax),co(0:nhmax,0:ntmax)
real*8 Qginst(0:ninstmax),tinst(0:ninstmax)
real*8 mb,Vplume,Clo,tau,cgB00n,Qg

```

```

real*8 cbo

real*8 par(1)
integer i

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

par(11) = cgB00n
par(12) = mb
par(13) = Clo
par(14) = Vplume
par(15) = tau
par(16) = cbo
par(17) = Qg
par(18) =Qginst(0)
par(19) =tinst(0)

do i=1,ninstmax
    par(19+i) = Qginst(i)
enddo
do i=1,ninstmax
    par(19+i+ninstmax) = tinst(i)
enddo

return
end

*****

subroutine CheckSteadyState(istatus,cg,co,cp,nh,it,nt,tau,tout)

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

c    d1 = abs (cg(nh,it) - cg(nh,it-1))
c    d2 = abs (co(nh,it) - co(nh,it-1))
c    d3 = abs (cp(nh,it) - cp(nh,it-1))

c    if(d1.le.TOLERR.and.d2.le.TOLERR.and.d3.le.TOLERR) then

```

```

c      if(d1.le.TOLERR.and.d2.le.TOLERR.and.d3.le.TOLERR.
c      & or.cgB00.lt.(1e-4)) then

*      As there is a constantly changing inlet concentration,
*      steady state will never be reached, so a check is put into
*      place instead
          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(npamax),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,cbb
      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

c      print*,'mb=',mb
c      print*,'Clo=',Clo
c      print*,'Vplume
c      print*,'Qg

c      do i=1,10
c          print*,'i,Qginst(i),tinst(i)

```

```

enddo

call get_cb0(t,cbb,mb,Clo,Vplume,Qg,Qginst,tinst,cgB00n,tau)

do i = 1,nh
  cb = y(i)
  cbpr = cbb
  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

  der1 = (cb-cbpr)/dz

  ydot(i)= -der1/porosity-betaB*fun1-beta*fun2
enddo

offset = nh
do i = 1,nh
  cb = y(i)
  co = y(i+nh)
  copr = coo
  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)

  der2 = (co-copr)/dz

  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
c
      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
c jacobian of the vectorfield
c
      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)
      enddo

```

```

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,mB,effdelta)
implicit none

real*8 avcg,delta,effectB,effectO,mB,avcgu
real*8 effdelta

avcgu = avcg*1000.

```

```

if (mB.eq.0.27) then
  effdelta = 0.014*(avcgu**3)-0.281*(avcgu**2)+1.837*avcgu+6.097
elseif (mB.eq.0.216) then
  effdelta = 0.016*(avcgu**3)-0.309*(avcgu**2)+1.905*avcgu+6.330
elseif (mB.eq.0.135) then
  effdelta = 0.015*(avcgu**3)-0.276*(avcgu**2)+1.658*avcgu+7.085
elseif (mB.eq.0.081) then
  effdelta = 0.008*(avcgu**3)-0.145*(avcgu**2)+1.110*avcgu+7.854
else
c      do nothing
      print*, 'Error, mb not in the list'
      stop
endif

*      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 Dilution.f

OBJ=main.o Model.o Print.o Init.o ReadParam.o Util.o FindDelta.o Dilution.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

```

51    Flowrate
340   Clo
1000  Volume of Plume
8     Number of points
160   5.1
330   6.375
450   8.5
570   10.2
690   12.75
810   17.0
930   25.5
1138  51.0
*****Toluene*****
100   B0           [Kg]
1.50  Miou i       [1/h]
11.03e-3 Ki        [Kg/m3]
78.94e-3 KiI       [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]
0.270 m            [-]
34.4  mO           [-]
275   C[o]         [gr/m3]
14.62 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
0.50  dt (dt of integration)

```

*include**parameters.h*

```

integer nhmax,ntmax,ninstmax
c    parameter(ntmax=100)
    parameter(ntmax=700)
    parameter(nhmax=20)
    parameter(ninstmax=50)

```

integer neqmax,nparmax,liwmax,lrwmax,colstatus
 character *80 fileprev
 parameter(neqmax=3*nhmax,nparmax=150,lrwmax=5000,liwmax=100)

integer FRESH,OLD
 parameter (FRESH=10,OLD=20)

real*8 TOLERR
 parameter (TOLERR=1.e-3)

integer TRUE,FALSE
 parameter (TRUE=1,FALSE=0)

real*8 zero
 parameter (zero=0.0)

integer izero
 parameter (izero=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

real*8 cgB00n,cgo00n
 common/concen1/cgB00n,cgo00n

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

```

```

real*8 effdelta
common/aux1/effdelta

```

```

real*8 cbb0,coo0
common/aux2/cbb0,coo0

```

all

```

real*8 omegaB,eta
common /omega/eta,omegaB
real*8 epsilnB,epsilnO
common /epsiln/epsilnB,epsilnO
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

```

```

real*8 mB,mO
common/henry/mB,mO
real*8 cgB00,cgo00
common /cgas/cgB00,cgo00

```

```

real*8 delta
common/del/delta

```

```

real*8 cgasB,cbasO
common /cgas/cgasB,cbasO

```

```

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

```

system.h

```

integer iprnewton
common/flowcontrol/iprnewton

```

unfold.h

```

real*8 epsilnB,epsilnO,gamaB,betaB,betaO,beta,psi,porosity,dz,an
real*8 cgB00n ,mb,Clo,Vplume,tau,Qg
real*8 Qginst(0:ninstmax),tinst(0:ninstmax)
real*8 coo

```

```

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)

```

```

cgB00n = par(11)
mb = par(12)
Clo = par(13)
Vplume = par(14)

```

```
tau      = par(15)
coo      = par(16)
Qg       = par(17)

Qginst(0) = par(18)
tinst(0) = par(19)

do i=1,ninstmax
  Qginst(i) = par(19+i)
enddo

do i=1,ninstmax
  tinst(i) = par(19+ninstmax+i)
enddo

1      continue
```


REFERENCES

- Androutsopoulou, H. 1994. *A Study of the Biofiltration Process Under Shock-Loading Conditions*, MS Thesis, Department of Chemical Engineering, Chemistry and Environmental Science, 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. 1995. "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 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.
- Chang, A.N. and H. Yoon. 1995. "Biofiltration of Gasoline Vapors," *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: 123-130.
- Cohen M.L. 1996. *Sensitivity Analysis and Design Calculations with Biofiltration Models*, MS Thesis, Department of Civil and Environmental Engineering, New Jersey Institute of Technology, Newark NJ.
- 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.
- 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.
- 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.

- Hein G.L., Hutzler N.J., and J.S. Gierke. 1994. "Quantification of the Mechanisms Controlling the Removal Rate," *Proceedings of the 1994 National Conference on Environmental Engineering Boulder, USA* Publ. By ASCE, CO: 556-563.
- Hodge, D.S. and J.S. Devinny. 1994. "Biofilter Treatment of Ethanol Vapors," *Environ. Progress*, **13**: 167-173.
- Hodge, D.S. and J.S. Devinny. 1995. "Modeling Removal of Air Contaminants by Biofiltration," *ASCE J. Environ. Eng.*, **121(1)**: 21-32.
- 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.
- Leson, G. and B.J. Smith. 1995. "Results from the PERF Field Study on Biofilters for Removal of Volatile Petroleum Hydrocarbons," *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: 99-113.
- Li, D.X. 1995. "In Situ Biofiltration for Treatment of Petroleum Hydrocarbon Vapors," *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: 1-17.
- McKenzie. 1990. *Extraction of Volatile Organic Chemicals from Unsaturated Soil*, MS Thesis, Michigan Technological University, Houghton, Michigan.
- 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.
- Reynolds F.E. Jr, and D.S. Hodge. 1995. "Foreword," *Proceedings of the 1995 Conference on Biofiltration (an Air Pollution Control Technology)*, pp.iv-v.
- Saberian, A.G., M.A. Wilson, E.O. Roe, J.S. Andrienas, C.T. Esler, G.H. Kise and P.E. Reith. 1994. "Removal of Gasoline Volatile Organic Compounds via Air Biofiltration: A Technique for Treating Secondary Air Emissions From Vapor-Extraction and Air-Stripping Systems," pp. 1-11 in *Hydrocarbon Bioremediation*, R.E. Hinchee, B.C. Alleman, R.E. Hoepfel and R.N. Miller (eds.), Lewis Publishers, Boca Raton, FL.
- Sellers K.L., R.P. Schreiber. 1993. "Air Sparging Model for Predicting Groundwater Clean up Rate," *Proceedings of the 1992 Petroleum Hydrocarbons and Organic Chemicals in Ground Water: Prevention, Detection, and Restoration*, Camp Dresser and McKee Inc., MA: 365-376.

Sephehr M., and Z.A. Samari. 1993. "In Situ Soil Remediation using Vapor Extraction Wells, Development and Testing of a 3-D Finite-Difference Model," *Ground Water*, **31**: 425-436.

Shareefdeen , Z.M. 1994. *Engineering Analysis of a Packed-Bed Biofilter for Removal of Volatile Organic Compound (VOC) Emissions*. PhD Dissertation, 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.

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.

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.