# EXPANSION OF A MULTICOMPONENT MIXED-BED ION EXCHANGE MODEL TO INCLUDE MORE COMPLEX SOLUTION CHEMISTRY

By

YUNQUAN LIU

Bachelor of Science
Dalian University of Technology
Dalian, China
1985

Master of Science
Dalian University of Technology
Dalian, China
1988

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
December, 1998

# EXPANSION OF A MULTICOMPONENT MIXED-BED ION EXCHANGE MODEL TO INCLUDE MORE COMPLEX SOLUTION CHEMISTRY

Thesis Approved:

Thesis Advisor

Randy S. Lewis

Arlind H. Johannsen

Mark S.

Wayne B. Powell

Dean of the Graduate College

# PREFACE

This study addresses the generalization of a mixed-bed ion exchange (MBIE) model to include more complex solution chemistry. Based on the discussion of mass transfer mechanisms of weak electrolytes in ion exchange column, a generalized rate model that can handle multicomponent mixed-bed ion exchange, especially systems with multivalent dissociations has been developed. The resultant model was used to predict the breakthrough behavior of industrial MBIE columns, and satisfactory agreement was achieved between model predictions and real plant experience.. The model was further extended to handle silica – one of the most complicated dissociative species, and the predicted results agree favorably with what would be expected. Finally, the model was extended to tackle some special cases in MBIE operations; such as variable inlet concentrations, variable inlet flow rate, layered-bed and bed cleaning.

I wish to express my deepest appreciation to my major advisor Dr. Gary L. Foutch, for his guidance, inspiration, patience and invaluable helpfulness throughout my doctorate program. Thanks are also extended to Dr. Arland H. Johannes, Dr. Martin S. High, Dr. Frank W. Chambers, and Dr. Randy S. Lewis for serving on my advisory committee and for their helpful suggestions and technical assistance in the completion of this study.

I would like to thank Sree K. Vinay Sunkavalli, and Dennis F. Hussey for their valuable contributions in developing the generalized rate model. Particular thanks go to Jidong Lou for his help and suggestions throughout my study.

Special gratitude and appreciation are expressed to my parents and my brothers for their encouragement, understanding and sacrifice. I owe a lot to my wife, Yanying Zhang, my son Yang, and my daughter Nancy. Without their encouragement and sacrifice, this work would never have been finished.

Financial assistance from the School of Chemical Engineering at Oklahoma State University, and Electrical Power Research Institute for completion of this study is gratefully appreciated. Finally, I would like to express my special thanks to all my friends for their encouragement in completing this manuscript.

# TABLE OF CONTENTS

# LIST OF TABLES

Table Page

Chapter I

Chapter III

Table Page

Chapter IV

Chapter V

# LIST OF FIGURES

Figure Page

Chapter I

Chapter III

Figure Page

## Chapter IV

Figure Page

Chapter V

# NOMENCLATURE

| | |
|---|---|
| $A_i$ | parameter of ion i |
| $a_s$ | interfacial area ($L^2/L^3$) |
| $B_i$ | parameter of ion i |
| $C_i$ | concentration of species i ($meq/L^3$) |
| $C_i^*$ | concentration of species i at the surface of the resin ($meq/L^3$) |
| $C_i^o$ | concentration of species i in the bulk solution ($meq/L^3$) |
| $C_T$ | total equivalent concentration ($meq/L^3$) |
| $C_{TDI}$ | total concentration of type I dissociative species ($meq/L^3$) |
| $C_{TDII}$ | total concentration of type II dissociative species ($meq/L^3$) |
| $C_{TDIII}$ | total concentration of type III dissociative species ($meq/L^3$) |
| $d_p$ | particle diameter (L) |
| $D_i$ | self-diffusivity of species i ($L^2/T$) |
| $D_e$ | effective diffusivity ($L^2/T$) |
| F | Faraday's constant (coulombs/mole) |
| FAR | fraction of anionic resin |
| FCR | fraction of cationic resin |
| $J_i$ | flux of species i in the film ($meq/T/L^2$) |
| k | reaction rate constant ($T^{-1}$) |
| K | representative mass transfer coefficient (L/T) |
| $K_A^B$ | resin selectivity coefficient for ion B in solution compared to A in the resin |
| $K_a$ | dissociation constant of amine |

| | |
|---|---|
| $K_b$ | dissociation constant of amine |
| $K_w$ | water dissociation equilibrium constant |
| $K_1$ | first dissociation constant of $CO_2$ |
| $K_2$ | second dissociation constant of $CO_2$ |
| m | number of coions |
| $N_i$ | relative valence of ion i |
| n | number of counterions |
| P | exponent |
| $q_i$ | concentration of species i in the resin (meq/$L^3$) |
| Q | capacity of the resin (meq/$L^3$) |
| r | radius of the film ($L^{-1}$) |
| Re | particle Reynolds number |
| Sc | Schmidt number |
| T | temperature (°C) |
| t | time (T) |
| $u_s$ | superficial velocity in axial flow packed bed (L/T) |
| V | volume of the packed resin ($L^3$) |
| $X_i$ | concentration fraction in liquid phase |
| $Y_i$ | concentration fraction in the resin phase |
| $Z_i$ | charge on species i |
| $Z_j$ | charge on species j |
| $Z_Y$ | mean coion valence |

Greek Letters

| | |
|---|---|
| $\delta$ | film thickness (L) |
| $\varepsilon$ | bed void fraction |
| $\tau$ | dimensionless time coordinate |

| | |
|---|---|
| $\xi$ | dimensionless space coordinate |
| $\phi$ | electric potential (ergs/coulomb) |
| $\mu$ | solution viscosity (cp) |
| $\omega$ | +1 for cations; -1 for anions |
| $\rho$ | solution density ($M/L^3$) |

Superscripts

| | |
|---|---|
| bar | refers to resin phase |
| * | interfacial equilibrium condition |
| f | column feed condition |
| o | bulk phase condition |

Subscripts

| | |
|---|---|
| c | reference ion |
| i | counterion species |
| j | coion species |
| p | resin particle |
| T | total concentration |
| Mol | molecular form of weak electrolyte |
| Ion | ionic form of weak electrolyte |

# CHAPTER I

# INTRODUCTION

Water is essential to human life. Early civilizations in India, China, Egypt and Greece have recognized the need for pure and clean water. Ancient Egyptians and Chinese were known to have used domestic filters of unglazed earthenware or of sandstone to get purer water for drinking (Wegmann, 1912). Clarification of muddy water by siphoning from one vessel to another through the capillary action of porous material, such as a strip of cloth was also recorded in some Chinese history books. These are probably the earliest activities performed by human beings in purifying water.

Later, with the progress of civilizations, various methods were developed to purify water. In addition to using charcoal for filtration and odor removal, chemical agents, such as chlorine, were often added to water for disinfection and sterilization. Other physical methods such as boiling or distillation to purify water were also popular (James, 1965). In the case of distillation, most of dissolved and suspended impurities are removed, as well as the bacteria being killed. It seems that distillation is one of the perfect methods for water purification. However, the cost of distillation makes it prohibitive for all practical purpose as long as large supplies are required. In addition, the water quality obtained by distillation is only in the level of parts-per-million *(ppm)* range (Porteous, 1975), which is not suitable for specific applications where 'ultrapure' water is desired. Therefore, more advanced technologies such as electrolysis, reverse osmosis, and ion exchange were developed later to obtain purer and purer water.

The water used by industry for boiler feed or process purposes is often taken from public supplies, or abstracted directly from wells, lakes or rivers. Natural water supplies often contain dissolved salts that dissociate in water to form charged species called ions. These ions are usually present in relatively low concentrations and permit the water to conduct electricity. These ionic impurities could lead to problems in cooling and heating systems, steam generation, and manufacturing processes (e.g. wafer fabrication) if the water were used without treatment. The ions that are commonly encountered in most waters include the positively charged ions (called cations) like calcium ($Ca^{2+}$), magnesium ($Mg^{2+}$), and sodium ($Na^{+}$); and the negatively charged ions (called anions), such as alkalinity, sulfate ($SO_4^{2-}$), chloride ($Cl^{-}$), and silica. As soft water (or deionized water) is more desirable for most industrial purposes, the ionic impurities should be removed before using the water. To remove the ionic impurities in water, ion exchange operation is often employed, since

- ion exchange resins have relatively high capacities for ions even at low concentrations;
- resins are stable and readily regenerated; and
- process is excellent for both large and small installations.

This dissertation addresses the modeling and simulation of Mixed-Bed Ion Exchange (MBIE) for ultrapure water processing. The objective is to generalize computer models to handle multicomponent MBIE with more complex solution chemistry.

# Ultrapure Water

Ultrapure water is an integral part of today's high technology industries like the pharmaceutical, biotechnological, environmental, nuclear power and semiconductors. The word *ultrapure* implies free of nearly all impurities. Generally, water with dissolved ionic impurity levels of less than one part-per-billion (ppb) with correspondingly low levels of particulate and microbial contaminants (Sadler, 1993) is termed ultrapure water. The degree to which ultrapure water is required in state-of-art technologies makes it essentially a 'high-tech' material, rather than a simple utility. For instance, ultrapure water is used as a special 'chemical' in pharmaceutical and biotechnological processes, rinse water in microchip manufacturing, and make-up and condensate polishing water in nuclear power generators.

Different users have different standards for the quality of ultrapure water, with the semiconductor industry probably the highest. Table I gives the comparison of ultrapure water with typical city water (Blume, 1987). From this Table, we understand that city water has about 200 times Total Organic Carbon (TOC), 6000 times total solids, 1000 times dissolved silica, 100 times particulates, and 100 ~ 7,000,000 times other ionic impurities relative to ultrapure water. Therefore, city water is 'dirty' for ultrapure water users.

Over the past several decades, the semiconductor industry has put forward the most stringent requirements for ultrapure water. Table II shows the progressive demands for the ultrapure water quality (Zoccolante, 1990) in this industry. To meet these new standards, the technologies to produce ultrapure water are accordingly becoming more and more sophisticated.

Table I. Ultrapure Water vs. City Water Quality (Blume, 1987)

| | | Ultrapure Water | City Water |
|---|---|---|---|
| Total Organic Carbon (TOC) | ppb | 20.0 | 4,000 |
| Total Solids (Residue) | ppb | 50.0 | 288,000 |
| Silica (Dissolved) | ppb | 5.0 | 5,000 |
| $Na^+$ | ppb | 0.05 | 37,000 |
| $K^+$ | ppb | 0.10 | 5,300 |
| $Zn^{2+}$ | ppb | 0.02 | 4 |
| $Cu^{2+}$ | ppb | 0.02 | 2 |
| $Cl^-$ | ppb | 0.05 | 13,000 |
| Bacteria Cell | Per 100 mL | 0 | 0 |
| Resistivity | MΩ -cm | 18.0 | 0.025 |
| Particles (>0.2 micron) | #/L | 10,000 | 1,000,000 |

Table II. Progressive Demands for High-Quality Water in Semiconductor Industry (Zoccolante, 1990)

| Year | Water-quality Goal by application | | | |
|---|---|---|---|---|
| | Total Organic Carbon (ppb) | | Bacteria (cfu/100ml) | |
| | Average | Advanced | Average | Advanced |
| 1975 | <1,000 | <500 | 500 | 100 |
| 1980 | <500 | <200 | 100 | 50 |
| 1985 | <200 | <100 | 50 | 20 |
| 1990 | <20 | <5 | 20 | <1 |

Another leading user of ultrapure water is the electric power industry. This industry also has constant demands for improved water quality. Three major concerns of utility industry are corrosion, scale formation and structural defects. All of these are influenced by water quality. Therefore, impurity limits for the power industry are also in the parts per billion (ppb) ranges. To reduce corrosion, proper pH (usually 9.0 ~ 9.6) of the steam condensate needs to be maintained, which is usually realized by adding ammonia or amines to the system. Specifications for the condensate polisher water requirements are given in Table III, which is for Boiling Water Reactors (BWR). However, the criteria for pressurized water reactors (PWR) is also approaching the similar limit (Harries, 1987).

Table III. Typical Specifications for Ultrapure Water in Power Industry (Spiegler and Laird, 1980)

| Constituent | Contaminant Concentration (ppb) |
|---|---|
| Total dissolved solids | 10 |
| Silica | 5 |
| Fe | 2 - 5 |
| Cu | 0 - 2 |
| Oxygen | 5 |
| Na | < 1 |
| pH | 6.5 - 7.5 |
| Conductance, (MΩ -cm) | 10 |

The manufacture of ultrapure water requires a sophisticated processing, which includes the removal of particulate, inorganic, organic, dissolved, colloidal and microbial impurities. Currently, process operations like ion exchange, electrodialysis, and membrane techniques, or their combinations are often used to produce ultrapure water from pre-treated water. As one of the key steps in obtaining ultrapure water, mixed-bed ion exchange (MBIE) deionlization has been used in industry for several decades. However, theoretical studies of MBIE, especially modeling of multicomponent MBIE process is still very limited. Therefore, more theoretical work on mixed-bed ion exchange, especially modeling such a process to improve process efficiency and product quality is required.

## Ion Exchange

Ion exchange is a phenomenon that reversible exchange of ions takes place between a solid and a liquid with no substantial change in the structure of the solid. The solids are called ion exchangers, which are usually polymeric resins with electrically charged sites. Most ion-exchange resins are made by the copolymerization of styrene and divinylbenzene (DVB), in which the styrene molecules provide the basic matrix of the resin, while the DVB is used to cross-link the polymer to improve physical strength, change properties such as affinities, and act in determining pore size.

Records indicate that the concept of ion exchange existed at least back to Aristotle's time, and possibly biblical times when Moses sweetened the waters of Marah using a tree; *Exodus* 15: 23-25 (Kunin and Myers, 1950; Helfferich, 1962). Moreover, alchemists searched for a material that would change seawater to drinking water long before Arrhenius proposed the existence of ions in water (circa 1887). Aristotle (circa 340 B.C.) used 20 earthen vessels containing a material with ion exchange properties to produce fresh water.

The modern history of ion exchange dates back only to the last century. Two English chemists, Thompson (1850) and Way (1850) first studied the exchange of ions on cultivated soils. They found that soil can absorb the ammonium in manure, and the ammonium salt was then released when lime was added to the soil. Although subsequent investigations by other authors verified their findings, it was not until the beginning of this century that significant breakthrough was achieved. German chemists, Harm and Rupler (1906), developed the first synthetic industrial ion exchange process. Shortly after this, another German chemist, Gans, extended ion exchange to water treatment. Gans (1906) produced synthesized ion exchange minerals called "permutits" by fusing clay, sand, and sodium carbonates, and used them to purify water. Another milestone was a process achieved by Adams and Holmes (1935a,b) who discovered the feasibility of using synthetic organic ion exchange resins. This process helped the development of sulfonated, cross-linked, polystyrene resins by D'Alelio (1944).

The 1940's and 1950's saw comprehensive studies of ion exchange phenomenon using newly discovered resins synthesized from functionalized copolymers of styrene and divinylbenzene (DVB). During this period of time, ion exchange equilibria, selectivity, and diffusion kinetics were studied extensively in a number of laboratories around the world, which further accelerated these materials to a variety of applications.

Today, ion exchange becomes one of the most widely used technologies with applications in chemistry, biotechnology, food processing, pharmaceutical manufacturing, soil-water interactions, trace analysis, and industrial and municipal water treatment (Liberti and Millar, 1985; Streat, 1995). In water treatment area, ion exchange is often used for deionization of industrial waters to obtain ultrapure water in power and semiconductor industries.

Ion exchange is a stoichiometric reaction, that is, when ions in the bulk liquid are 'adsorbed' into resin, an equivalent amount of ions will be released from the resin. With this characteristic, the electric neutrality is maintained in the ion exchange solution, which is a very important assumption for developing model. In practical operations, most ion exchange units are operated as fixed beds called column operations; and the solution can pass the column in down, up and counter flow, with down-flow the most common.

The materials packed in ion exchange column can be made of either cationic or anionic resins, or both cationic and anionic resins. If the former two are the cases, the bed is called mono (or homogeneous) bed, whereas the latter is called mixed bed.

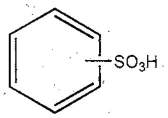
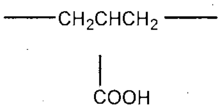
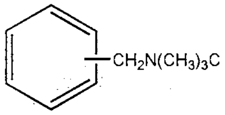
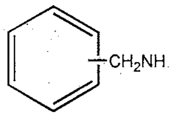
Synthetic ion exchange resins (also called ion exchangers) are usually classified as four major types based on their functional groups: 1) strong-acid cation exchange resin, 2) weak-acid cation exchange resin, 3) strong-base anion exchange resin, and 4) weak-base anion exchange resin. Their typical configurations, functional groups and characteristics are shown in Table IV. Whichever type, all ion exchange resins have the following two common physical properties.

### *Capacity*

The total capacity of an ion exchange resin is defined as the total number of chemical equivalents available for exchange per unit weight or unit volume of resin. Usually, capacity is expressed in terms of milliequivalents per dry gram of resin (meq/g) or in terms of millequivalents per milliliter of wet resin (meq/ml).

Typical resin capacities are in the range of 1.0 ~ 2.5 meq/ml, with cationic capacities in the upper half of this range and anionic capacities in the lower half. Strong acid or strong base resins tend to have constant capacities during the exchange process. However, for weak acid, or weak base resins, the maximum degree of exchange (i.e. the total capacity) also depends upon the pH in the solution phase (Grimshaw, 1975). Because of this, the modeling of ion exchange with weak acid or base resins is more complex. In this dissertation, the ion exchange resin is assumed to have constant capacity. However, the adsorption of the nonionic forms of weak electrolytes onto the resin surface is considered, hence extra amount of species is absorbed by the resin.

Table IV. Classification of Major Ion Exchange Resins (Wachinski, 1997)

| Type | Functional group | Configuration | Features |
|---|---|---|---|
| | **Cationic Resins** | | |
| Strong acid | Sulfonic acid |  | *Useful on all waters*<br>*Complete cation removal*<br>*Variable capacity*<br>*Good physical stabilities*<br>*Low initial cost* |
| Weak acid | Carboxylic acid |  $\overset{}{\underset{\mathrm{COOH}}{-\!\!-\mathrm{CH_2\underset{|}{C}HCH_2}-\!\!-}}$ | *Very high capacity*<br>*High operating efficiency*<br>*Partial cation removal*<br>*Fixed operating capacity*<br>*High initial cost*<br>*Poor kinetics* |
| | **Anionic Resins** | | |
| Strong base | Quaternary ammonium |  | *Complete anion removal*<br>*(including silica & $CO_2$)*<br>*Lower initial cost*<br>*Variable efficiency*<br>*Excellent kinetics*<br>*Shorter rinses*<br>*Limited life* |
| Weak base | Secondary amine<br>(or Tertiary amine) |  | *High operating capacity, regeneration efficiency*<br>*Excellent organic fouling resistance*<br>*Good thermal stability*<br>*Only partial Ani removal*<br>*High initial cost* |

### *Selectivity (or Affinity)*

The selectivity is the preference of resins for one ion relative to another. The selectivity is usually defined in terms of the selectivity coefficient ($K_A^B$), which is the expression of mass action equilibrium (Kunin, 1960).

The selectivity of ion-exchange resins is influenced by properties of the resin bead, the ions being exchanged, and the solution in which the ions are present. According to Helfferich (1962), ion exchange resins have greater selectivities for ions with higher valence or increasing charge. While among ions with the same charge, higher affinities are observed for ions with a higher atomic number (Kunin, 1960). However, the affinity relationships are reversed in concentrated solutions, which makes regeneration of exhausted resins possible.

### *Several terms in ion exchange*

#### 1) Breakthrough

Breakthrough of a bed is defined by the effluent concentration of certain ion. If the effluent concentration of the ion is greater than the specified criterion, then we say the bed is broken through.

#### 2) Ionic throw

In an ion exchange column, ion A which presents in much higher concentration, or has a higher valence than the ion B, forces liberation of some of the ion B, which results in a peak in the effluent concentration curve, is called 'ionic throw' of the ion B.

## Mixed-Bed Ion Exchange

Mixed-bed ion exchange (MBIE) technology is typically used when highly pure water is desired (Bungay, 1989). A MBIE column consists of an intimate mixture of cationic and anionic resins, by which the ionic impurities in water can be reduced to very low concentrations (with a conductivity of less than 0.2 μS/cm), and the dissolved silica can also be removed to less than 2 ppb (Arden, 1968). The mechanism of MBIE is: the cationic resin in the column offers hydrogen ions ($H^+$) to exchange with other positive ions in water, while the anionic resin provides hydroxide ions ($OH^-$) to exchange with negative ions. Then the liberated hydrogen and hydroxide ions react to form water, which is called neutralization. The neutralization reaction decreases the amount of hydrogen and hydroxide ions in the bulk phase, and thus minimizes the reverse exchange of $H^+$ and $OH^-$ with resin.

A MBIE column can be operated in different cycles, depending on the process objectives. The two common cycles often used in the power industry are the hydrogen cycle (HOH cycle) and amine cycle. In hydrogen cycle, the cationic resin is in the hydrogen form while the anionic resin is in the hydroxyl form. After exchange, the excess hydrogen and hydroxide are consumed by the water equilibrium reaction. For amine cycle, the operation is realized either by adding amines to the feed water or using the cationic resin in the amine form. Amine cycle is used for corrosion reduction in the condensate polisher, since it results in alkaline effluent water pH. Besides being operated in different cycles, mixed-bed columns can also be operated in series or in combination with other desalination apparatus to achieve the water processing goals.

Although MBIE is efficient in deionization, it has one drawback – difficult to regenerate. The regeneration of mixed bed requires a series of steps, namely, separation of resin particles by backwashing, individual resin regeneration using strong acids or bases, and remixing of resins by air. Since complete separation of cationic and anionic resins is difficult, the cross contamination of resins during regeneration is inevitable.

Overall, mixed-bed ion exchange has the following major characteristics:

- small space requirements (combining cationic and anionic beds in one unit);
- low operating costs due to smaller amounts of regenerant and water; (Applebaum, 1968)
- a higher discharge rate of ultrapure water;
- constant readiness for operation;
- a low initial investment; and
- difficulty of regeneration (Arden, 1968)

## Mixed Bed Ion Exchange Modeling

MBIE technology has been used for water processing in semiconductor and power industries for several decades; however, theoretical studies of the MBIE process are still limited, especially in modeling MBIE. Caddell and Moison (1954) conducted theoretical study of MBIE. They investigated the variables influencing the breakthrough of mixed-bed, and an empirical relationship between leakage and capacity is developed.

Their work indicated the general applicability of mixed-bed deionization over a wide range of flow rates and concentrations. Frisch and Kunin (1960) further conducted an experimental study on the kinetics of mixed-bed deionization. The effects of influent concentration, flow rate, bed depth, and temperature on bed performance was investigated. They concluded that the ion exchange rate was controlled by a liquid-film mass transfer mechanism at low concentrations.

Haub and Foutch (1986a,b) were the first to model rate limited mixed-bed ion exchange. They developed a model for hydrogen cycle MBIE at ultra-low concentrations with the dissociation of water considered. With this model, the prediction of effluent concentration history of ions is possible. The major advantages of this model are the considerations of separate material balances for each resin and the inclusion of water dissociation effects. But the model was limited to only binary monovalent systems such as $Na^+$ - $Cl^-$ at 25 °C. Divekar et al. (1987) extended this model to incorporate temperature effects, the correlations of physical properties such as diffusivity, viscosity, and dissociation constant as a function of temperature for certain species were developed and implemented in the model. Unfortunately, the temperature dependence of selectivities is not available due to lack of experimental data. Hence the model's capability to handle different temperature remains limited. Zecchini (1990, 1991) extended the above models to deal with a ternary system of monovalent ions with amines. In his work, the amine cycle was first simulated. Pondugula (1994) extended the model to handle divalent ternary systems, and the effect of sulfate generation on column performance was incorporated. Bulusu (1994) further extended the model to handle 3 cations and 5 anions with dissociative carbonates. The flux expression developed by

Franzreb et al. (1993) was incorporated in the model for the first time, which allows the handling of arbitrary valence. Bulusu's model is the prototype of OSU's multicomponent MBIE model. Based on Bulusu's work, a more generalized multicomponent MBIE model was developed by Sunkavalli (1996), and the number of strong electrolytes which can be handled by the model was expanded to 8 cations and 8 anions. Hussey (1996) developed an equilibrium model to calculate the interfacial concentrations for ionic species in MBIE column. A pseudo ion concept is introduced as reference to guarantee the convergence of solution in determining the interfacial concentration. The algorithm developed by Hussey (1996) was successfully incorporated in Sunkavalli's (1996) multicomponent model. Nevertheless, the capability of the model to handle dissociative species is still very limited – only monovalent amines and carbonic species are handled by the model. Therefore, more work on generalizing the model, especially, generalizing the water chemistry for different dissociative species is needed for further expanding the model.

The modeling of MBIE needs not only the understanding of ion exchange equilibrium, but also, and the most important, knowledge of ion exchange kinetics. As we know, equilibrium relationship determines only the final state of a process, but does not indicate the time to reach the state. For a dynamic process, it is the system kinetics that determines the rate to reach equilibrium. Hence to model MBIE, rate-controlling step needs to be determined first. In an ion exchange column, since the rate of exchange is usually much faster than that of diffusion, most ion exchange processes are diffusion-limited. Within the diffusion-limited regime, as shown in Figure 1, the possible rate-controlling mechanisms (Helfferich 1962) could be: i) film diffusion, ii) particle diffusion, and iii) combined film and particle diffusion.

The rate at which ion exchange proceeds is a complex function of physicochemical conditions. The characteristics of the resin, influent impurity concentrations and column operating conditions determine the mechanism. Helfferich (1962) has given a criterion, called Helfferich number *He*, to predict rate-determining step

$$He = \frac{X\overline{D}\delta}{CDr_0}(5 + 2\alpha_B^A) << 1, \quad \text{particle diffusion control}$$

$$He = \frac{X\overline{D}\delta}{CDr_0}(5 + 2\alpha_B^A) >> 1, \quad \text{film diffusion control}$$

$$He \approx 1, \quad \text{both film and particle diffusion controls affect the rate}$$

Recently Bunzl (1995) also discussed this criterion, and based on his experimental work, the criterion has been extended to predict batch processes where the boundary condition is a finite solution volume.

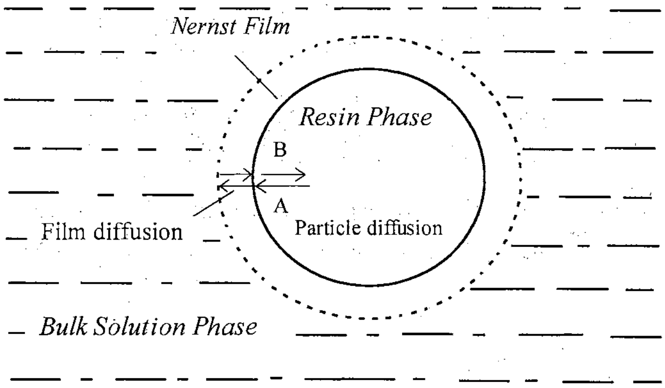


Figure 1. Ionic Diffusion Process Around a Resin Particle

It is believed (Dorfner, 1972) that in the region of low bulk phase concentrations ($\leq$ 0.001 N), the rate is determined by the diffusion of ions through the liquid film surrounding the resin particle (i.e. film diffusion control). While at high bulk solution concentration (about 0.3 N), diffusion of the ions through the resin particle itself is the rate-controlling step (i.e. particle diffusion control). However, Harries (1987) studied the kinetics of ion exchange experimentally. He concluded that in dilute solution (<0.01M), diffusion of ion through the liquid film between the bulk solution and the bead surface is the rate-controlling step. Zecchini (1990) also discussed the effects of solution concentration on the rate-determining step. He concluded that particle diffusion is the rate-controlling step if the bulk phase ionic concentration is above 0.5 M. Below this concentration, the rate-controlling step is most probably film diffusion.

Although different authors gave different criteria of concentration range in predicting the rate-controlling step, we may still conclude that for ultrapure water processes in MBIE, mass transfer is exclusively film diffusion controlled, since the concentration within the column is extremely low (in the level of ppb), which is much less than the above concentration ranges.

With the rate-controlling step decided, the next step of modeling MBIE is to quantify the rate of mass transfer from bulk solution to resin phase. This is usually realized by calculating the ionic flux. For the diffusion of charged species, the Nernst-Planck equation is most often used to describe mass transfer rate, although de la Vega et al (1996) recently showed the equivalence between the Nernst-Planck equation and the "corrected" Fick's law in modeling mixed bed ion exchange processes. They concluded that ion exchange can be simulated with Fick's approach, but the "corrected" Fick diffusivity has to be introduced to achieve the equivalence. In this work, an analytical flux expression developed by Franzreb et al. (1993), which is a solution of the Nernst-Planck equation, is implemented in the model to determine the ionic flux. One of the major advantages of Franzreb's flux expression is that it can handle the general case of arbitrary number of counterions with arbitrary valences. In addition, the analytical form of the expression makes it easy to implement in our model.

The final step of modeling MBIE is to combine the flux expression, dissociation equilibrium, and differential column material balance equations to determine the effluent concentrations, which requires us to solve the partial differential equations and other mathematical expressions governing the process. To achieve this goal, necessary assumptions and appropriate numerical methods are required.

## Objectives

Several specific problems will be addressed in this dissertation; however, expanding the MBIE model to handle multicomponent, multivalent species, especially multivalent dissociative species, is the focus. Dissociative species are also called weak electrolytes. To model ion exchange with weak electrolytes, dissociation equilibrium, which is usually a function of temperature and pH, has to be introduced. This greatly increases model's complexity. In addition, the introducing of multi-step dissociation equilibrium for multivalent species into the model significantly increases the difficulty in solving model equations, because the order of the polynomials to be solved in the model is directly related to the number of dissociation steps, and also the number of species. Hence, when multicomponent multivalent dissociative species are introduced in the model, the program run time taken to solve the model equations will be longer; and the instability of solution will be worse because of the increase of the stillness in model equations. Therefore, more robust numerical method is required to solve the model.

As the goal is to develop a generalized model for multicomponent MBIE with dissociation species, understanding the characteristics of dissociation equilibrium and transport mechanism is very important. For this purpose, a detailed review of ion exchange with weak electrolytes, and a discussion of transport mechanisms of weak electrolytes were presented in Chapter II. The rate expressions for both ionic and nonionic forms of weak electrolytes were also proposed.

Chapter III is the core of this dissertation, in which the development of a generalized rate model for multicomponent, multivalent mixed-bed ion exchange is presented. In addition, the validation of the model by industrial data is also performed. To study the sensitivity of the model to some parameters, the effects of temperature, resin fouling, C/A resin ratio, particle size, and other operating conditions on MBIE column performance are also discussed in this Chapter. As special cases such as variable inlet concentrations, variable inlet flowrate, layer bed (including homogenerous bed and beds in series), bed cleaning, and upset challenge are also often encountered in industry, expanding the model to handle these cases is required. The modeling of special cases will be addressed specifically in Chapter IV.

Silica is another concern of this dissertation, because the semiconductor industry is concerned about the amount of silica in the rinse water. Due to the complexity of silica dissociation chemistry, there has been no successful attempt in modeling MBIE operation with silica. Although Pamarthy (1995) did some preliminary work in predicting silica breakthrough for a MBIE column, the polymeric silicic species (or polysilicates) had not been considered in his model, which is, however, reported the existence by many researchers (Stumm et al., 1967; Iler, 1979). In this work, based on a thorough literature search for silica, a better model that can handle mixed-bed ion exchange with silicic species was developed. In addition, the silica subroutine has been successfully implemented in the multicomponent, multivalent model developed in Chapter III. Simulations also showed good match of predicted results with plant experience. This information is presented at length in Chapter V.

Overall, expanding model to handle multicomponent MBIE, especially systems with multicomponent dissociative species is the main objective of this study. We expect that the developed model can handle most cases of MBIE operations, and the time to reach breakthrough, and the concentration for a given time should be predicted.

# CHAPTER II

# TRANSPORT OF WEAK ELECTROLYTES IN ION EXCHANGE COLUMN

## Abstract

In this chapter, a detailed discussion of transport mechanisms of weak electrolytes in ion exchange column was presented. Methods have been proposed to calculate the mass transfer rate of weak electrolytes in ion exchange process. It is found that the transport of nonionic (molecular) forms of weak electrolytes is important and should be considered in ion exchange with weak electrolytes, especially when the concentrations of nonionic forms are higher. The factors that influence the mass transfer rates of weak electrolytes were also discussed.

## Introduction

Weak electrolytes are substances that have a low percentage (less than 100%) of ionization or dissociation in solution; that is, in a weak electrolyte solution, a dissociation equilibrium exists between the dissociated (ionic) and undissociated (molecular) forms of the electrolytes, with protons or hydroxyl ions liberated (Brady and Holum, 1984). Weak electrolyte species are often encountered in industrial ion exchange processes. For instance, to prevent the corrosion of equipment, amines (a kind of weak electrolyte) are often added to system to adjust the water pH. In addition, in some areas the water resources to be treated by ion exchange often contain rich amount of carbonic or silicic species (both weak electrolytes). Therefore, knowing the transport mechanisms and correctly quantifying the mass transfer rate of weak electrolytes in an ion exchange process are vital for developing a model to handle multicomponent ion exchange.

For ion exchange with weak electrolytes, since both ionic and nonionic forms of species existed in the system, the transport of weak electrolytes into resin is complex. Both the ionic and nonionic forms of weak electrolytes may transfer into resin due to migration or diffusion. Therefore, to describe the transport of different forms of weak electrolytes, specific methods are required. This article addresses the transport of weak electrolytes in ion exchange column. Emphasis will be put on the development of rate equations for weak electrolyte transport.

## Literature on Ion Exchange with Weak Electrolytes

Samuelson (1963) studied the ion exchange with weak electrolytes. He treated the sorption of weak electrolytes, especially the undissociated part (molecular form) as nonionic adsorption. Helfferich (1962) also investigated the sorption of weak electrolytes. He indicated that the amount of weak electrolytes adsorbed into resin could exceed the ion-exchange capacity of the resin, which is greatly different from what we have observed for strong electrolyte systems.

The ion exchange of weak electrolytes is also complicated by the fact that weak electrolytes occur in various pH dependent, ionic states, so that the uptake of weak electrolytes into the ion exchange resin is dependent on the pH of solution. Recently, Jansen et al. (1996b, 1997) studied the effects of pH, solution concentration and dissociation equilibrium on column dynamics of weak electrolyte ion exchange. They found that changing either pH or concentration, while keeping the other variable constant, lead to considerable fluctuations of effluent concentration and pH, respectively. These phenomena, they thought, were caused mainly by the uptake of undissociated (molecular form) electrolytes. Thus, they concluded that the exchange of counterions and sorption of neutral species are the two major factors determining the ion exchange behavior of weak electrolytes.

Jansen et al. (1996a) have also proposed a rigorous equilibrium model to handle ion exchange with strong and weak electrolytes. In their work, besides counterions, the uptake of coions and neutral species has also been taken into account. By employing expressions of selectivity coefficients for all ionic pairs, and distribution coefficients for all neutral species, the model is capable of handling binary or multicomponent systems, especially when weak electrolytes are involved, and electrolyte concentrations greatly exceed the resin capacity.

Helfferich and Hwang (1985) investigated the kinetics of acid uptake by weak-base anion exchangers. The mechanism of proton transfer was proposed to explain the higher mass transfer rates for weak or polybasic acids. Based on their studies, a mathematical model has been developed to describe such processes.

Höll and Sontheimer (1977) discussed the ion exchange kinetics of the protonation of weak-acid ion exchange resins. A mathematical model that incorporated the protonation of weak-acid ion exchange resins with various acids was developed. They concluded that the diffusion of hydrogen ions within a resin sphere is coupled with two major chemical reactions – the protonation of the fixed carboxylic groups and the dissociation of acid molecules.

Wagner and Dranoff (1967) modeled the mass transfer of weak electrolyte into a strong-acid resin. Liquid-film diffusion controlled transport of ammonia was investigated. In their model, the uptake of both dissociated (ionic) and undissociated (molecular) forms of the ammonia were considered. The assumption that the resin surface concentrations of undissociated ammonia and ammonium ion can be ignored was made in order to solve the model equations. The analytical solution of their model has been validated by their limited experimental data.

Helfferich and Bennett (1984) studied the effect of association-dissociation reactions on pH effects in ion-exchange columns. The systems they treated were acetate-acetic acid and sodium carbonate-bicarbonate. Based on their analysis of the mass transfer process, an equilibrium model was developed to predict the column performance, but the sorption of coions and nonionic species had not taken into account.

Separation of amino acid (another kind of weak electrolytes) using ion exchange chromatography was also investigated by Helfferich (1990). In his work, non-electrolyte sorption mechanism was considered. He also found that amino acid dissociation is influenced by the solution pH; and the neutral species in the equilibrium mixture are also adsorbed by ion exchange resin.

Hubner and Kadlec (1978), and Bhandri et al. (1992) studied the mass transfer in weak-base resins. They found that transport of both dissociated and undissociated forms of weak electrolytes are important. The process was modeled by using either double layer or pore equilibrium theory. The model parameter of pore diffusivity in their model was calculated by fitting experimental data. However, their studies indicated that film mass transfer resistance can be neglected. This is because the solutions studied by them are in the high concentration range.

Removal of amine (or ammonia) from aqueous solution via ion exchange has been investigated by many other researchers (Koon and Kaufman, 1975; Dobbs et al., 1975; Gaspard and Martin, 1983; Bolden et al., 1989; Haralambouus et al., 1992; Bernal and Lopez-Real, 1993; Lin and Wu 1996). Gaspard and Martin (1983) firstly examined the ammonia removal characteristics from drinking water by clinoptilolite (a natural zeolite). They found that the transport of ammonia is pH dependent. Bolden et al. (1989) modeled ammonia removal from aqueous solution, the shrinking-core model was used to correlate their experimental data, but the transport of different forms of amine into the resin was not considered. Comparison of the ammonia removal by both the natural and synthetic zeolites from aqueous solution was attempted by Haralambous et al. (1992). They concluded that clinoptilolite has only limited capability for ammonia removal. Recently, Lin and Wu (1996) conducted experimental study of ammonia removal from aqueous solution using synthetic ion-exchange resin. Effects of operating variables such as pH, temperature and initial ammonia concentration on the exchange capacity were examined. A squared driving force model was developed to represent the mass transfer process.

Removal of amine from wastewater by strong cation-exchange resins has also been studied by Yoshida and Kataoka (1987), and Yoshida et al. (1990). In their work, ten representative amines including ammonia was used to be adsorbed on different $H^+$ form ion exchangers. They found that the selectivity for the amine adsorption on the resin is fairly high. Amines of $C_6$-$C_{12}$ species adsorbed onto the resin by multilayer mechanism. But the adsorption capacity of the amine did not change with time, and intraparticle diffusivity of the amine was constant during the adsorption process. Based on these facts, a model was developed to model the amine adsorption process. In the model, the undissociated amine was treated as immobilized by the resin with a neutralization reaction, that is, the uptake of the undissociated amine is also realized by the ion exchange reaction rather than the physical adsorption. By considering both external and internal mass transfer resistances, the breakthrough curves predicted by the model matched favorably with the experimental data.

Meichik et al. (1989) studied the kinetics of ion exchange with weak acid or weak base resin. They indicated that both internal diffusion and chemical reaction contribute to the ion exchange rate. Therefore, concentrations of all species involved in the reaction have to be considered in solid-liquid interface.

Nakanishi (1978) measured the diffusivities of nonelectrolytes in dilute solution based on a generalized Hammond-Stokes plot. A new general correlation is proposed to predict the diffusion coefficients for nonelectrolytes. The average deviation of this correlation from experimental results is only ± 9.1%.

Zecchini (1990) modeled the ion exchange with amines at ultra-low concentrations. The effect of water dissociation on the amine exchange was studied. However, the transport of non-ionic form to resin phase has not been considered in the model.

Chowdiah (1996) studied the transport of weak electrolytes, particularly amine transfer from solution to the resin phase in a mixed-bed ion exchange column. The uptake of both the ionic and molecular form of amines by the resin has been considered in his work. He used Fick's law to describe the transport of molecular (nonionic) form, and the Nernst-Planck equation to the ionic form. The predicted results also compare favorably with industrial experience.

Lou (1997) recently studied the kinetics of boron sorption and desorption in boron thermal regeneration systems (BTR). He concluded that the migration of molecular boron is the dominant mass transfer process, so that the contribution of borates to the total boron sorption capacity can be ignored. A mathematical model is developed to predict the BTR performance.

Huang et al. (1991) studied the sorption of boric acid on weak-base anion exchange resin. The equilibrium isotherms and ionic migration velocity were measured at both low and high concentration regions. They showed that the migration velocity of boric acid could be calculated using simple solute movement equation. The sorption of molecular species, rather than ion exchange of the counter-ions, is the major process of mass transfer inside the pore of the resin was concluded.

Sahin (1996) recently investigated boron adsorption from wastewater of boric acid and borax using Amberlite XE-243 resin in an ion exchange column. He concluded that boron removal via ion exchange was affected by the flow rate, solution pH, and initial concentration of solution, because all of these factors influence the sorption capacity of resin. Based on his experimental study, an empirical equation relating boron concentration with filtrate resin volumes was developed.

Summarizing the above literature, the following conclusions can be made regarding the ion exchange of weak electrolytes.

1. The uptake of weak electrolytes by ion exchange resins has several mechanisms:
    - Transport of ionic form into resin as an ion exchange counter-ion
    - Protonation of the molecular form into resin, which takes up ion exchange capacity
    - Pure physical adsorption of the molecular forms to the resin, which takes no exchange capacity
2. The factors influencing the mass transfer of weak electrolytes could be:
    - valence of the weak electrolyte
    - pH of the solution
    - particle size of resin
    - degree of cross-linkage of resin

- the concentration of solution
- dissociation equilibrium

## Transport Rate of Weak Electrolytes into Resin

As both ionic and non-ionic forms of weak electrolytes may transfer to the resin via different mechanisms in an ion exchange column, the mass transfer rates for both ionic and nonionic species have to be estimated to model MBIE with weak electrolytes.

Usually the transport rate of a species is directly related to its flux. Hence the flux expressions for both ionic and nonionic species have to be developed first. Due to combined transfer mechanisms of both ionic and non-ionic forms of weak electrolytes, the mass transfer flux for weak electrolytes should be the summation of both, that is

$$J_{Total} = J_{Ion} + J_{Mol} \qquad \text{(II-1)}$$

where $J_{Ion}$ indicates the flux of ionic form; $J_{Mol}$ is the flux of nonionic form.

### Flux for Ionic Forms

Mass transfer of ionic species in aqueous solutions has been recognized as greatly different from that of nonionic form for a long time. The diffusion characteristics of an ionic species in dilute solution are usually described by the Nernst-Planck equation (Helfferich, 1962), which is

$$J_i = -D_i\left[\frac{\partial C_i}{\partial r} + \frac{Z_i C_i F}{RT}\frac{\partial \phi}{\partial r}\right] \qquad \text{(II-2)}$$

where the first term denotes the mass transfer rate due to the concentration gradient; the second term represents the effect of electric potential gradient.

Traditionally, integration is used to solve the above equation to get the flux expression for each species. Pleijel (1910) was probably the first to engage in this study. He integrated Eq (II-2) for several ionic species over the Nernst film. Schlögl (1954) extended this integration method to cases with electric current. All these studies dealt with diffusion across a stagnant film between two totally mixed electrolyte solutions in which all concentrations are given.

Schlögl and Helfferich (1957) further applied the Nernst-Planck equation to the film-diffusion controlled ion exchange process. They presented solutions to the Nernst-Planck equation for the cases with and without consideration of the electric field. Unfortunately, their solution is restricted only to binary systems with monovalent species and no selectivity of the resin. To get more generalized solutions, Kataoka et al. (1968), and Turner and Snowdon (1968) presented solutions for binary exchange with arbitrary valences. Rahman (1979) and Kataoka et al. (1987) further gave out analytical approximations for the case of ternary systems with arbitrary valences. However, these approximations are too complex to be expanded to general multicomponent cases.

The most significant progress in describing film-controlled mass transfer in ion exchange was made by Franzreb et al. (1993). In their work, a generalized solution for ionic fluxes of a multicomponent system was developed. Unlike many previous studies, differentiation rather than integration was used to solve the Nernst-Planck equation, so that an analytical flux expression for general case of multicomponent, arbitrary valence systems was obtained.

In this work, the method developed by Franzreb et al. (1993) is followed to derive the flux expression for the ionic form of weak electrolytes, and the following assumptions are made.

1) The ion exchange is controlled by the diffusion across a Nernst film;
2) The curvature of the stagnant Nernst film can be neglected because of the small thickness of the film compared to the particle size;
3) The bulk phase is totally mixed and may contain any number of counterions and coions of arbitrary valences;
4) The loadings on the resin and the compositions in the bulk phase are known;
5) Individual diffusion coefficients are constant at the specific temperature;
6) All activity coefficients are unity because of the dilute solution;
7) The electric potential has only a gradient in *r* (radial) direction;
8) Pseudo-steady state and complete Donnan exclusion; and
9) No coion flux, and no net electric current flow.

Based on the above assumptions, and started derivations from the Nernst-Planck equation (Eq II-2), an ionic flux expression for any species is developed (see Appendix C for the detailed derivation)

$$J_i = \frac{D_i}{\delta}\left((1-\frac{N_i}{P})(C_i^* - C_i^o) + N_i A_i (1+\frac{1}{P})(C_T^* - C_T^o)\right) \qquad \text{(II-3)}$$

where $N_i = -\frac{Z_i}{Z_Y}$ (II-4)

$$P = \frac{\sum_{i=1}^{n} N_i D_i \left(X_i^* - X_i^o\right)}{\sum_{i=1}^{n} D_i \left(X_i^* - X_i^o\right)} \qquad \text{(II-5)}$$

$$C_T^* = \left( \frac{\sum_{i=1}^{n} (1+N_i) D_i X_i^o}{\sum_{i=1}^{n} (1+N_i) D_i X_i^*} \right)^{1/P+1} C_T^o \qquad \text{(II-6)}$$

$$A_i = \frac{1}{C_T^o}\left(Z_i C_i^o - B_i \left(C_T^o\right)^{-P}\right) \qquad \text{(II-7)}$$

$$B_i = \omega \frac{X_i^* - X_i^o}{\left(C_T^*\right)^{-P-1} - \left(C_T^o\right)^{-P-1}} \qquad \text{(II-8)}$$

Since the total interfacial concentration is incorporated in the flux expression – Eq (II-3), which is also dependent on the ion exchange equilibrium, the iteration method has to be used to calculate the ionic flux. In this study, the fluxes for a given system are calculated by the following iteration methods:

1) Selection of an initial value for the total interfacial concentration, usually, let $C_T^* = C_T^o$;
2) Calculation of the mole fractions at the interface by means of the resin loadings using the local equilibrium;
3) Calculation of P and an improved value for $C_T^*$ with Eq (II-5) and Eq (II-6). If the difference between the new and old values of $C_T^*$ exceeds the chosen tolerance, repeat steps 2) and 3);
4) Calculation of $A_i$ and $B_i$ for all counterions using Eq (II-7) and Eq (II-8);
5) Calculation of the ion fluxes using Eq (II-3).

Flux for Nonionic forms

The transport of nonionic (molecular) form species is usually described by Fick's second law, that is

$$J_{Mol} = -D_{Mol} \frac{dC_{Mol}}{dz} \tag{II-9}$$

Integrating Eq (II-9) yields

$$J_{Mol,i} = \frac{D_{Mol,i}}{\delta}(C^{o}_{Mol,i} - C^{o}_{Mol,i}) = k_{Mol,i}(C^{o}_{Mol,i} - C^{o}_{Mol,i}) \tag{II-10}$$

where $k_{Mol,i} = \frac{D_{Mol,i}}{\delta}$ is molecular mass transfer coefficient.

Frequently in a multicomponent system with dissociative species, the concentration of molecular form is much greater than that of ionic form, so that the effect of molecular concentration on the total mass transfer rate has to be taken into account (Jasen, 1996b). To model such cases, an enhancement factor is introduced in this work to account for the effect of high molecular concentration on the mass transfer rate, that is

$$\alpha = \mathrm{Log}\left(\frac{D_{Mol}C_{Mol}}{D_{Ion}C_{Ion}}\right) \tag{II-11}$$

where $D_{Mol}$ and $C_{Mol}$ are the diffusivity and concentration of the molecular form

$D_{Ion}$ and $C_{Ion}$ are the diffusivity and concentration of the ionic form.

Therefore, the flux for the molecular form of weak electrolyte is

$$J_{Mol,i} = (1+\alpha)\frac{D_{Mol,i}}{\delta}(C^{o}_{Mol,i} - C^{*}_{Mol,i}) \tag{II-12}$$

Then the total flux for the weak electrolyte is

$$J_{\text{total},i} = \frac{D_i}{\delta}\left((1-\frac{N_i}{P})(C_i^{*} - C_i^{o}) + N_i A_i (1+\frac{1}{P})(C_T^{*} - C_T^{o})\right)$$

$$+ (1+\alpha)\frac{D_{\text{Mol},i}}{\delta}(C^{o}_{\text{Mol},i} - C^{*}_{\text{Mol},i}) \qquad \text{(II-13)}$$

However, when $C_{\text{Mol}} \leq C_{\text{Ion}}$, the effect of molecular concentration on the transfer rate of weak electrolytes is not so important that the enhancement factor should not be incorporated. Thus the total flux for the weak electrolytes should be calculated by

$$J_{\text{total},i} = \frac{D_i}{\delta}\left((1-\frac{N_i}{P})(C_i^{*} - C_i^{o}) + N_i A_i (1+\frac{1}{P})(C_T^{*} - C_T^{o})\right)$$

$$+ \frac{D_{\text{Mol},i}}{\delta}(C^{o}_{\text{Mol},i} - C^{*}_{\text{Mol},i}) \qquad \text{(II-13A)}$$

Rate Equations

The rate of mass transfer relates to its flux by the following equation

$$\frac{\partial q_i}{\partial t} = -J_i a_s \qquad \text{(II-14)}$$

where $q_i$ is the amount of weak electrolytes “i” in resin phase; $a_s$ is the interfacial area between the bulk solution and resin phase.

According to this equation, the rate of weak electrolyte “i” transporting to resin can be determined as long as the flux of weak electrolyte “i” is obtained.

Interfacial Concentrations

From Eq (II-3), we see that the interfacial concentration $C^*$ for species “i” of weak electrolyte needs to be determined to calculate ionic flux. In this work, for the interfacial concentrations of ionic species, the local equilibrium is assumed to exist between bulk and resin phase, so that the selectivity expression can be used to calculate the interfacial concentration, that is

$$K_A^B = \left(\frac{q_B}{C_B^*}\right)^{Z_A}\left(\frac{C_A^*}{q_A}\right)^{Z_B} \tag{II-15}$$

Rewrite it in equivalent fraction form, we have

$$K_A^B = \left(\frac{Y_B}{X_B^*}\right)^{Z_A}\left(\frac{X_A^*}{Y_A}\right)^{Z_B} Q^{(Z_A-Z_B)}C_T^{*(Z_B-Z_A)} \tag{II-16}$$

In general, for species "i", it is

$$X_i^* = Y_i\left(K_A^i\right)^{-1/Z_A}\left(\frac{X_A^*}{Y_A}\right)^{Zi/Z_A}\left(\frac{Q}{C_T^*}\right)^{1-Zi/Z_A} \tag{II-17}$$

However, for the interfacial concentration of molecular forms, we use the following expression to calculate the interfacial concentrations, which is based on the dissociation equilibrium between the interfacial concentrations of molecular and ionic forms.

$$[AH]^* = \frac{[A^-]^*[H^+]^*}{K_a} = \frac{[A^-]}{K_a}\frac{K_w}{[OH^-]} \qquad \text{(for acids)} \tag{II-18}$$

$$[BOH]^* = \frac{[B^+]^*[OH^-]^*}{K_b} = \frac{[B^+]}{K_b}\frac{K_w}{[H^+]} \qquad \text{(for bases)} \tag{II-19}$$

where the superscript "$*$" denotes interface. Therefore, $[A^-]^*$, $[B^+]^*$, $[H^+]^*$, and $[OH^-]^*$ are the ionic concentrations at interface, which are calculated from the Eq (II-17).

## Factors Influencing the Transport Rate of Weak Electrolytes

Generally for interfacial mass transfer, the transport rate is directly affected by the driving force – concentration difference between two phases. The larger the concentration difference, the higher the mass transfer rate. Therefore, the first factor influencing the mass transfer of weak electrolytes should be the concentration of each form on the resin and solution phase.

The uptake rate of weak electrolytes by ion exchange resins from an ambient solution and the transport of species from solution to the resin phase are also affected by the dissociation equilibrium and the equilibrium between the solution and resin phase. Because dissociation equilibrium determines the amount of each form of dissociative species in the solution, while the equilibrium between the resin and solution phase limits the maximum amount of the species going to resin.

In addition, since the dissociation of weak electrolytes is pH dependent, the amount of both ionic and nonionic forms will be influenced by the solution pH, given the total concentration of weak electrolytes. Hence mass transfer rate of weak electrolytes between the solution and resin phase is also influenced by solution pH. Generally, if the dissociation is favored by acidic conditions, lower pH leads to higher ionic concentration, so that the mass transfer rate for ionic form will be faster; at the same time, the mass transfer rate for the molecular form will be slower.

Another factor influencing the rate of mass transfer of weak electrolytes is the resin properties. Both the particle size and degree of cross-linkage of resin will influence the transport rate of weak electrolytes, because the smaller the resin particle, the larger the surface area of resin, thus the higher the mass transfer rate.

## Conclusions

The transport of weak electrolytes in an ion exchange column has been discussed. Methods have been proposed to calculate the mass transfer rate of weak electrolytes in ion exchange process. It is founded that the transport of nonionic (molecular) form of weak electrolytes to resin is important, especially when the concentrations of molecular form are higher. The mass transfer rate of weak electrolytes is the summation of both ionic and nonionic rates.

A generalized ionic flux expression in the form of Eq (II-3) was developed. However, to calculate the ionic flux using this expression, iteration has to be used.

The factors influencing the transport of weak electrolytes include solution pH, dissociation equilibrium, resin properties and both the ionic and nonionic concentrations of weak electrolytes.

## CHAPTER III

# GENERALIZATION OF A MIXED-BED ION-EXCHANGE MODEL WITH MULTICOMPONENT DISSOCIATIONS

## Abstract

A generalized rate model for multicomponent mixed-bed ion exchange (MBIE) has been developed. Apart from an arbitrary number of strong electrolytes, the model was extended to handle most weak electrolytes; such as monovalent, divalent and trivalent amines, organic acids, carbonates and phosphates. The total number of species handled by the model is 18 cations and 18 anions. The model was used to predict the effluent history of MBIE columns. Simulations showed satisfactory agreement of predictions with real plant experience. The effects of mass transfer coefficient, cation/anion ratio, particle size and other operating conditions on column performance were studied.

## Introduction

Pure water is important for many applications, such as power generation, food processing, and semiconductor chip rinsing. Water containing impurities tolerable to household use is frequently unacceptable for industry. For instance, scale or corrosion can hinder the performance of boilers and pipes, leading to decreased production or shutdown. Impurities can also impair the taste of foodstuff and the operability of microprocessor chips. Therefore, removing ionic impurities in water to obtain ultrapure water is very important to these industries.

Mixed-bed ion exchange (MBIE), where cationic and anionic resins are intimately mixed together for service flow, and separated hydraulically for regeneration, is an effective way for deioniziation. Since its invention in 1950's, MBIE has been widely used in industry for ultrapure water processing; but in most cases, the water treated by MBIE columns often contain multicomponent multivalent ions. Therefore, a generalized model that can handle MBIE with arbitrary number of species and arbitrary valence is required.

OSU began modeling mixed-bed ion exchange as early as 1982, and significant effort has been made (Haub, 1984; Divekar, 1987; Zecchini, 1990; Yoon, 1990; Noh, 1992; Pondugula, 1994; Bulusu, 1994; Sunkavalli, 1996; and Hussey, 1996). By the end of 1996, the MBIE model developed by OSU was able to handle as many as 8 cations and 8 anions of strong electrolytes; but the number of weak electrolytes handled by the model was still limited. Only monovalent amines and carbonates were addressed. The objective of this work is to generalize the MBIE model for systems with multicomponent, multivalent species. Specifically, extending the model to handle various dissociative species like monovalent amines, organic acids, carbonates and phosphates is the focus of this work. With the developed model, the time for a MBIE column to reach breakthrough and the effluent concentrations at specific time should be predicted for a real multicomponent system. Furthermore, the generalized model should also be able to handle some special cases of MBIE operations; such as variable inlet concentrations, variable inlet flowrate, bed cleaning, and layered-bed. The discussion of special cases will be presented separately in Chapter IV.

## Literature on Multicomponent Ion Exchange

Two theories are often used to describe multicomponent ion exchange – equilibrium and rate. Equilibrium theory assumes that local equilibrium exists between the resin and bulk phase, and the mass transfer resistance between two phases can be neglected. In addition, the absence of axial diffusion, constant separation factors, uniform presaturation, and constant feed are also assumed. According to this theory, ion exchange equilibrium is described by the standard mass action law (Kunin, 1958), and the effluent ionic concentrations are predicted by solving the material balance equations along with the equilibrium relationship.

As local equilibrium is not usually attained for most ion exchange processes, rate theory, which is based on the rate law of diffusion, is more popular. In applying rate theory, Fick's law and the Nernst-Planck equation are often employed to describe the rate of interfacial mass transfer. Moreover, column material balance, equilibrium relationship, and kinetic law expressions are needed to combine together to predict the effluent concentrations.

In rate theory, film and particle diffusion control are the two most commonly used mechanisms, although sometimes both film and particle diffusion are considered (Helfferich, 1990). In addition, the linear driving force approximation (Glueckauf and Coates, 1947; Vermulen and Hiester, 1954, Wong and Niedawiecki, 1982) is also assumed. In recent years, nonlinear driving force expressions for the rate receive more attention (Franzreb et al., 1993; Lin and Wu, 1996).

## Multicomponent Ion Exchange Kinetics

To simulate multicomponent mixed-bed ion exchange, the kinetics of ion exchange must be understood first. Boyd et al. (1947) studied ion exchange kinetics, Nernst concept of a liquid diffusion layer was applied to model the process. In addition, they indicated that particle and film diffusions are the two major rate-controlling steps in ion exchange. Since then, many other investigators (Helfferich, 1965; Kataoka et al., 1976; Wagner and Dranoff, 1967; Omatete et al., 1980a,b; Wildhagen et al., 1985; Yoshida and Kataoka, 1987) have studied ion exchange kinetics or column dynamics; and numerous models were presented to describe ion exchange. Haub (1984), Zecchini (1990) and Yoon (1990) have conducted extensive review of this topic. Therefore, only those efforts related to this study will be reviewed here.

Studies on liquid-side mass transfer in binary ion exchange systems have been carried out by many researchers (Copeland et al. (1967); Copeland and Marchellow, (1969); Glasski and Dranoff (1963); Kataoka et al. (1968, 1971, 1973); Schlögl and Helfferich (1957); Smith and Dranoff (1964); Turner and Snowdon (1968); and Chowdiah (1996). Schlögl and Helfferich (1957) were the first to apply the Nernst-

Planck (N-P) equation to describe the fluxes of ionic species. In their work, the kinetics of a binary system was studied. They showed that the electric field caused by the difference of the diffusivities has a great effect on ion exchange rate.

In order to extend the Nernst-Planck equation to handle multicomponent systems, Kataoka et al. (1987) studied the film-diffusion controlled liquid-side mass transfer in a ternary system. Flux expressions for the ions with equal valences and ions with different valences have been developed separately. The numerical solution of their model agreed reasonably well with the experimental results. However, the model cannot be extended to more than three species.

Yoshida and Kataoka (1987) studied the intraparticle mass transfer in ternary ion exchange systems. The effects of electric field caused by the difference of counterion diffusivities, the change of the volume of resin particle, and the change of the activity coefficient of ionic species on the process dynamics has been considered. Based on their study, a mathematical model was developed to predict the mean ionic concentration profile in the resin phase.

The effective diffusivity is a very important parameter in modeling multicomponent ion exchange process. Wildhagen et al. (1985) measured the effective diffusivity in multicomponent systems. Various definitions of effective liquid-side diffusivity in binary ion exchange systems have been tested using their experimental data. They showed that the effective diffusivity defined by film theory could be correlated using available mass transfer correlations. They also indicated that the theory developed by them could be extended to treat ternary systems.

Copeland et al. (1967) studied the influence of resin selectivity on film diffusion-controlled ion exchange process. They found that selectivity has a remarkable effect on the rate of exchange for a wide range of diffusivities. Like many other researchers, a mathematical model based on the film diffusion mechanism was developed.

Yu and Wang (1989) developed a general rate equation model to simulate the column dynamics of multicomponent adsorption and ion exchange. The model took into account axial dispersion, film and intraparticle diffusion, size exclusion and interference (competition of solutes for sorbent sites) effects for systems with nonlinear isotherms (or variable separation factors). To solve the model equations, a gradient-directed moving finite element method was developed, and the solid phase concentration was related to the bulk phase concentration through the equilibrium isotherms.

The most important progress in modeling film-diffusion controlled liquid-phase mass transfer in multicomponent ion exchange is probably the work by Franzreb et al. (1993). By introducing a pseudo ion concept and total equivalent concentration, the Nernst-Planck equation was solved analytically, and a generalized solution of ionic flux is presented for multicomponent systems with an arbitrary number of species and valences. They also performed experiments to validate their generalized solution.

Haub and Foutch (1986a,b) was the first to model mixed-bed ion exchange (MBIE). They developed a rate model for hydrogen cycle MBIE at ultra-low concentrations with the dissociation of water considered. However, the model was limited to only binary monovalent systems such as $Na^+$ - $Cl^-$ at 25 °C. Divekar et al. (1987) extended this model to incorporate the temperature effect, and correlations of physical properties such as diffusivity, viscosity, and dissociation constant as a function of temperature for certain species were developed and implemented in the model. Zecchini (1990, 1991) extended the above models to deal with a ternary system of monovalent ions with amine. In his work, the amine cycle was simulated successfully. Pondugula (1994) extended the model to handle divalent ternary systems; and the effects of sulfate generation on column performance was incorporated. Bulusu (1994) further extended the model to handle 3 cations and 5 anions with dissociative species – carbonates. The flux expression developed by Franzreb et al. (1993) was incorporated in his work, which makes the model handle arbitrary valence possible. Sunkavalli (1996) further extended Bulusu's model to handle 8 cations and 8 anions. Dissocative species like monovalent amines as well as carbonates can be simulated using this model.

Summarizing the above literature, we may conclude that of all the models developed to date, no model can handle mixed-bed ion exchange with arbitrary number of dissociative species. In addition, such systems are industrially significant for ultrapure water processing. Therefore, a model that can handle multicomponent mixed-bed ion exchange with arbitrary number of species and arbitrary valency, especially multivalent dissociative species is required.

## Model Development

A mathematical model is a collection of equations that describes some aspects of the behavior of the system being investigated. Generally, to model a process, knowledge of mathematics and insights of the problem are required.

By analyzing the physical process in a multicomponent mixed-bed ion exchange column (both strong and weak electrolytes may be involved in), we model the MBIE process into the following four integrated parts:

- Equilibrium between the bulk solution and resin surface (local equilibrium)
- Mass transfer from bulk solution to resin phase (flux expression)
- Bulk phase neutralization (weak electrolyte dissociation equilibrium, water dissociation effect, and charge balance)
- Column material balance (the relationship of concentration changing between the bulk solution and resin phase)

Specifically, the Nernst-Planck equation is used to determine the ionic flux, since it incorporates the effect of electric potential (Newman, 1973). Meanwhile, Fick's law is used to describe the flux of nonionic form for weak electrolytes. The interfacial concentrations, which are needed in calculating mass transfer rates, are determined from the selectivity expression (for ionic form), or from dissociation equilibrium (for molecular form) separately. To account for the bulk phase neutralization, the dissociation equilibria of weak electrolytes and water dissociation equilibrium are incorporated in the charge balance. Based on these efforts, the effluent concentrations can be determined by solving the column material balance equations along with the rate expressions numerically. In this chapter, only the major equations relevant for computations are presented. Detailed derivations of these equations are presented in the Appendixes A through D.

## Assumptions

As usual, assumptions have to be made to model a process. In this work, the number of assumptions has been kept to a minimum for model generalization. Of all the assumptions made, the most important is that ion exchange process is film-diffusion controlled, because industrial MBIE columns are usually operated at high flow rates and low influent ionic concentrations.

The other important assumptions are:

- Ions with higher valence are preferred by ion-exchange resins. Between ions with the same valence, the ion with higher selectivity is preferred.
- Ionic forms of species are exchanged with resin as counterions.
- The transport of nonionic (molecular) forms of species has two mechanisms – protonation to resin exchange site and molecular adsorption to resin matrix.
- The protonation of nonionic form is assumed being adsorbed by an ion-exchange-resin site-sorbate interaction, which takes the ion exchange capacity.
- The adsorption of nonionic form is a pure physical adsorption process that does not use any ion exchange capacity.
- For weak electrolytes, if the concentrations of the molecular form $C_{Mol}$ is greater than that of the ionic form $C_{Ion}$, the mass transfer rate of nonionic species is enhanced by a factor, $\mathrm{Log}\left(\frac{D_{Mol}C_{Mol}}{D_{Ion}C_{Ion}}\right)$, so that the actual mass transfer rate is $(1+\mathrm{Log}\left(\frac{D_{Mol}C_{Mol}}{D_{Ion}C_{Ion}}\right))$. Where $D_{Mol}$ and $D_{Ion}$ are the diffusivities of molecular and ionic form.

- The overall mass transfer rate for weak electrolytes is the addition of both ionic and molecular transfer rates.

Besides the above, other assumptions have also been made. Table I lists all the assumptions that are used in the model.

## Model Equations

Ion exchange is a diffusion process involving charged species. The Nernst-Planck equation (Newman, 1973) is used to describe the ionic diffusion flux, which is

$$J_i = -D_i \left[ \frac{\partial C_i}{\partial r} + \frac{Z_i C_i F}{RT} \frac{\partial \phi}{\partial r} \right] \qquad \text{(III-1)}$$

where the first term denotes the mass transfer rate due to the concentration gradient; the second term represents the effect of the electric potential gradient.

However, for the nonionic (molecular) forms of weak electrolytes, Fick's second law is employed to describe the mass transfer flux, which is

$$J_{Mol} = -D_{Mol} \frac{dC_{Mol}}{dz} \qquad \text{(III-2)}$$

TABLE I

Model Assumptions

---

1. Process is film diffusion controlled.
2. The Nernst-Planck model incorporates all interactions between diffusing ionic species.

3. Nonionic (molecular) form of weak electrolytes can be adsorbed into the resin by two mechanisms – protonation and physical adsorption.
4. Fick's law is used to model the transfer rate of nonionic form into the resin.
5. The adsorbed amount (equilibrium value) of molecular form on the resin can be determined by a Freundlich isotherm.
6. An ion with a higher valence is preferred to an ion with a lower valence by the ion-exchange resin.
7. Between ions with the same valence, the ion with higher selectivity is preferred to the ion with lower selectivity by the ion-exchange resin.
8. Pseudo steady state exchange (variations of concentration with space are much more important than with time).
9. No net current flow and no net coion flux within the film.
10. No coion flux across the particle surface.
11. Local equilibrium at solid - film interface.
12. Curvature of the film is negligible.
13. Selectivity coefficients are constant and temperature independent.
14. Binary selectivity coefficients can be used for multicomponent ion exchange.
15. Bulk phase neutralization.
16. Uniform concentrations across the resin bead.
17. Uniform bulk phase concentrations.
18. Ion exchange reactions are instantaneous compared to the overall rate of exchange.
19. Activity coefficients are constant and unity.
20. Negligible axial dispersion and plug flow.
21. Isothermal, isobaric operation.

---

## Dissociation Equilibrium

To generalize the model for multicomponent dissociative systems, three types of most common dissociative species have been considered; namely, type I; type II; and type III dissociation species. Their dissociation characteristics are expressed in the following generalized equilibrium equations. We expect that any species with the similar dissociation chemistry should be handled by the model.

### *Type I dissociative species*

Monovalent amine (e.g. ammonia) and carboxylic acid (e.g. acetic acid) are typical examples of type I cationic and anionic species. This type of species has only one-step dissociation equilibrium, that is,

For monovalent amines

$$RNH_2 + H_2O \xleftrightarrow{K_b} RNH_3^+ + OH^- \qquad \text{(III-3)}$$

$$K_b = \frac{[RNH_3^+][OH^-]}{[RNH_2]} \qquad \text{(III-4)}$$

For organic acids

$$RCOOH \xleftrightarrow{K_a} RCOO^- + H^+ \qquad \text{(III-5)}$$

$$K_a = \frac{[RCOO^-][H^+]}{[RCOOH]} \qquad \text{(III-6)}$$

Eq (III-4) and Eq (III-6) have the same form. Therefore, they can be combined into a generalized equation, that is

$$K = \frac{[TypeI\_Ion][Dum\_Ion]}{[TypeI\_Mole]} \qquad \text{(III-7)}$$

Where, Dum_Ion represents either $H^+$ or $OH^-$ depending on whether the type I species is acid or base.

In the computer code, Eq (III-7) was implemented in the computational algorithm. Therefore, any weak electrolytes with the similar dissociation equilibrium like Eq (III-7) should be handled by the model.

*Type II dissociative species*

Type II species have the characteristics of two step dissociation. For instance, diprotic acid – carbonate is an example of type II dissociative species. Divalent amines that dissociate in two steps are other examples of type II dissociative cationic species. The dissociation equilibrium of carbonates can be expressed as

$$\mathrm{CO_2 + H_2O \longleftrightarrow H_2CO_3} \tag{III-8}$$

$$\mathrm{H_2CO_3 \xleftrightarrow{K_1} H^+ + HCO_3^-} \tag{III-9}$$

$$\mathrm{HCO_3^- \xleftrightarrow{K_2} H^+ + CO_3^{2-}} \tag{III-10}$$

$$K_1 = \frac{[H^+][HCO_3^-]}{[H_2CO_3]} \tag{III-11}$$

$$K_2 = \frac{[H^+][CO_3^{2-}]}{[HCO_3^-]} \tag{III-12}$$

Hence, the dissociation equilibrium for type II dissociative species can be generalized by the following dissociation equations

$$K_1 = \frac{[\mathrm{TypeII\_MonoIon}][\mathrm{Dum\_Ion}]}{[\mathrm{TypeII\_Mole}]} \tag{III-13}$$

$$K_2 = \frac{[\text{TypeII\_DiIon}][\text{Dum\_Ion}]}{[\text{TypeII\_MonoIon}]} \qquad \text{(III-14)}$$

With the Eq (III-13) and Eq (III-14) implemented in the computer code, any weak electrolytes having the same dissociation equilibrium as the above should be handled by this model.

*Type III dissociative species*

Type III species have the characteristics of three-step dissociation. For instance, triprotic acid – phosphate is a typical type III dissociation species. The dissociation equilibrium relationships for phosphate are

$$H_3PO_4 \xleftrightarrow{K_1} H^+ + H_2PO_4^- \qquad \text{(III-15)}$$

$$H_2PO_4^- \xleftrightarrow{K_2} H^+ + HPO_4^{2-} \qquad \text{(III-16)}$$

$$HPO_4^{2-} \xleftrightarrow{K_3} H^+ + PO_4^{3-} \qquad \text{(III-17)}$$

$$K_1 = \frac{[H^+][H_2PO_4^-]}{[H_3PO_4]} \qquad \text{(III-18)}$$

$$K_2 = \frac{[H^+][HPO_4^{2-}]}{[H_2PO_4^-]} \qquad \text{(III-19)}$$

$$K_3 = \frac{[H^+][PO_4^{3-}]}{[HPO_4^{2-}]} \qquad \text{(III-20)}$$

In general, the dissociation equilibrium of type III species can be represented by

$$K_1 = \frac{[\text{Dum\_Ion}][\text{TypeIII\_MonoIon}]}{[\text{TypeIII\_Mole}]} \qquad \text{(III-21)}$$

$$K_2 = \frac{[\mathrm{Dum\_Ion}][\mathrm{TypeIII\_DiIon}]}{[\mathrm{TypeIII\_MonoIon}]} \tag{III-22}$$

$$K_3 = \frac{[\mathrm{Dum\_Ion}][\mathrm{TypeIII\_TriIon}]}{[\mathrm{TypeIII\_DiIon}]} \tag{III-23}$$

By implementing Eq (III-21) through (III-23) into the charge balance equation, any weak electrolytes with the dissociation equilibrium like Eq (III-21) through Eq (III-23) should be handled by this model.

## Interfacial Concentration

Interfacial concentrations for each species are required to calculate the ionic fluxes or the exchanging rates. In this model, the interfacial concentrations of ions are determined by ion exchange equilibria (local equilibrium is assumed at solid-liquid interface). Specifically, the selectivity expression, which is based on the mass action law, is employed to calculate the interfacial concentrations.

Generally, for the exchange of species A with B, the selectivity is defined as

$$K_A^B = \left(\frac{q_B}{C_B^*}\right)^{Z_A} \left(\frac{C_A^*}{q_A}\right)^{Z_B} \tag{III-24}$$

The above equation can be rewritten in terms of equivalent fractions, total resin capacity and total interfacial concentration as

$$K_A^B = \left(\frac{Y_B}{X_B^*}\right)^{Z_A} \left(\frac{X_A^*}{Y_A}\right)^{Z_B} Q^{(Z_A - Z_B)} C_T^{*(Z_B - Z_A)} \tag{III-25}$$

Generalizing the above equation for any ion *i*, we have

$$X_i^* = Y_i \left(K_A^i\right)^{-1/Z_A} \left(\frac{X_A^*}{Y_A}\right)^{Z_i/Z_A} \left(\frac{Q}{C_T^*}\right)^{1-Z_i/Z_A} \tag{III-26}$$

In general, for *n* counterions in the bulk liquid replacing ion A in the resin, we can write *n* such expressions. However, given the resin loading, resin capacity and total interfacial concentration, we will have *n*+1 unknown interfacial fractions. The extra equation needed to completely specify the system is obtained from a material balance at the solid-film interface (summation relationship), which is

$$\sum_{i=1}^{n+1} X_i^* = 1.0 \tag{III-27}$$

From Eq (III-26), it is evident that for an arbitrary valence case, the ion exchange equilibrium depends on the resin capacity *Q* and total interfacial concentration $C_T^*$. Therefore iteration is required to determine the interfacial concentrations. However, before we can determine these individual interfacial concentrations, we need to have an expression for the total interfacial concentration, $C_T^*$. This is discussed in the following section. The detailed equations for calculation of interfacial concentrations are presented in Appendix B.

## Flux Expression

The flux expression of ions is developed based on the method proposed by Franzreb et al. (1993). The Nernst-Planck equation is the basis of the derivation,

$$J_i = -D_i \left[\frac{\partial C_i}{\partial r} + \frac{Z_i C_i F}{RT} \frac{\partial \phi}{\partial r}\right] \tag{III-28}$$

in which the first term on the right hand side of Equation (III-28) is the concentration gradient and the second term is the electric potential. The electric potential term in the Nernst-Planck equation is eliminated using the assumption of no net coion flux in the film. With the introduction of total equivalent concentration, $C_T$, the electric potential term can be written as

$$\frac{d\phi}{dr} = -\frac{RT}{Z_Y F}\frac{1}{C_T}\frac{dC_T}{dr} \qquad \text{(III-29)}$$

where $Z_Y$ is the mean coion valence, which is defined in Appendix C. Substituting of Eq (III-29) into Eq (III-28) and assuming a pseudo steady state exchange, we have

$$J_i = -D_i\left(\frac{dC_i}{dr} - \frac{C_i Z_i}{C_T Z_Y}\frac{dC_T}{dr}\right) \qquad \text{(III-30)}$$

After a series of mathematical manipulations (see Appendix C), the final form of the flux expression is as follows:

$$J_i = \frac{D_i}{\delta}\left((1-\frac{N_i}{P})(C_i^* - C_i^o) + N_i A_i (1+\frac{1}{P})(C_T^* - C_T^o)\right) \qquad \text{(III-31)}$$

where $$N_i = -\frac{Z_i}{Z_Y} \qquad \text{(III-32)}$$

$$P = \frac{\sum_{i=1}^{n} N_i D_i (X_i^* - X_i^o)}{\sum_{i=1}^{n} D_i (X_i^* - X_i^o)} \qquad \text{(III-33)}$$

$$C_T^* = \left(\frac{\sum_{i=1}^{n} (1+N_i) D_i X_i^o}{\sum_{i=1}^{n} (1+N_i) D_i X_i^*}\right)^{1/P+1} C_T^o \qquad \text{(III-34)}$$

From Equations (III-26) and (III-34) we know that the total interfacial concentration ($C_T^*$) and the individual interfacial equivalent fractions ($X_i^*$) are relevant. Thus, an iterative solution has to be used to determine these quantities and subsequently the ionic fluxes. The solution strategy adopted in the code to determine the ionic fluxes is presented in the following Table.

TABLE II. Solution Strategy for Calculation of Ionic Fluxes

1. Assume $C_T^* = C_T^o$.
2. Calculate $X_i^*$'s using Equations (III-26) and (III-27).
3. Calculate $C_T^*$ using Equation (III-34).
4. If the difference between new and old $C_T^*$ exceeds the chosen tolerance, repeat steps 2 and 3.
5. Calculate the ionic fluxes using Equation (III-31).

Equation (III-31) is used to determine the overall effective diffusivity defined as:

$$D_e = \frac{\sum_{i=1}^{n} | J_i \delta |}{\sum_{i=1}^{n} | C_i^* - C_i^o |} \qquad \text{(III-35)}$$

The film thickness in Equation (III-31) can be eliminated using the relation

$$\delta = D_e / K \qquad \text{(III-36)}$$

where K is a mass transfer coefficient calculated from correlation developed by Dwivedi and Upadhyay (1977), which is

$$\mathrm{Sh} = \mathrm{Sc}^{\frac{1}{3}}\,\mathrm{Re}\left[\frac{0.765}{(\varepsilon\,\mathrm{Re})^{0.82}} + \frac{0.365}{(\varepsilon\,\mathrm{Re})^{0.386}}\right] \tag{III-37}$$

or provided by the user from experiments.

## Particle Rates

The rate of exchange is related to the flux of the species ‘i’ by

$$\frac{\partial q_i}{\partial t} = -J_i a_s \tag{III-38}$$

where $J_i$ is the flux of species i. For weak electrolytes, $J_i$ is the summation of both ionic and nonionic forms, that is $J_i = J_{\mathrm{Ionic}} + J_{\mathrm{Mol}}$

However, the resin phase concentration $q_i$ can be represented as:

$$q_i = y_i Q \tag{III-39}$$

where $y_i$ is the fraction of resin phase concentration, Q is the resin capacity.

Substituting Eq (III-39) into Eq (III-38), yields

$$\frac{dy_i}{dt} = \frac{-J_i a_s}{Q} \tag{III-40}$$

Therefore, the rate of ions loading into resin can be determined by Eq (III-40), once the individual ionic fluxes are known.

## Column Material Balances

The material balance equation in an ion exchange column is the same as that in a fixed-bed adsorption column. If neglecting axial dispersion, the column material balance equation can be expressed as

$$\frac{u_s}{\varepsilon}\frac{\partial C_i}{\partial Z}+\frac{\partial C_i}{\partial t}+\frac{(1-\varepsilon)}{\varepsilon}\frac{\partial q_i}{\partial t}=0 \qquad \text{(III-41)}$$

In addition, this equation can be simplified into the dimensionless form for solving (the detailed derivation is shown in Appendix D).

For the ionic species, the dimensionless form is

$$\frac{\partial x_i}{\partial \xi}+\frac{\partial y_i}{\partial \tau}=0 \qquad \text{(III-42)}$$

where $x_i = C/C_T$, $y_i = q_i/Q$.

For the nonionic (molecular) forms of species, it is

$$\frac{\partial x_i^*}{\partial \xi}+\frac{C_T}{C_{Weak}^F}\frac{\partial y_i}{\partial \tau}=0 \qquad \text{(III-43)}$$

where $x_i^* = C/C_{Weak}^F$, $y_i = q_i/Q$.

Rate Expressions

Based on the material balance equation (III-41), the dimensionless form of the rate expressions can be derived (shown in Appendix D). The final form of the dimensionless rate equations is:

$$\frac{\partial y_i}{\partial \tau_c}=\left(-\frac{6J_i}{K_iC_f}\right)\frac{K_i}{K_c}\frac{d_{pa}}{d_{pc}}\frac{Q_a}{Q_c} \qquad \text{(Cations)} \qquad \text{(III-44)}$$

$$\frac{\partial y_i}{\partial \tau_c}=\left(-\frac{6J_i}{K_iC_f}\right)\frac{K_i}{K_c} \qquad \text{(Anions)} \qquad \text{(III-45)}$$

where $J_i$ is the flux. For the ionic forms, it is calculated by Franzreb's expression (III-31). For the nonionic forms of weak electrolytes, it is calculated by Fick's law (III-2).

The effluent concentrations from the column are determined by solving the above set of equations. The method of characteristics is employed to solve this system of equations. The resin and bulk phase fraction equations are then solved using the first order Euler's method and fourth order Gear's method (Gear, 1971).

## Temperature Dependent Parameters

There are four model parameters that are temperature dependent: selectivity, diffusivity, viscosity, and dissociation (or ionization) constant. Because of lack of experimental data, there are no correlations available for selectivities. However, limited correlations for diffusivities and dissociation constants for some species are available. For instance, Divekar et al. (1987) developed the correlations of diffusivities and dissociation constants for $Na^+$, $Cl^-$, $H^+$, and $OH^-$, and incorporated them into the model of Haub and Foutch (1986a,b) to account for temperature effects. Their study indicated that temperature has appreciable effects on column performance..

Zecchini (1990), Bulusu (1994) and Sunkavilli (1996) further developed some correlations for the species relevant to their studies; such as $Ca^{2+}$, $SO_4^{2-}$, $HCO^{3-}$, $CO_3^{2-}$, $NH_3$ and Morpholine. However, to develop a generalized model, more correlations for other ions are required. In this work, quite a few temperature relations of dissociation constants and diffusivities for other ions have been developed. Table III is the summary of all the correlations used in this model.

## TABLE III

### Correlations for Temperature Dependent Parameters

Ionic Diffusion Coefficients (cm$^2$/s)

| | |
|---|---|
| Hydrogen[1] | $D_H = 8.931*10^{-10}(T+273.16)(221.71+5.52T-0.0144T^2)$ |
| Hydroxide[1] | $D_{OH} = 8.931*10^{-10}(T+273.16)(104.74+3.807T)$ |
| Sodium[1] | $D_{Na} = 8.931*10^{-10}(T+273.16)(23.005+1.0642T+0.003319T^2)$ |
| Potassium[5] | $D_K = 8.931*10^{-10}(T+273.16)(40.22+1.278T+0.00271T^2)$ |
| Silver[5] | $D_{Ag} = 8.931*10^{-10}(T+273.16)(33.12+1.065T+0.003538T^2)$ |
| Calcium[3] | $D_{Ca} = 8.931*10^{-10}(T+273.16)(23.27+1.575T)/2$ |
| Barium[5] | $D_{Ba} = 8.931*10^{-10}(T+273.16)(34.0+1.042T+0.00568T^2)$ |
| Magnesium[3] | $D_{Mg} = 8.931*10^{-10}(T+273.16)(28.81+0.819T+0.00542T^2)/2$ |
| Ammonia[2] | $D_{NH_4} = 8.931*10^{-10}(T+273.16)(39.1537+1.4055T)$ |
| Chloride[1] | $D_{Cl} = 8.931*10^{-10}(T+273.16)(39.649+1.392T+0.00332T^2)/2$ |
| Sulfate[3] | $D_{SO_4} = 8.931*10^{-10}(T+273.16)(35.76+2.079T)/2$ |
| Bicarbonate[3] | $D_{HCO_3} = 8.931*10^{-10}(T+273.16)\times 44.5$ |
| Carbonate[3] | $D_{CO_3} = 8.931*10^{-10}(T+273.16)(36.0+1.44T)/2$ |
| Nitrate[5] | $D_{NO_3} = 8.931*10^{-10}(T+273.16)(40.03+1.161T+0.003882T^2)$ |
| Fluoride[5] | $D_F = 8.931*10^{-10}(T+273.16)(107.7-6.61T+0.1804T^2)$ |
| Bromide[5] | $D_{Br} = 8.931*10^{-10}(T+273.16)(42.52+1.326T+0.004104T^2)$ |
| Perchloride[5] | $D_{ClO_4} = 8.931*10^{-10}(T+273.16)(36.96+1.136T+0.003443T^2)$ |

Acetate[5] $D_{CH_3COO} = 8.931 * 10^{-10}(T + 273.16)(201 + 0.8169T + 0.000603T^2)$

where T is temperature in °C.

## Dissociation Constants

Water[1] $pK_w = -6.0875 + 0.0176T + \dfrac{4470.99}{T}$

Ammonia[2] $pK_{NH_3} = 4.8601 + 6.31 \times 10^{-5} T - 5.98 \times 10^{-3} / T^2$

Morpholine[2] $pK_{Mor} = 5.7461 + 8.095 \times 10^{-5} T - 1.3881 \times 10^{-2} / T^2$

Carbonates

1st. dissociation[3] $pK_1 = \dfrac{17052}{T} + 215.21 Log(T) - 0.12675T - 545.56$

2nd. Dissociation[3] $pK_2 = \dfrac{2902.39}{T} + 0.02379T - 6.498$

Phosphate:

1st dissociation[5] $pK_1 = \dfrac{799.31}{T} - 4.5535 + 0.013486T$

2nd dissociation[5] $pK_2 = \dfrac{2073.0}{T} - 5.9887 + 0.020912T$

3rd dissociation[5] $pK_3 = 12.1152 - 0.00908T + 0.000127T^2$

where *T* is temperature in °K.

## Solution Properties

Viscosity[1] (cp) $\mu = 1.5471 - 0.0317109T + 2.3345 \times 10^{-4} T^2$

Density[4] (g/cc)

$$\rho = \left[ \frac{1 + 0.134248 * SS2 - 3946263 * 10^{-3} * SS1}{3.1975 - 0.3151548 * SS2 - 1.203374 * 10^{-3} * SS1 + 7.48908 * 10^{-13} * SS1^4} \right]^{1/2}$$

where $SS1 = 374.11 - T$

$SS2 = SS1^{1/3}$

T is temperature in °C in the above correlations

---

Note* 1 – Divekar (1987); 2 – Zecchini (1990); 3 – Bulusu (1994); 4 – Sunkavalli (1996)
5 – This work

However, for the ionic diffusivities of ions whose experimental data are not available, the following generalized correlation is used to estimate the diffusivities for these species at temperatures other than 25 °C (Reid et al., 1987).

$$D_T = D_{298} \frac{T}{334\eta_w} \qquad \text{(III-46)}$$

where $D_{298}$ = ionic diffusivity at 25 °C (298 K)

$T$ = temperature, K

$\eta_w$ = Viscosity of water at T °K, CP

It should be noted that this correlation has an average error of 17%, and maximum error of 40%.

## Desulfonation of strongly acidic cation resin

In some power plants, the desulfonation of cationic resin has a great effect on the effluent concentration of sulfate. So the effect of desulfonation on the column performance has to be considered in the model, and the model should give users the option whether they want to incorporate the desulfonation effect or not. Fisher and Burke (1993) have performed experiments to quantify the decomposition of cationic resin, and the following rate constant was generated to describe the desulfation rate

$$K = (7.5 \times 10^6)\text{EXP}(-10278.6/(T + 273.16)) \qquad \text{(III-47)}$$

It is easy to see that the desulfonation rate is a function of temperature. Based on Fisher and Burke's (1993) data, and data from other researchers, Palino (Pondugula, 1994) developed the following expression to estimate the amount of sulfate released from the cationic resin

$$\mathrm{DS} = (7.5\times 10^{6}\,\mathrm{e}^{(-10278.6/(\mathrm{T}+273.16))}\times \mathrm{HT}\times \pi \mathrm{D}^{2}\times \mathrm{Q}_{\mathrm{c}})(\mathrm{V}_{\mathrm{s}}\times \mathrm{d}_{\mathrm{pa}})$$

$$\times \mathrm{FCR}/(\mathrm{NT}\times 3600\times \mathrm{FR}\times \mathrm{k}_{\mathrm{Lc}}\times (1-\varepsilon)) \qquad \text{(III-48)}$$

In this model, Eq (III-48) is employed to account for the desulfation effects.

### Model Algorithm

Based on the model equations, an algorithm was developed to predict the effluent concentrations of a MBIE column for multicomponent systems. The detailed schematic diagram is shown in Appendix E.

## Simulation Results and Discussion

Using the model developed above, several practical cases from industry were simulated. Those are: 1) multicomponent system with amines (ANO case); 2) multicomponent system with carbonates (PP&L case); and 3) multicomponent system with phosphates (Assumed). For each case, the effects of model parameters such as temperature, resin fouling, variable inlet concentration and cation/anion resin ratio on column performance were also investigated.

## Case I. Multicomponent System with Amines (ANO case)

In Arkansas Nuclear One (ANO), ethanolamine (ETA) and ammonia are used to adjust water pH to minimize erosion and corrosion of the service water to the equipment. The input concentrations of condensate polishers are typically 700 parts-per-billion (ppb) of ETA, and 300 ppb of ammonia. Rohm and Haas Amberjet 4400 ($OH^-$ form) is the strong base resin. The cationic resin is Amberjet 1500 ($H^+$ form). Regeneration is performed when polisher effluent reaches 0.05 ppb (or 50 ppt) sodium. The bed dimensions are 9 ft in diameter and 4 ft in depth with void fraction of 0.35. The flow rate is 2220 gpm per bed. The operating temperature is 52 °C, and the estimated influent pH is 8.5. For this practical case, the following simulations were performed to study the effects of various model parameters on the column performance: (1) normal operating conditions; (2) fouled cation and anion resin effects; (3) temperature effects; and (4) resin ratio effects. The input data and model parameters are summarized in Table IV.

### (1) *Normal Operating Condition (Base Case)*

Figure 1 presents the simulation results for the base case. From this plot, we see that $Na^+$ will reach 50 ppt (breakthrough concentration) in about 16.5 days. This prediction is very close to the plant experience of 16 days for Rohm and Haas Amberjet 1500 resins. This result indicates that our model works very well in predicting the MBIE with amines. Figure 1 also shows the expected breakthrough curves for ammonia and ETA, from which we know that ammonia and ETA started breakthrough in about 11 days and 18 days, respectively. These two results are also in good agreement with plant observations.

Table IV. Input Data and Model Parameters for ANO Case

| Item | Value |
|---|---|
| Bed diameter (cm) | 274.0 |
| Bed depth (cm) | 121.0 |
| Bed void fraction | 0.35 |
| Resin bead diameter (cm) | |
| Cation (Amberjet 4400) | 0.068 |
| Anion (Amberjet 1500) | 0.058 |
| Resin capacity (meq/ml) | |
| Cation | 2.1 |
| Anion | 1.0 |
| Cation/Anion ratio | 0.611 : 0.389 |
| Influent pH | 8.5 |
| Temperature (°C) | 52 |
| Influent concentration (ppb) | |
| Sodium ($Na^+$) | 0.1 |
| Ammonia ($NH_3$) | 300 |
| ETA | 700 |
| Sulfate ($SO_4^{-2}$) | 0.6 |
| Chloride ($Cl^-$) | 0.2 |
| $CO_2$ (total) | 4.26 |
| Initial loadings on the resin (%) | |
| Sodium ($Na^+$) | 0.3 |
| Ammonia ($NH_3$) | 0.01 |
| ETA | 0.01 |
| Sulfate ($SO_4^{-2}$) | 4.0 |
| Chloride ($Cl^-$) | 0.5 |
| Bicarbonate ($HCO_3^-$) | 0.1 |
| Carbonate ($CO_3^{-2}$) | 0.1 |
| Influent flow rate ($cm^3/s$) | 1.4E+5 (220gpm) |

| Physical properties | Diffusivity | Selectivity |
|---|---|---|
| Species: | ($cm^2/s$) | |
| Sodium ($Na^+$) | 1.334E-5 | 1.5 |
| Ammonia ($NH_3$) | 1.957E-5 | 1.95 |
| ETA | 1.124E-5 | 2.04 |
| Sulfate ($SO_4^{-2}$) | 1.065E-5 | 20 |
| Chloride ($Cl^-$) | 2.032E-5 | 22 |
| Bicarbonate ($HCO_3^-$) | 1.185E-5 | 6 |
| Carbonate ($CO_3^{-2}$) | 0.923E-5 | 12 |

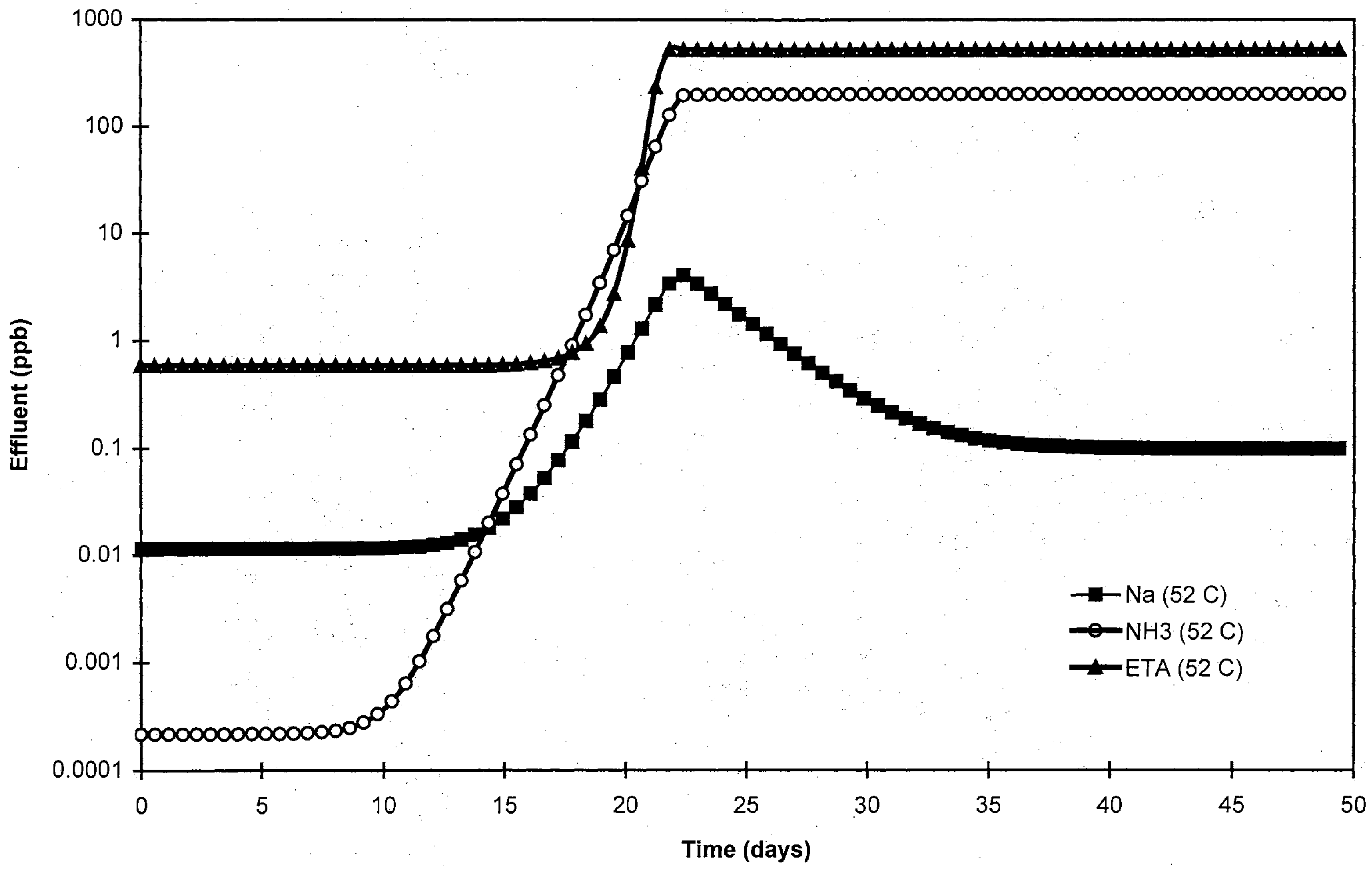


Figure 1. Cation breakthrough curves at normal operational conditions

### (2) *Degraded Resin (Resin Fouling) Effect*

Figure 2 and 3 present the breakthrough curves for the case of both cationic and anionic resins fouled. The simulations were performed by reducing both cationic and anionic resin mass transfer coefficients (MTC's) to 70% and 50% of the initial, unfouled values. With reduced MTC's, we observed that all species broke through earlier. For sodium, when MTC is reduced by half, the expected run time until 50 ppt is reduced from 16.5 to 13.7 days, a reduction of 17%.

Figure 4 is another way to present the effect of MTC on sodium breakthrough. Ten different simulations were performed to generate this plot. Each run used identical operating conditions with only the percentage of resin fouling adjusted. Of particular note in this chart is that even when the MTC is 20% that of unfouled resin, the expected run time until 50 ppt $Na^+$ is 9.2 days; a reduction of about 44% run-time. Based on this prediction, we suggest resins with such a low MTC should be removed from service due to the inability to handle a major tube leak effectively, since reduction in run time has a direct economic impact on the cost associated with an increased number of regenerations. However, the reduction in run-time is not directly proportional to the decrease in MTC due to the equilibrium characteristics of ion exchange. If MTC continues to drop toward zero, the days to reach 50 ppt will also approach zero since no other mechanism for the removal of sodium is included in the model. We believe that repeating the simulations at higher flow rates will result in a steeper curve, while operating at lower flow rates will give a more horizontal curve.

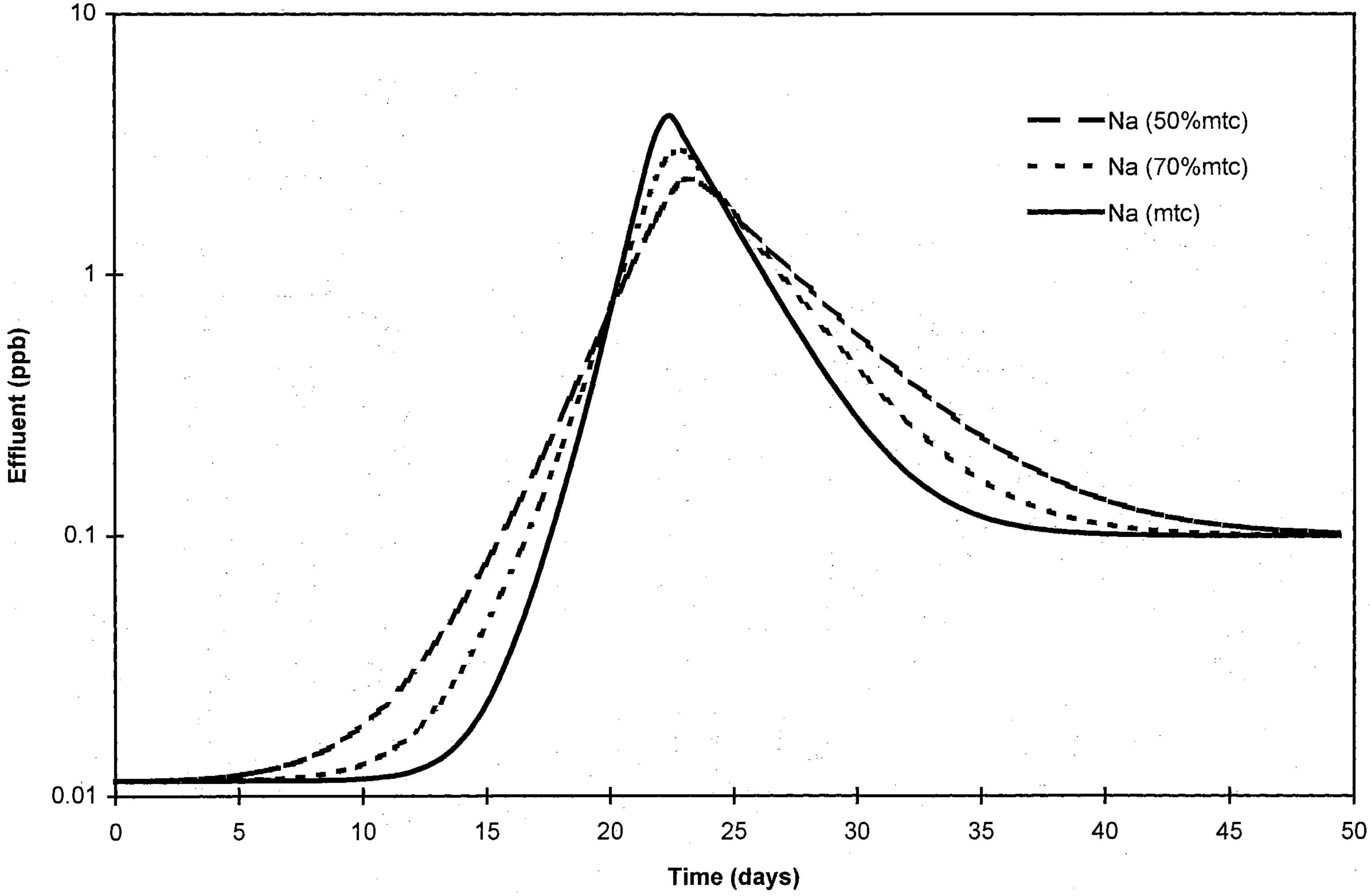


Figure 2. Effect of reduced cation MTC on $Na^+$ breakthrough curves

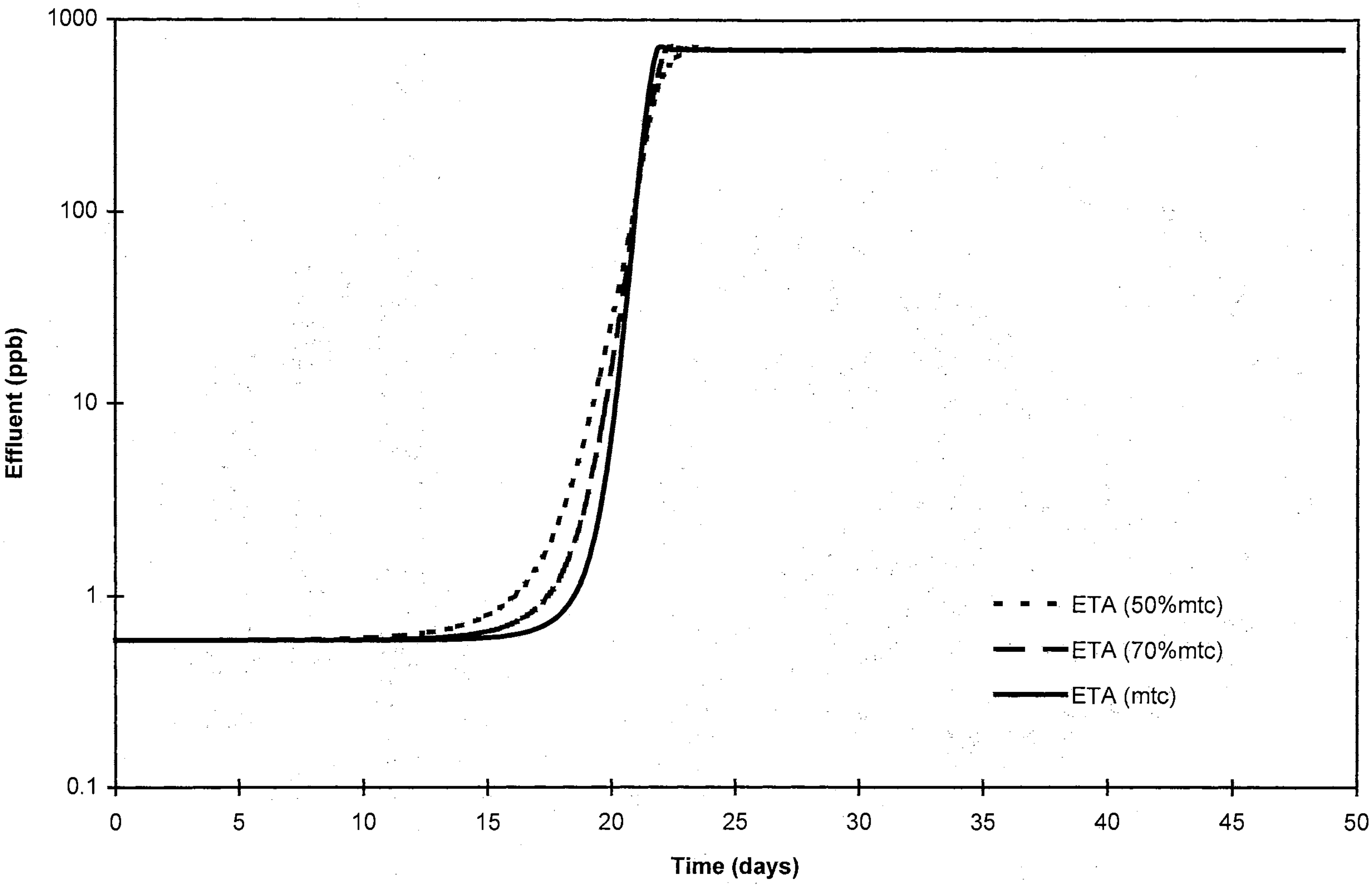


Figure 3. The effect of reduced cation MTC on ETA breakthrough curves

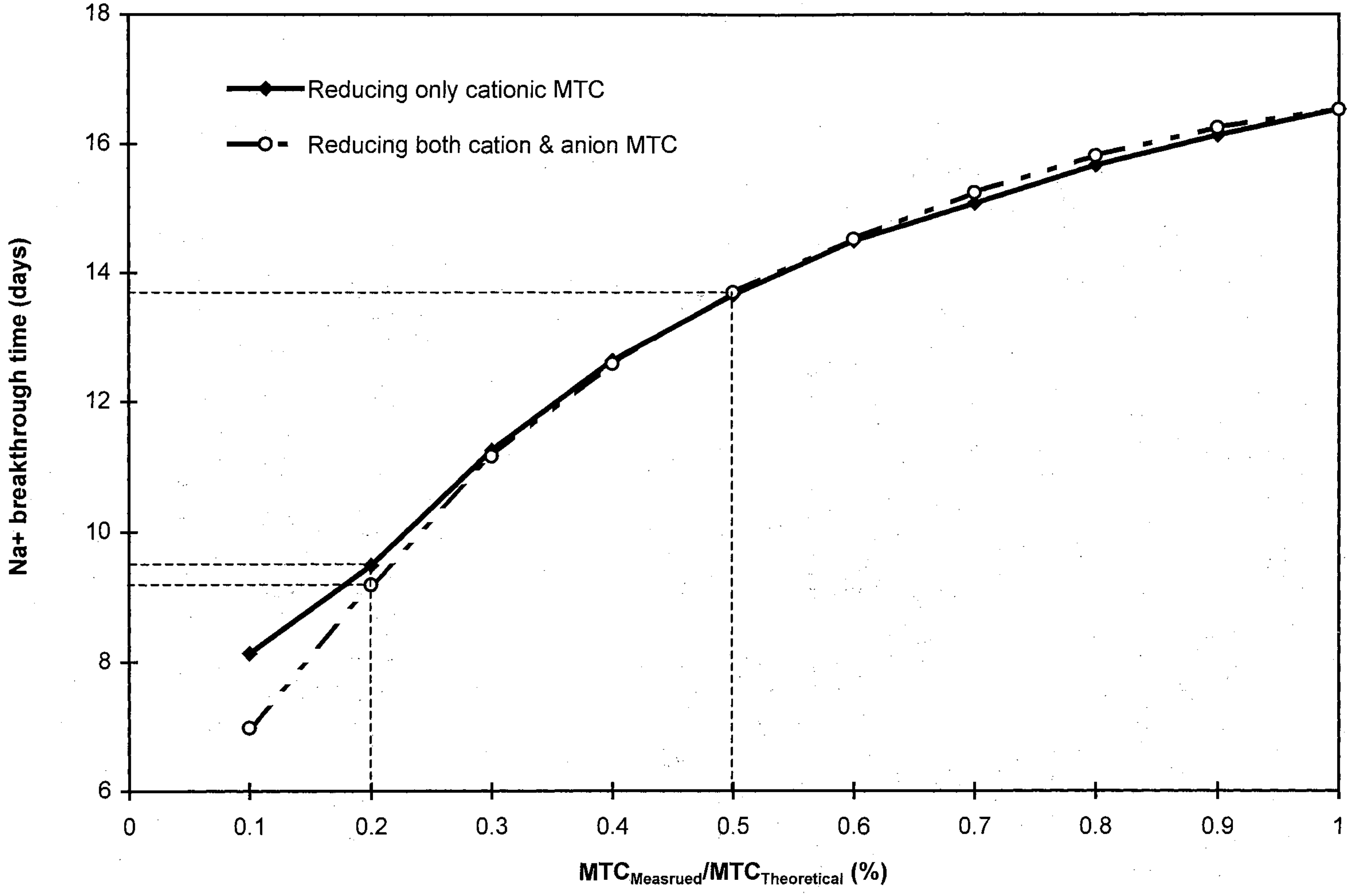


**Figure 4. Effect of reduced MTC on the $Na^+$ breakthrough time**

### (3) *Temperature Effect*

Figure 5 is the breakthrough comparison for cations at 52 °C and 25 °C. When the temperature increases, initial leakage will be higher and the ions will break through at different times. Higher temperature seems to increase the time to 50 ppt, which is consistent with the conclusion of Zecchini (1990). However, temperature has a complex influence on breakthrough time, which is difficult to quantify (Foutch, 1991). The predictions presented here are plant specific, and should not be extended to other plants directly.

### (4) *Resin Ratio Effect*

Simulations are also performed to optimize the ion exchange resin ratio. For this case, we evaluated the C/A ratio required to minimize the total sodium and chloride concentration in the polisher effluent while achieving the 50 ppt $Na^+$ requirement. This is presented as a minimum value in a plot of the total concentration of ($[Na^+]+[Cl^-]$) divided by the breakthrough time for $Na^+$ to 50 ppt versus the cation to anion (C/A) resin ratio. Currently, the column is operated at C/A $\approx 1.6$, corresponding to a breakthrough time of 16.5 days for sodium, at which time the chloride concentration is 2.2 ppt. Figure 6 indicates that this operating ratio is near the optimum C/A for this water chemistry.

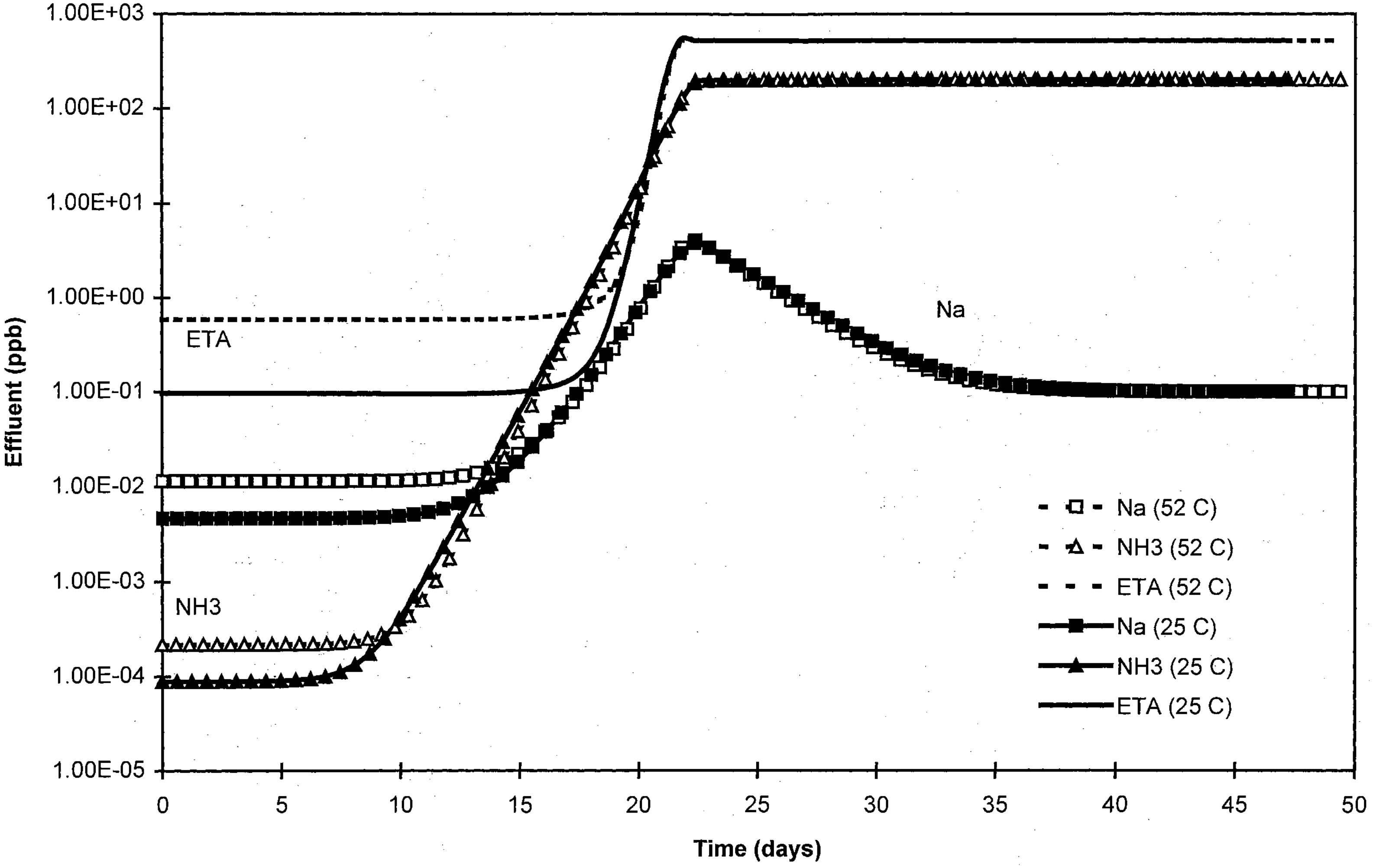
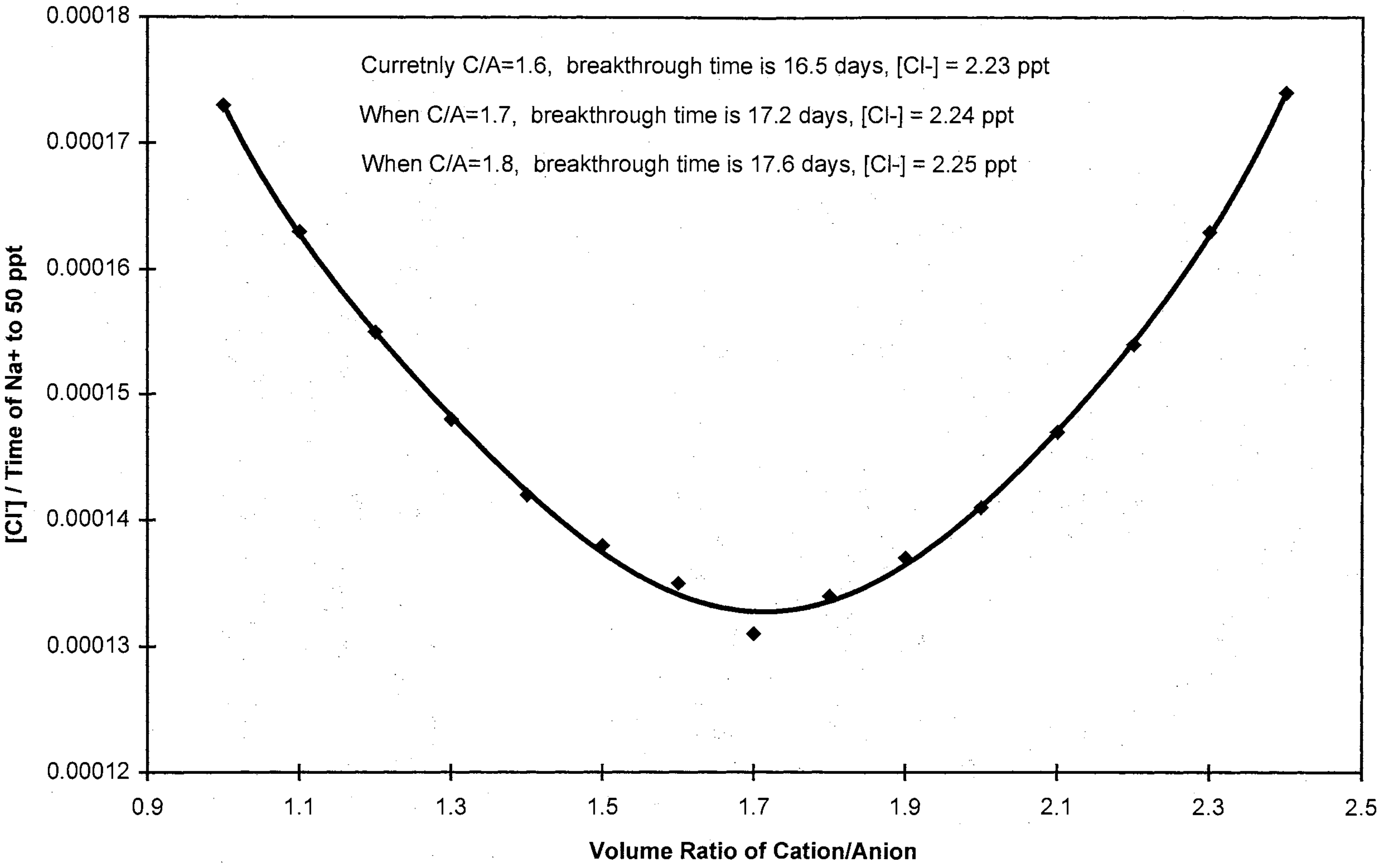


Figure 5. Temperature effect on cation breakthrough curves



Curretnly C/A=1.6, breakthrough time is 16.5 days, [Cl-] = 2.23 ppt

When C/A=1.7, breakthrough time is 17.2 days, [Cl-] = 2.24 ppt

When C/A=1.8, breakthrough time is 17.6 days, [Cl-] = 2.25 ppt

**Figure 6. Effects of cation anion ratio on column performance**

## Case II. Multicomponent System with Carbonates (PP&L Case)

At Susquehanna Steam Electric Station of Pennsylvania Power and Light (PP&L), the condensate polishing system contains seven spherical vessels in parallel, each containing about 280 cubic feet of mixed cation and anion resin. With six beds normally in service, the water flux is about 50 gpm/ft$^2$. The bed depth is about 3 feet. The resin ratio is specified to provide equal capacities of hydrogen-form cation resin and hydroxide-form anion resin. The polishers operate at high pressure (up to 600 psia). Condensate temperature averages ~105°F in winter and ~125°F in summer, and peaks can reach 135°F. Because of the iron oxides from corrosion and erosion in the steam cycle which accumulate on the beds during operation, and increase bed flow resistance and system pressure drop, the resins in each vessel are removed from service about once a month for ultrasonic resin cleaning (URC).

In addition, chemistry specifications for BWR reactor water are stringent. Corrosive impurities such as chloride and sulfate are limited to $\leq$5 ppb, with typical values in the range of <1 to 3 ppb. Other impurities, such as sodium, nitrate and chromate, are typically well under 20 ppb. At Susquehanna, reactor water is continuously processed through a filter-demineralizer system at about 300 gpm. This limits the buildup of soluble, non-volatile impurities in reactor water to 100 times their concentration in feedwater. Therefore, for normal operation, condensate polisher effluent is expected to contain well under 0.05 ppb (50 ppt) of corrosive impurities. During challenges, such as condenser leakage, the polishers must limit corrosive impurities to less than 1 ppb to avoid a plant shutdown.

Among anionic impurities, besides $Cl^-$ and $SO_4^{-2}$, dissociative species like carbonates also exist in the system, which makes the water treated by the mixed-bed demineralizer (condensate polisher) is a complex multicomponent system. Therefore, whether this model can handle such cases is a good test of the model's capability. In this work, the following scenarios for condensate polishers at Susquehanna were simulated using the developed MBIE model: 1) well mixed bed (normal operating conditions); 2) degraded anion resin (simulated by reduced mass transfer coefficients); 3) temperature effect; 4) a 6 inch underlay of either cationic or anionic resin; and 5) effect of periodic resin cleaning on bed performance and operating life. The input data and model parameters are summarized in Table V through VII, all of which are plant supplied.

1) *Normal Operating Conditions - Well Mixed-Bed (Base Case)*

Figure 7 and 8 showed the predicted anion and cation concentrations in the polisher effluent for a well mixed-bed exposed to "Service" conditions at 25°C. At the start of the run, the model predicts a neutral effluent containing sodium bicarbonate with ~10 ppt sodium. Sodium is the lead impurity to break, reaching 200 ppt in ~660 days and causing the effluent to become slightly alkaline. Bicarbonate is predicted to break through next, and to reach 100 ppt in the effluent after ~1290 days. However, bicarbonate is expected to break down to $CO_2$ in the reactor and to pass over to the steam, so it will not impact reactor water chemistry. Chloride and sulfate do not reach significant concentrations over the simulation time of 1600 days. The predicted sulfate concentration is not from equilibrium leakage, but from temperature-dependent desulfonation of the cation resin that is built into the model. The predictions also show that calcium and magnesium will not reach significant concentrations until ~1200 days, which agree well with the plant facts.

TABLE V. Influent Concentrations and Initial Loadings for Normal Service

| Ions | Influent Concentration | Initial Loading (%) |
|---|---|---|
| | Service (ppb) | |
| $Na^+$ | 0.39 | 0.5 |
| $Ca^{2+}$ | 0.97 | 0.1 |
| $Mg^{2+}$ | 0.24 | 0.1 |
| $SO_4^{2-}$ | 1.14 | 0.05 |
| $Cl^-$ | 0.066 | 0.25 |
| $HCO_3^-$ | 2.63 | 2.0 |
| $CO_3^{2-}$ | - | 0.1 |
| pH (25°C) | 7.0 | |
| Cond. (μS/cm) | 0.065 | |

TABLE VI. Resin Properties and Selectivities

| | Anionic Resin Dowex 550A | Cationic Resin Dowex 650C |
|---|---|---|
| Functionality | q-ammonia | $RSO_3^-$ |
| Crosslinkage | type 1 | 10% |
| Capacity (meq/ml) | 1.1 | 2 |
| Bead Diameter (cm) | 0.055 | 0.0625 |

TABLE VII. Selectivities and Diffusivities for Species

| Ions | Selectivities | Diffusivities ($cm^2/s \times 10^5$) |
|---|---|---|
| $Ca^{+2}$ | 4.4 | 0.792 |
| $Na^+$ | 1.6 | 1.334 |
| $K^+$ | 2.6 | 1.957 |
| | | |
| $Cl^-$ | 22 | 2.032 |
| $SO_4^{2-}$ | 20 | 1.902 |
| $HCO_3^-$ | 6 | 1.185 |
| $CO_3^{2-}$ | 12 | 0.923 |

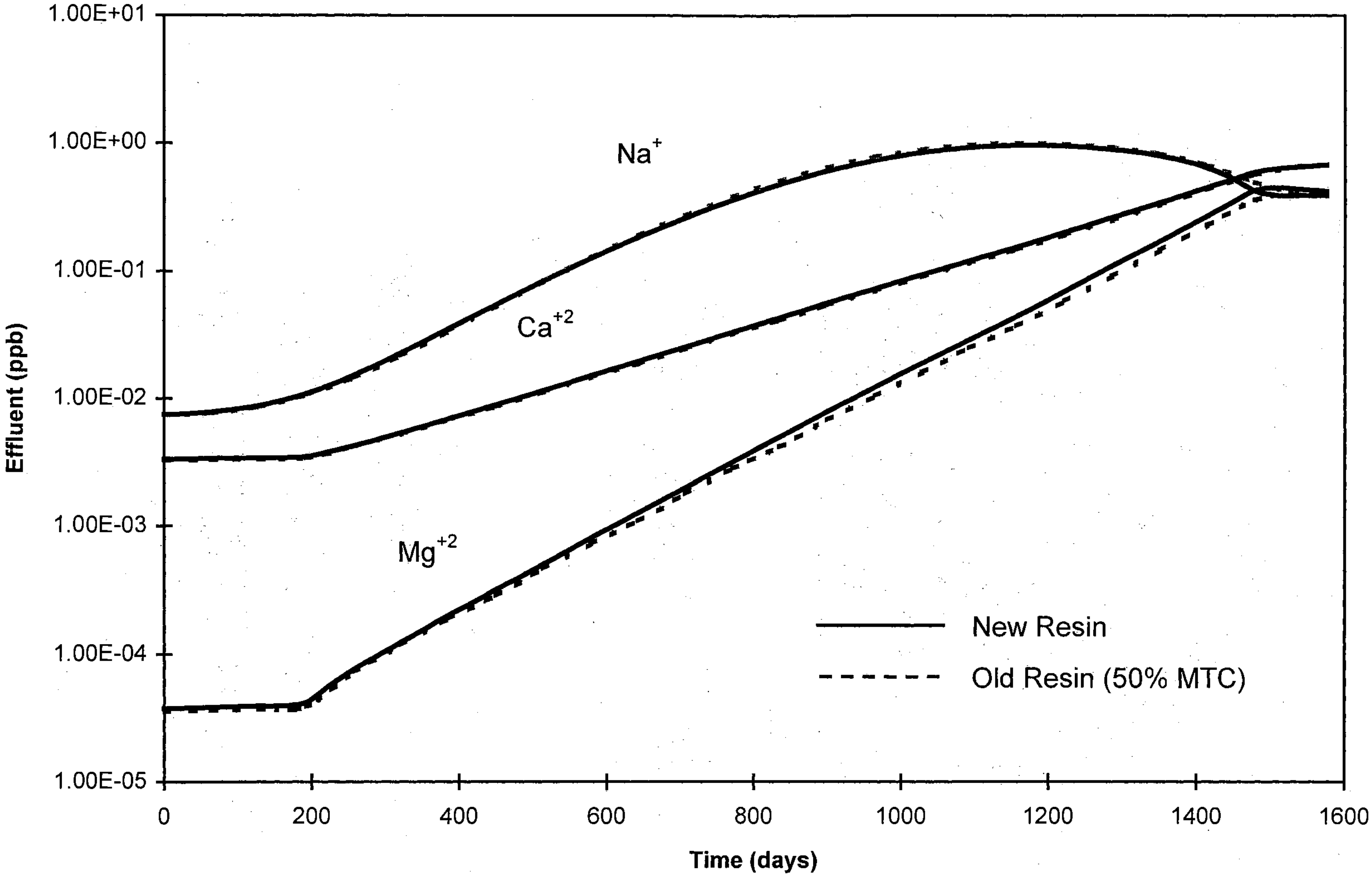


Figure 7. Cation effluent profile for new and 50% fouled anionic resin at 25 °C

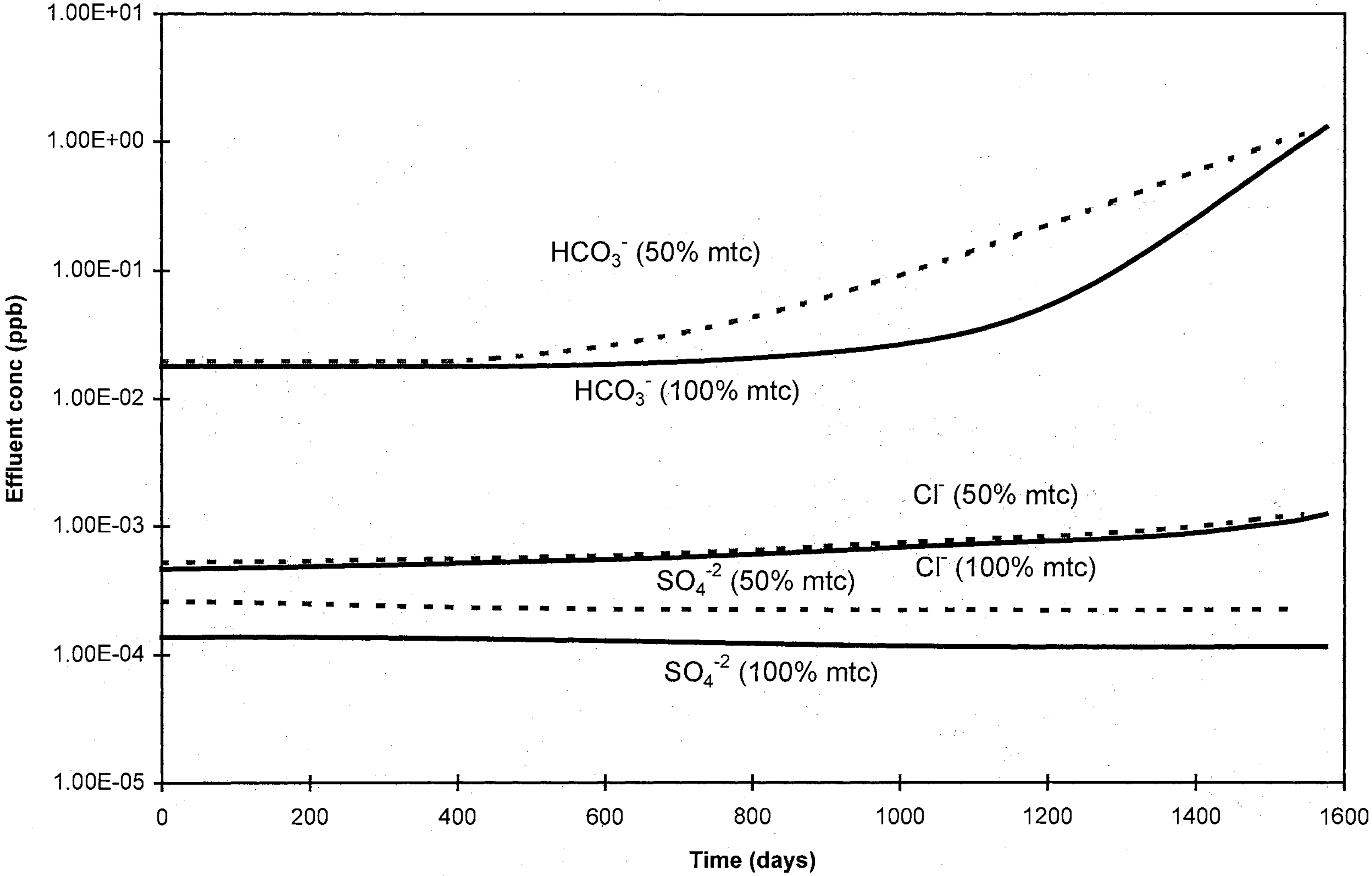


Figure 8. Anion effluent profile for new and 50% fouled anionic resin at 25 °C

### 2) *Temperature Effect*

Figures 9 and 10 show the effect of increasing operating temperature to 55°C (131°F) on the column performance. Effluent concentrations are observed increased for all species, and run lengths are shortened. For sodium, a run length of ~330 days to 200 ppt in the effluent is predicted. For bicarbonate, the predicted breakthrough time (time to reach 100 ppt ) is ~1125 days. The increase in the effluent appears to be primarily due to the increased ionization of water at higher temperature, which provides higher concentration of hydronium and hydroxide to compete with impurity ions for sites on the resin. However, as the ion selectivities were not adjusted for temperature because of the lack of data, these results only indicate the trend for temperature effects, but may not accurately predict the magnitude. Temperature effects in this study are limited to ionic diffusivities, water dissociation constant and water viscosity.

### 3) *Fouled Resin Effect (Degraded Resin)*

In industry, resin fouling (resin degradation) is also often encountered. The direct effect of resin fouling is a measurable reduction of the overall mass transfer coefficient, which leads to a higher equilibrium leakage and earlier breakthrough time. The effects of anionic resin fouling were simulated by comparing service cycle operation with new resin and anionic resin modeled by reducing the anion mass transfer coefficient by 50% (no changes were made in the cationic resin mass transfer coefficient) in this work. It is showed that there is only minor difference in the effluent cationic when anionic resin is fouled, and pH differences are also small.

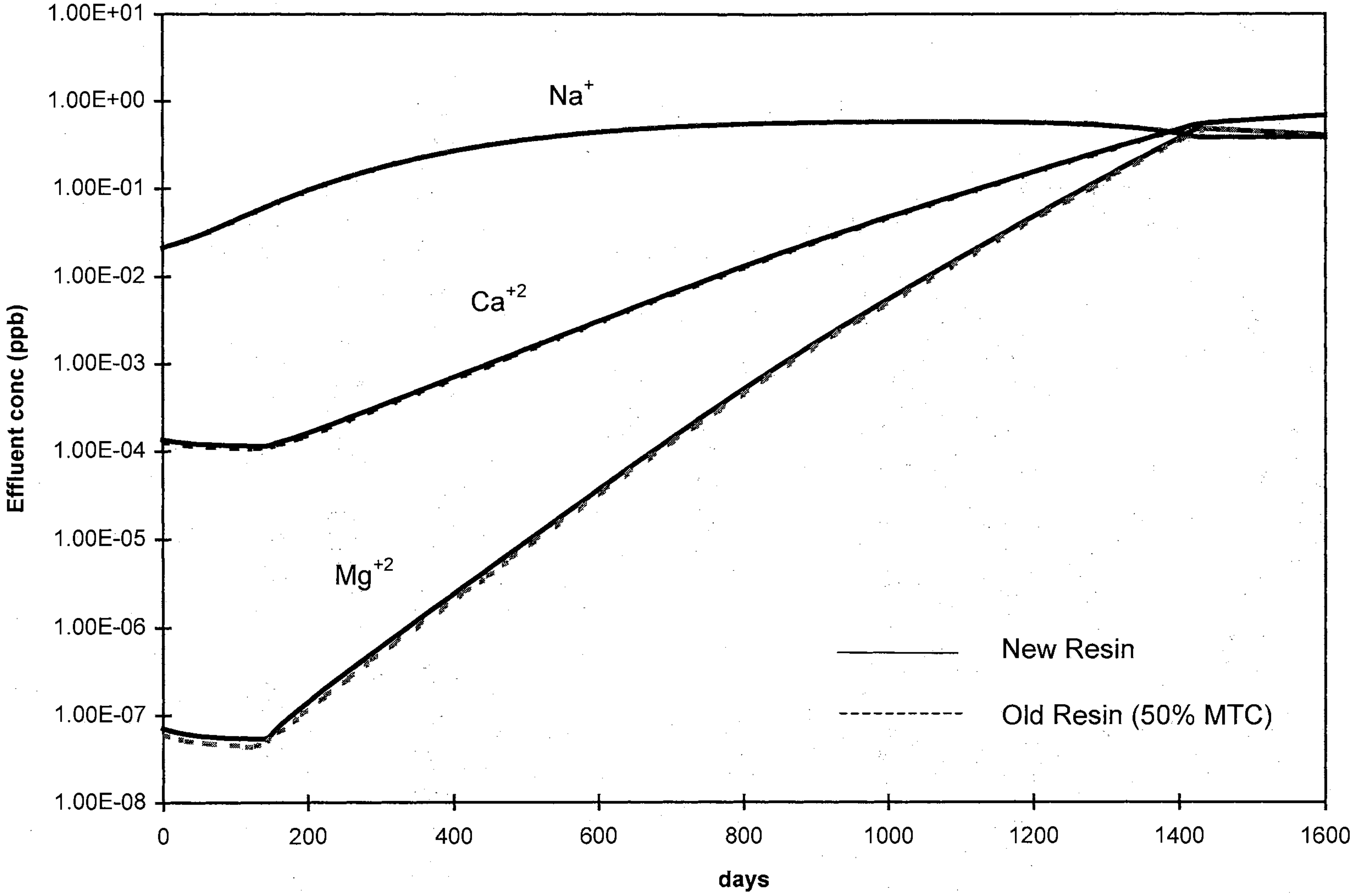


Figure 9. Cation effluent profile for new and 50% fouled anionic resin at 55 °C

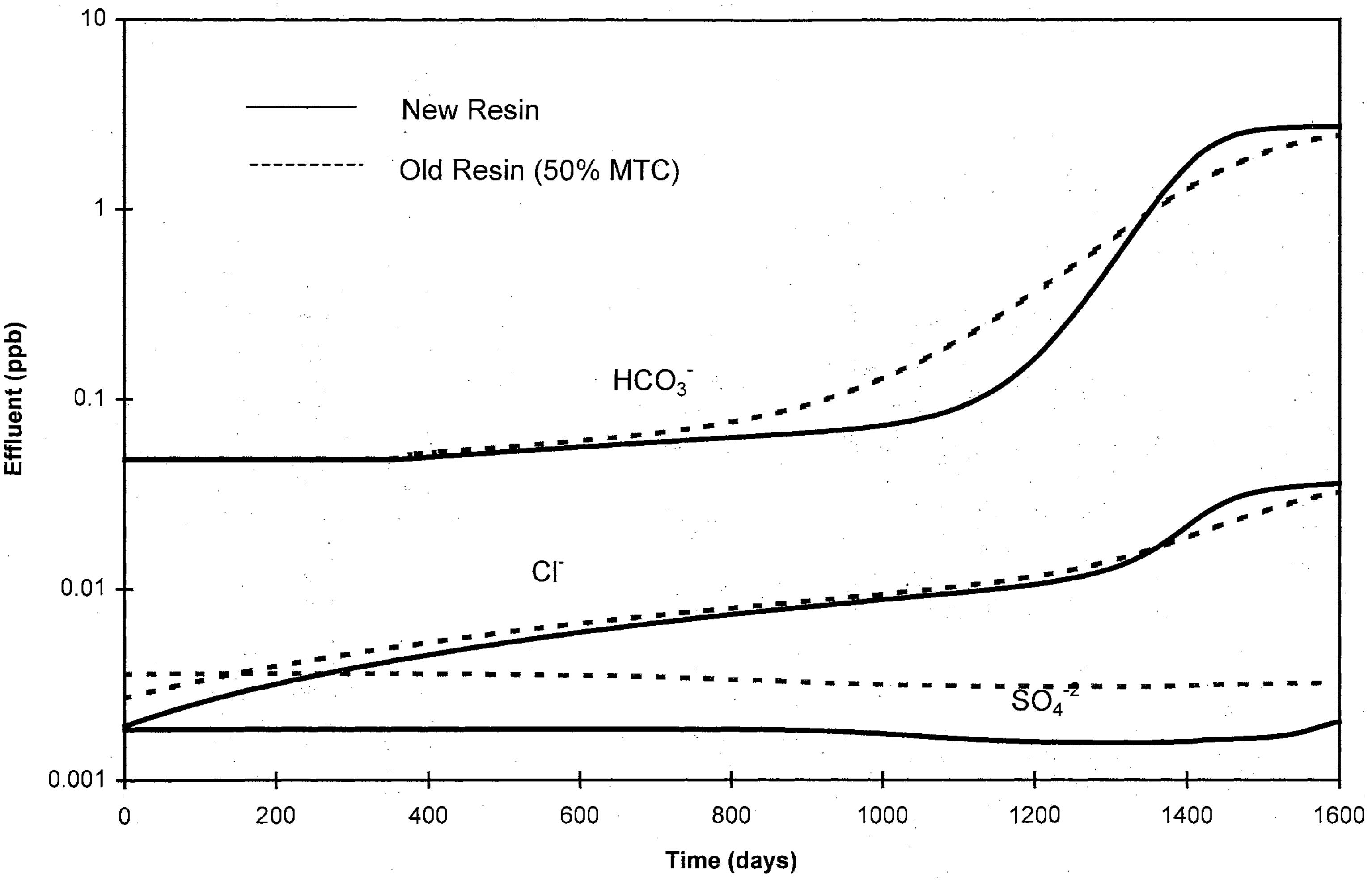


Figure 10. Anion effluent profile for new and 50% fouled anionic resin at 55 °C

Figure 7 also presents the anion breakthrough curves for both new and fouled anion resin at 25 °C. Typically, reducing the mass transfer coefficient results in an early, gradual breakthrough without the loss of exchange capacity, as seen with the $HCO_3^-$ curves. However, the primary initial effect of resin fouling is the initial sulfate leakage increase from 0.14 ppt to 0.26 ppt at 25 °C and 1.85 to 3.63 ppt at 55 °C; approximately doubling the sulfate leakage expected for new resin. A summary of the initial sulfate leakage is presented in Table VIII. Bicarbonate follows the expected breakthrough curves, with the time to 0.1 ppb effluent concentration decreasing from 1290 days to 1020 days at 25 °C; and from 1125 days to 928 days at 55 °C, when resin is fouled. These predictions are important, which tell us resin fouling may have serious effects on the column performance and column life.

TABLE VIII. Initial Sulfate Leakage at Different Simulation Conditions

| | $SO_4^{-2}$ (ppt) | | |
|---|---|---|---|
| Temperature | 25 °C | 40 °C | 55 °C |
| New Resin | 0.14 | 0.53 | 1.85 |
| Fouled Resin | 0.26 | 1.02 | 3.63 |
| Cationic Heels | 0.63 | 4.5 | 18.5 |
| Anionic Heels | 0.11 | 0.41 | 1.22 |

4) *Resin Heel Effect*

Cation resin has a higher density and greater terminal settling velocity than anion resin. Therefore, segregation of cation resin to the bottom (outlet) of the vessel can occur if the system design for resin mixing and sluicing is poor or if these processes are not conducted properly. A cation resin heel can result in significantly higher levels of anionic impurities (especially sulfate) in the effluent, which is more serious at elevated temperatures (see Figure 12). If good mixing cannot be achieved, an anion resin underlay has to be used to improve bed performance by intercepting the releases of sulfate from the cation resin.

The effects of anionic and cationic resin heels at 55 °C are predicted in this work. Figures 11 and 12 are the simulation results. The observed effects are limited primarily to the sulfate leakage from the bed due to the cationic resin generation of sulfate. Anionic resin heels reduce sulfate from the 1.85 ppt of base case to 1.22 ppt; a reduction of 34%. The impact of cationic heels is much more significant. Sulfate leakage increases from the 1.85 ppt base case to 18.5 ppt; about ten times the amount of sulfate leaching from the bed. However, there is little effect of either cationic or anionic heels on the cationic effluent concentrations predicted.

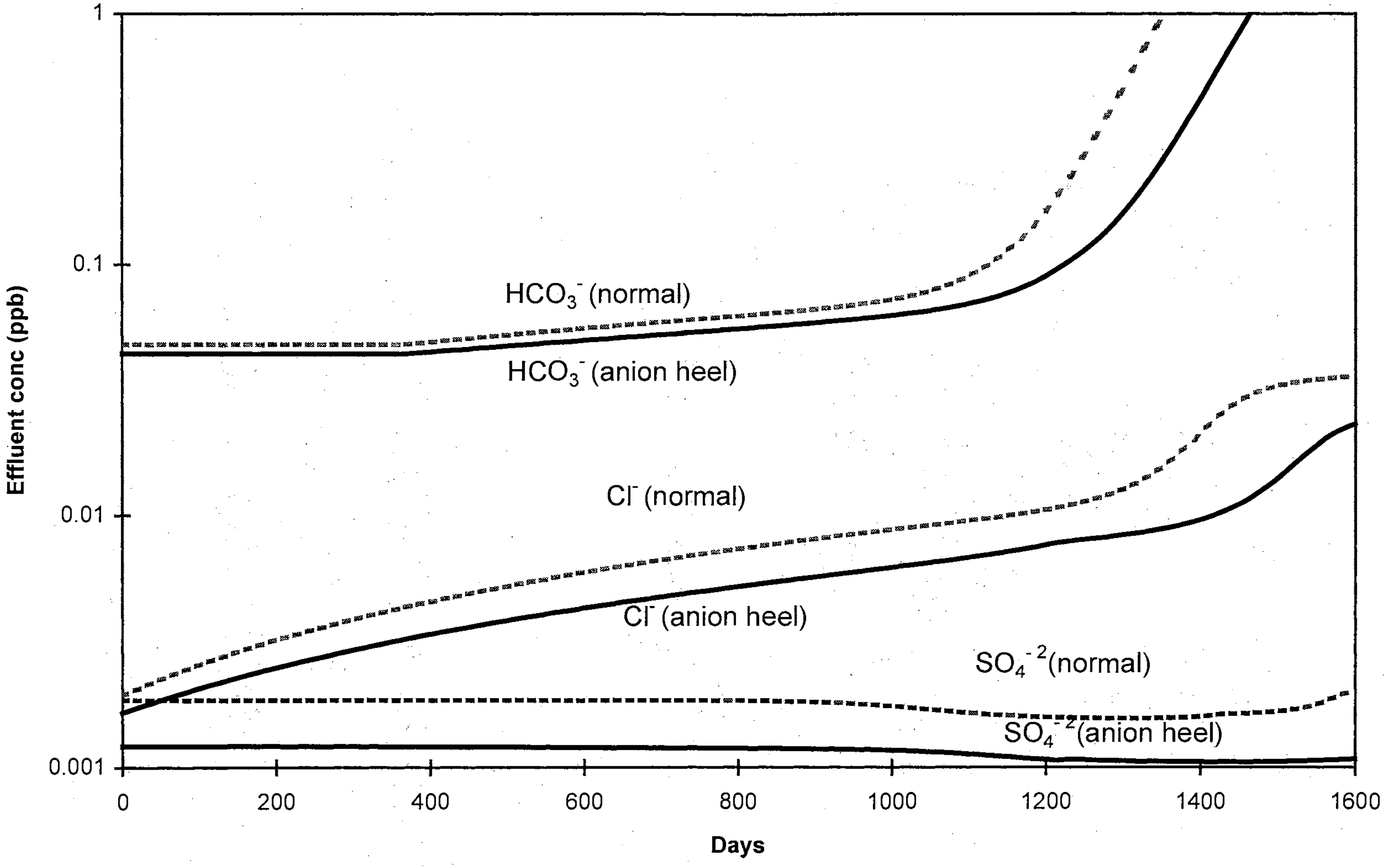


Figure 11. Anion effluent with and without 6" anionc resin heels at 55 °C

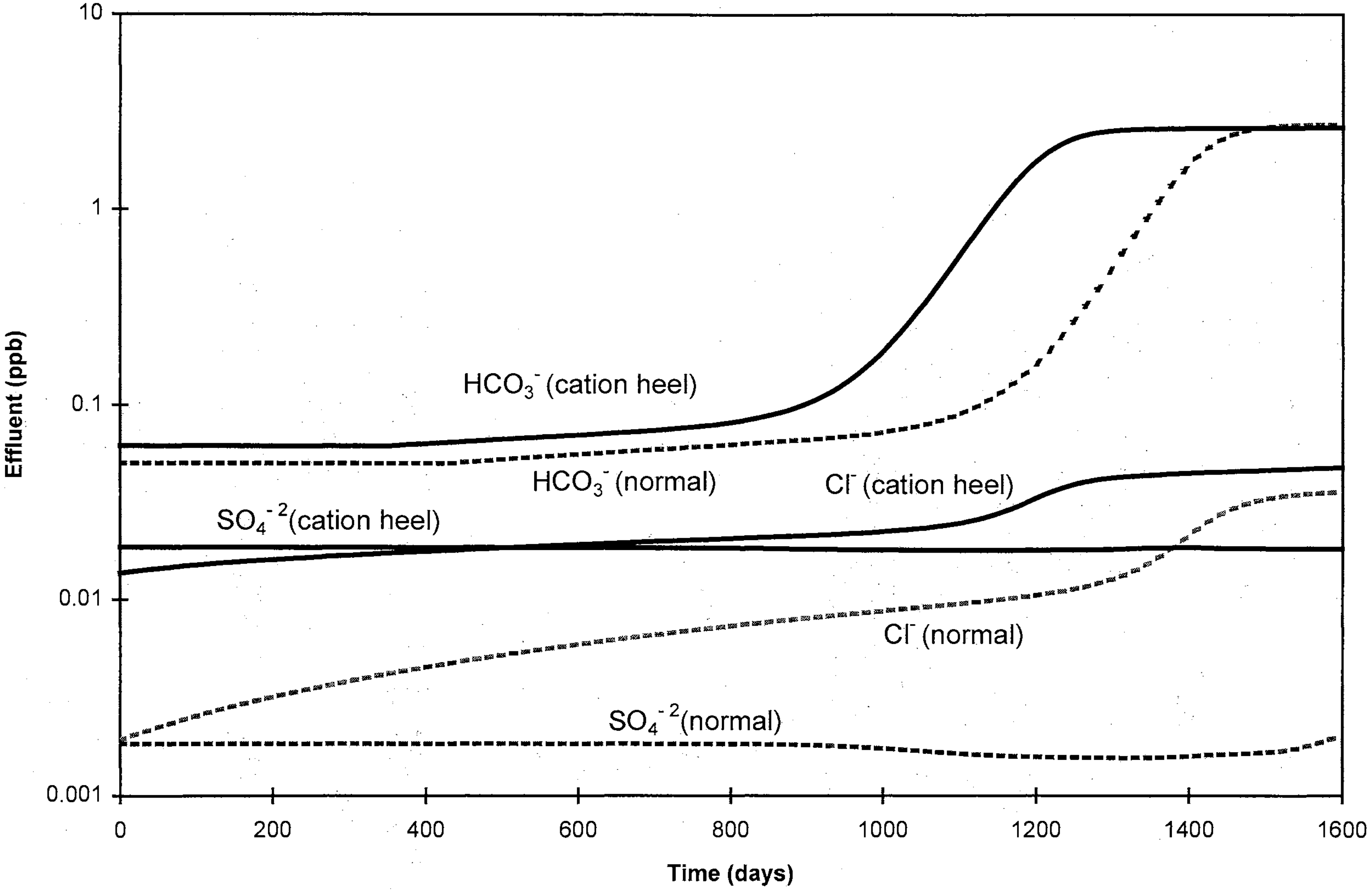


**Figure 12. Anion effluent with and without 6 " cationic resin heels at 55 °C**

5) *Bed Cleaning Effect*

Figures 13 presents the expected effluent concentration profiles when beds are removed from service at 30-day intervals for ultrasonic resin cleaning at 55 °C. The ladder effect observed in these figures is due to remixing the loading profile every 30 days of operation to give a uniform loading profile. Obviously, the most significant impact is the large reduction of time to achieve target breakthrough concentrations. Specifically, the time for sodium effluent concentration to reach 200 ppt is reduced from approximately 330 to 195 days at 55°C. The bicarbonate breakthrough (to reach 100 ppt) drops from 1125 to 42 days at 55°C. Little change is noted for sulfate, since the sulfate generation term from cationic resin is essentially constant and sulfate from the resin is significantly greater than kinetic or equilibrium leakage from the exchange sites.

The above predictions are generally consistent with Susquehanna's experience. Resins that have operated for up to fifteen months (450 days) with some exposure to condenser leakage show increased levels of sodium and chloride leakage. Currently at Susquehanna Unit 1, sodium levels in reactor water are in the range of 10 to 15 ppb (implying 100 to 150 ppt in the polisher effluent), and chloride levels are ~1.5 ppb. Reactor water pH has shifted upward slightly. With new condensate polisher resins, sodium is 2 ~ 5 ppb and chloride <0.5 ppb in reactor water.

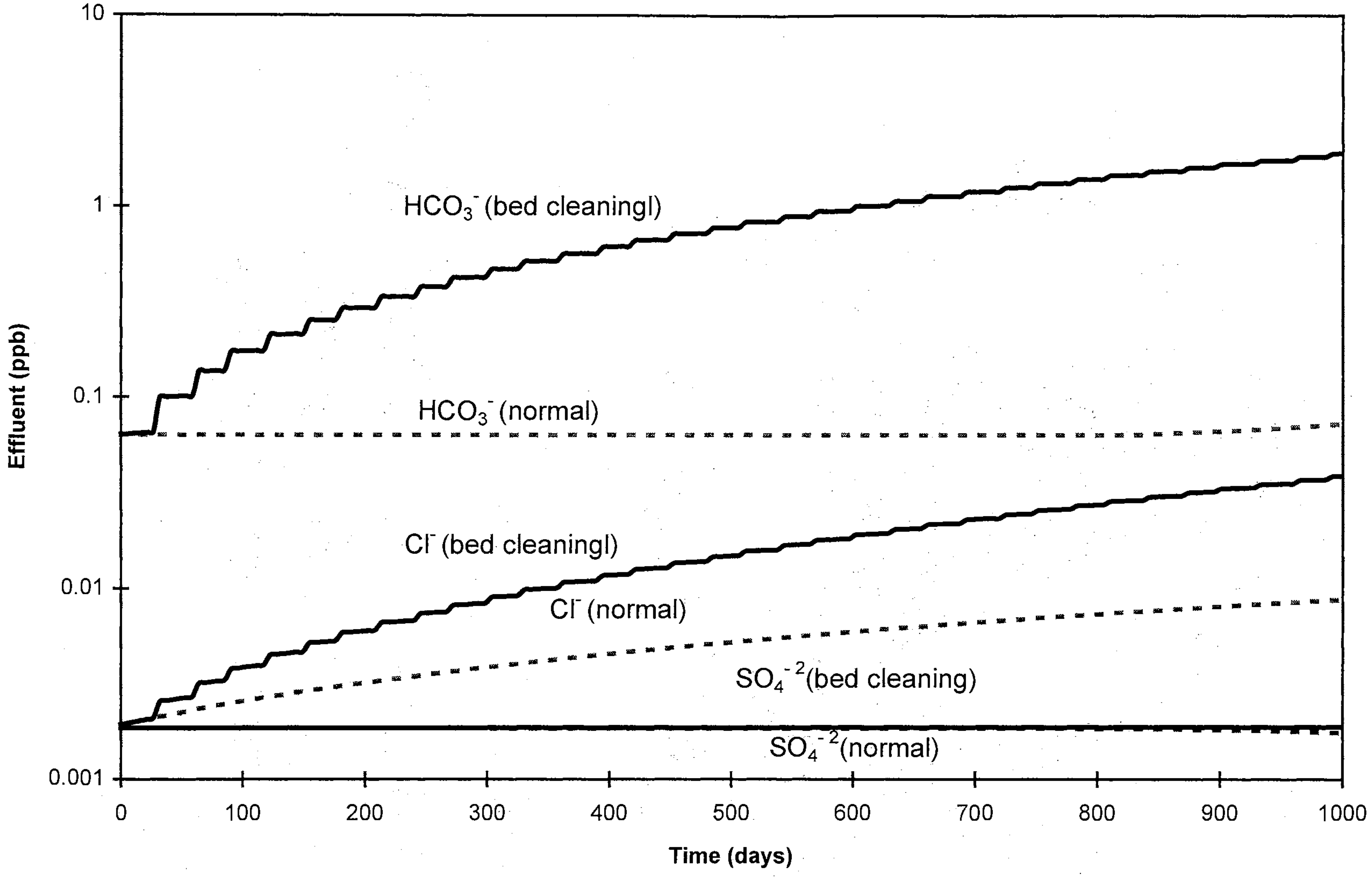


Figure 13. Anion effluent with and without bed cleaning every 30 days at 55 °C

### Case III. Multicomponent System with Phosphates (Assumed Case)

Phosphoric acid ($H_3PO_4$) is a triprotic acid which is importance to water and wastewater chemistry. In some power plants, the water to be treated by mixed-bed ion exchange columns sometimes contain phosphate species too, therefore extending the MBIE model to handle trivalent dissociative species like phosphates is also required.

The dissociation of phosphate in water is a complex three-stage process. Several different phosphate ions may exist in solution besides the molecular phosphate acid; and the relative amount of each kind of phosphate depends on solution pH. Figure 14 shows how solution pH influences the relative amount of each kind of phosphate species. This chart was generated based on the dissociation equilibrium of phosphates.

The purpose of this simulation is to show that the model can handle mixed-bed ion exchange with trivalent dissociative species like phosphates properly. However, because of lack of real plant data, no practical cases are simulated. In this study, a multicomponent system with phosphates was assumed and simulated. Table IX lists all input data and model parameters used for simulations. The following are the simulation results.

Table IX. Input Data and Model Parameters for the Case with Phosphates

| Item | Value |
|---|---|
| Bed diameter (cm) | 274.0 |
| Bed depth (cm) | 121.0 |
| Bed void fraction | 0.35 |
| Resin bead diameter (cm) | |
| Cation | 0.068 |
| Anion | 0.058 |
| Resin capacity (meq/ml) | |
| Cation | 2.0 |
| Anion | 1.0 |
| Cation/Anion ratio | 0.333 : 0.667 |
| Influent pH | 7.5 |
| Temperature (°C) | 25 |
| Influent concentration (ppb) | |
| Calcium ($Ca^{+2}$) | 1.6 |
| Sodium ($Na^{+}$) | 3.45 |
| Ammonia ($NH_3$) | 0.85 |
| Sulfate ($SO_4^{-2}$) | 3.36 |
| Chloride ($Cl^{-}$) | 8.65 |
| Phosphates (total) | 3.92 |
| Initial loadings on the resin (%) | |
| Calcium ($Ca^{+2}$) | 0.03 |
| Sodium ($Na^{+}$) | 0.01 |
| Ammonia ($NH_3$) | 0.01 |
| Sulfate ($SO_4^{-2}$) | 4.0 |
| Chloride ($Cl^{-}$) | 0.5 |
| ($H_2PO_4^{-}$) | 0.1 |
| ($HPO_4^{-2}$) | 0.1 |
| ($PO_4^{-3}$) | 0.1 |
| Influent flow rate ($cm^3/s$) | 1.4E+5 (220gpm) |

| Physical properties | Diffusivity ($cm^2/s$) | Selectivity |
|---|---|---|
| Species: | | |
| Calcium ($Ca^{+2}$) | 0.792E-5 | 4.4 |
| Sodium ($Na^{+}$) | 1.334E-5 | 1.5 |
| Ammonia ($NH_3$) | 1.957E-5 | 1.95 |
| Sulfate ($SO_4^{-2}$) | 1.065E-5 | 20 |
| Chloride ($Cl^{-}$) | 2.032E-5 | 22 |
| ($H_2PO_4^{-}$) | 0.879E-5 | 2.26 |
| ($HPO_4^{-2}$) | 0.439E-5 | 4.0 |
| ($PO_4^{-3}$) | 0.612E-5 | 5.25 |

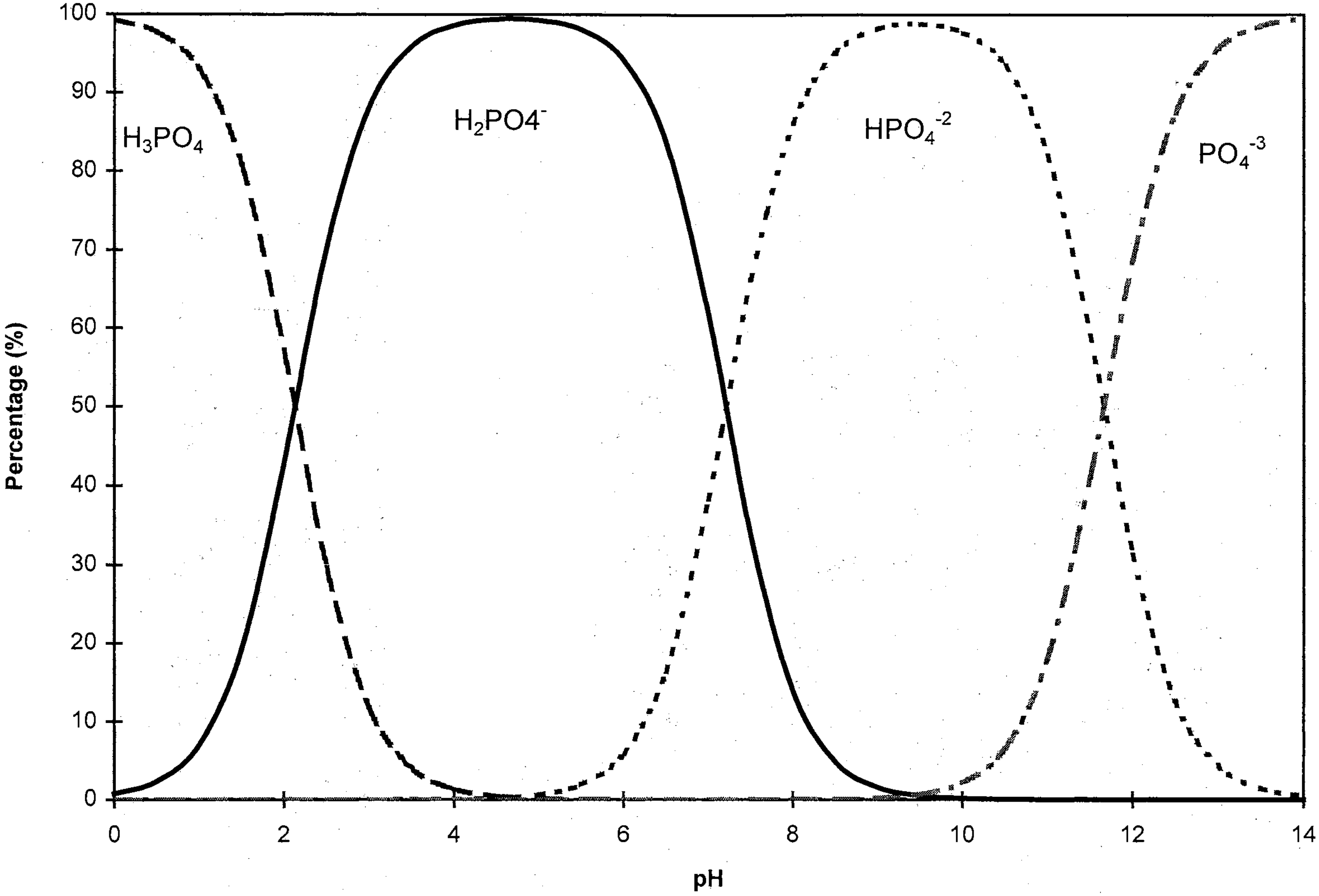


Figure 14. Distribution of various phosphate species with pH at 25 °C

As shown in Figure 15, $Na^+$ starts breakthrough in about 150 days; and $NH_4^+$ in about 300 days. But $Ca^{2+}$ will not break through until 800 days. These predictions are generally consistent with the concept that divalent species has the priority over the monovalent species in exchanging with resin; that is, monovalent ions will break through first. Also from Figure 15, ionic throws were observed for monovalent ions in their breakthrough curves.

Figure 16 is the breakthrough curves for phosphate species, from which we see that among various phosphate species, monovalent phosphate $H_2PO_4^-$ breakthrough first, because it has a lower selectivity. The divalent $HPO_4^{-2}$ is next, and then $PO_4^{-3}$. These results are also consistent with what we expected.

Figure 17 gives the breakthrough curves for all anions. This simulation was performed by considering the effect of desulfonation of cationic resin. From this chart, we can see that the effluent concentration of sulfate ($SO_4^{-2}$) exhibits no changes over the whole simulation time of 1200 days. This is because the sulfate is generated from the resin itself, (desulfonation term built in the model), not from the kinetic leakage of the influent.

Figure 18 is the corresponding pH profile for the case considering desulfonation, from which, we observe that the solution pH goes basic first because cations break through first (see Figure 15) in this case. Also, over the entire simulation period, the solution pH is in the range of 6.5 ~ 7.5, therefore, most phosphates should be in the forms of $H_2PO_4^-$ and $HPO_4^{-2}$ (see Figure 14), and only small amount of phosphates is $PO_4^{-3}$. Figure 16 and 17 confirmed this prediction.

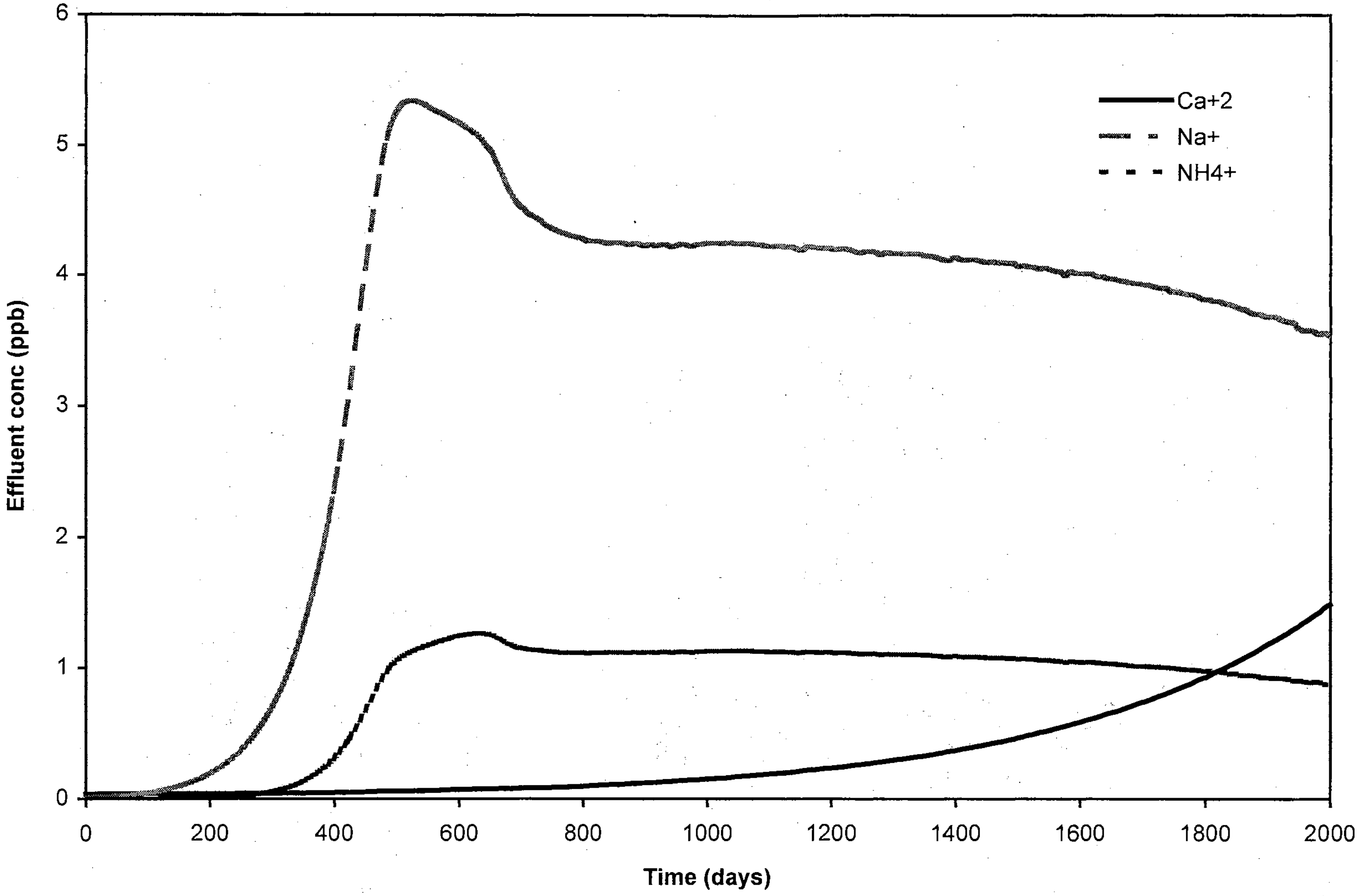


Figure 15. Cation breakthrough curves for the case with phosphates

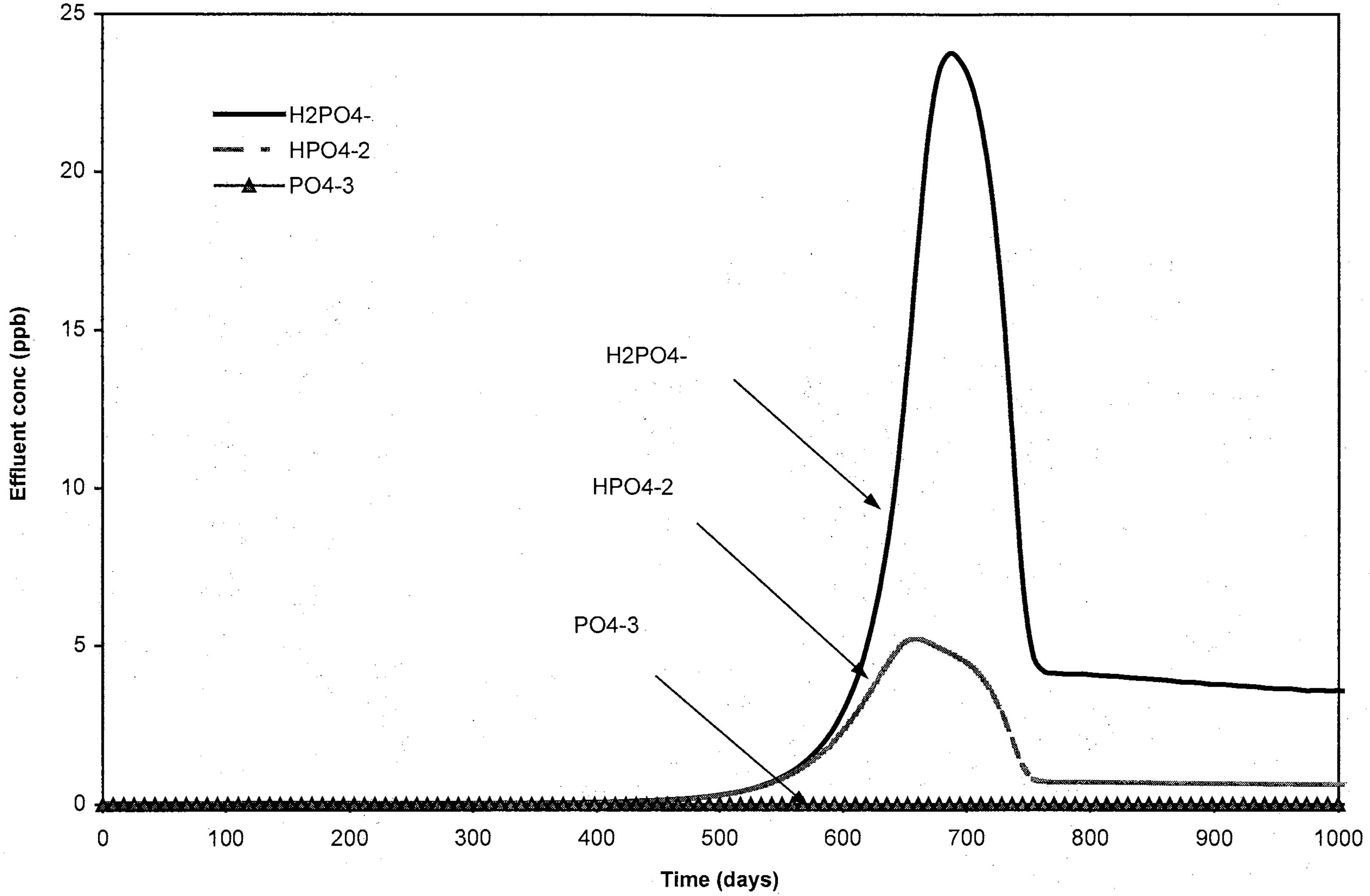


Figure 16. Breakthrough curves for various phosphate species

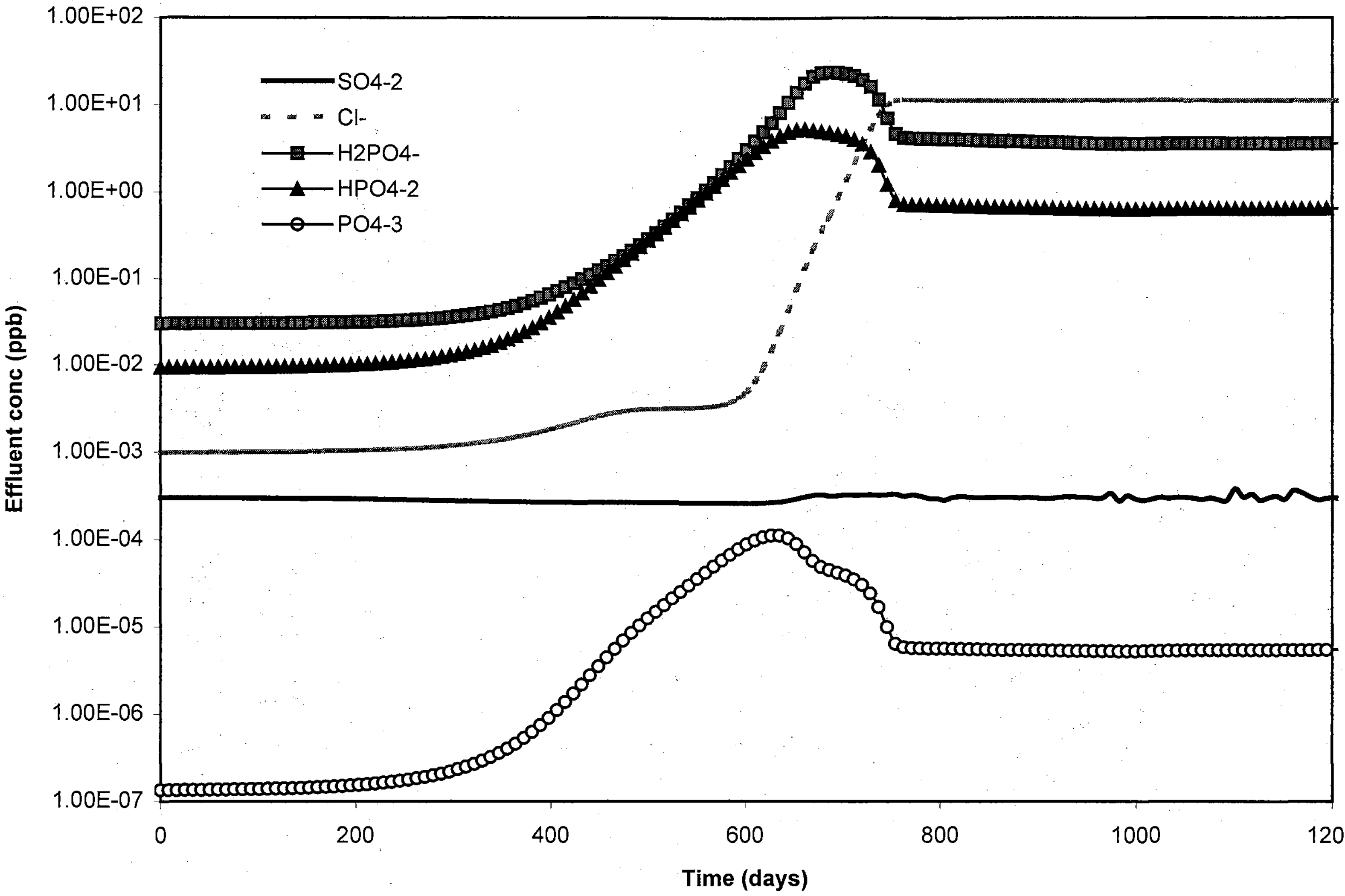


Figure 17. Anion breakthrough curves when considering desulfonation effect

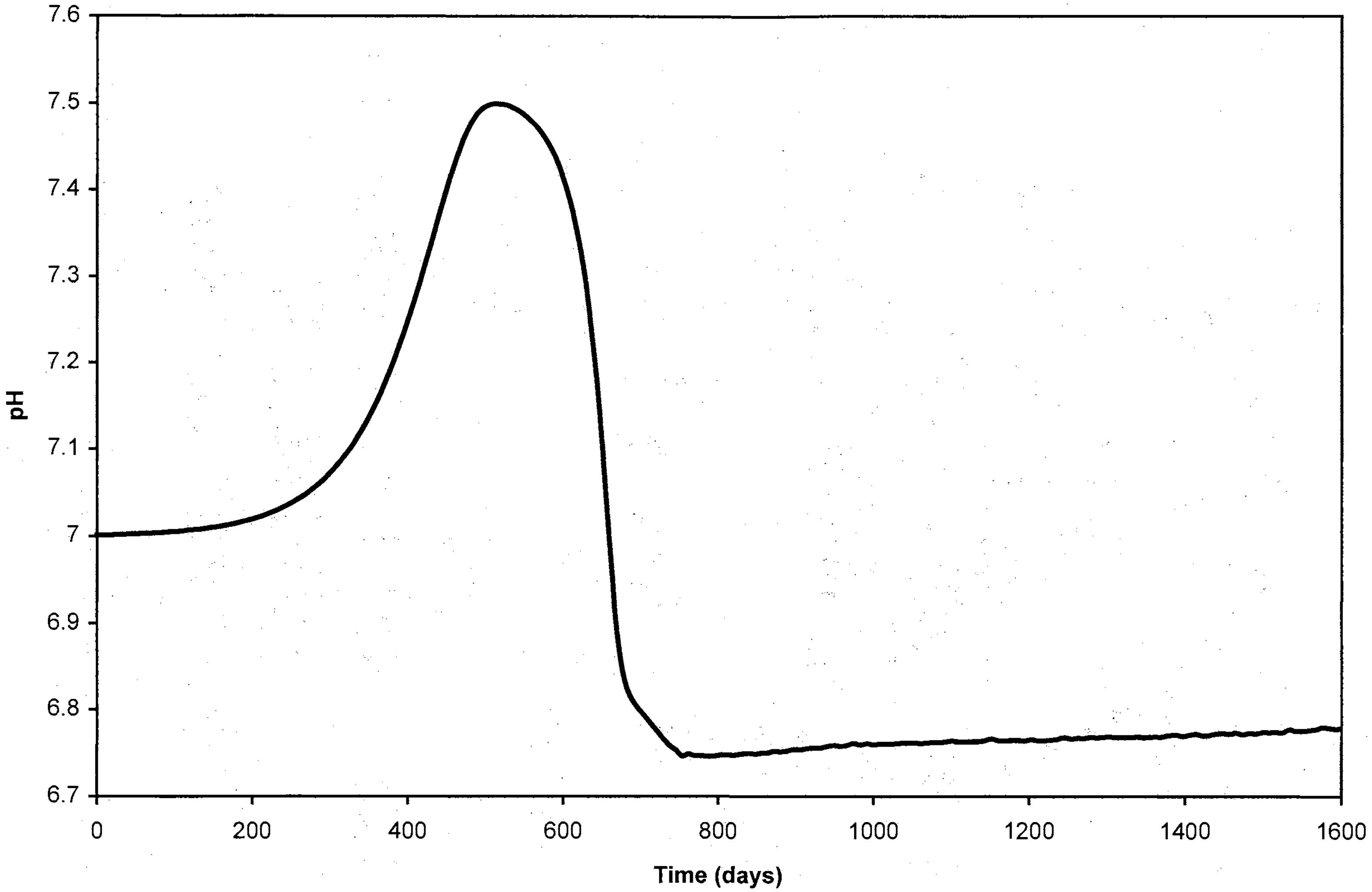


Figure 18. pH profile for the case with phosphate species

## Conclusions

A generalized rate model that can handle multicomponent mixed-bed ion exchange was developed. In addition to handling an arbitrary number of strong electrolytes, the model was extended to handle most weak electrolytes, such as type I, II and III dissociative species. Based on extensive tests, the total number of species that can be handled by the model is 18 cations and 18 anions.

The model was used to predict the breakthrough behavior of industrial MBIE columns. Several practical cases from industry were simulated. Predicted results agree favorably with plant experience.

Simulations also showed that the column performance is sensitive to the model parameters like MTC, inlet ion concentrations and flow conditions. When resins are fouled (MTC's are reduced), the kinetic leakage of ions tend to be higher; and the breakthrough time is earlier. In addition, when the inlet concentrations or the flowrate increase, the column will also break through earlier.

It was found that, besides the ion exchange reactions, physical adsorption and protonation of the nonionic forms of weak electrolytes into the resin play an important role in ion exchange with weak electrolytes. They should be considered in modeling multicomponent MBIE with weak electrolytes.

The incorporation of temperature dependent correlations for diffusivity, ionization constant, and viscosity in the model makes the evaluation of temperature effects on the column performance possible. However, to further improve the model, temperature dependent correlations for selectivity are required.

# CHAPTER IV

# MODELING OF SPECIAL CASES IN MULTICOMPONENT MIXED-BED ION EXCHANGE

## Abstract

This article addresses special case handling in modeling multicomponent mixed-bed ion exchange. The model developed in Chapter III was extended to handle some special cases in MBIE, such as variable inlet concentration, variable inlet flow rate, layered-bed and bed cleaning. To model these cases, efforts were made in modifying the computer code. Moreover, additional assumptions have been supplemented to the original model developed in Chapter III for each case. The contents presented here concentrate on the coding efforts and discussions of the simulation results.

## Introduction

Many multicomponent mixed-bed ion exchange systems do not exhibit uniform feed composition or flowrate due to fluctuation, or something unexpected during operations. In addition, segregation of cation resin to the bottom of the column to form cation heels can also occur if the system design for resin mixing and sluicing is poor or if these processes are not performed properly. Heels occur because cation resin has a higher density and greater terminal settling velocity than anionic beads. The column with resin heel or head is a layered-bed. Besides the above, in some power plants, the condensate polishers (mixed-beds) sometimes need to be removed out of service periodically for ultrasonic cleaning to eliminate iron oxides or other impurities accumulated in the bed. This is the case referred as bed cleaning. Therefore, extending the multicomponent MBIE model (see Chapter III) to handle these special cases is required. The objective of this work is to develop appropriate methods to handle these cases.

## Variable Inlet Concentrations Case

### Modeling Efforts

#### *Assumptions*

Most assumptions for modeling variable inlet concentrations are similar to those employed in Chapter III for the general case. However, besides the assumptions made in Chapter III, the following additional assumptions have to be supplemented.

- Only one-step change in inlet concentrations
- The time when inlet concentrations change is known
- The ionic physical properties remain constant with changing concentration

- Inlet concentrations changes are within the range of ultra-low conditions, so that film diffusion control still applies

*Coding efforts*

In a mixed-bed ion exchange column, when ionic inlet concentrations change after operating for some time, the total inlet concentration CF for all ions will also change, thus leading the change of column life (operating time) and dimensionless time – TAUMAX, which controls the code run time. To cope with these changes, a subroutine called "NEWCONCEN" is developed and implemented in the code to recalculate those variables depending on the total inlet concentration CF. The following are the major efforts made in modifying the computer program.

1) Determine the dimensionless time that the column has operated for the old concentrations (before concentration change).

```
TAUMAX_OLD1 = DAY_NEW1/DMAX*TAUMAX
```

Where Day_new1 is the time (days) when inlet concentrations change; Dmax is the maximum time the column can operate based on the old inlet concentrations.

2) Adjust the capacities of the column (deduct the capacity already consumed by the old concentrations) to calculate the operating time based on the new concentrations.

```
ANI_CAPNEW = (1-DAY_NEW1/DMAX)*ANI_CAP
CAT_CAPNEW = (1-DAY_NEW1/DMAX)*CAT_CAP
```

3) Call subroutine "NEWCONCEN" to recalculate those values depending on the total inlet concentration CF.

```
CALL NEWCONCEN (CF, A, B, C, …)
```

(see Appendix F for the detailed computer code). Based on the return values of this subroutine, the new practical time that the column can operate for both old and new concentrations is calculated by

```
            DMAX_PRAC = DMAX_NEW+DAY_NEW1
```

Where DMAX_NEW is the new time that column can operate for the new concentrations; DAY_NEW1 is the time the column has already operated for the old concentrations.

4) Then within the time loop, add the following statements to control the code execution (the details see Appendix F - Computer Source Code)

```
IF (TAUTOT.GT.TAUMAX_OLD1) THEN
     IF (CFLAG.EQ.'Y'.OR.CFLAG.EQ.'y') THEN
      TAUMAX = TAUMAX_NEW + TAUMAX_OLD1
      CF_DUM = CF
      CF = CF_NEW
       ......
       ......
     ENDIF
ENDIF
```

5) Finally, at the end of the time loop, add the following statements to recalculate the variable TAUTIM (days) which is required by the output.

```
IF (CFLAG.EQ.'Y'.OR.CFLAG.EQ.'y') THEN
   IF (TAUTOT.GT.TAUMAX_OLD1)THEN
     DALTTAU1=(TAUTOT-TAUMAX_OLD1)
      DALTIME1=DALTTAU1*ANI_DIA*ANI_CAP/(MTC_REF*CF*60.)/1440.
     TAUTIM=DAY_NEW1+DALTIME1
   ENDIF
ENDIF
```

## Simulation Results

Based on the efforts above, a multicomponent system with 2 strong cations ($Ca^{+2}$, $Na^{+}$), 2 strong anions ($SO_4^{-2}$, $Cl^{-}$), 1 amine ($NH_3$), and 2 organic acids (format/acetic acids) was simulated. Table I lists all model parameters and input data. The following are the detailed simulation results.

### 1) *Normal Operating Condition (Base Case)*

Figure 1 is the simulation results for the cations under normal operating conditions (no inlet concentration change). As you can see, sodium will break through in about 5 days, $NH_4^{+}$ 6.5 days, and $Ca^{+2}$ 16 days. In addition, monovalent cations like $Na^{+}$ and $NH_4^{+}$ exhibit 'throws' in their breakthrough curves. This is because all monovalent ions are kicked out by divalent cation $Ca^{2+}$, due to their lower preference by the cationic resins compared to $Ca^{+2}$. Likewise, all monovalent anions $Cl^{-}$, $HCOO^{-}$, $CH_3COO^{-}$ are kicked out by $SO_4^{-2}$ (see Figure 2), and showed throws in their breakthrough curves. From Figure 2, acetate ($CH_3COO^{-}$) breaks through in about 6 days, formate ($HCOO^{-}$) 7 days, $Cl^{-}$ 16 days and $SO_4^{-2}$ 37 days. The relative order of breakthrough time obtained for these ions is as what would be expected.

Table I. Input Data and Model Parameters for the Base Case

| Item | Value | |
|---|---|---|
| Bed diameter (cm) | 150.0 | |
| Bed depth (cm) | 50.0 | |
| Bed void fraction | 0.35 | |
| Resin bead diameter (cm) | | |
| Cation | 0.08 | |
| Anion | 0.06 | |
| Resin capacity (meq/ml) | | |
| Cation | 2.1 | |
| Anion | 1.0 | |
| Cation/Anion ratio | 0.5: 0.5 | |
| Influent pH | 7.0 | |
| Temperature (°C) | 60 | |
| Influent concentration (ppb) | | |
| Calcium ($Ca^{+2}$) | 60.0 | |
| Sodium ($Na^{+}$) | 69.0 | |
| Ammonia (total) | 102.0 | |
| Sulfate ($SO_4^{-2}$) | 144.0 | |
| Chloride ($Cl^{-}$) | 106.0 | |
| Formic acid (total) | 124.0 | |
| Acetic acid (total) | 198.0 | |
| Initial loadings on the resin (%) | | |
| Calcium ($Ca^{+2}$) | 0.1 | |
| Sodium ($Na^{+}$) | 0.1 | |
| Ammonia ($NH_3$) | 0.1 | |
| Sulfate ($SO_4^{-2}$) | 0.1 | |
| Chloride ($Cl^{-}$) | 0.1 | |
| Formic acid ($HCOO^{-}$) | 0.1 | |
| Acetic acid ($CH_3COO^{-}$) | 0.1 | |
| Influent flow rate ($cm^3/s$) | 2.56E+4 (40gpm) | |
| Physical properties | Diffusivity (25 C) | Selectivity |
| Species: | ($cm^2/s$) | |
| Calcium ($Ca^{+2}$) | 0.792E-5 | 4.4 |
| Sodium ($Na^{+}$) | 1.334E-5 | 1.5 |
| Ammonia ($NH_3$) | 1.957E-5 | 1.95 |
| Sulfate ($SO_4^{-2}$) | 1.065E-5 | 20 |
| Chloride ($Cl^{-}$) | 2.032E-5 | 22 |
| Formic acid ($HCOO^{-}$) | 0.823E-5 | 4.6 |
| Acetic acid ($CH_3COO^{-}$) | 1.089E-5 | 3.2 |

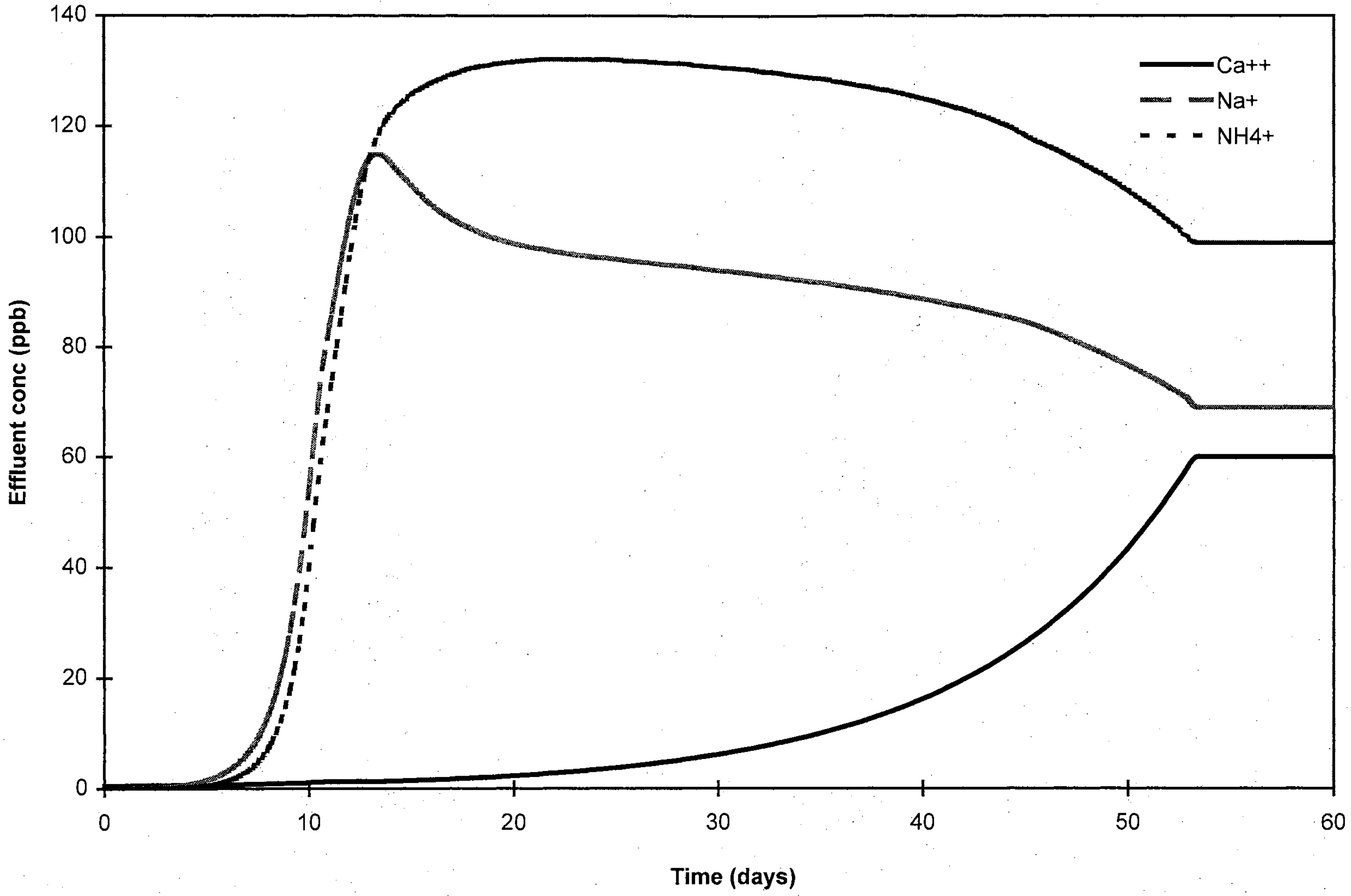


Figure 1. Breakthrough curves of cations in the normal case

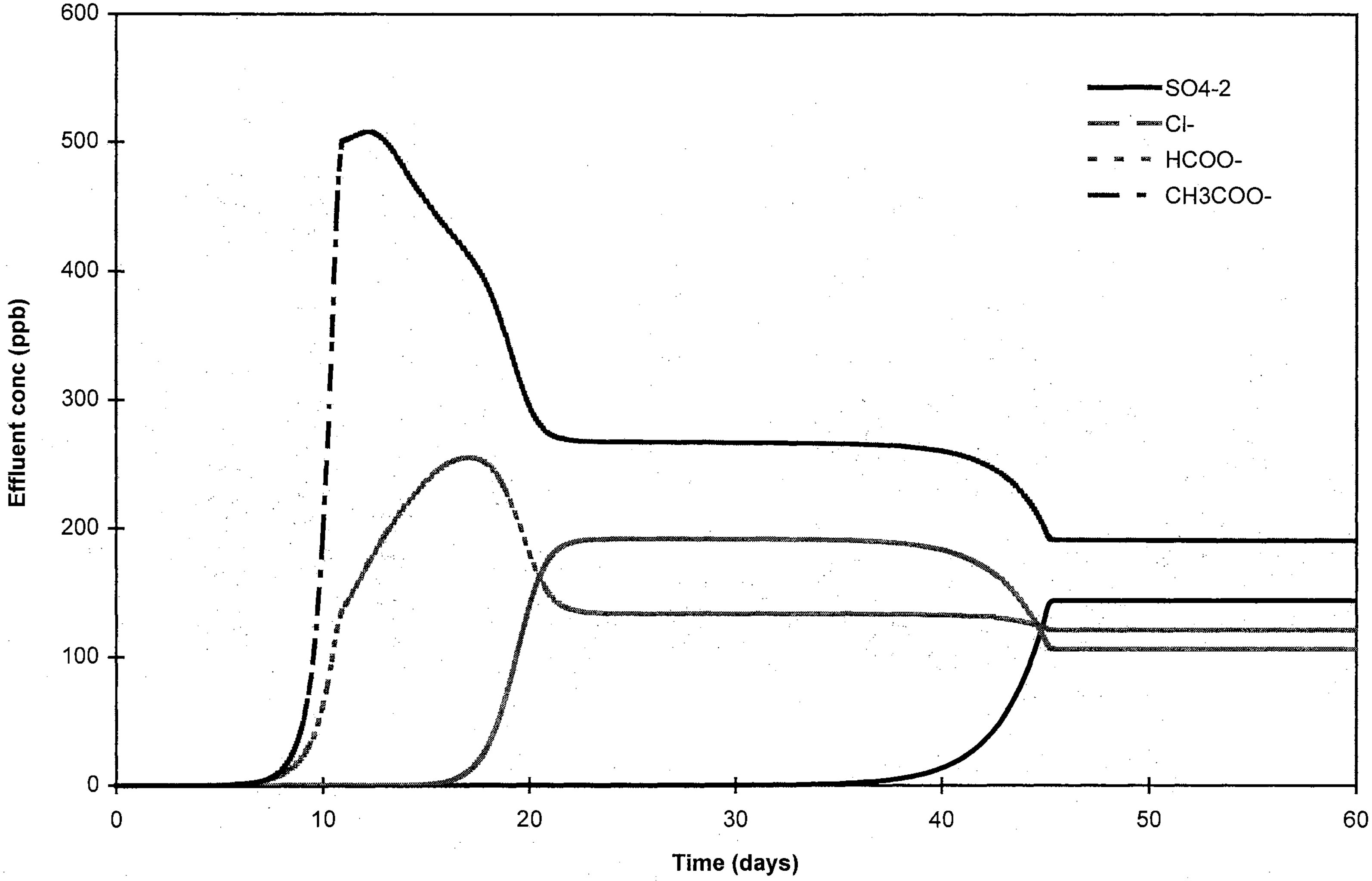


Figure 2. Breakthrough curves of anions in the normal case

2) *Variable Inlet Concentrations*

Two cases were simulated for variable inlet concentrations: i) inlet concentrations increase 20 % at the 10th operating day; and ii) inlet concentrations decrease 20% at the 10th operating day. Table II summarizes all these concentrations.

Table II. Old and New Inlet Concentrations Used in the Simulations

| Ions | Normal inlet conc (meq/ml) | New inlet conc after increased 20% (meq/ml) | New inlet conc after decreased 20% (meq/ml) |
|---|---|---|---|
| $Ca^{+2}$ | 3.0E-6 (60 ppb) | 3.6E-6 (72 ppb) | 2.4E-6 (48 ppb) |
| $Na^{+}$ | 3.0E-6 (69 ppb) | 3.6E-6 (83 ppb) | 2.4E-6 (55 ppb) |
| $NH_4^{+}$ | 6.0E-6 (102 ppb) | 7.2E-6 (122 ppb) | 4.8E-6 (82 ppb) |
| $SO_4^{-2}$ | 3.0E-6 (144 ppb) | 3.6E-6 (173 ppb) | 2.4E-6 (115 ppb) |
| $Cl^{-}$ | 3.0E-6 (106 ppb) | 3.6E-6 (127 ppb) | 2.4E-6 (85 ppb) |
| $HCOO^{-}$ | 2.7E-6 (124 ppb) | 3.24E-6 (149 ppb) | 2.16E-6 (99 ppb) |
| $CH_3COO^{-}$ | 3.3E-6 (198 ppb) | 3.96E-6 (238 ppb) | 2.64E-6 (158 ppb) |

As shown in Figure 3 through Figure 6, ions tend to break through earlier when inlet concentrations increase, and vice versa. For example, the breakthrough time for $Ca^{+2}$ decreased from 16 days to 12 days when the inlet concentration of $Ca^{+2}$ increased by 20%; while the breakthrough time of $Ca^{+2}$ will increase from 16 to 20 days when the inlet concentration decrease by 20% (see Figure 3).

In addition, ionic throws (bumps or plateaus in the figures) are observed for all monovalent ions; and these throw peaks move forward or backward when the inlet concentrations are increased or decreased. For instance (see Figure 4), the throw peak of $NH_4^{+}$ moves from the 20th day to 19th day when the inlet concentration increased by 20%; and moves from 20th day to 21st day when the inlet concentration decreased by 20%. These results are easy to understand and generally consistent with our expectations.

Moreover, we found that the peak value of throw increases with the increase or decreases with the decrease of new inlet concentrations. For the system given in Table I, the peak value of $NH_4^+$ throw is 132 ppb in the normal case. But when inlet concentration increased by 20%, it is increased to 159 ppb; and when inlet concentration decreased by 20%, it is decreased to 105 ppb. Similar phenomena are also observed for other monovalent ions.

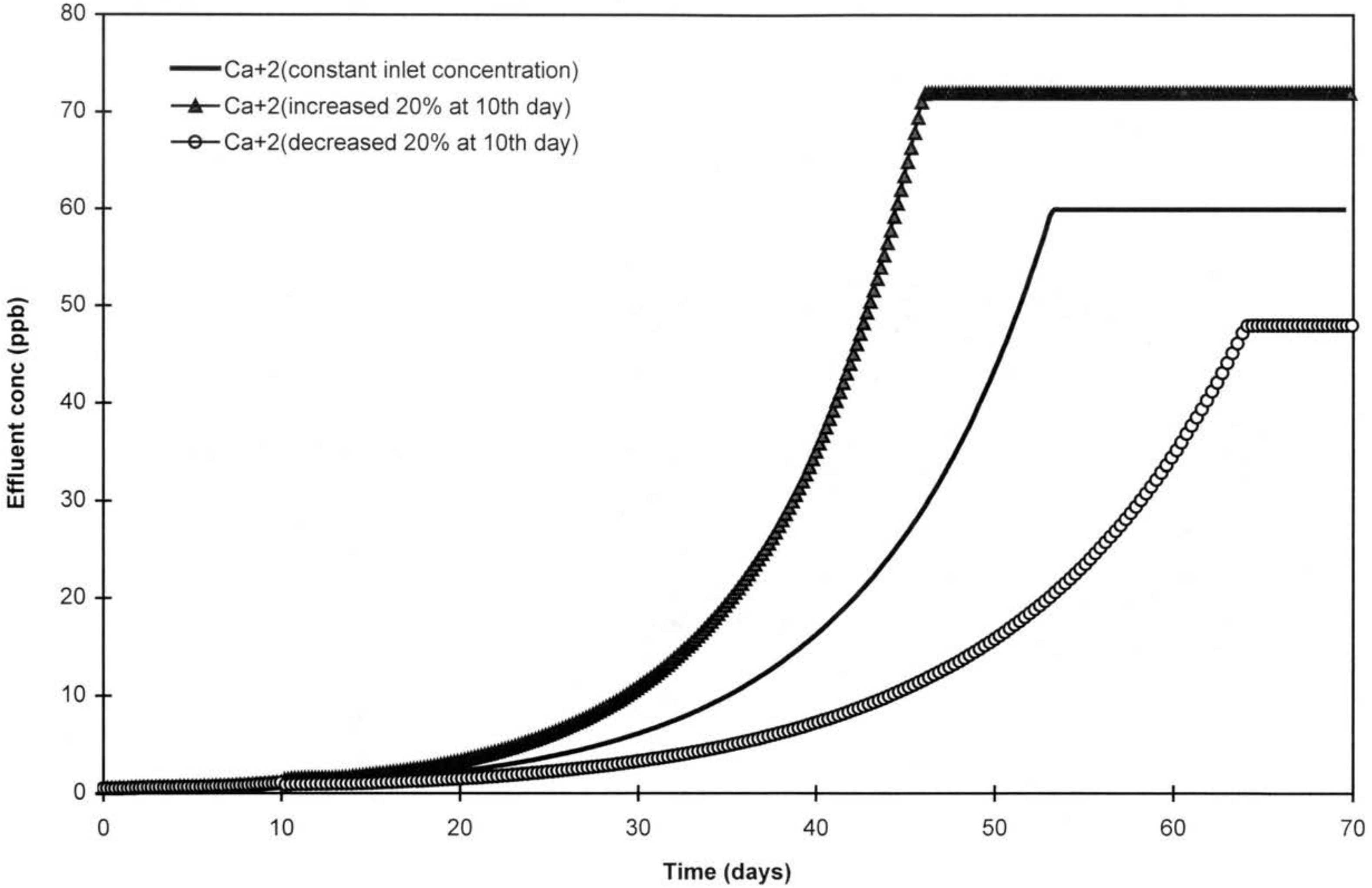


Figure 3. Effect of variable inlet concentration on $Ca^{+2}$ breakthrough curves

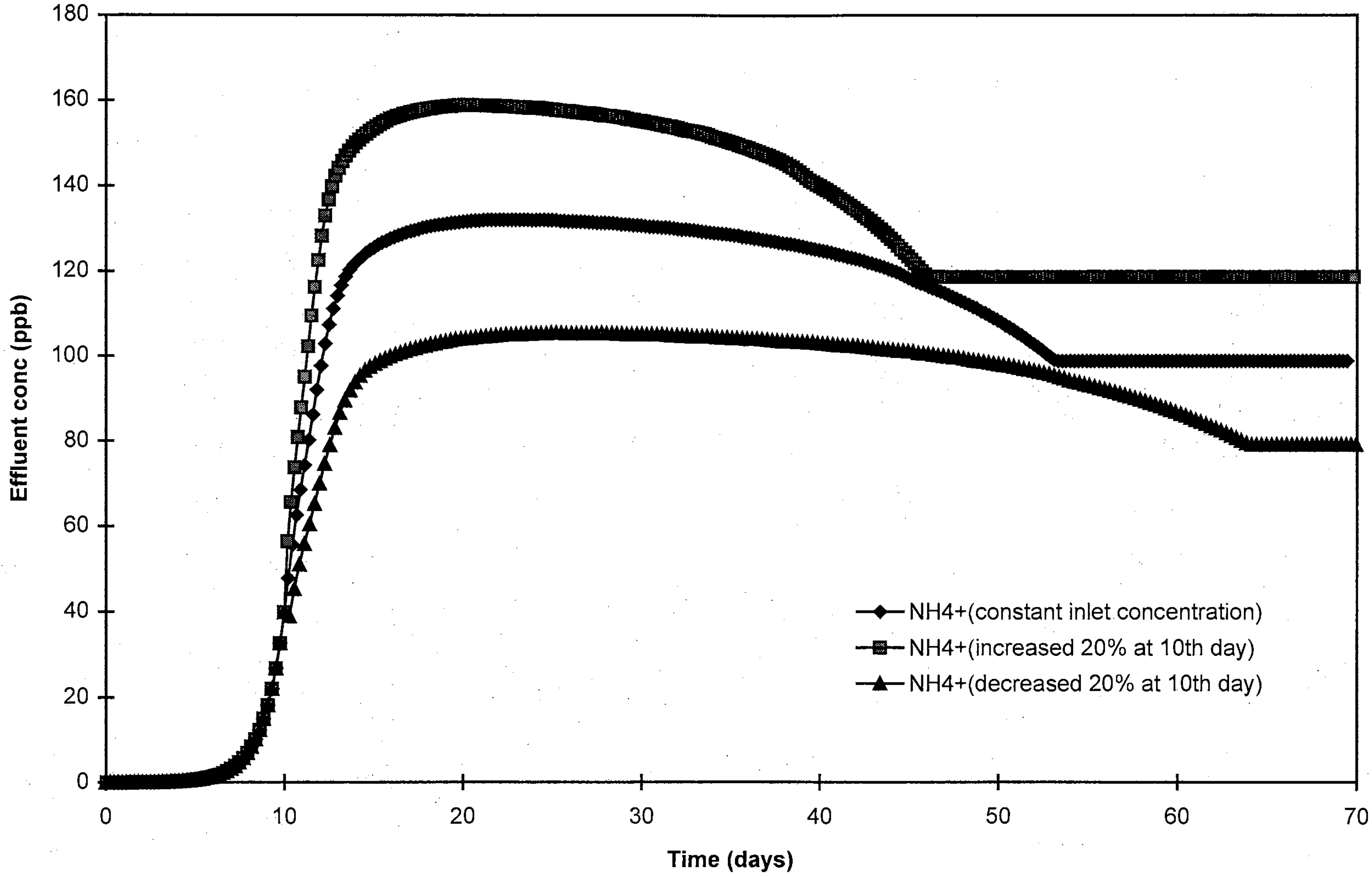


Figure 4. Effect of variable inlet concentration on $NH_4^+$ breakthrough curve

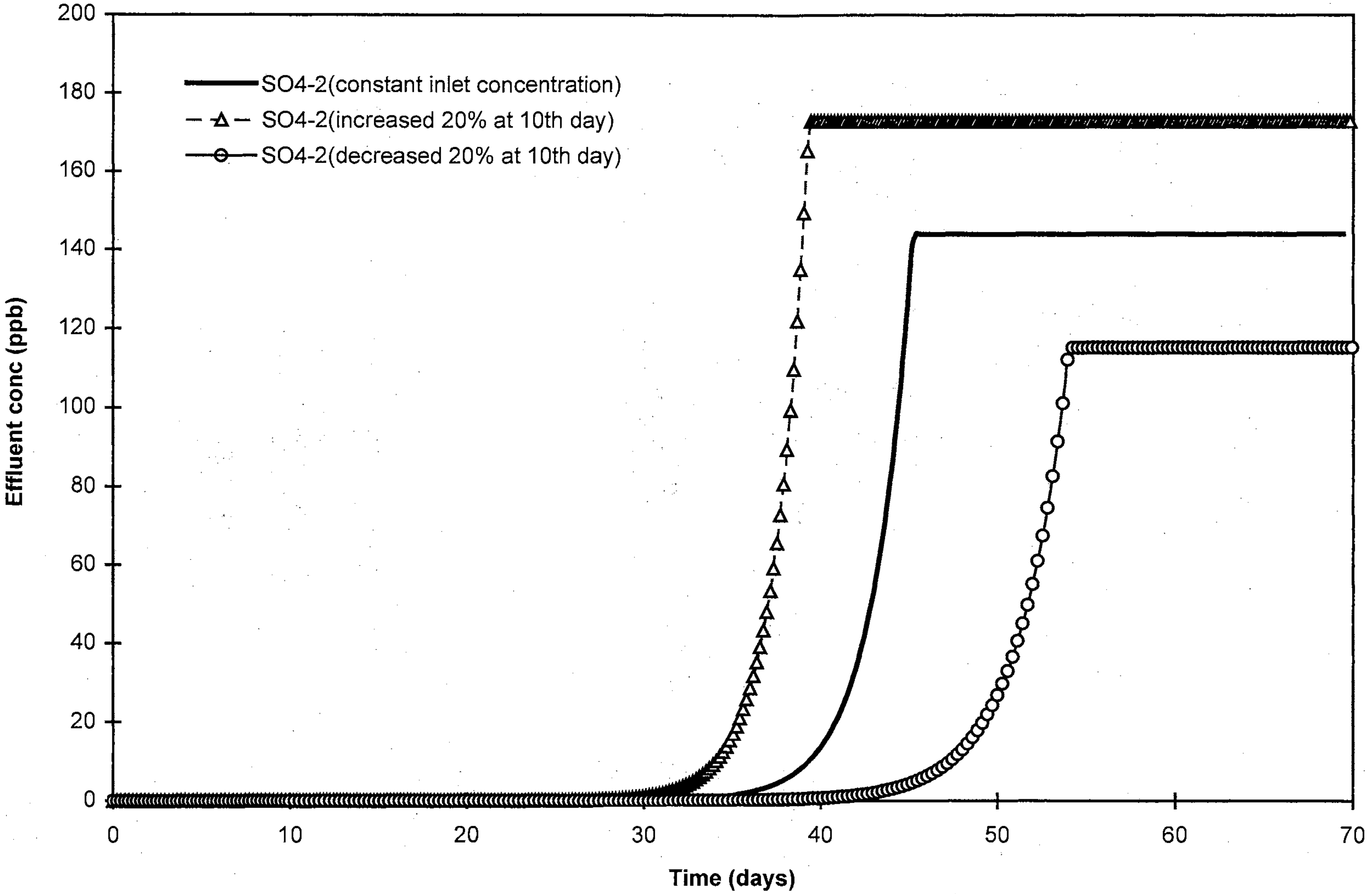


**Figure 5. Effect of variable inlet concentration on $SO_4^{-2}$ breakthrough curves**

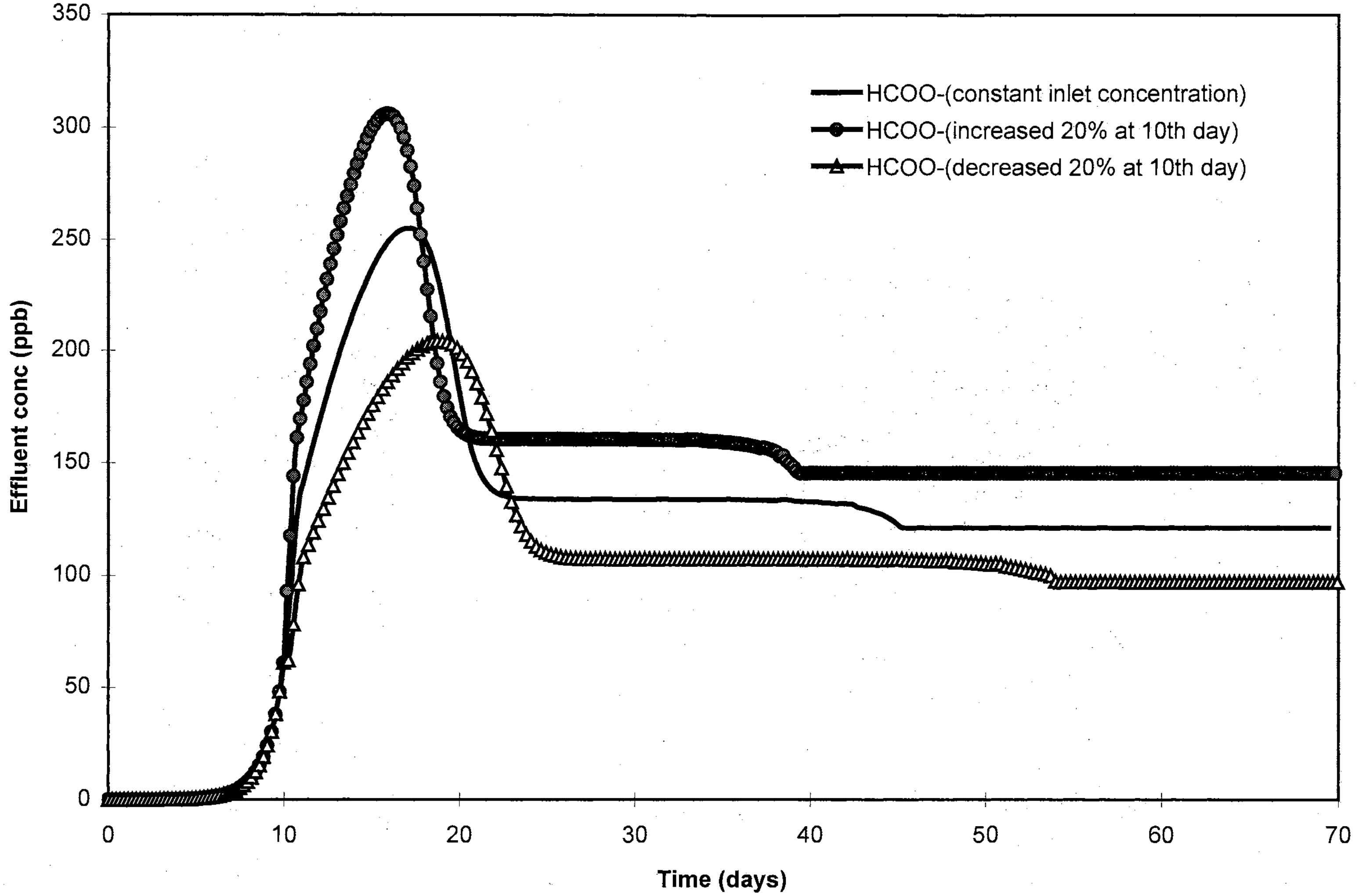


Figure 6. Effect of variable inlet concentration on $HCOO^-$ breakthrough curve

## Variable Inlet Flowrate Case

### Modeling Efforts

Modeling variable inlet flowrate is more complex than modeling variable inlet concentrations, because the change of flowrate results in the change of superficial velocity $V_S$, which modifies the number of slices that controls the distance loop (the number of slices depends on both superficial velocity and distance step used). Given a constant distance step, the number of slice decreases with increasing flowrate, and vice versa. Because of the change of number of slices, those variables – such as fraction of bulk phase concentration $X_i$ and resin loading $Y_i$ need to be recalculated on the new slices.

#### *Assumptions*

Like the case of variable inlet concentrations, most assumptions for modeling variable inlet flowrate are the same as those in Chapter III. In addition, the following additional assumptions are made.

- Only one step change in inlet flowrate during operation
- The time when inlet flowrate changes is known
- The physical properties of species are the same before and after flowrate changes
- The change of loading or bulk phase concentrations as a function of bed depth are continuous, so that the interpolating method can be used

*Coding efforts*

To develop a procedure to handle variable inlet flowrate, a subroutine called "NEWFLOW" was developed and implemented in the MBIE code. In addition, the natural cubic spline method (Cheney and Kincaid, 1985) was used to interpolate the values of resin loadings $Y_i$ and bulk phase concentration fractions $X_i$ on the new slices. The following are the detailed efforts made in modifying the code.

1) Determine the dimensionless time that column has operated at the old flowrate (before flowrate change)

```
TAUMAX_OLD2 = DAY_NEW2/DMAX*TAUMAX
```

2) Adjust the capacities of the column (deduct the capacity already consumed by the old flowrate) to calculate the new maximum run time TAU_MAX

```
ANI_CAPNEW2 = (1-DAY_NEW2/DMAX)*ANI_CAP
CAT_CAPNEW2 = (1-DAY_NEW2/DMAX)*CAT_CAP
```

3) Call subroutine "NEWFLOW" to recalculate those variables depending on the flowrate (VOL_FLOW). These variables include superficial velocity ($V_s$), number of slices (NOSLICE) and mass transfer coefficients (MTC's).

```
CALL NEWFLOW (A, B, C, …)
```

4) A subroutine named "CUBSPLINE" is developed and used to interpolate the resin loadings and bulk phase concentration fractions on the new slices. This subroutine is called when the dimensionless time TAUTOT is greater than TAUMAX_OLD2 (the details see Appendix F - Computer Source Code).

```
IF (TAUTOT.GT.TAUMAX_OLD2) THEN
   IF (VFLAG.EQ.'Y'.OR.VFLAG.EQ.'y') THEN
      TAUMAX=TAUMAX_NEW+TAUMAX_OLD2
      CHTD=CHTDNEW
```

```
                NOSLICE=NOSLICE_NEW
                .........
                .........
             DO 1115 I=1, NC
                do 1116 K=1, MT
                   XNC(K)=XBC(I,K)
1116            continue
                CALL cubspline(CHTD,XI1,XI,MT,NMT,XNC,U,V,Z,W,B,S,T)
                do 1117 K=1, NMT
                   XBC(I,K)=S(K)
1117            continue
1115         CONTINUE
             .........
             .........
            ENDIF
          ENDIF
```

5) Finally, add the following statements to recalculate the TAUTIM (days), which is required by the output.

```
            IF (VFLAG.EQ.'Y'.OR.VFLAG.EQ.'y') THEN
                IF (TAUTOT.GT.TAUMAX_OLD2)THEN
                  DALTTAU2=(TAUTOT-TAUMAX_OLD2)
                  DALTIME2=DALTTAU2*ANI_DIA*ANI_CAP/
                              (MTC_REF*CF*60.)/1440.
                  TAUTIM=DAY_NEW2+DALTIME2
                ENDIF
            ENDIF
```

## Simulation Results

Using the input data and model parameters in Table I, two cases are simulated for variable inlet flowrate: i) inlet flowrate increases 20% on the 5th day, and ii) inlet flowrate decreases 20% on the 5th day. Figures 7 through 10 are the simulation results. From Figure 7, it is observed that due to the change of flowrate at the day 5, the time for

$Ca^{+2}$ to reach breakthrough will be different. Generally, increasing flowrate results in earlier breakthrough, while decreasing flowrate leads to late breakthrough. However, no matter how the flowrate changes, the breakthrough curves will finally come to the same value (equal to the inlet concentration). These predictions are also in agreement with what would be expected. For monovalent ions, the same trends are observed except throws appear before they come to the inlet concentrations (see Figure 8 and 10). The reason of ionic throw is that monovalent ions are kicked out by other higher valence ions.

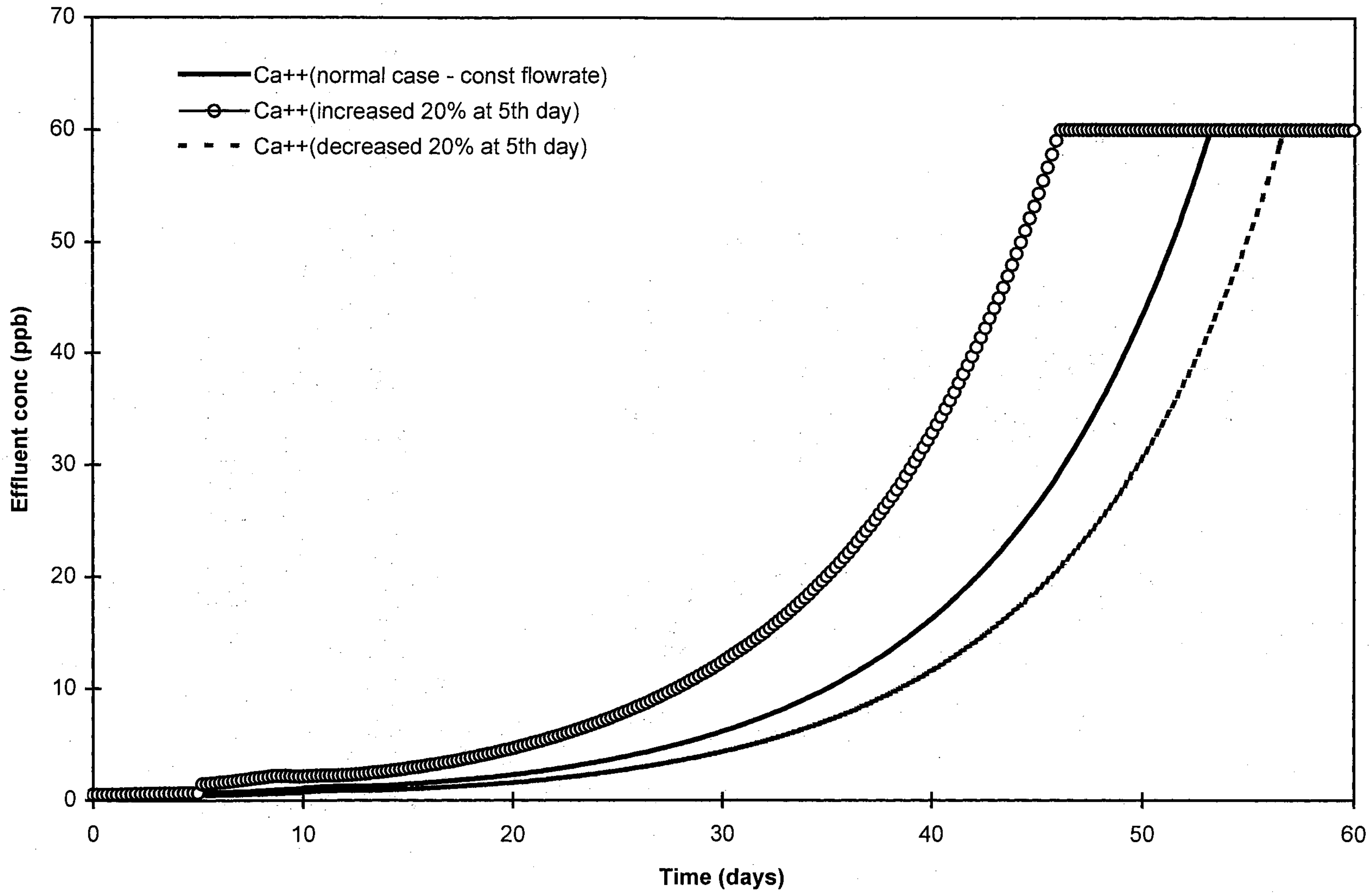


Figure 7. Effect of variable inlet flowrate on $Ca^{+2}$ breakthrough curve

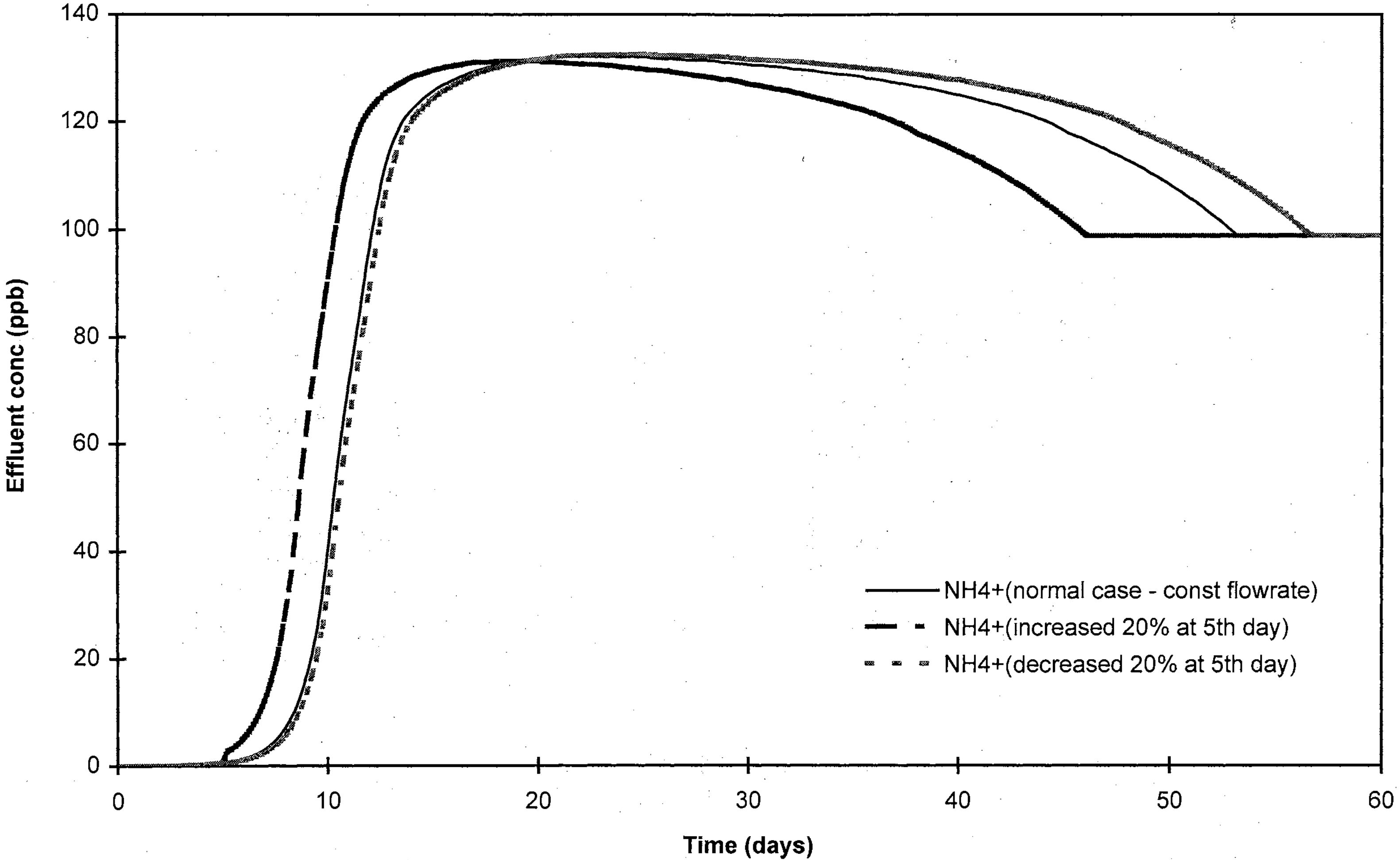


Figure 8. Effect of variable inlet flowrate on $NH_4^+$ breakthrough curve

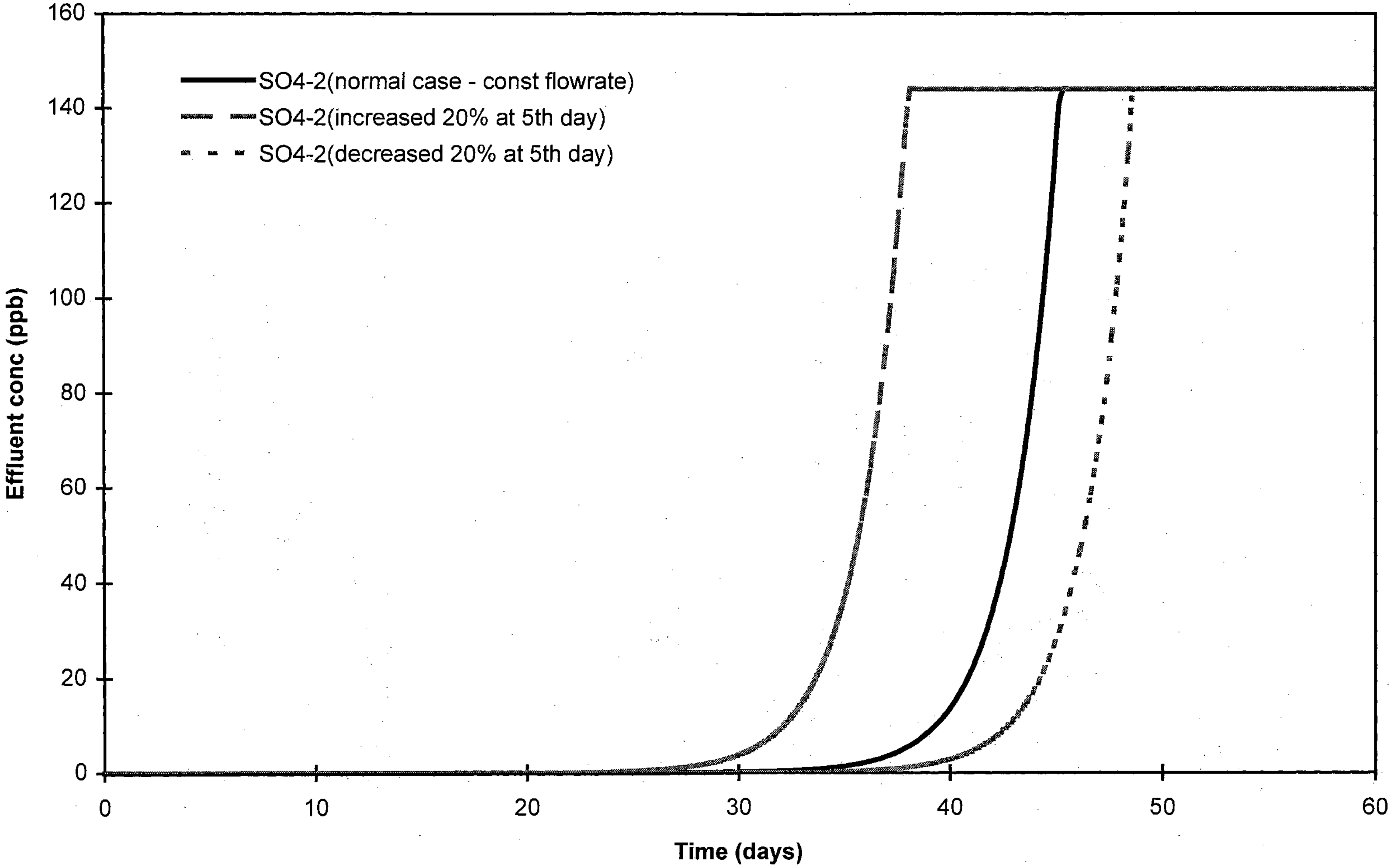


Figure 9. Effect of variable inlet flowrate on $SO_4^{-2}$ breakthrough curve

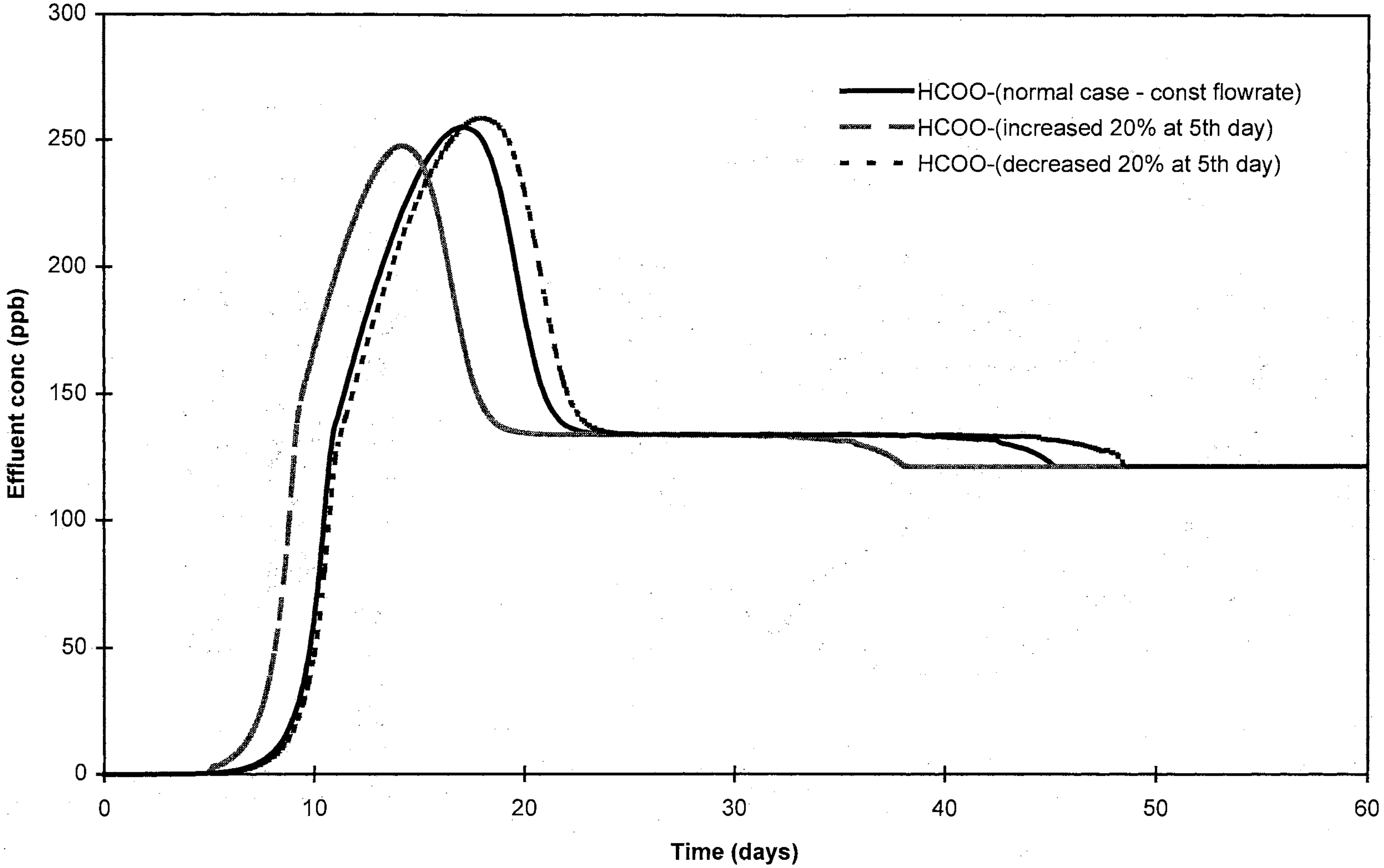


**Figure 10. Efflect of variable inlet flowrate on $HCOO^-$ breakthrough curve**

## Layered-Bed Case

### Modeling Efforts

Layered-beds are often encountered in industry, for instance, mixed-beds with a resin heel or head. In addition, several columns with different cation/anion ratios are also often operated in series (e.g. mono-bed followed by mixed-bed), which can also be viewed as a layered-bed. Therefore, modeling layered-bed is very important for MBIE operations.

In modeling layered-beds, because of different cation/anion ratio in each layer, the distance loop in the computer model has to be divided into several parts to incorporate the effects of cation/anion ratios (or FAR and FCR).

### *Assumptions*

Besides the assumptions made in Chapter III for the general case, several additional assumptions have been added to model layered-bed. For simplification, the current model is temporarily assumed to handle layered-bed alone; that is, the cases of variable inlet concentration or variable inlet flowrate cannot be handled simultaneously with layered-bed. The following are the additional assumptions made.

- Flow is continuous, that is, the flowrate is the same for different layers
- No inlet concentrations change
- No inlet flowrate change

*Coding efforts*

The key of handling a layered-bed is the distance loop. Therefore, our coding efforts will concentrate on the modification of the code within the distance loop. The following are what we have done in modifying the computer code.

1) Ask users to input the values of cation/anion ratios for the different layers, and calculate the resin ration fraction for each layer.

```
IF (RFLAG.EQ.'Y'.OR.RFLAG.EQ.'y') THEN
   WRITE(*,*)'input the C/A ratio for inlet part of column'
   READ(*,*) RATIO1
   WRITE(*,*)'input the length of the inlet part of column'
   READ(*,*)BED_HT1
   WRITE(*,*)'input the C/A ratio for middle part of column'
   READ(*,*)RATIO2
   WRITE(*,*)'input the length of the middle part of column'
   READ(*,*)BED_HT2
   WRITE(*,*) 'input the C/A ratio for outlet part of column'
   READ(*,*)RATIO3
   WRITE(*,*) 'input the length of the outlet part of column'
   READ(*,*)BED_HT3
     FAR1=1./(1+RATIO1)
     FCR1=1.-FAR1
     FAR2=1./(1+RATIO2)
     FCR2=1.-FAR2
     FAR3=1./(1+RATIO3)
     FCR3=1.-FAR3
ENDIF
```

2) Determine the dimensionless height or number of slices in each layer.

```
IF (RFLAG.EQ.'Y'.OR.RFLAG.EQ.'y') THEN
```

```
        GRPNUM = MTC_REF*(1.-VOID_FRAC)/(VS*ANI_DIA)
        CHTD1 = GRPNUM*BED_HT1
        CHTD2 = GRPNUM*BED_HT2
        CHTD3 = GRPNUM*BED_HT3
        NT1=CHTD1/XI-3+3*XI/XI1
        NT2=CHTD2/XI
        NT3=CHTD3/XI
        NOSLICE = NT1+NT2+NT3
     ENDIF
```

3) Recalculate the dimensionless time of 'TMAXC' and 'TMAXA'

```
    IF (RFLAG.EQ.'Y'.OR.RFLAG.EQ.'y') THEN
        GRPCAT = CAT_CAP*3.142*(BED_DIA/2.)**2/(VOL_FLOW*CF*60.)
        TMAXC1 = GRPCAT*BED_HT1*FCR1
        TMAXC2 = GRPCAT*BED_HT2*FCR2
        TMAXC3 = GRPCAT*BED_HT3*FCR3
        TMAXC=TMAXC1+TMAXC2+TMAXC3
        GRPANI = ANI_CAP*3.142*(BED_DIA/2.)**2/(VOL_FLOW*CF*60.)
        TMAXA1 = GRPANI*BED_HT1*FAR1
        TMAXA2 = GRPANI*BED_HT2*FAR2
        TMAXA3 = GRPANI*BED_HT3*FAR3
        TMAXA=TMAXA1+TMAXA2+TMAXA3
     ENDIF
```

4) Within the distance loop, implement the following statements to control the execution of the code.

```
IF (RFLAG.EQ.'Y'.OR.RFLAG.EQ.'y') THEN
    IF (K.LE.NT1) THEN
          FAR=FAR1
          FCR=FCR1
    ELSEIF (K.GT.NT1.AND.K.LE.(NT1+NT2)) THEN
          FAR=FAR2
```

```
                FCR=FCR2
            ELSEIF (K.GT.(NT1+NT2).AND.K.LE.(NT1+NT2+NT3)) THEN
                FAR=FAR3
                FCR=FCR3
                ENDIF
        ENDIF
```

## Simulation Results

Using the input data and model parameters in Table I, two cases of layered-bed are simulated: i) a 20-cm anion bed followed by a 30-cm mixed-bed; and ii) a 30-cm mixed-bed followed by a 20-cm cation bed. For these two cases, the total length of the mono-bed plus the mixed-bed is 50 cm, which is equal to the length of the base case column (well mixed-bed). Figure 11 through 14 are the simulation results. It can be seen that in the case of an anion bed followed by a mixed bed, cations will breakthrough earlier, while anions later. However, in the case of a mixed bed followed by a cation bed, cations will breakthrough later, while anions earlier. These results are reasonable and as expected.

Based on the above simulations results, we may conclude that to prevent anion leakage, a combination of an anion bed followed by a mixed bed; or a mixed bed with anion resin head is effective. However to control cation leakage, a mixed bed followed by a cation bed, or a mixed bed with cation resin heel can be used.

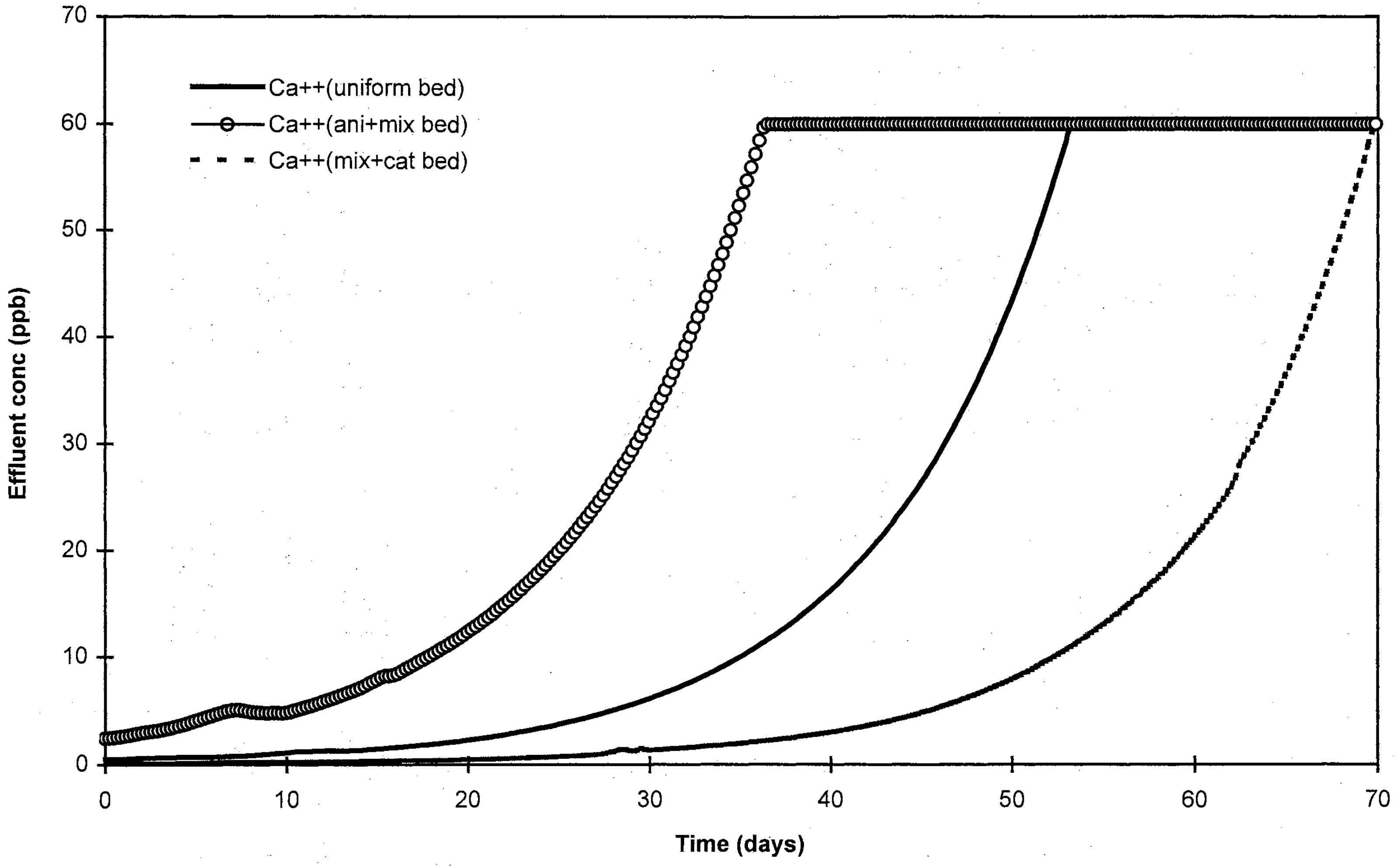


Figure 11. Comparison of $Ca^{+2}$ breakthrough curves for uniform bed and layered-bed

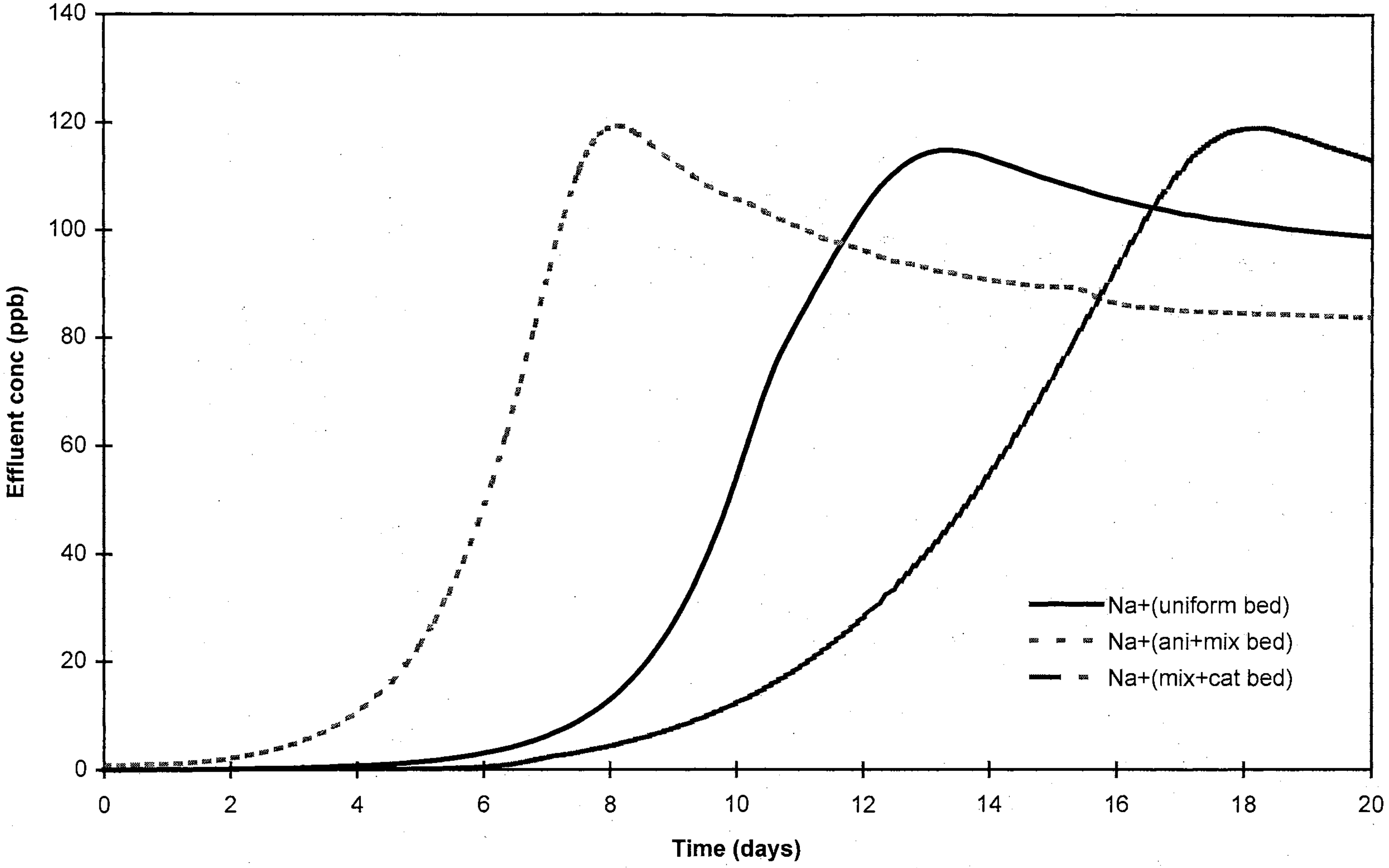


Figure 12. Comparison of Na+ breakthrough curves for uniform bed and layered-bed

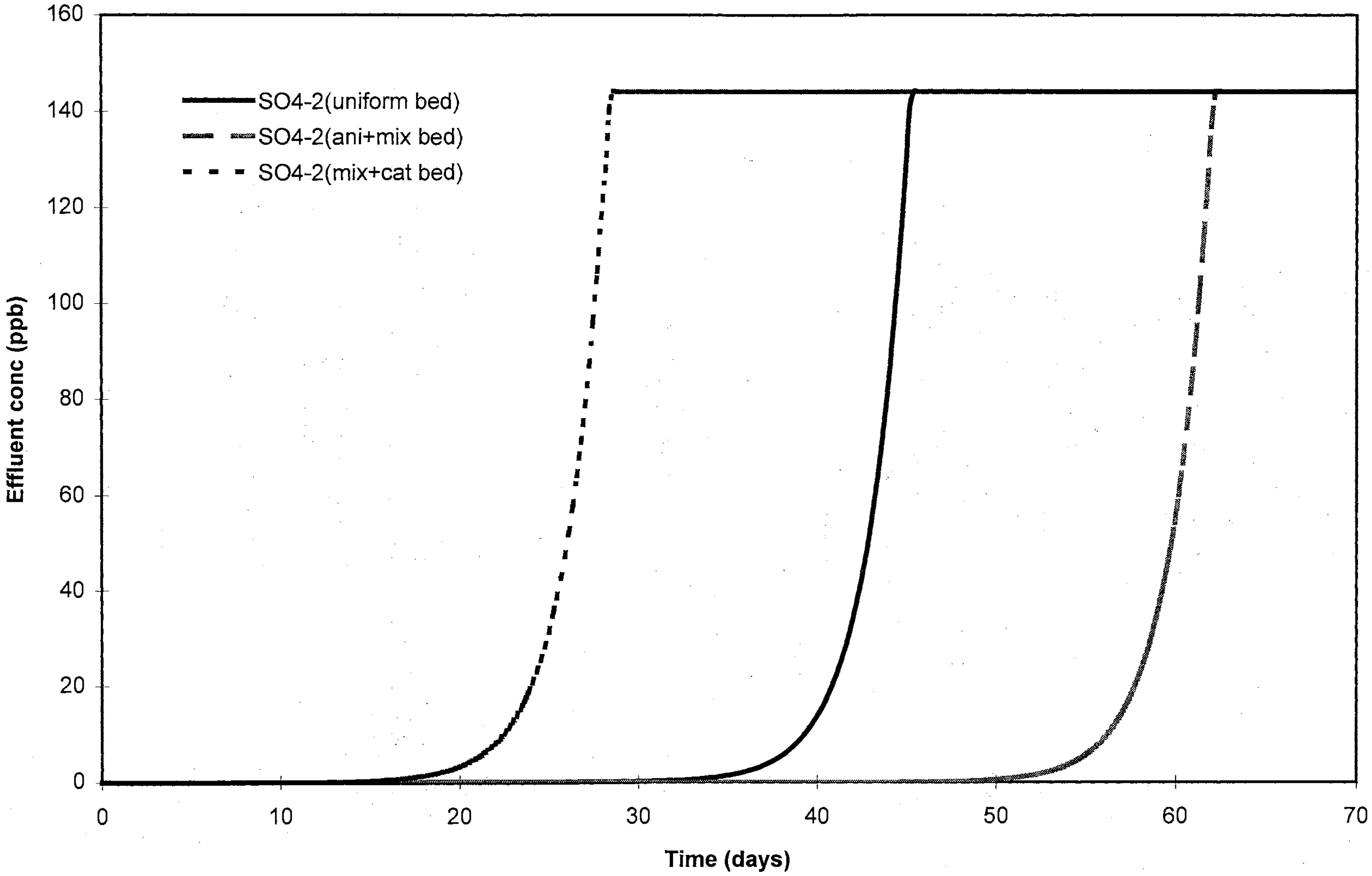


Figure 13. Comparison of $SO_4^{-2}$ breakthrough curves for uniform bed and layered-bed

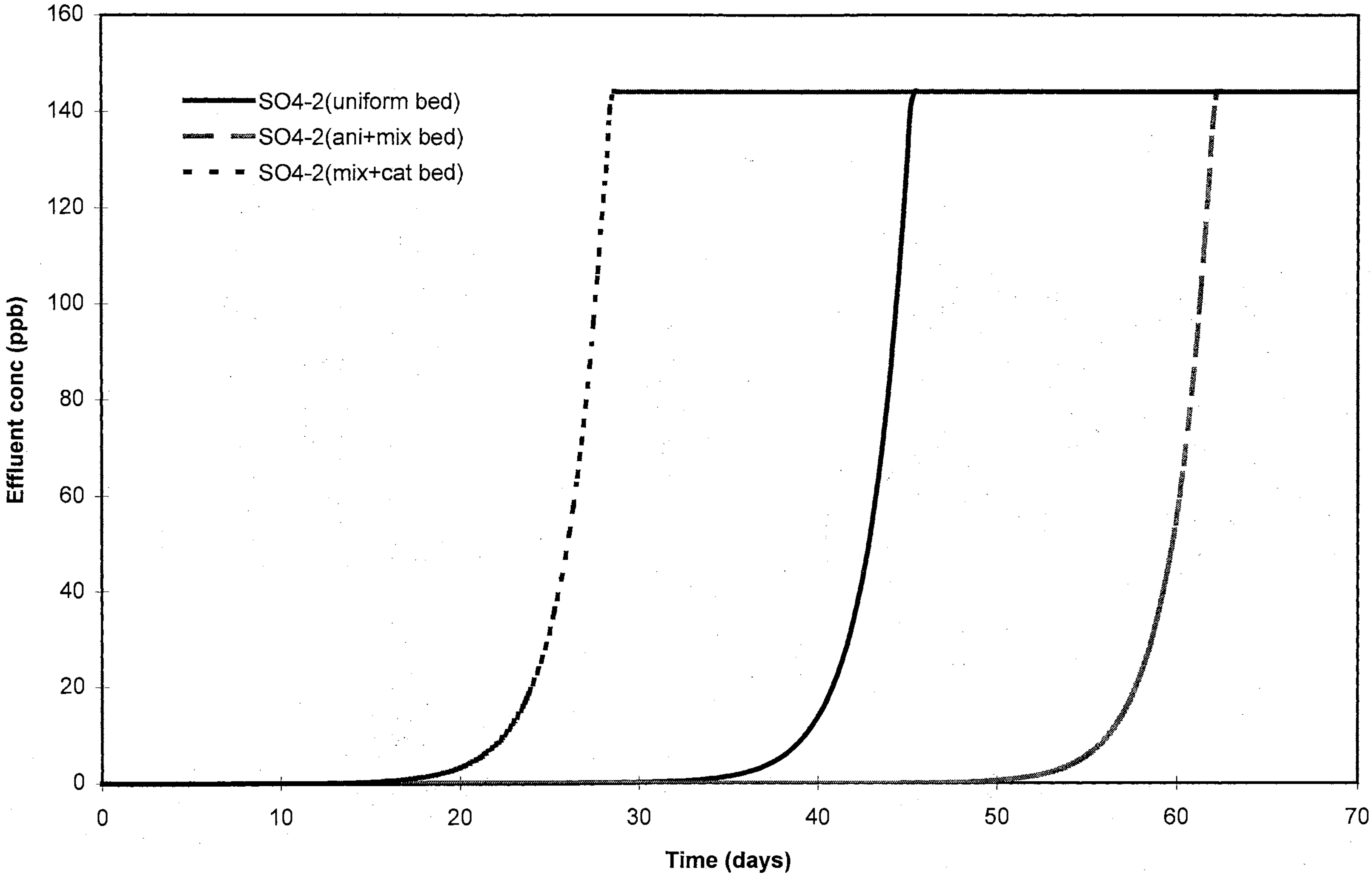


Figure 13. Comparison of $SO_4^{-2}$ breakthrough curves for uniform bed and layered-bed

## Bed Cleaning Case

### Modeling efforts

Bed cleaning is another special case that may be encountered in some power plants. For example, at Susquehanna Power Plant of Pennsylvania, iron oxides from corrosion and erosion in the steam cycle accumulate on the beds during operations, which increases bed flow resistance and raises system pressure drop. Therefore, to ensure proper operation, the resin bed must be removed from service and transferred outside for ultrasonic cleaning about once a month (every 30 days). For such a case, the following additional assumptions have been made to model it.

#### *Assumptions*

- No flow rate changes for the mixed-bed
- No inlet concentrations change
- Column is uniform (no layered bed)
- After cleaning, the loading of the bed are redistributed uniformly, so that the average loading from the top to the bottom can be taken as the initial loading for the cleaned bed

#### *Coding efforts*

To make the code handle the bed cleaning case, the following statements have been added into the computer model.

1) Ask user to input the time interval of bed cleaning, and then convert it (in days) into dimensionless time.

```
WRITE(*,*)
WRITE(*,*) 'Do you want to handle the bed cleaning case ?'
READ(*,501) BFLAG
IF (BFLAG.eq.'Y'.OR. BFLAG.eq.'y') THEN
      write(*,*)
      write(*,*) "input the time interval of bed cleaning"
      read(*,*) Day_clean
      Tclean_user = Day_clean*1440
      tau_clean = MTC_REF*CF*(Tclean_user*60.)/
                                  (ANI_DIA*ANI_CAP)
ENDIF
```

2) Within the time loop, implement the following statements to average the resin loadings when the cleaning happens, and assign it to all slices.

```
         IF (BFLAG.eq.'Y'.OR. BFLAG.eq.'y') THEN
            IF (MOD(tautot,tau_clean).lt.tau) THEN
                DO 1202 II = 1,NC
                     catloading(II) = 0
                     Do 1203 M = 1, NOSLICE+1
                        catloading(II) = catloading(II)+YRC(II,JD,M)
1203                 Continue
                     catloading(II) = catloading(II)/(NOSLICE+1)
1202            CONTINUE
                DO 1208 JJ = 1,NA
                      aniloading(JJ) = 0
                      Do 1209 M = 1,NOSLICE+1
                        aniloading(JJ) = aniloading(JJ)+YRA(JJ,JD,M)
1209                  Continue
```

```
                  aniloading(JJ) = aniloading(JJ)/(NOSLICE+1)
1208        CONTINUE
              DO 974 M = 1, MT
                 DO 966 II = 1,NC
                    YRC(II,JD,M) = catloading(II)
 966             CONTINUE
                  DO 970 JJ = 1, NA
                    YRA(JJ,JD,M) = aniloading(JJ)
970               CONTINUE
974           CONTINUE
          ENDIF
     ENDIF
```

Simulations Results

Figures 15 and 16 are the simulation results for the case of bed cleaning. When the bed is cleaned every 5 or 10 days, ions tend to break through earlier. The breakthrough curve obtained for the bed cleaning case is not a smooth curve, but contains ladders, which is different from the normal case. For sulfate (Figure 15), the time for its effluent concentration to reach 0.01 ppb is15.2 days when the bed is cleaned every 5 days, or 18.3 days if it is cleaned every 10 days. While for the normal case (no bed cleaning), the time for sulfate to reach 0.01 ppb is 24.7 days. The reason of earlier breakthrough is that after bed cleaning, the resins are remixed, so that a uniform loading is achieved for the new bed; and this initial loading of the new bed can be approximated by taking the average of the loadings from top to bottom for old bed. Therefore, earlier breakthrough is expected. In addition, we also found that cleaning frequency does not have significant effect on the breakthrough curve. Figure 16 is the simulation results for chloride, from which the same phenomena can be observed.

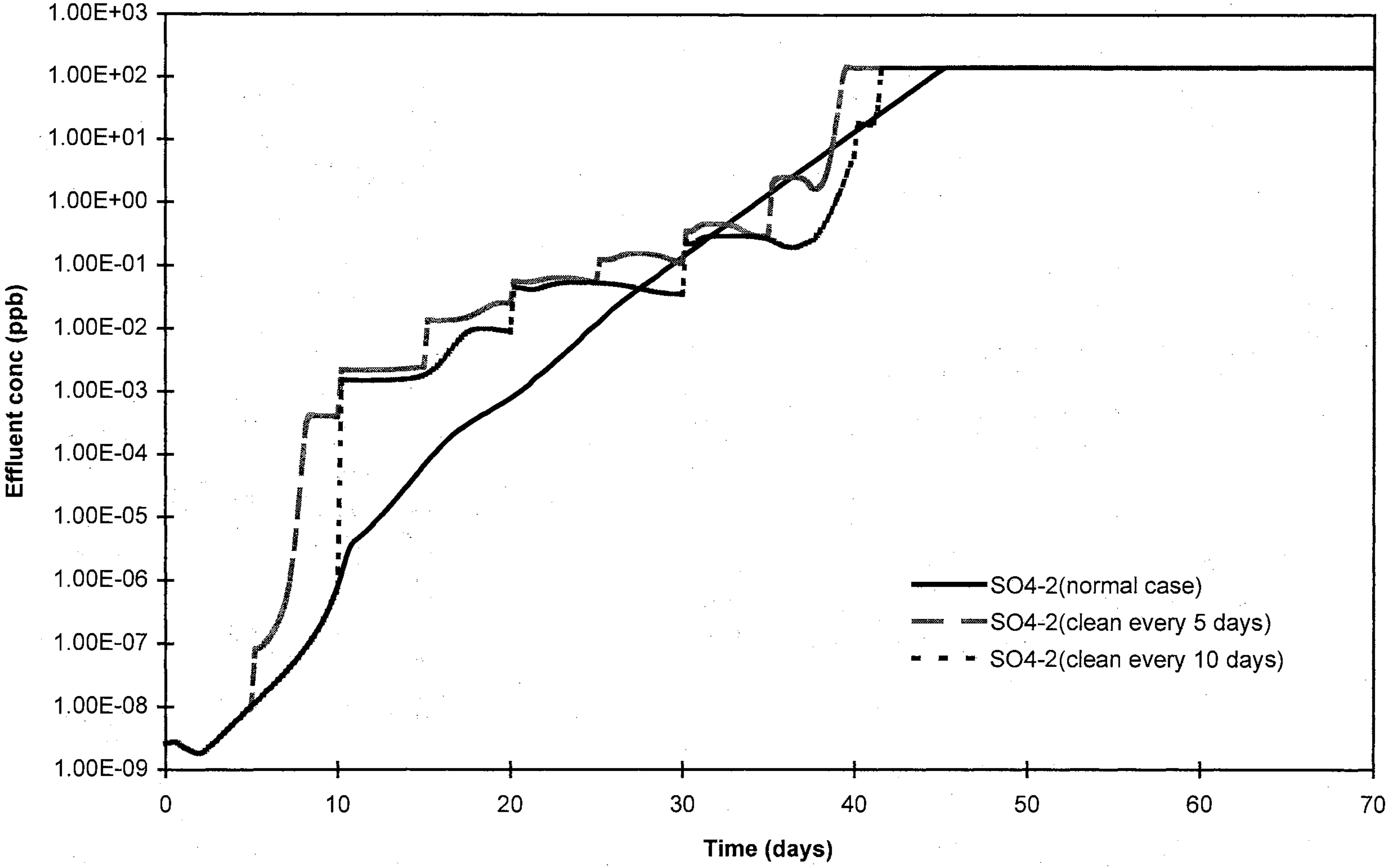


Figure 15. Effect of bed cleaning on $SO_4^{-2}$ breakthrough curve

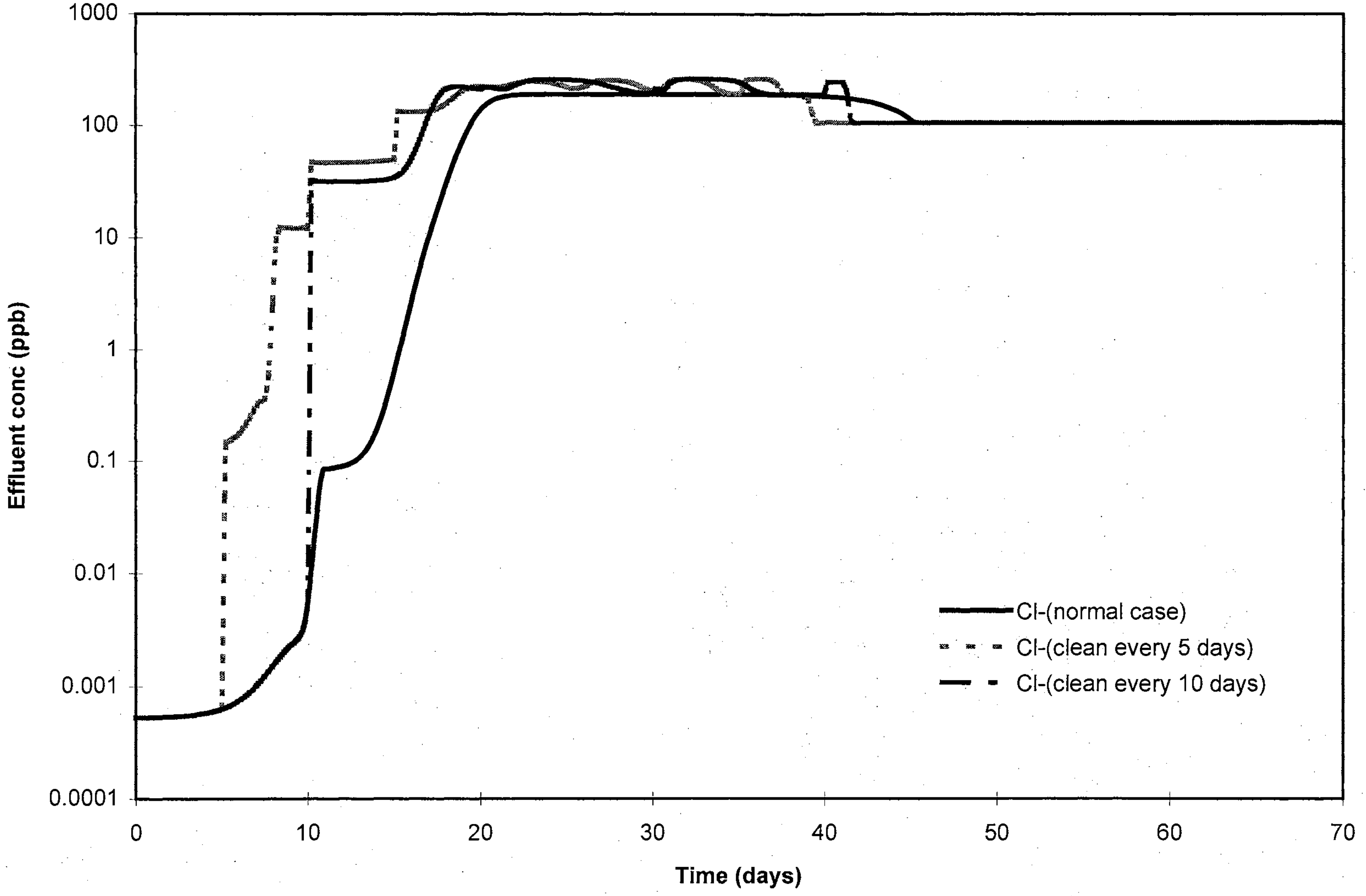


Figure 16. Effect of bed cleaning on Cl⁻ breakthrough curve

## Conclusions

In this chapter, special cases of multicomponent mixed-bed ion exchange (MBIE), such as variable inlet concentrations, variable inlet flowrate, layered-bed, and bed cleaning were studied. Appropriate methods were proposed to handle these cases. Specifically, different algorithms have been developed and implemented in the computer code, and simulations showed satisfactory results.

Due to the complexity of implementing these algorithms into the computer code, the simulator can only handle these cases independently. In addition, only step-change in inlet concentrations or inlet flowrate is handled by the model. Therefore, further work to extend the model to more complex cases; such as more than one step-change in inlet concentration or inlet flowrate, is required. Moreover, fine tuning the computer code to handle all above cases simultaneously is also required.

# CHAPTER V

# MODELING OF MULTICOMPONENT MIXED-BED ION EXCHANGE WITH SILICA

## Abstract

Silica is a major concern of the high-purity-water-users like power plants and semiconductor manufacturers. In this chapter, a rate model is developed to predict the breakthrough behavior of multicomponent mixed-bed ion exchange (MBIE) with silica. The silicic species considered in the model include monosilicic acid, monovalent silicate, divalent silicate, polymeric silicate, colloidal silica and particulate silica. Several possible transport mechanisms such as ion exchange, protonation and physical adsorption have been taken into account to model the removal of silica by the ion exchange resins. Model predictions showed satisfactory agreement with the plant observations.

## Introduction

Silica is one of the most abundant materials on earth. Yet not much is known about its chemistry, and particularly, its solubility behavior in water. The term silica refers to silicon in natural waters, and is usually represented by the hydrated form of the oxide: $H_4SiO_4$ or $Si(OH)_4$ – silicic acid, which is classified as a nonionic species, because it is a weak acid, and does not protolyze until the pH of water exceeds 9 (Iler, 1979).

There are three kinds of silica that are most commonly known: soluble, colloidal, and particulate. Soluble silica is also referred to as reactive silica, because it reacts rapidly with molybdate to form a heteropoly-acid that can be reduced to form a blue color; while colloidal silica is called non-reactive or polymeric silica, since it reacts very slowly with molybdate. The particulate silica is also known as suspended or granular silica. As most particulate silica can be removed by pretreatments like ultra-filtration, or reverse osmosis, efforts will be stressed on the modeling of first two kinds of silicic species (reactive and non-reactive silica) in this work. However, the amount of particulate silica can also be predicted using this model if the total silica amount is given.

Silica, whether colloidal or soluble, is a major concern of end-users producing high purity water. Power plants seek to stop or minimize deposition of silica on expensive equipment like turbines, since the deposited silica may hurt the generator efficiency and even cause other problems, such as underdeposit corrosion, blade failure, and corrosion fatigue (Pate, 1991). Meanwhile, semiconductor manufacturers concern that precipitates from silica will cause defective wafers, because the dissolved silica can affect the thermal oxidation on semiconductor wafer surface. The pharmaceutical industry is also sensitive to the amount of silica in water, since excessive silica may cause potential formation of siliceous precipitates, which are detrimental to pharmaceutical applications.

There are several ways to remove silica from water. For reactive (soluble) silica control, strong-base anion exchange at the makeup demineralizer is a common approach. A standard method for colloidal silica removal is the use of coagulation and precipitation of suspended solids, followed by mixed-media filtration (Harfst, 1992). In addition, reverse osmosis and ultrafiltration are also used to eliminate colloidal silica from water, but they cost more.

Both soluble (reactive) and colloidal (non-reactive) silica can be removed from water to varying low levels in neutral or slightly alkaline solution by coprecipitation with insoluble metal hydroxides, or by adsorption upon freshly formed hydroxides added into the water (Iler, 1979). However, for feed water of high pressure steam power plants, precipitation processes are served only as the initial step to reduce silica concentration to the level of parts-per-million (ppm) range. To further reduce the amount of silica in water to parts-per-billion (ppb) level, ion exchange process has to be employed. The strong-base type anionic resins are usually used to adsorb the soluble silica as silicate ions, however, it was reported that colloidal silica can also be adsorbed to the resins (Kun and Kunin, 1966; Drucker and Dale, 1988).

Mixed-bed ion exchanger, in which cationic and anionic resins are mixed together is currently widely used in ultrapure water processing for power and semiconductor industries. In addition to removing cations and anions in water, the silica impurities can also be eliminated to a low level (Arden, 1968). Although mixed-bed ion exchange has been used for silica removal in practice for a long time, very little is known about its mechanisms in removing silica; in addition, there is nearly no modeling work on mixed-bed ion exchange with silica reported in literature up to now. Therefore, modeling multicomponent mixed-bed ion exchange with silica is of great significance and challenging. It is the purpose of this work to develop a model to handle multicomponent mixed-bed ion exchange with silica. To achieve this goal, the water chemistry of silica, and transport mechanisms of silica into resin need to be reviewed first.

## Water Chemistry of Silica

### Solubility of Silica in Water

There are innumerable mineral sources of silica in natural waters. The greatest sources of silica are the feldspars, micas and clay minerals, or the hydrous aluminum silicates. For the most part, the solubility of quartz or amorphous silica determined its concentration in water. At 25 °C the solubility of quartz is small, only around 6.0 mg/l; whereas the solubility of amorphous silica is reported to be 115 mg/l (Faust and Aly, 1981). These solubility values may increase with the increase of water temperature or pH value.

The dissolution of $SiO_2$ (s) – solid silica, in water is not completely understood, since the dissolution of silica in water is complex. Siever (1962) thought that the dissolution of silica may be represented by the following reaction

$$SiO_2\,(s) + 2\,H_2O \Leftrightarrow H_4SiO_4 \qquad \text{(V-1)}$$

For silica glass (amorphous silica), the equilibrium constant corresponding to this reaction is

$$K_{eq} = \frac{[H_4SiO_4]}{[SiO_2(s)]} = 10^{-2.77} \qquad \text{(V-2)}$$

However, for silica quartz, the equilibrium constant is

$$K_{eq} = \frac{[H_4SiO_4]}{[SiO_2(s)]} = 10^{-3.76} \tag{V-3}$$

where $H_4SiO_4$ is known as silicic acid. It is easy to see that the amorphous form of silica (silica glass) is more soluble in water than its crystalline (quartz) counterpart. The same conclusion had also been made by Alexander et al. (1954).

Rimstidt and Barnes (1980) studied the hydration of amorphous silica. They measured its solubility in pure water. For the reaction equilibrium of

$$SiO_2\,(s) + 2\,H_2O \Leftrightarrow H_4SiO_4 \tag{V-4}$$

(amorphous silica)

$$\frac{[H_4SiO_4]}{[SiO_2(s)]} = 10^{-k_s} \tag{V-5}$$

they proposed, for temperatures between 0 and 300 °C, the following relationship

$$-k_s = 0.3380 - 7.889 \times 10^{-4}\,T - 840.1/T \tag{V-6}$$

to correlate the dissociation constant of amorphous silica with temperature. Where $k_S$ is the hydration constant (or solubility constant), $T$ the temperature (K). Specifically, $k_S$ = 2.716 at $T$ = 298 K (25 °C ). It should be noted that the above solubility assumes excessive solid (particulate) silica exist. Hence the amount of $Si(OH)_4$ is pH independent in this case.

However, monosilicic acid never exists alone. When this very weak acid species exists in undersaturated silica solutions, it has high tendency of polymerization; that is, when the solubility threshold is exceeded, monosilicic acid immediately polymerizes, which results in polymeric silica or colloids, and finally evolves into silica gel structures.

Van Lier (1965) studied the dissolution of quartz in water. He found that cleaned quartz has a well defined solubility in water at autogenous pressure, which can be expressed by the following equation:

$$\log C = 0.151 - 1162 / T \qquad \text{(V-7)}$$

where $C$ is the molar concentration of $Si(OH)_4$, and $T$ is absolute temperature ($K$).

For the solubility of amorphous silica, Alexander et al. (1954) also studied it experimentally. They indicated that the solubility of amorphous silica at pH 7 ~ 8 is between 100 and 150 ppm. In addition, the work of Goto et al. (1953) further indicated that the solubility of amorphous silica is constant from pH 2 to 8, and then it increases rapidly at higher pH. Specifically, when pH $>$ 11, silica is completely soluble, that is, all solid phase of amorphous silica dissolves to form soluble silicate, since at higher pH the concentration of $Si(OH)_4$ is greatly lowered by conversion to ionic species, so that no amorphous solid can remain in equilibrium.

Iler (1979) studied the dissolution of amorphous silica in water too. They indicated that the solubility of amorphous silica under constant pressure can be expressed by the following expression:

$$\log C = 4.82 - 810 / T \qquad \text{(V-8)}$$

where $C$ = solubility of silica, mg $SiO_2$/kg (or ppm)

$T$ = absolute temperature, $K$

Moreover, they also found, from pH 9 to 10.7, there is an apparent increase in the solubility of amorphous silica, which, they thought, is due to the formation of silicate ion $H_3SiO_4^-$ in addition to neutral monomer molecular $Si(OH)_4$.

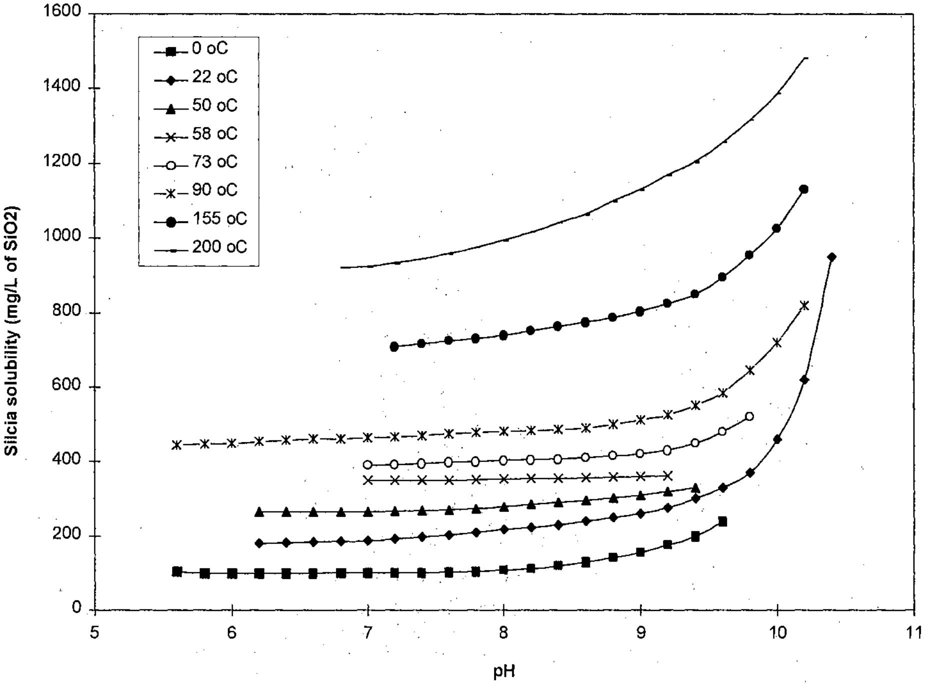


Figure 1. Solubility of amorphous silica versus pH at different temperature

The effect of pH on the solubility of silica (amorphous) at temperature rang of 0 ~ 200 °C was investigated by Goto (1955). He found that the solubility of amorphous silica was about 30% higher than that of most powders and gels. Figure 1 was generated based on his experimental study. From this chart, we can see that both temperature and pH influence the solubility of silica. However, at lower pH (< 8.5), the solubility depends only on temperature.

Ionization of Silicic Acid

The ionization of silicic acid is a complicated and multi-step process. One important factor influencing the ionization of silicic acid is the pH of solution. Usually, the following ionization reactions are believed to take place (Iler, 1979):

i) At 9 < pH < 10.7, silicic acid will ionize first to form monovalent silicate

$$\mathrm{Si(OH)_4} \longleftrightarrow \mathrm{H_3SiO_4^-} + \mathrm{H^+} \tag{V-9}$$

$$\frac{[\mathrm{H_3SiO_4^-}][\mathrm{H^+}]}{[\mathrm{Si(OH)_4}]} = 10^{-k_{d1}} \tag{V-10}$$

ii) Furthermore, at pH > 10.7, monovalent silicate ionizes into divalent silicate

$$\mathrm{H_3SiO_4^-} \longleftrightarrow \mathrm{H_2SiO_4^-} + \mathrm{H^+} \tag{V-11}$$

$$\frac{[\mathrm{H_2SiO_4^-}][\mathrm{H^+}]}{[\mathrm{H_3SiO_4^-}]} = 10^{-k_{d2}} \tag{V-12}$$

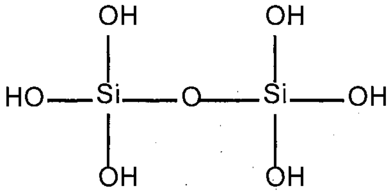
Since the solution is dilute, the unit activity is assumed and concentrations rather than activities for all species are used in the above equilibrium expressions. For these two ionization constants, Stumm et al. (1967) reported the values of $k_{d1}$ = 9.46, and $k_{d2}$=12.56 at 25 °C.

Polymerization of Silicic Acid

As mentioned before, monoorthosilicic acid has a very strong tendency to polymerize in solution. This polymerization involves spontaneous "dehydrating condensation" reactions, which can be represented by

$$2\mathrm{Si(OH)_4} \rightarrow \mathrm{Si_2O(OH)_6} + \mathrm{H_2O} \tag{V-13}$$

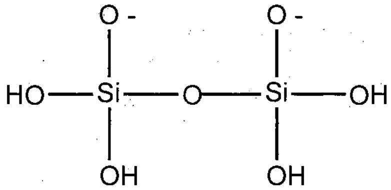
The first product is a disilicic acid $Si_2O(OH)_6$ which has the form (Ingri, 1959) of



Ionization of this acid yields the orthodisilicate ion, $Si_2O_3(OH)_4^{2-}$, with the equilibrium

$$\frac{[Si_2O_3(OH)_4^{2-}][H^+]^2}{[Si(OH)_4]^2} = 10^{-k_{p2}} \qquad \text{(V-14)}$$

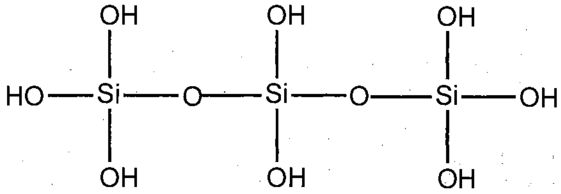
in which $Si_2O_3(OH)_4^{2-}$ has the structure of



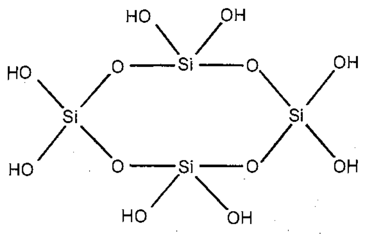
The equilibrium constant for this acid is $k_{p2} = 18.2$ (Lagerstrom, 1959). However, condensation reactions favor formation of siloxane bonds (Si-O-Si) inside the molecules and silanol groups (Si-OH) on the outside, which leads to form trisilicic acid $Si_3O_2(OH)_8$

$$Si_2O(OH)_6 + Si(OH)_4 \rightarrow Si_3O_2(OH)_8 + H_2O \qquad \text{(V-15)}$$

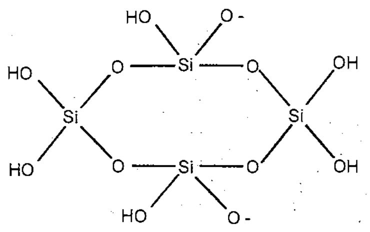
This acid is linear, and has the structure of



Moreover, condensation to form polymers higher than trisilicic acid can also take place at interior silanol groups, thus leading to cyclic structure, such as the tetrasilicic acid $Si_4O_4(OH)_8$ (Ingri, 1959). The structure of tetrasilicic acid $Si_4O_4(OH)_8$ is



This acid will also ionize to form the tetrasilicate ion



with the ionization equilibrium reaction

$$Si_4O_4(OH)_8 \longleftrightarrow Si_4O_6(OH)_6^{-2} + 2H^+ \tag{V-16}$$

That is

$$\frac{[Si_4O_6(OH)_6^{2-}][H^+]^2}{[Si(OH)_4]^4} = 10^{-k_{p4}} \tag{V-17}$$

Langestrom (1959) reported the equilibrium constant $k_{p4} = 12.57$ at 25 °C in the media of perchloric acid. But Ingri (1959) gave a value of 12.37 at 25 °C in 0.5 M NaCl. In this work, the value proposed by Ingri (1959) is used to develop the model.

Tetrasilicic acid is also easy to form three-dimensional structure under condensation, and becomes insoluble rapidly. In this case, the solution becomes cloudy and enters the domain of colloidal silica. Electrophoreses study shows that colloidal silica is negative charged (Iler, 1955) although it is insoluble. Therefore, it is believed that part of the colloidal silica in the solution can also be removed by ion exchange resin.

For better reference, the hydration and ionization constants for amorphous silica are summarized in Table I, which are the values at 25 °C. For values at other temeperatures, they are currently not available in literature.

TABLE I. Silica Ionization and Hydration Constants

| Authors | t (°C) | $k_s$ | $k_{d1}$ | $k_{d2}$ | $k_{p2}$ | $k_{p4}$ | Media |
|---|---|---|---|---|---|---|---|
| Stumm et al. (1967) | 25 | 2.716 | 9.46 | 12.56 | | | $NaClO_4$ (0.5 M) |
| Langerstrom (1959) | | | | | 18.12 | | $NaClO_4$ ($\mu$=3) |
| Ingri (1959) | | | | | | 12.37 | NaCl (0.5 M) |

To determine the proportions of different silicic species in solution and their changes with pH, the dissociation constants provided by Stumm et al. (1967) in Table I were used in calculation.

Rewriting the Eq (V-10) and (V-12), we have

$$[H_3SiO_4^-] = [H_4SiO_4]10^{pH-9.46} \quad \text{(V-10A)}$$

$$[H_2SiO_4^{-2}] = [H_4SiO_4]10^{2pH-22.02} \quad \text{(V-12A)}$$

If neglecting polymeric silicic species, the proportions of different silicic species as a function of pH can be determined by

$$Si(OH)_4\% = \frac{100}{1+10^{pH-9.46}+10^{2pH-22.02}} \quad \text{(V-18)}$$

$$H_3SiO_4^-\% = \frac{100\times 10^{pH-9.46}}{1+10^{pH-9.46}+10^{2pH-22.02}} \quad \text{(V-19)}$$

$$H_2SiO_4^{2-}\% = \frac{100\times 10^{2pH-22.02}}{1+10^{pH-9.46}+10^{2pH-22.02}} \quad \text{(V-20)}$$

Figure 2 is a plot of the above relationships. It can be seen that below pH = 8, most of the silicic species is in the form of silicic acid; starting from pH = 8, monosilicic acid begins to ionize to form monovalent silicate ion; starting from pH = 10, it begins to ionize divalent silicate ion.

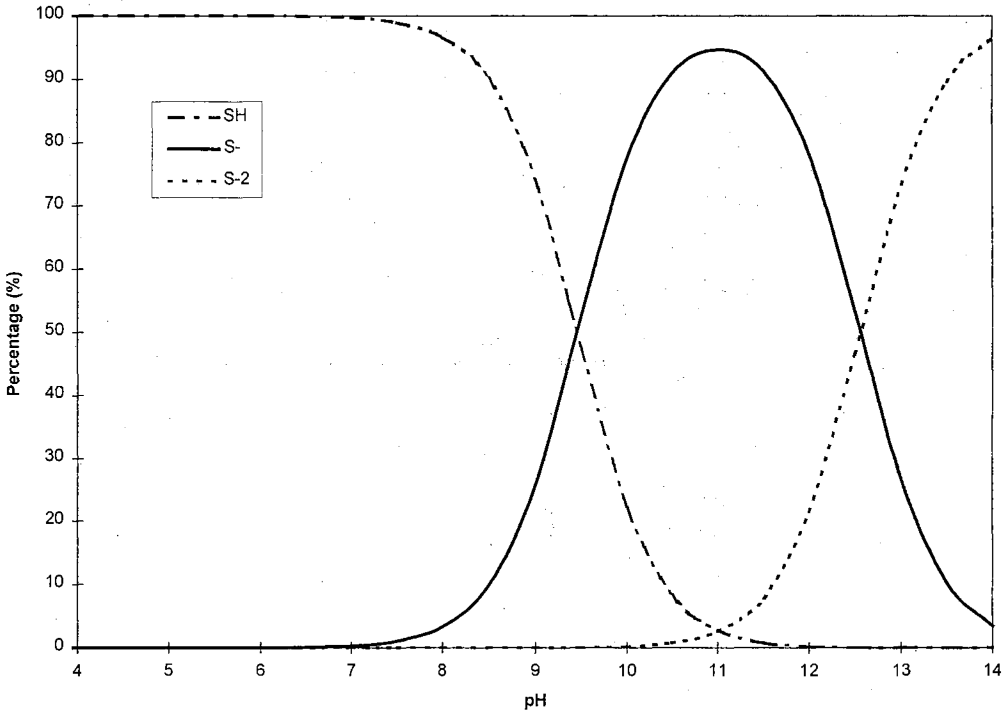


Figure 2. Distribution of various silicic species as a function of pH

## Transport of Silica into Resin

As we know, ion exchange is the principal method currently used in producing very low-silica feed water for high-pressure boilers (Henley, 1992). Figure 3 is a typical setup of demineralization system used in industry for silica removal. From this chart, we can see that mixed-bed is usually placed at the end of the deionization system, which can remove ionic species (including various silica species) further; in addition, it also serves as the protector to guard against accidental sodium leaks and errors in anion resin regeneration. It should be noted that in this schematic, most of silicic species are removed by the strong-base anionic resin bed (which is usually put in front of mixed-bed). But it is the mixed-bed that further reduces the amount of silica in solution to less than 0.2 ppb (Arden, 1968).

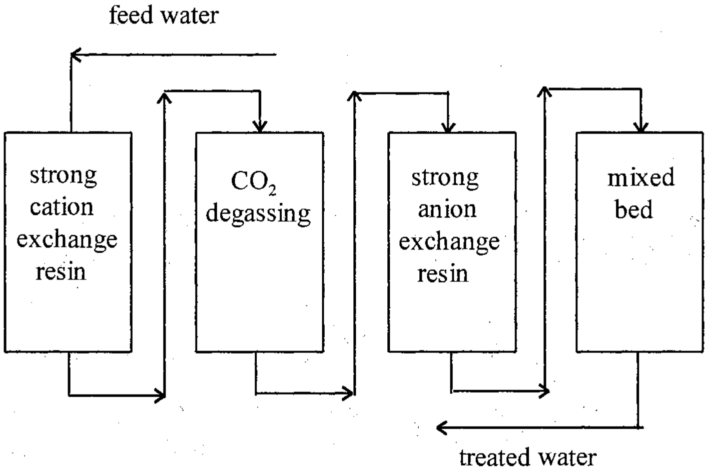


Figure 3. Schematic diagram of silica removal by ion exchange process

Since the objective of this work is to model mixed-bed ion exchange with silica, description of the transport of silicic species in mixed-beds is our major concern. However using this model, strong-base anionic bed can also be simulated, since mono bed is only a special case of mixed-bed. What we need to do is letting the FCR (cationic resin fraction) be a small value, say, 0.000001 (but never 0 to avoid the mathematical error of divided by zero) in doing anion bed simulations.

In a mixed-bed ion exchange column, even though the silica solution is neutral, it may contain charged species such as monovalent silicate ion, divalent silicate ion, and polymeric silicates, in addition to molecular form species of silicic acid $Si(OH)_4$. The existence of various silicic species depends on the pH of solution. Usually, when pH is between 7 ~ 8, only silicic acid and colloidal silica exist; but when pH is greater than 9, $Si(OH)_4$ will ionize to form monovalent silicate $H_3SiO_4^-$; furthermore, when pH increases to 11, divalent silicate ion $H_2SiO_4^{2-}$ will be produced.

When the solution of silica is passed through a mixed-bed with strong-base anion- and strong acid cation-exchange resins, the monovalent and divalent silicate ions will be removed by resins as counterions. In addition, silicic acid (molecular form of weak electrolytes) will also be adsorbed to resin in the mechanism similar to the protonation of amine or in the mechanism of physical adsorption.

Many other researchers (Kun and Kunin, 1966; Drucker and Dale, 1988) also indicated that the nonionic forms of silica including $Si(OH)_4$ and colloidal can also be adsorbed to resins via physical adsorption. Therefore, in mixed-bed ion exchange column, silica can be removed by ion exchange resin in the following ways:

- all soluble silicate ions are adsorbed to resin by ion exchange reactions;
- molecular form silica $Si(OH)_4$ are adsorbed to resin in the mechanisms of protonation and physical adsorption;
- colloidal silica are adsorbed to resin by physical adsorption only.

## Model Development

### Assumptions

Like the model developed in Chapter III, the major assumption of this model is that ion exchange is film-diffusion controlled, so that the Nernst-Planck equation can be used to describe the process. This is based on the characteristics of ultra-low concentrations in MBIE column. The other important assumptions are the bulk phase neutralization, and local equilibrium between solid-liquid interface. As silica has a very strong tendency of polymerization, polysilicate ions are also considered in the model. However for the simplicity of mathematics, it is assumed that tetrasilicic acid ion, $Si_4O_6(OH)_6^{2-}$ is the only polymeric species that is in non-negligible amount (Stumm, 1967), and it is incorporated in the model development. Besides above, the following transport mechanisms for silica are assumed:

- All ionic silica are adsorbed to resin by ion exchange reactions;
- Molecular form silica are adsorbed to resin by protonation and physical adsorption;
- Colloidal silica is adsorbed to resin by physical adsorption only.

Table II lists all the assumptions employed in this model.

TABLE II. Model Assumptions

---

1. Process is film diffusion controlled.
2. The Nernst-Planck model incorporates all interactions between diffusing ionic species.
3. Molecular form of silica $Si(OH)_4$ is adsorbed into the resin by two mechanisms – protonation and physical adsorption.
4. Tetrasilicic acid ion, $Si_4O_6(OH)_6^{2-}$, is the only polymeric species existed in solution in non-negligible amount.
5. Colloidal and particulate silica are adsorbed to resin in the mechanism of physical adsorption.
6. Fick's law is used to model the transfer rate of various silica species except silica ions.
7. The interfacial concentrations of nonionic silicic species are determined by the Freundlich law.
8. For ion exchange process, an ion with a higher valence is preferred to an ion with a lower valence by the ion-exchange resin.
9. Between the ions with same valence, the ion with higher selectivity is preferred to the ion with lower selectivity by the ion-exchange resin.
10. Pseudo steady state exchange (variations of concentration with space are much more important than with time).
11. No net current flow and no net coion flux within the film.

12. No coion flux across the particle surface.
13. Local equilibrium at solid - film interface.
14. Curvature of the film is negligible.
15. Selectivity coefficients are constant and temperature independent.
16. Binary selectivity coefficients can be used for multicomponent ion exchange.
17. Bulk phase neutralization.
18. Uniform concentrations across the resin bead.
19. Uniform bulk phase concentrations.
20. Ion exchange reactions are instantaneous compared to the overall rate of exchange.
21. Activity coefficients are constant and unity.
22. Negligible axial dispersion and plug flow.
23. Isothermal, isobaric operation.

---

## Model Equations

### *1) Silica Equilibrium*

In addition to reactive and non-reactive (colloidal) silica, particulate silica is also considered in the model development. For simplification, we define the following symbols to represent various silica species

ST ---- Total silica in the solution, $SiO_2$ (total)

SP ----- Particulate silica (nonsoluble)

SC ---- Colloidal silica

SH ---- Molecular silica, silicic acid $Si(OH)_4$

$S^-$ ---- Monovalent silicate ion, $H_3SiO_4^-$

$S^{2-}$ ---- Divalent silicate ion, $H_2SiO_4^{2-}$

$PS^{2-}$ --- Polysicate, in the form of polymer $Si_4O_6(OH)_6^{2-}$

Obviously, the total silica balance should be

$$[ST] = [SP]+[SC]+[SH]+[S^-]+[S^{2-}]+[PS^{2-}] \qquad \text{(V-21)}$$

and the total soluble silica can be determined by Eq (V-6), that is

$$[SH]+[S^-]+[S^{2-}]+[PS^{2-}]=10^{-ks} \qquad \text{(V-22)}$$

From Eq (V-10) and (V-12), the silicate ions can be expressed as

$$[S^-] = 10^{-k_{d1}} \frac{[SH]}{[H^+]} \qquad \text{(V-23)}$$

$$[S^{2-}] = 10^{-(k_{d1}+k_{d2})} \frac{[SH]}{[H^+]^2} \qquad \text{(V-24)}$$

and From (V-17) the colloidal silica can be expressed as

$$[PS^{2-}] = 10^{-k_{p4}} \frac{[SH]^4}{[H^+]^2} \qquad \text{(V-25)}$$

Substituting Eq (V-23) through (V-25) into Eq (V-22), and using the definition of

$$[H^+] = 10^{-pH} \qquad \text{(V-26)}$$

yields

$$10^{-k_s} = (1+10^{pH-k_{d1}}+10^{2pH-(k_{d1}+k_{d2})})[SH]+10^{2pH-k_{p4}}[SH]^4 \qquad \text{(V-27)}$$

This is a polynomial regarding [SH], we can use Newton-Raphson method to solve it as long as the pH (or $[H^+]$) of solution is given.

Let's assume the solution of Eq (V-27) is

$$[SH] = f([H^+]) \qquad \text{(V-28)}$$

Back substituting the value of [SH] from Eq (V-28) into Eq (V-23) through (V-25), then the concentrations of [S$^-$], [S$^{2-}$], [PS$^{2-}$] can be determined. Therefore, all the concentrations of [SH], [S$^-$], [S$^{2-}$] and [PS$^{2-}$] are dependent on the pH of solution. To determine the pH of solution, charge balance equation has to be solved, which is

$$\sum \mathrm{Cat}^+ + [\mathrm{H}^+] = \sum \mathrm{Ani}^- + [\mathrm{S}^-] + [\mathrm{S}^{2-}] + [\mathrm{PS}^{2-}] + [\mathrm{OH}^-] \qquad \text{(V-29)}$$

where $\sum \mathrm{Cat}^+$ = summation of concentrations of other cations except $H^+$ in solution

$\sum \mathrm{Ani}^-$ = summation of concentrations of other anions except silica and $OH^-$ in solution

Substituting Eq (V-23) through (V-26) into Eq (V-29), and using the following relation

$$[\mathrm{OH}^-] = \frac{\mathrm{K_w}}{[\mathrm{H}^+]} = \mathrm{K_w} 10^{\mathrm{pH}} \qquad \text{(V-30)}$$

yields

$$\sum \mathrm{Cat}^+ + 10^{-\mathrm{pH}} = \sum \mathrm{Ani}^- + 10^{\mathrm{ph}-(k_s+k_{d1})}[\mathrm{SH}] + 10^{2\mathrm{pH}-(k_s+k_{d1}+k_{d2})}[\mathrm{SH}]$$

$$+ 10^{2\mathrm{pH}-k_{p4}}[\mathrm{SH}]^4 + \mathrm{K_w} 10^{\mathrm{pH}} \qquad \text{(V-31)}$$

Eq (V-31) has only one unknown pH, since $[\mathrm{SH}] = f([\mathrm{H}^+]) = f(\mathrm{pH})$. So we can solve it by using an iterative method.

With the Eq (V-31) solved, the concentrations of [SH], [S$^-$], [S$^{2-}$] and [PS$^{2-}$] can be calculated. If the total silica concentration [ST] is given, and the colloidal silica concentration [SC] also given by a percentage of the total, then the particulate silica concentration is determined by

$$[\mathrm{SP}] = [\mathrm{ST}] - ([\mathrm{SC}] + [\mathrm{SH}] + [\mathrm{S}^-] + [\mathrm{S}^{2-}] + [\mathrm{PS}^{2-}]) \qquad \text{(V-32)}$$

2) *Interfacial Concentrations*

For ionic species, the interfacial concentrations are calculated from the selectivity expression based on the local equilibrium at solid-liquid interface.

$$X_i^* = Y_i \left(K_A^i\right)^{-1/Z_A} \left(\frac{X_A^*}{Y_A}\right)^{Z_i/Z_A} \left(\frac{Q}{C_T^*}\right)^{1-Z_i/Z_A} \tag{V-33}$$

For nonionic form species, the interfacial concentrations are determined by the Freundlich isotherm, which is in the form of

$$X_i^* = KC^{\frac{1}{n}} \tag{V-34}$$

or

$$X_i^* = \left(\frac{1}{a_1}C\right)^{\frac{1}{e_1}} \tag{V-34A}$$

where K, $a_1$ = Freundlich adsorption coefficient,

n, $e_1$ = Freundlich adsorption index.

The values of K and n (or $a_1$ and $e_1$ ) used in the model for various silica are given in the Table III.

TABLE III. Freundlich Adsorption Coefficients and Index for Various Silica

| Name of Species | Freundlich Coefficient | Freundlich Index |
|---|---|---|
| Silicic acid (SH) | $K = 1.0$ | $n = 3.0$ |
| Colloidal silica (SC) | $a_1 = 0.5$ | $e_1 = 0.2$ |
| Particulate silica (SP) | $a_1 = 3.0$ | $e_1 = 0.12$ |

3) *Flux Expression*

The ionic flux is developed from the Nernst-Planck equation. Here the ionic flux expression developed by Franzreb et al. (1993) for multicomponent systems (see Chapter III) is implemented in the model, which is

$$J_i = \frac{D_i}{\delta}\left( (1-\frac{N_i}{P})(C_i^* - C_i^o) + N_i A_i (1+\frac{1}{P})(C_T^* - C_T^o) \right) \qquad \text{(V-35)}$$

The detailed derivation of this expression is shown in Appendix C.

However for the flux of non-ionic species, Fick's law is used to describe it. The flux expression for nonionic form species is

$$J_{i,\mathrm{Nonion}} = \frac{D}{\delta}(C_i^* - C_i^o) = k_{L,\mathrm{Nonion}}(C_i^* - C_i^o) \qquad \text{(V-36)}$$

where $k_{L,\mathrm{Nonion}}$ is the mass transfer coefficient of the nonionic form.

4) *Particle Rate*

The rate of exchange is related to the flux of species by

$$\frac{\partial q_i}{\partial t} = -J_i a_s \qquad \text{(V-37)}$$

As the resin phase concentration $q_i$ can be represented by

$$q_i = y_i Q \qquad \text{(V-38)}$$

Equation (V-37) can be rewritten as

$$\frac{\partial y_i}{\partial t} = \frac{-J_i a_s}{Q} \qquad \text{(V-39)}$$

In this work, the mass transfer rate is calculated separately for each kind of silicic species:

(i) Rate of ionic silica ($S^-$, $S^{2-}$, $PS^{2-}$)

For ionic species, the rate is calculated using Eq (V-39), in which the flux $J_i$ is calculated from Eq (V-39), that is

$$\frac{\partial y_i}{\partial t} = -\frac{D_i}{\delta}\left( (1-\frac{N_i}{P})(C_i^* - C_i^o) + N_i A_i (1+\frac{1}{P})(C_T^* - C_T^o) \right)\frac{a_s}{Q} \qquad \text{(V-40)}$$

(ii) Rate of silicic acid (SH)

This rate includes two parts --- protonation rate and adsorption rate.

$$\frac{\partial q_i}{\partial t} = k_{L,Mol} a_s (C^*_{Mol,prot} - C^o_{Mol,prot}) + k_{L,Mol} a_s (C^*_{Mol,sorp} - C^o_{Mol,} \quad ) \tag{V-41}$$

(iii) Rate of colloidal silica (SC)

Colloidal silica going to the resin mainly in the mechanism of adsorption

$$\frac{\partial q_{SC}}{\partial t} = k_{L,SC} a_s (C^*_{SC,sorp} - C^o_{SC,sorp}) \tag{V-42}$$

(iv) Rate of particulate silica (SP)

$$\frac{\partial q_{SP}}{\partial t} = k_{L,SP} a_s (C^*_{SP,sorp} - C^o_{SP,sorp}) \tag{V-43}$$

5) *Column material balance*

The column material balance determines the concentration profile of the ion exchange column and estimates the outlet concentration for each ionic species. The column material balance equation is in the following form

$$\frac{u_s}{\varepsilon} \frac{\partial C_i}{\partial Z} + \frac{\partial C_i}{\partial t} + \frac{(1-\varepsilon)}{\varepsilon} \frac{\partial q_i}{\partial t} = 0 \tag{V-44}$$

in which the axial dispersion is neglected.

## Simulation Results and Discussion

The model developed above was used to predict the breakthrough behavior of multicomponent mixed bed ion exchange with silica. Several practical cases from industry were simulated. In addition to predicting the effluent concentrations of silicic acid and monovalent silicate ion, this model can also predict the amount of molecular, colloidal and particulate silica that are adsorbed into the resin and their breakthrough time. The following are the detailed simulation results.

### Case I. Intel's Ireland Plant

The MBIE column operated at Intel's new plant at Ireland has dimensions of 152.4 cm in diameter and 91.6 cm in depth with void fraction of 0.35. The flow rate is 210 gpm. The operation temperature is 25 °C, and the estimated value of influent pH is 6.32. The influent concentrations for various species in the system are listed in Table IV. The resin properties and selectivities are summarized in Table V and VI.

TABLE IV. Influent Concentrations and Initial Loadings at Normal Operating Conditions

| Species | Influent Concentrations (ppb) | Initial loadings (%) |
|---|---|---|
| Cations: | | |
| $Ca^{2+}$ | 10 | 0.03 |
| $Na^{+}$ | 160 | 0.03 |
| $K^{+}$ | 20 | 0.03 |
| Anions: | | |
| $HCO_3^-$ | 320 | 0.1 |
| $Cl^-$ | 60 | 0.5 |
| NO3- | 80 | 0.5 |
| $SiO_2$ (total) | 20 | 0.1 |
| $CO_2$ | 260 | 0.1 |
| pH ( 25 °C) | 6.32 | |

TABLE V. Resin Properties

| | Anionic Resin | Cationic Resin |
|---|---|---|
| Name | Ambersep 440 | Ambersep 132 |
| Capacity (meq/ml) | 1.0 | 2.0 |
| Bead Diameter (cm) | 0.050 | 0.060 |
| Volume Fraction | 0.34 | 0.66 |

TABLE VI. Selectivities and Diffusivities for Species

| | Selectivities | Diffusivities ($cm^2/s \times 10^5$) |
|---|---|---|
| $Ca^{+2}$ | 3.9 | 0.792 |
| Na+ | 1.5 | 1.334 |
| $K^+$ | 2.6 | 1.957 |
| | | |
| $Cl^-$ | 22 | 2.032 |
| $NO_3^-$ | 65 | 1.902 |
| $HCO_3^-$ | 6 | 1.185 |
| $CO_3^{2-}$ | 12 | 0.923 |
| $S^-$ | 1.3 | 0.915 |
| $S^{-2}$ | 1.1 | 0.823 |

Using the data and parameters above, the following scenarios are simulated: (1) normal operating conditions (base case); (2) fouled resin effect; (3) temperature effect; (4) particle size effect; (5) inlet concentration effect; (6) flow rate effect; and (7) cation/anion ratio effect.

(1) Normal operating conditions (base case)

Figure 4 and Figure 5 are the predicted breakthrough curves for anions and cations at normal operating conditions. It can be seen that silica starts breakthrough in about 15.5 days; while sodium breaks through in about 25 days. These two predictions are closer to the plant experience of 14 days and 21 days respectively, although they are about 20% of overpredicting. The reason of overpredicting, we think, is probably caused

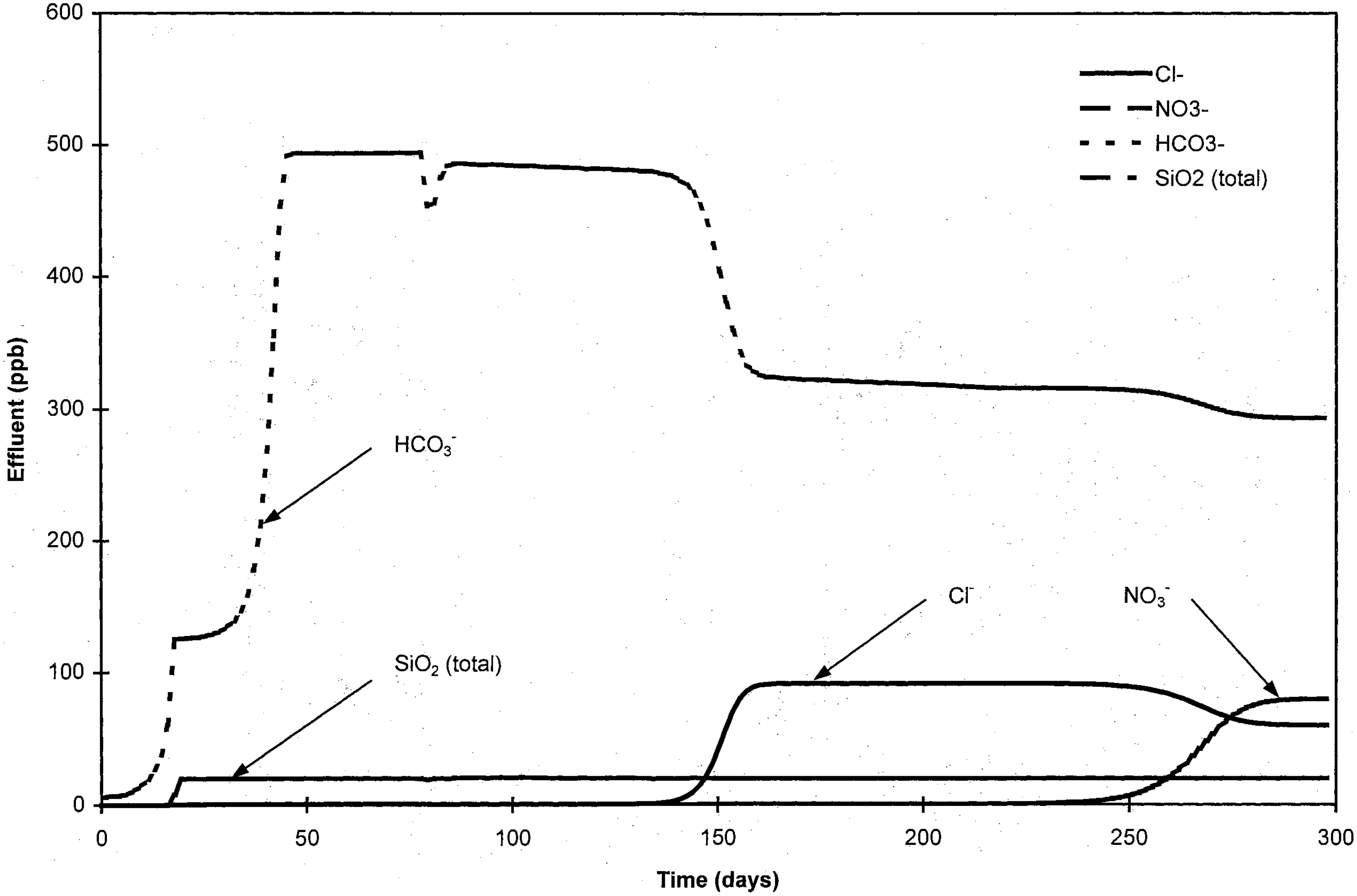


**Figure 4. Effluent history of anions for Intel's plant at Ireland**

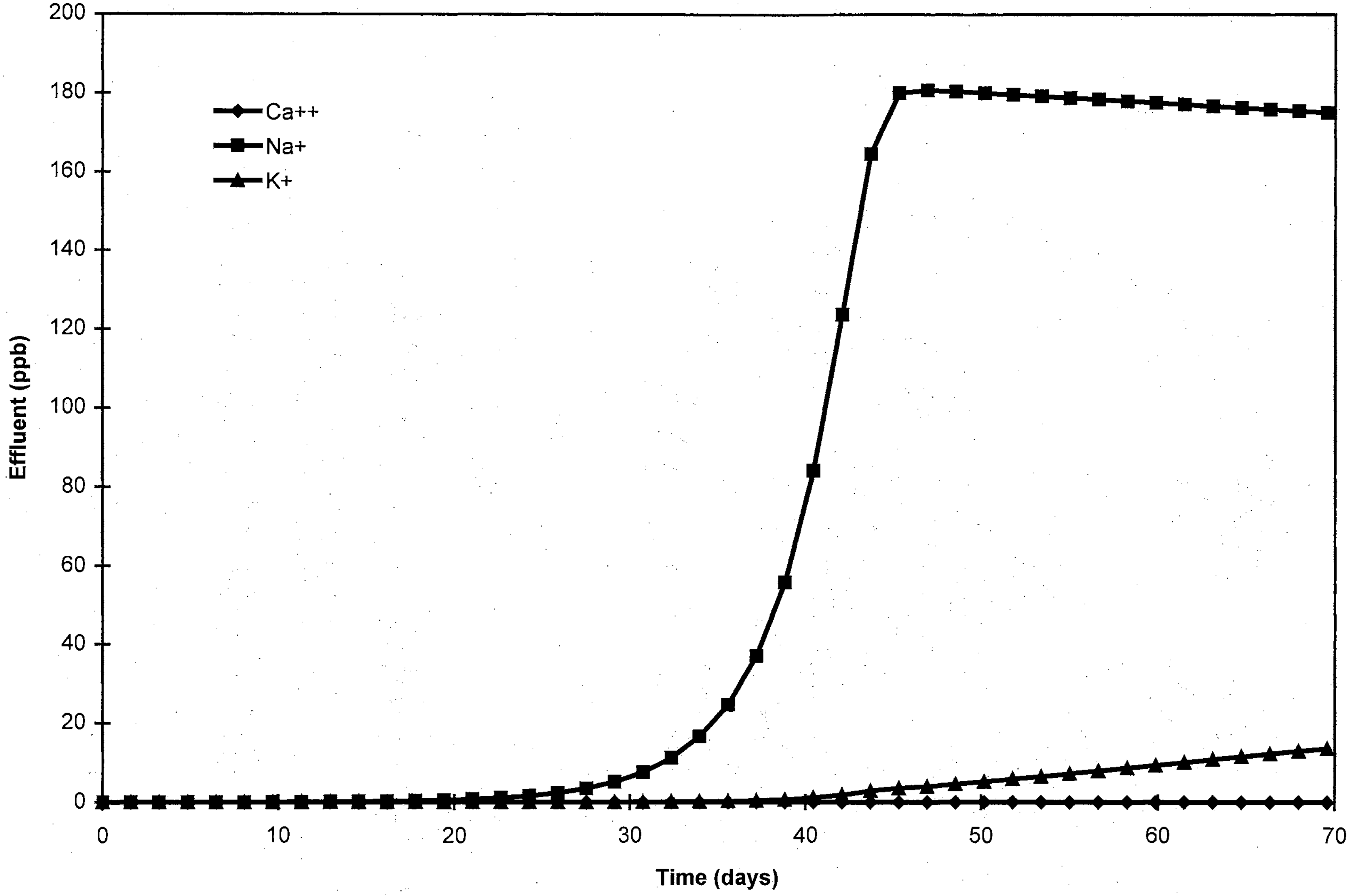


Figure 5. Effluent history of cations for Intel's plant at Ireland

by the system problem. Since in industry, the plant is difficult to maintain the flow rate consistently or if there are periods when flow is stopped or reduced.

As the dissociation and ionization of silica are pH dependent, various silicic species may exist in the system. Figure 6 are the predicted breakthrough curves for monovalent silicate, silicic acid and colloidal silica. From this plot, we can see that, for an input of 20 ppb of total silica, most of them are in the form of silicic acid, only small amount are monovalent silicate ion and colloidal silica. In addition, no particulate silica exists according to this prediction, and divalent silicates are also in trace amount. These results are also in agreement with the plant observations, and can be explained by the pH of solution, which is shown is Figure 7. Since pH < 8 during the exchange process, most of silica species should be in the form of silicic acid according to Figure 2.

(2) Fouled resin effect

The effect of fouled (or degraded) resin on silica breakthrough is shown in Figure 8. It is founded that when the anionic resin degraded or fouled, the time for silica to reach breakthrough tends to be earlier. These simulations were performed by multiplying the theoretical mass transfer coefficients of anions by a percentage called fouling factor. When the anionic resin is fouled by 30% (i.e. 70% of original MTC), the breakthrough of silica is about 13.5 days; when the resin is degraded by 50% (i.e. 50% of the original MTC), the breakthrough time for silica is around 11.5 days. These results are reasonable and easy to understand, since when the resin are fouled, the mass transfer rate between the bulk and resin phase is lower, thus leading to an earlier breakthrough.

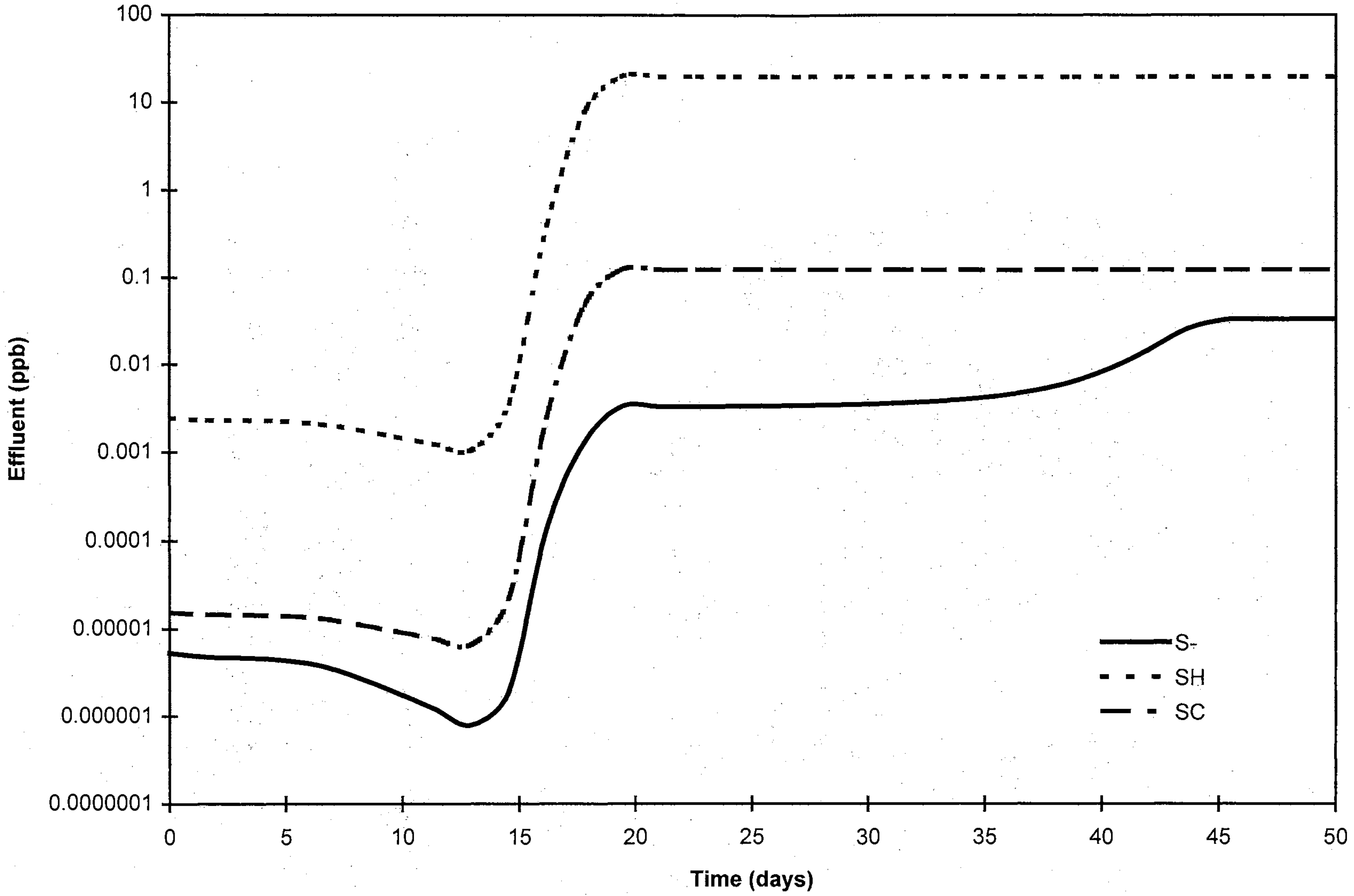


Figure 6. Predicted breakthrough curves for various silica

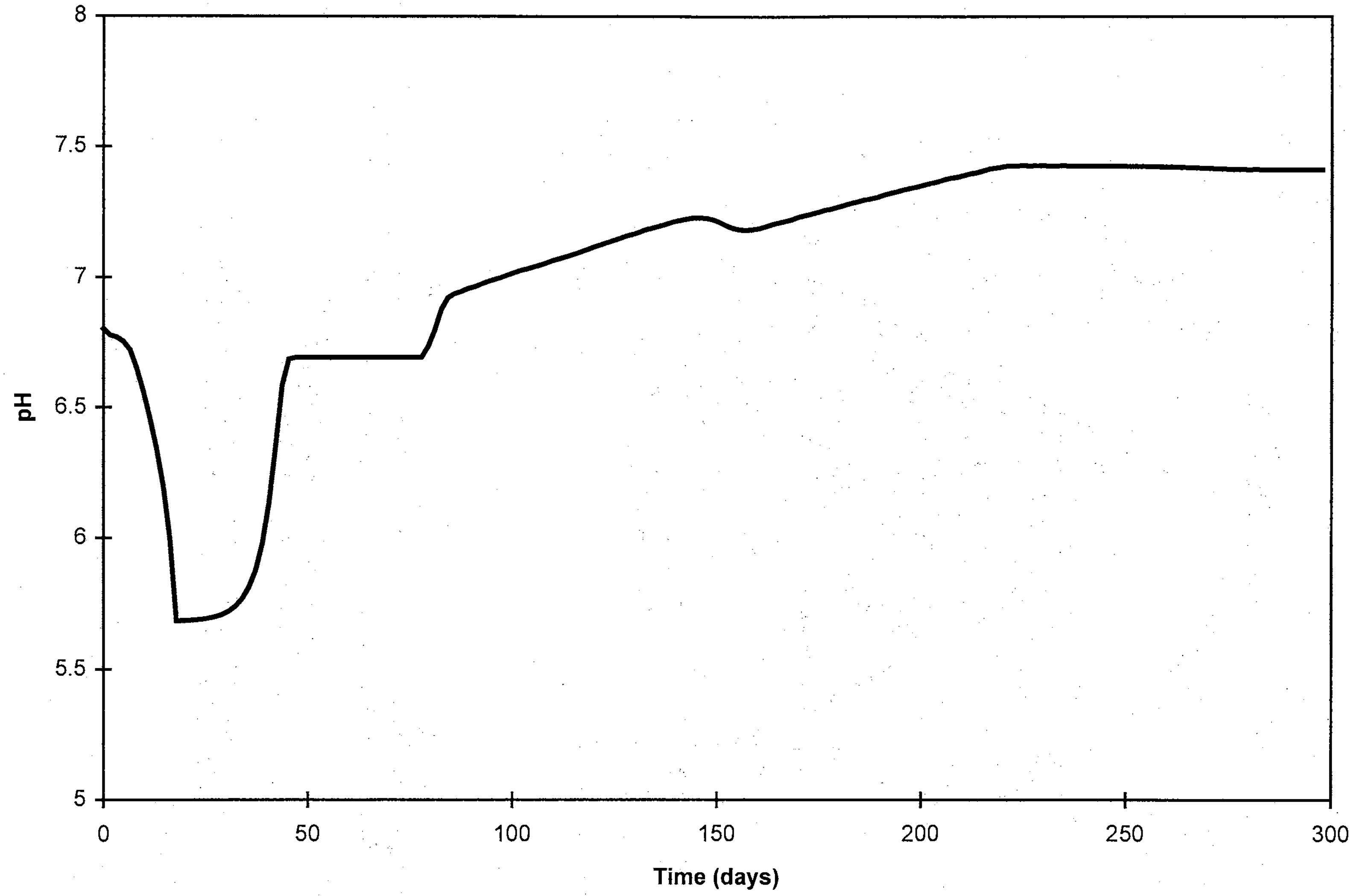


Figure 7. pH profile with the operation of the bed

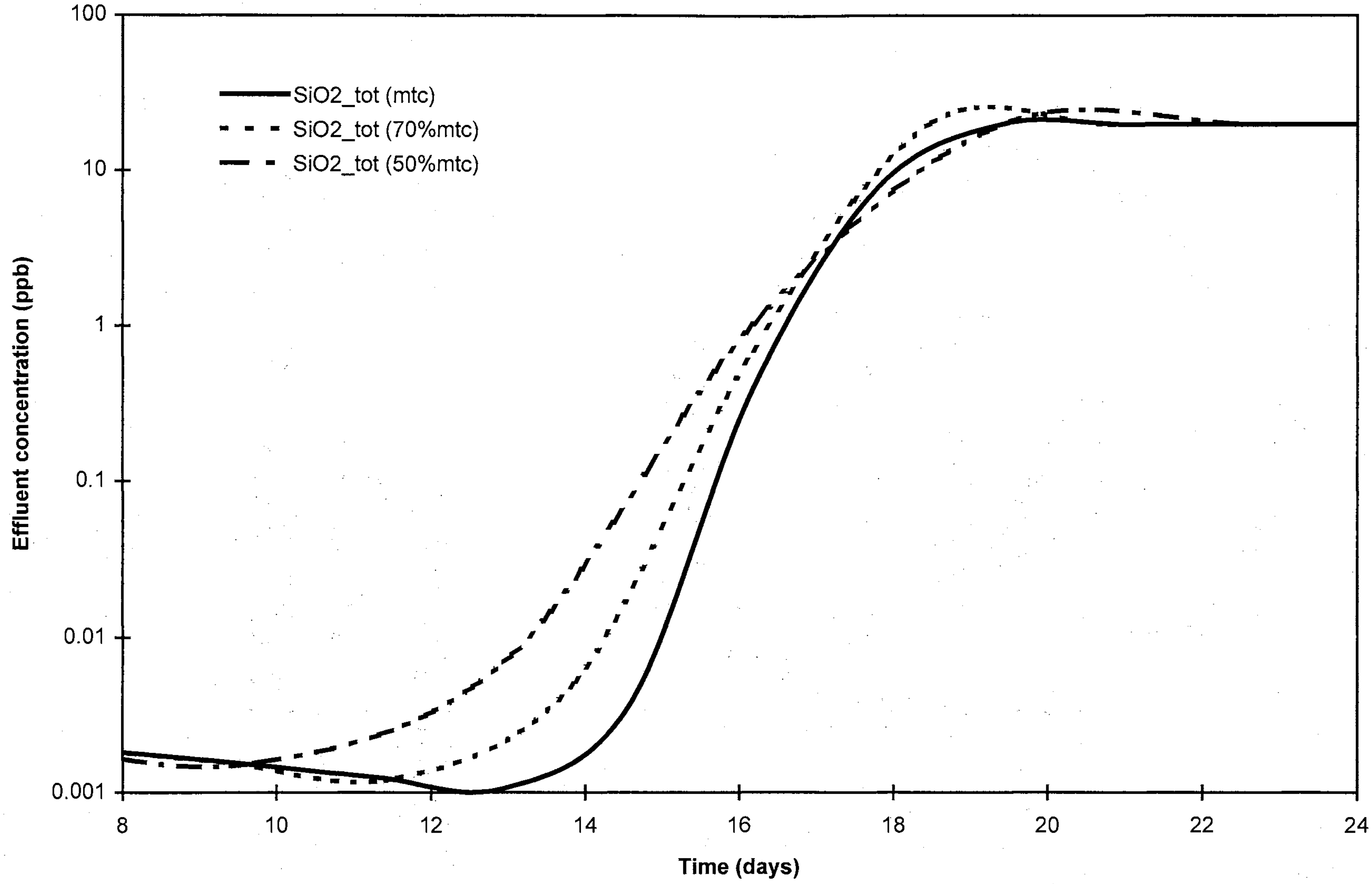


Figure 8. Effect of resin fouling on silica breakthrough

(3) Temperature effect

As we know (see Chapter III), there are several model parameters (usually physical properties) that are temperature dependent; For example, diffusivity, selectivity, dissociation constant, viscosity and density. Therefore, when temperature changes, the breakthrough time for silica will be different. Figure 9 is the simulation results of temperature effect. It can be seen that when temperature of the service water is high, silica tends to breakthrough earlier. This is because higher temperature results in lower selectivity; in addition, at higher temperature, the silica adsorbed into the resin is easy to desorb back into the bulk phase. Thus, it takes less time for silica to break through. In these simulations, three cases are studied, 25 °C (base case), 40 °C, and 60 °C. As we can see, at 25 °C, the breakthrough time is about 15.5 days, while at 40 °C and 60 °C, they are 15 days and 14.5 days respectively.

(4) Particle size effect

Another important factor influencing the silica breakthrough is the anionic resin particle size. It is founded that the particle diameter of the anionic resin affects the silica breakthrough appreciably. Three different cases are considered here. The base case is the anion particle diameter of 0.05 cm. The other two cases are anion particle diameters of 0.07 and 0.09 cm respectively. Figure 10 is the simulation result. It is observed that the larger the particle, the earlier the breakthrough time. This is because larger particle size results in a small surface area per unit volume of resin, so that the adsorption rate of silica into the resin is decreased. Furthermore, an increase in resin diameter decreases the

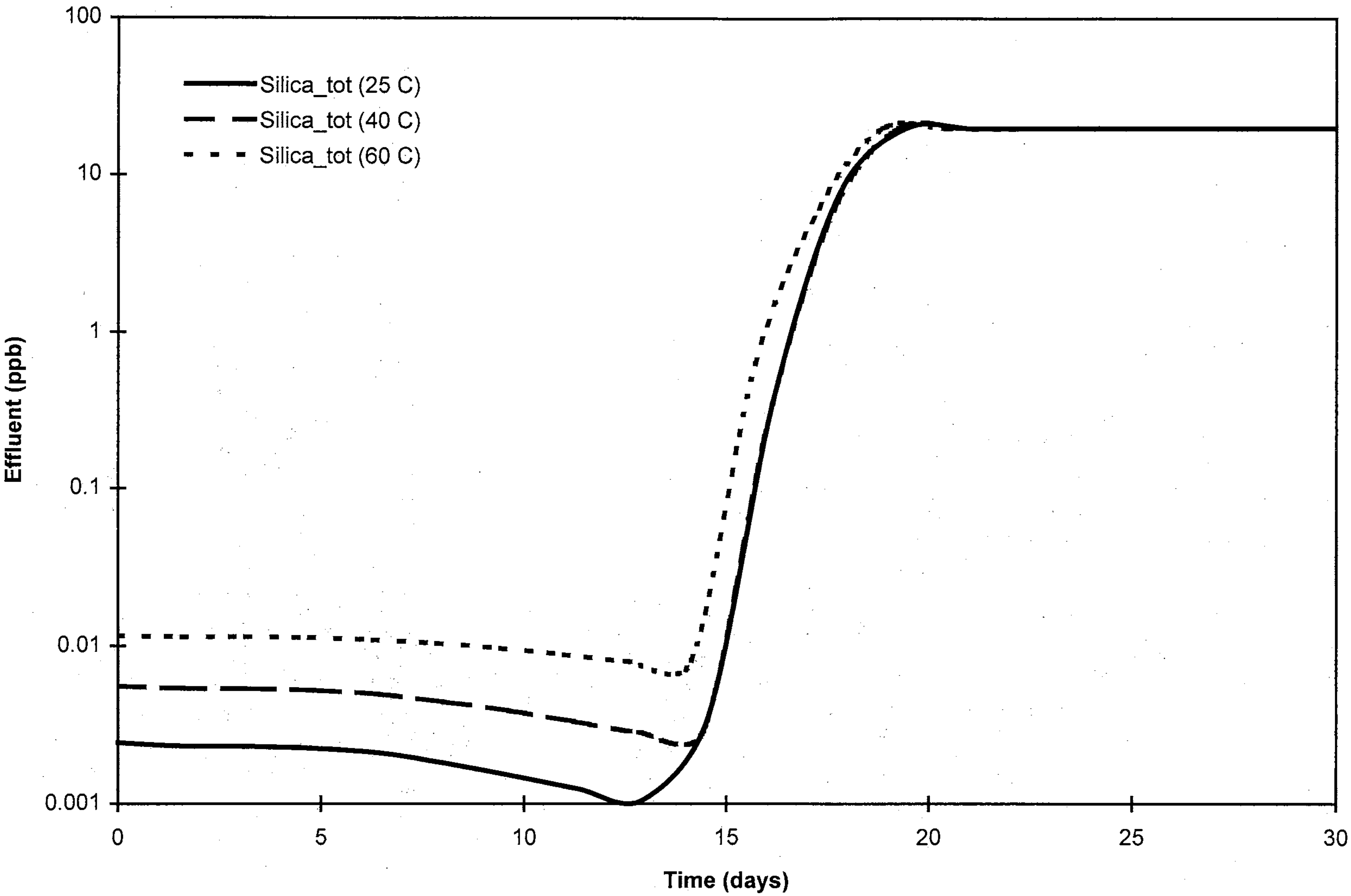


Figure 9. Effect of temperature on silica breakthrough

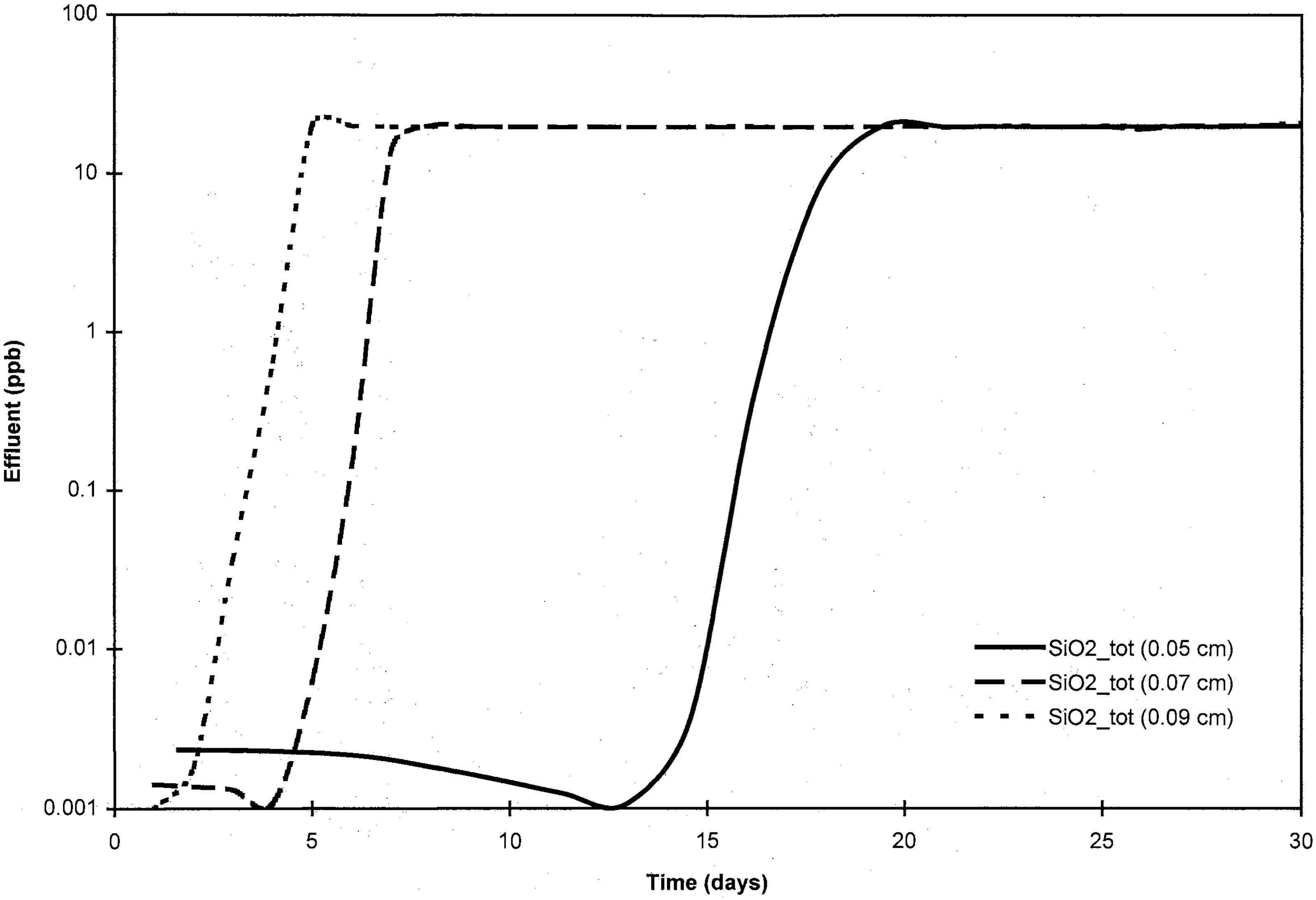


Figure 10. Effect of anion resin diameter on silica breakthrough

particle mass transfer coefficient ( $k_L \propto d_p^{-(0.386\sim0.82)}$ in the D & U equation ). The combined effects of the above two reasons lead to an earlier breakthrough time.

(5) Influent concentration effect

Figure 11 is the plot of influent concentration on silica breakthrough curve. It can be seen that when the influent concentration of silica increases, the breakthrough time will decrease. In addition, the peak value of silica 'throw' is also proportional to the influent concentration. As you can see, three different inlet concentrations of 20 ppb, 40 ppb and 60 ppb are simulated. The 20 ppb inlet concentration is the base case, which corresponds to a breakthrough time of 15.5 days; while the breakthrough time for 40 ppb and 60 ppb inlet concentrations are 12.5 days and 9.5 days respectively. These results are reasonable and easy to understand, since the total resin capacity is the same. Hence higher concentration leads to an early breakthrough.

(6) Flow rate effect

Figure 12 is the simulation results of inlet flow rate effect. The breakthrough time for silica is found to increase with a decrease in the influent flow rate of service water and vice versa. Three different inlet flow rates of 200 gpm, 210 gpm and 220 gpm are simulated, among which 210 gpm is the base case. From this figure, it is seen that the breakthrough time for 210 gpm flowrate is about 15.5 days; however when the flow rate is increased to 220 gpm, the breakthrough time is 14.5 days; and when the flowrate is decreased to 200 gpm, the breakthrough time is 17.0 days. The reason why higher flowrates reduce breakthrough time can be explained by the following equation, which is derived from the material balance, that is

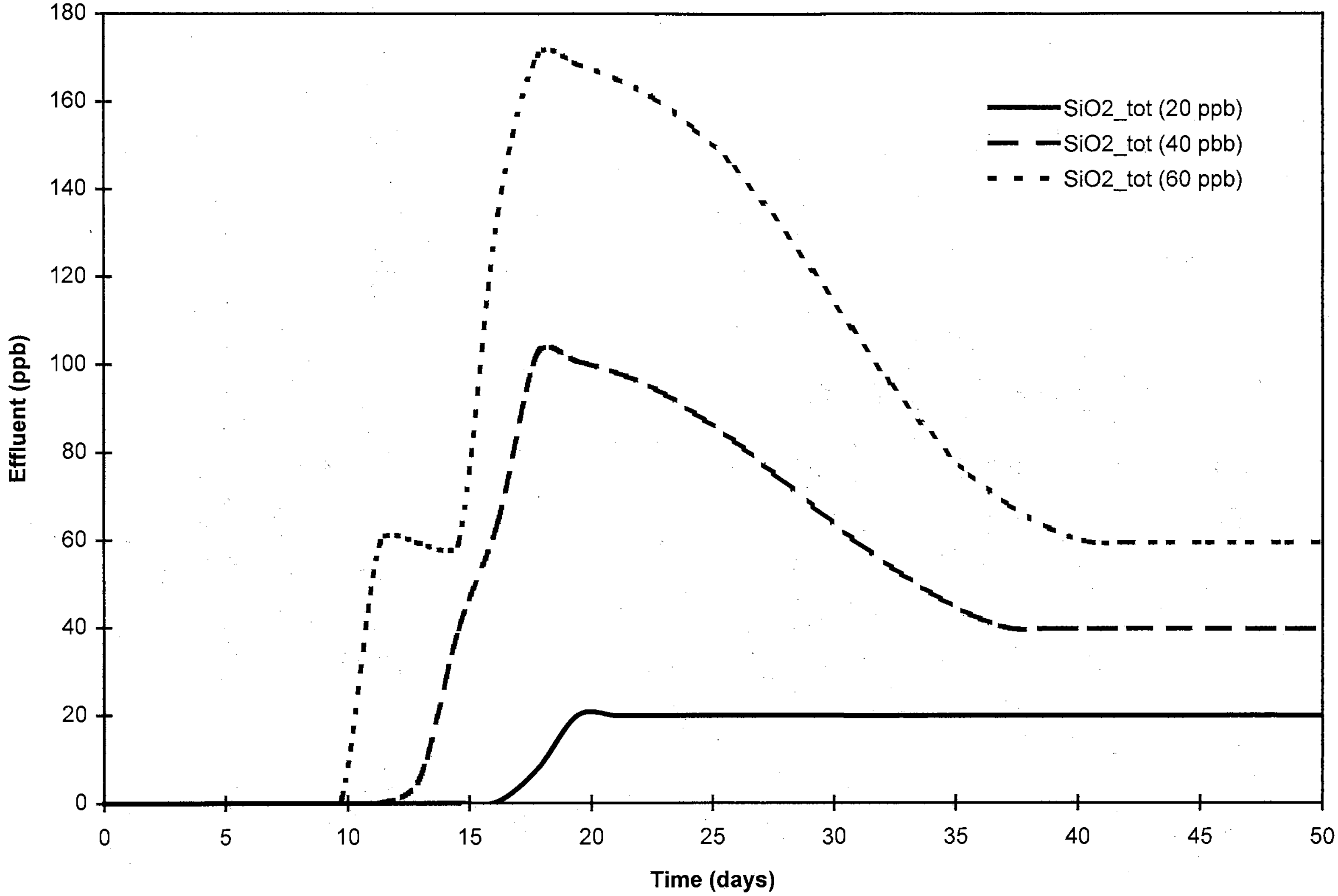


Figure 11. Effect of influent concentration on silica breakthrough

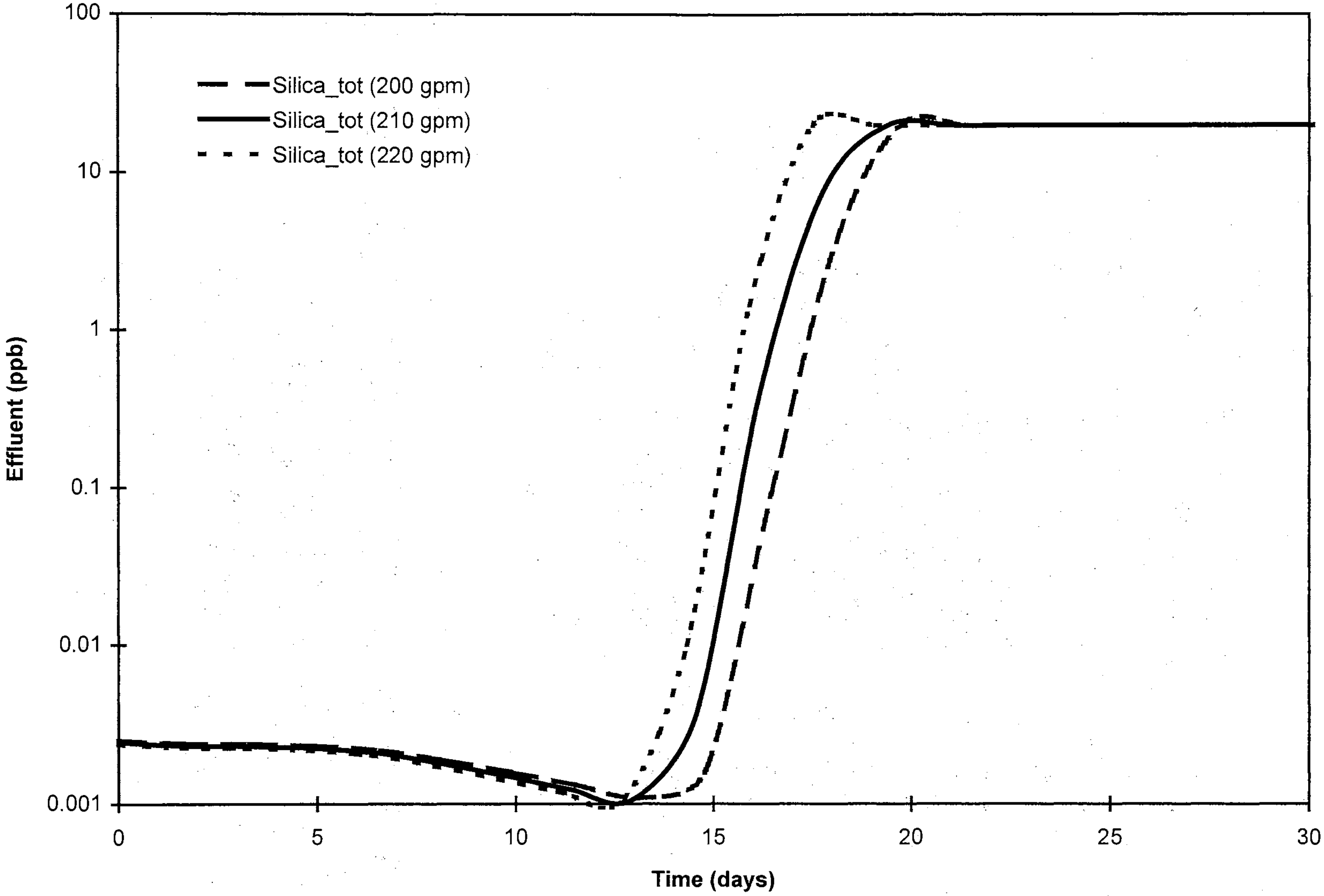


Figure 12. Effect of inlet flowrate on silica breakthrough

$$T\,(\text{days}) = \frac{Q_p \cdot \frac{\pi}{4} D^2 H \cdot FR}{V_F \cdot C_T^F \cdot 60} \cdot \frac{1}{1440} \qquad \text{(V-42)}$$

where $V_F$ = volume flow rate, ($cm^3/s$)

H = column height, (cm)

D = column diameter, (cm)

$Q_p$ = capacity of resin, (meq/ml)

FR = fraction of resin,

$C_T^F$ = total inlet concentration of ions, (meq/ml)

T = column operating time, (days)

As the column operating time *T* is proportional to the inverse of the volume flow rate $V_F$ in the above equation, higher flowrate will result in early breakthrough.

(7) Cation/anion ratio effect

The cation/anion resin ratio in the mixed-bed is also termed as the volume fraction of resin. It is found that the volume fraction of the anion resin (FAR) has a remarkable effect on the breakthrough of silica. For an anion fraction of FAR = 0.66 (basecase), the breakthrough time is about 15.5 days. But for FAR = 0.75 and FAR = 0.50, the breakthrough time are 20.5 days and 12.8 days respectively. These observations are expected and reasonable, since the number of active sites (for ion exchange or adsorption) available increase with the increase of anion resin fraction (FAR). With an increase in the number of exchange sites or adsorption sites, the resin could exchange more ionic silica or adsorb more molecular silica. The simulation results for this case are shown in Figure 13.

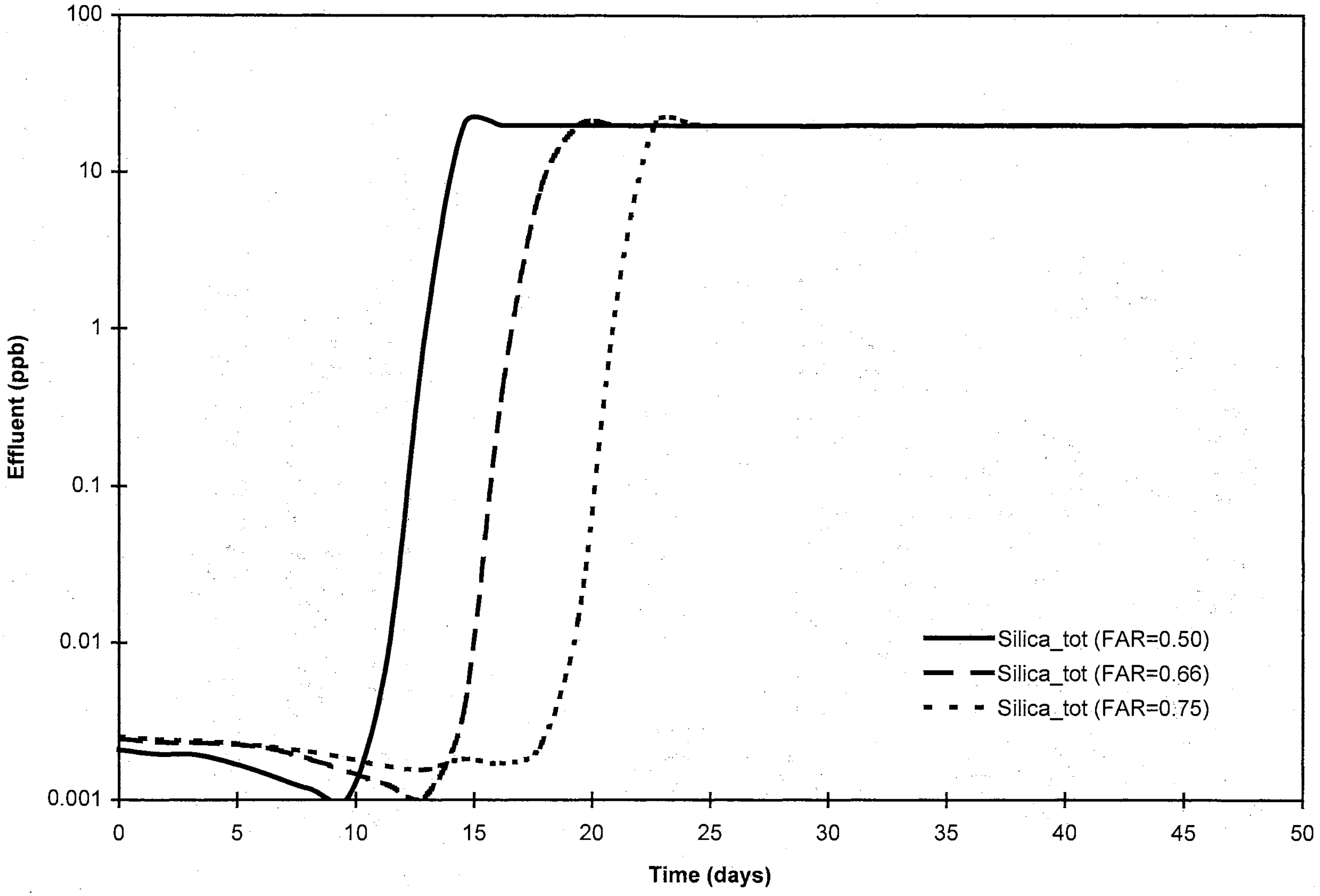


Figure 13. Effect of anion fraction on silica breakthrough

## Case II. NETCO Water Treatment Plant

At the water treatment plant of Northeastern Technology Corporation (NETCO), mixed bed ion exchange is used to remove silica. The vessel diameter is 54 inches, and the resin depth is 2.516 ft. The flowrate is about 300 gpm, and the influent silica concentration is about 3327 ppb. The operating temperature is 75 °F, and the influent pH is 5.0. Other operating conditions and model parameters are listed in Table VII.

Using the model developed in this Chapter, simulations were performed for this case. Figure 14 through Figure 16 are the simulation results. It can be seen that among the cations, $Mg^{+2}$ starts breakthrough first, in about 100 days; and $Ca^{+2}$ in about 130 days. However, $Al^{+3}$ will not break through until 160 day. In addition, all divalent ions showed ionic throws in their breakthrough curves. This is because they are kicked out by trivalent cation $Al^{+3}$. For anions, silica ($SiO_2$) is the first to breakthrough, which is about 4 days; then $F^-$ is the next (5.5 days); and then $Cl^-$ (7.5 days). However, $SO_4^{-2}$ does not break through until 400 days. These simulation results, we think, are reasonable and conceptually true, because they are in consistent with what we expect. However, the model's quantitative ability of predicting breakthrough time for this case has not been validated by the industrial experience from NETCO yet. Therefore we can only conclude that the model predicts silica qualitatively true. To further improve model's quantitative ability, more simulations and fine tuning the model parameters are required.

Table VII. Input Data and Model Parameters for NETCO Case

| Item | Value | |
|---|---|---|
| Bed diameter (cm) | 137.2 | |
| Bed depth (cm) | 76.78 | |
| Bed void fraction | 0.35 | |
| Cation resin diameter (cm) | 0.068 | |
| Anion resin diameter (cm) | 0.058 | |
| Cation resin capacity (meq/ml) | 2.1 | |
| Anion resin capacity (meq/ml) | 1.0 | |
| Cation/Anion ratio | 0.5: 0.5 | |
| Influent pH | 5.0 | |
| Temperature (°C) | 23.8 | |
| Influent flow rate ($cm^3/s$) | 1.388E+4 (220gpm) | |
| **Influent conc (ppb) and resin loadings (%)** | **Conc (ppb)** | **Resin loading (%)** |
| Aluminum ($Al^{+3}$) | 10.0 | 0 |
| Calcium ($Ca^{+2}$) | 1.1 | 0 |
| Magnesium ($Mg^{+2}$) | 1.0 | 0 |
| Fluoride ($F^-$) | 8.1 | 0 |
| Chloride ($Cl^-$) | 19.1 | 0 |
| Sulfate ($SO_4^{-2}$) | 13.9 | 0 |
| Silica total ($SiO_2$) | 3327 | 0 |
| **Physical properties** | **Diffusivity (25 C) ($cm^2/s$)** | **Selectivity** |
| Species: | | |
| Aluminum ($Al^{+3}$) | 0.541E-5 | 5.62 |
| Calcium ($Ca^{+2}$) | 0.792E-5 | 3.9 |
| Magnesium ($Mg^{+2}$) | 0.706E-5 | 2.7 |
| Fluoride ($F^-$) | 1.475E-5 | 1.6 |
| Chloride ($Cl^-$) | 2.032E-5 | 22 |
| Sulfate ($SO_4^{-2}$) | 1.065E-5 | 60 |
| Monovalent silicate ($S^-$) | 0.915E-5 | 2.5 |
| Divalent silicate ($S^{-2}$) | 0.823E-5 | 1.1 |
| Polymeric silicate ($PS^{-2}$) | 0.685E-5 | 0.5 |
| Silicic acid (SH) | 0.743E-5 | |
| Colloidal silica (SC) | 0.582E-5 | |
| Particulate silica (SP) | 0.471E-5 | |

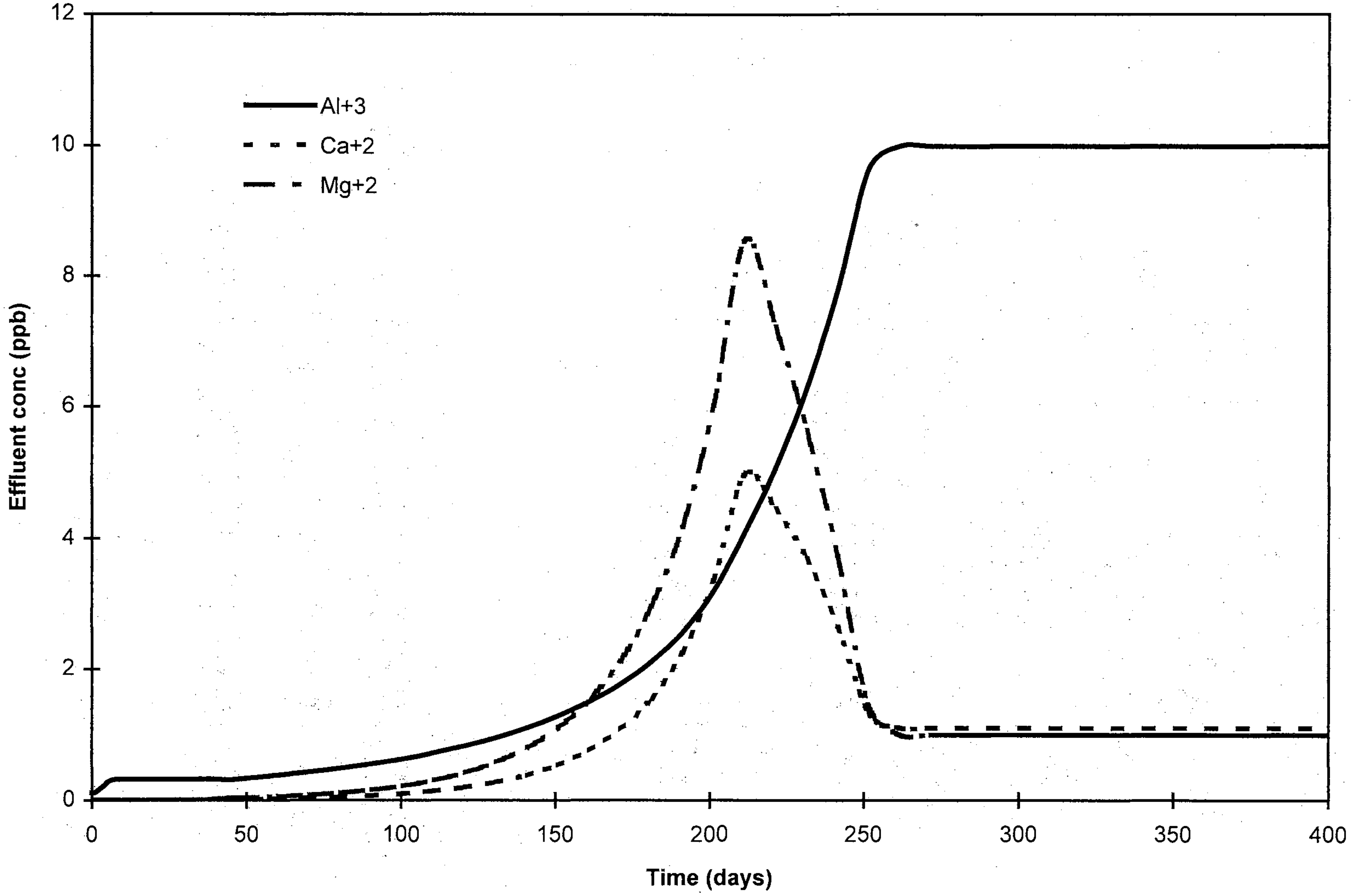


Figure 14. Cation breakthrough curves for NETCO case

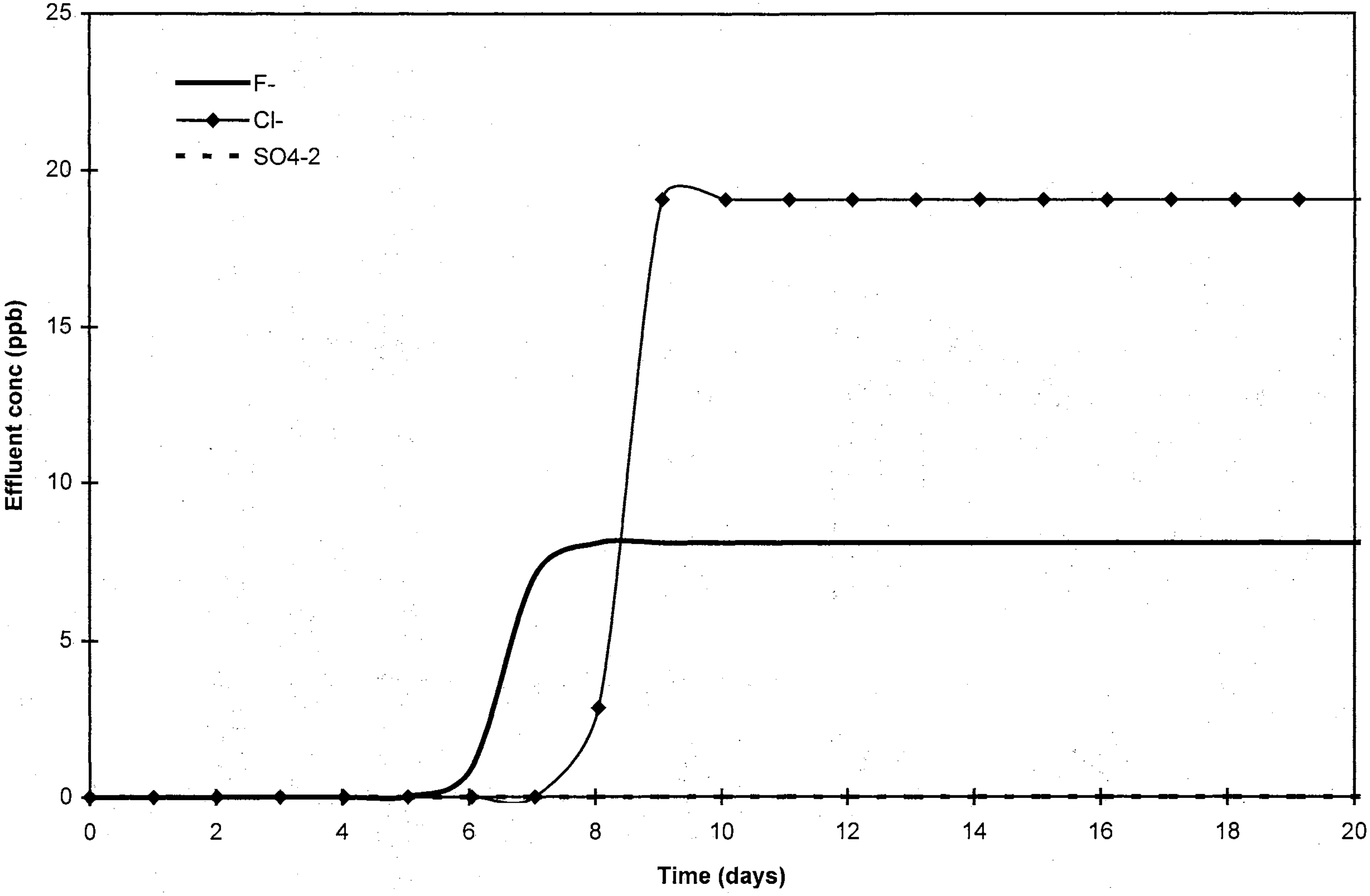


Figure 15. Anion breakthrough curves for NECTO case

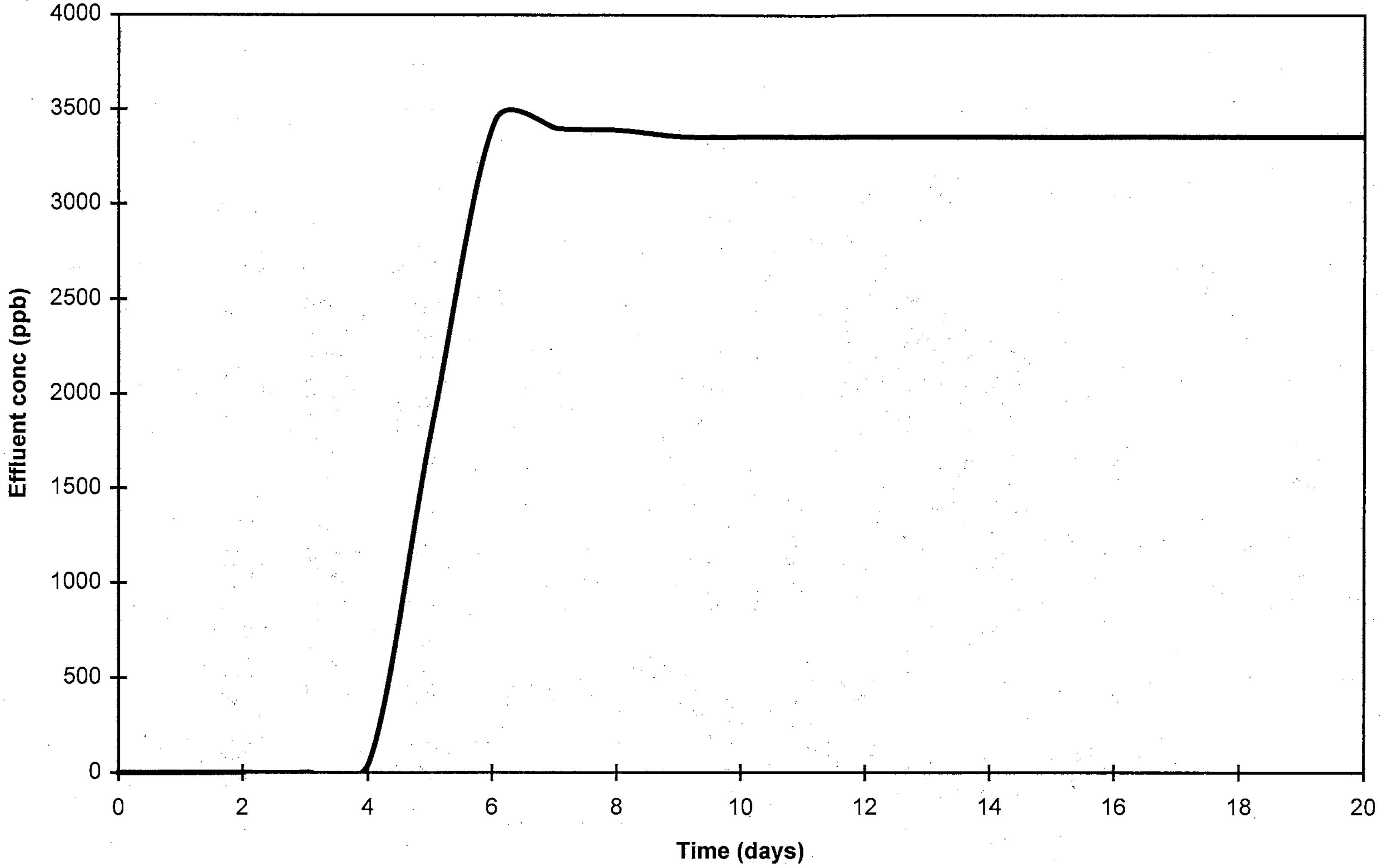


Figure 16. Breakthrough curve for the total silica (SiO2_tot)

## Sensitivity Analysis of the Model Parameters

There are several model parameters that are important in determining model's performance. The first one is the criterion of maximum amount of silica capable of being adsorbed by the resin (not the exchange capacity). As we know, under pH<8, most of silica is in the form of silicic acid. So the amount of molecular silica that can be adsorbed by the resin decides the silica removal limit. In this model, we set the criterion of maximum amount of silica adsorbed to the resin a fraction of 0.05 (relative to resin's capacity). Actually, it could be other values depending on resin's properties. For the NETCO case, simulations were performed to study its effects. Figure 17 shows the effect of this criterion on the breakthrough time predicted. As you can see, the larger the criterion, the later the breakthrough time. But when $f_a > 0.10$, there is no much change in the breakthrough time. For the operating conditions in NETCO, the value of 0.05 was chosen, since it matches the breakthrough time well. For other plants, different values may be chosen, but usually they are in the range of 0.01 ~ 0.10, because the maximum amount of silica which can be adsorbed by the resin is about 10 % of the resin capacity (Iler, 1955).

The adsorption coefficient and adsorption index for the Freundlich isotherm also influence model performance. For different values of $K$ and $n$ (or $a_1$ and $e_1$), the predicted breakthrough time is different. Figure 18 shows the effect of these parameters on breakthrough behavior. From this plot, we can see that, parameters of $K$ and $n$ (or $a_1$ and $e_1$) have great effects on the breakthrough time. In this study, we chose the following values: $K = 1.0$, $n = 3.0$; $a_1 = 0.5$, $e_1 = 0.2$; $a_2 = 3.0$, $e_2 = 0.12$ for the molecular, colloidal, and particulate silica correspondingly.

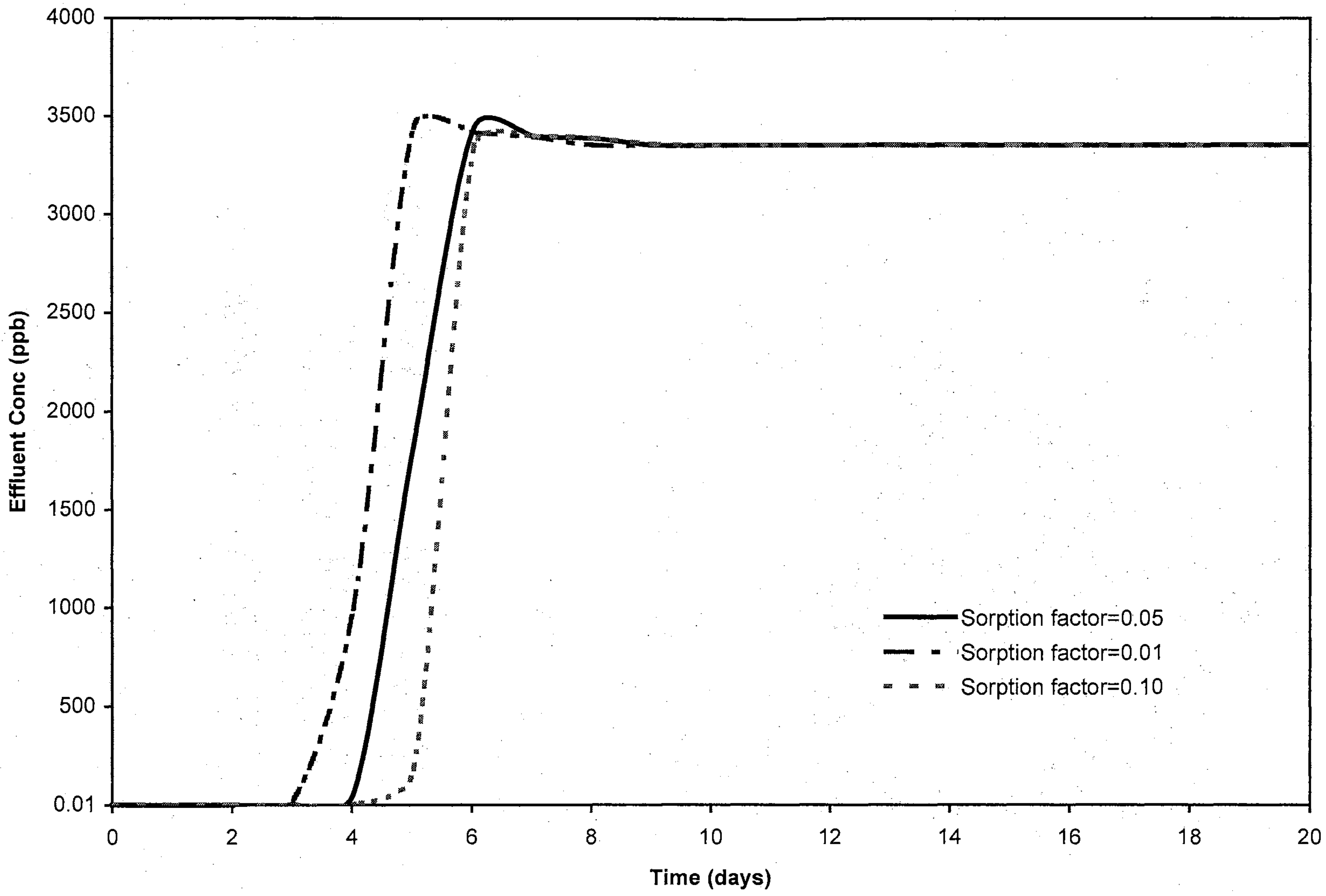


Fig. 17. Effect of adsorption factor on silica breakthrough curve

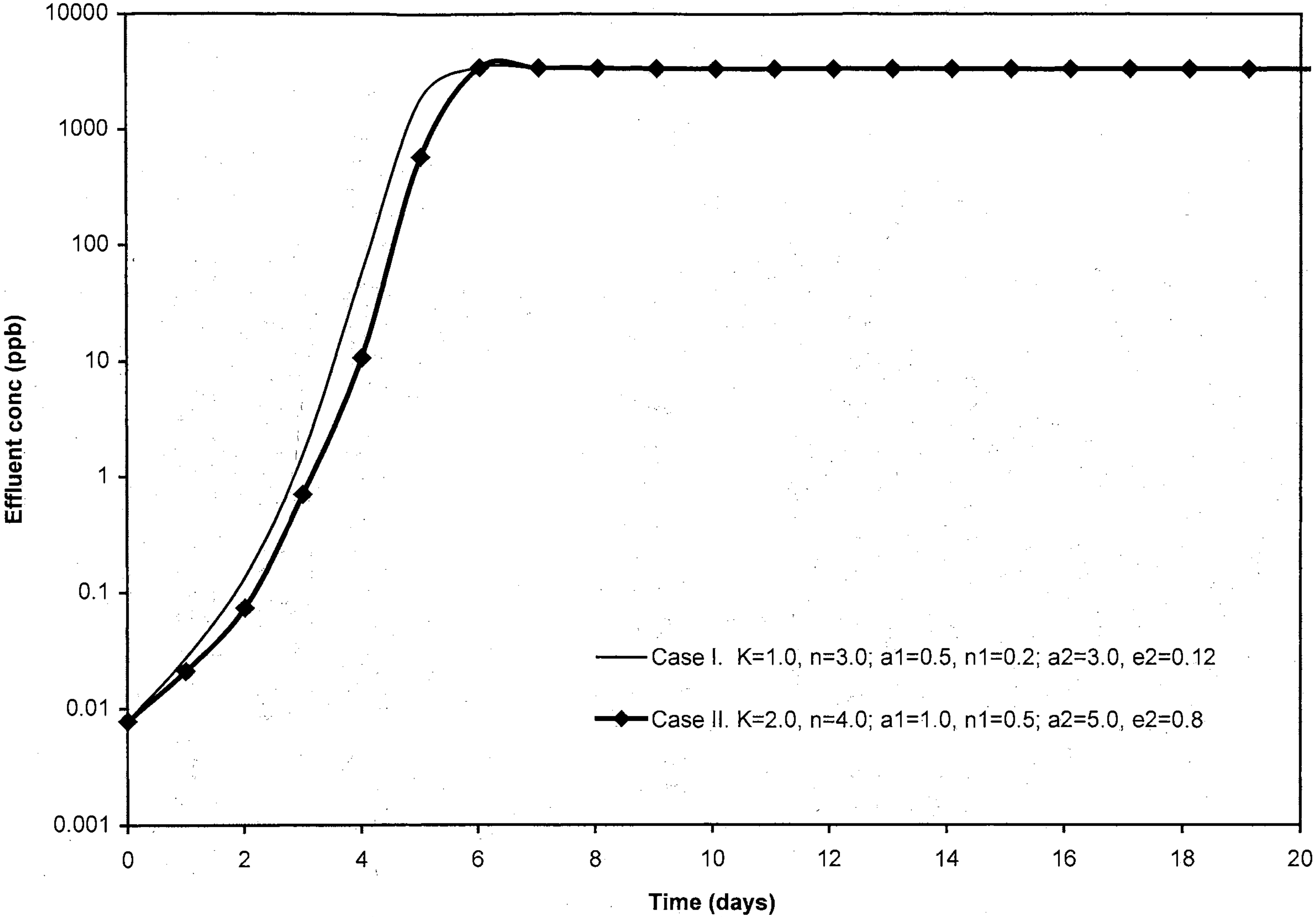


Figure 18. Effect of Freundlich adsorption coefficients and index on silica breakthrough

## Conclusions

A model that can handle multicomponent mixed-bed ion exchange with silica has been developed with the following features:

- The species handled by the model can be of arbitrary number and with arbitrary valence.
- Compared to the work of Pamarthy (1995), the model accounts for various silicic species in the silica equilibrium, and therefore, is a more generalized model.
- Besides the silicic acid and monovalent silicate ion, this model can be used to predict the removal of other silicic species such as colloidal and particle silica.
- Industrial column simulations showed the predicted results agree with plant experience.

Several transport mechanisms, such as ion exchange, protonation and physical adsorption have been considered in modeling silica removal by ion exchange resins. A Freundlich adsorption isotherm was employed to determine the interfacial concentrations for various nonionic silicic species. Model predictions showed that this approach is an effective way to characterize the mass transfer rate of nonionic silicic species. But the model parameters such as Freundlich adsorption coefficients and Freundlich indexes have great effects on model performance, and should be chosen properly.

The simulation of temperature, particle size, inlet flow rate, inlet concentration, cation/anion ratio and resin fouling effects on mixed-bed performance are as expected. However, due to lack of experimental data, these results have not been validated. Therefore, they should not be used as absolute guidelines for plant operators, instead, they can only be served as qualitative guidelines.

More work on validating the model is needed. In addition, program run time improvement is required, because the incorporation of silica dissociation equilibrium greatly increases model complexity, so that run time is increased.

# CHAPTER VI

# CONCLUSIONS AND RECOMMENDATION

This dissertation deals with the expansion of a mixed-bed ion exchange model to include more complex solution chemistry. Specifically, the model was extended to handle multivalent weak electrolytes, such as type I, II, and III dissociative species and silica. In addition, the model was also extended to handle some special cases in MBIE operations, such as resin fouling, different cation/anion ratio (layered-bed), variable inlet concentrations, variable inlet flowrate, layered-bed and bed cleaning.

Chapter 2 presented the study of mass transfer mechanisms of weak electrolytes in an ion exchange column. The transport of both ionic and nonionc forms of weak electrolytes to the resin has been considered. To model ion exchange with weak electrolytes, methods have been proposed to describe the mass transfer rate of weak electrolytes, in which an enhancement factor was introduced to account for the effect of higher nonionic concentration on the mass transfer rate.

Chapter 3 addressed the development of a generalized rate model. Based on the discussion of mass transfer mechanisms in Chapter 2, a generalized rate model that can handle multicomponent MBIE, especially systems with dissociative species was developed. The success of the model was achieved by implementing the generalized dissociation equilibrium equations and considering the transport of both ionic and nonionic forms of weak electrolytes.

Chapter 4 demonstrated how to handle special cases in modeling multicomponent MBIE. In this chapter, additional assumptions have been supplemented, and a lot of effort has been made in modifying the computer code. With these endeavors, the model was extended successfully to handle special cases like resin fouling, variable inlet concentrations, variable inlet flowrate, layered-bed, and bed cleaning.

Chapter 5 presented further expansion of the model to handle silica. Due to the complexity of silica dissociation chemistry, a detailed discussion of silica dissociation and ionization in water was presented. Based on which, a model that considers various possible silicic species in water was developed. In the model, different transport mechanism was applied to the different silicic species; and a solubility correlation for silica (Rimstidt and Barnes, 1980) was employed to develop the model.

Summarizing the above work, the following conclusions can be made.

## Conclusions

- A generalized rate model that can handle multicomponent mixed-bed ion exchange, especially systems with dissociative species has been developed. Extensive tests showed that the model handles as many as 18 cations and 18 anions, including most weak electrolytes and silica.
- The transport of nonionic forms of weak electrolytes into resin is important and should be considered in modeling ion exchange with weak electrolytes. The mass

transfer rate of weak electrolytes is the summation of the rates for both ionic and nonionic forms.

- It was found that the solution pH, dissociation equilibrium, resin properties, and both ionic and nonionic concentrations of weak electrolytes have great effects on the mass transfer rate of weak electrolytes.
- Simulations of industrial cases, such as Arkansas Nuclear One (ANO) and Pennsylvania Power and Light (PP&L), showed that the model predictions agree favorably with the real plant experience, which partially validated the model.
- The model's capability of handling temperature effects was expanded. A lot of temperature dependent correlations for diffusivity, viscosity, density and dissociation constant were developed and incorporated in the model, which makes the model can handle most species within the temperature range of 25 ~ 90 °C.
- The developed model has features of handling special cases in MBIE, such as variable inlet concentrations, variable inlet flowrate, layered-bed and bed cleaning. In addition, the model also gives users the option of incorporating the effect of cation resin desulfonation.
- Studies of silica dissociation chemistry showed that various silicic species; such as silicic acid, monovalent silicate, divalent silicate, polymeric silica, colloidal silica, and particulate silica may exist in a system containing silica. The relative amount of each species depends on the solution pH and dissociation equilibrium.
- At the normal pH range (6.5 ~ 7.5), most silica is in the form of silicic acid (about 98%), and the monovalent silicate is about 1%. Therefore, other kinds of silicic

species are in trace amount, and their effects on the silica effluent concentration can be ignored.

- Temperature, particle size, inlet flow rate, inlet concentrations, cation/anion ratio, and resin fouling have great effects on MBIE column performance. Higher temperature, larger particle size, higher flow rate, higher inlet concentrations, larger cation/anion ratio, and severe resin fouling tend to result in earlier breakthrough.

## Recommendations

Although a generalized rate model was developed, there are still some areas in which improvements need to be made:

- Numerical instability and program run time

Extensive simulations showed that some instability problems still exist and the program run time is higher when several kinds dissociative species are handled simultaneously, especially when silicic species are involved. This is due to the stiffness of the equations involved in the model. Therefore, better numerical methods or robust algorithms are required. Moreover, a method that uses variable step size will probably be helpful for the run time improvement.

- Quantitative capability of handling silica

As shown in Chapter 5, the predicted breakthrough curves for silica is conceptually accurate; however, they have not been validated fully by industrial data. Therefore, more simulations and fine tune the model parameters are needed to improve the model's quantitative ability of predicting silica.

- Temperature dependence data of selectivity

In the current model, the assumption of temperature independence of selectivity has been made due to a lack of experimental data. Therefore, to improve the model's capability of handling temperature effects, there is a need for temperature dependence selectivity data.

- Handling special cases simultaneously

The current model can only handle some special cases of multicomponent MBIE independently. However, in practical situations all these cases could happen concurrently. Hence further effort in modifying the code to handle these cases simultaneously is required.

- Further expansion of the model

Although the current model can handle most weak electrolytes, the borate dissociation chemistry has not been included. Therefore, borate cannot be handled temporarily by this model. Further work to incorporate borate dissociation chemistry, such as the work by Lou (1997), is required. In addition, there is a need to expand the model to handle metal oxides.

# BIBLIOGRAPHY

Adams, B. A. and Holmes, E. L. (1935a). Adsorptive properties of synthetic resins, J. Soc. Chem. Ind., 54, 1-9

Adams, B. A, and Holmes, E. L. (1935b). Base-exchange resins, Chem. Age, 38, 117

Alexander, G. B. Heston, W. M. and Iler, R. K. (1954). The solubility of amorphous silica in water, J. Phys. Chem., 58, 453-455

Applebaum, S. B. (1968). Demineralization by Ion Exchange in Water Treatment and Chemical Processing of Other Liquids, Academic Press, New York

Arden, T. V. (1968). Water Purification by Ion Exchange, pp76, Butterworth, London

Bhandari, M. V., Juvekar, V. A., and Patwardhan, S. R. (1992). Sorption studies on ion exchange resins. 2. Sorption of weak acids on weak base resins, Ind. Eng. Chem. Res., 31, 1073-1080

Bernal, M. P. and Lopez-Real, J. M. (1993). Natural zeolites and sepiolite as ammonium and ammonia adsorbent materials. Bioresoures Technology, 43, 27-33

Blume, R. (1987). Preparing ultrapure water, Chem. Eng. Prog., 83(12), 55-57

Bolden, W. B., White, T. and Groves, Jr. F. R. (1989). Continuous fixed bed ligand exchange: The shrinking-core model, AIChE. J., 35(5), 849-852

Boyd, G. E., Adamson, A. W., and Myers, Jr. L. S. (1947). The exchange adsorption of ions from aqueous solutions by organic zeolites II. Kinetics, J. Am. Chem. Soc., 69, 2836-2848

Brady, J. E. and Holum J. R. (1984). Fundamentals of Chemistry, 2nd Ed., John Wiley & Sons, Inc., New York

Bulusu, R. (1994). Development of a column model to predict multicomponent mixed bed ion exchange breakthrough, MS Thesis, Oklahoma State University, Stillwater, OK

Bungay, H. R. (1989). Mixed bed exchangers, Basic Biochemical Engineering, 25, 10-26

Bunzl K. (1995). Calculation of the Helfferich number to identify the rate-controlling step of ion exchange for a batch process, Ind. Eng. Chem. Res., 34, 2584-2587

Caddell, J. R. and Moison, R. L. (1954). Mixed-bed deionization at high flow rates. Chem. Eng. Progr. Sympo. Ser., 50(14), 1-6

Cheney, W. and Kincaid, D. (1985). Numerical mathematics and computing, Chapter 7, P268. Brooks/Cole Publishing Company, Pacific Grove, California.

Chowdiah, V. N. (1996). Liquid-film diffusion controlled ion-exchange modeling: Study of weak electrolyte mass transport and film mass-transfer kinetics, PhD. Dissertation, Oklahoma State University, Stillwater, OK

Copeland, J. P., C. L. Henderson, and J. M. Marchellow (1967). Influence of resin selectivity on film-diffusion controlled ion exchange. AIChE J., 13, 449-452

Copeland, J. P. and J. M. Marchellow (1969). Film-diffusion controlled ion exchange with a selective resin. Chem. Eng. Sci., 24, 1471-1474

D'Alelio, G. F. (1944). U.S Patent 2,366,007

Divekar, S. V., Foutch, G. L., and Haub, C. E. (1987). Mixed bed ion exchange at concentrations approaching the dissociation of water. Temperature effects. Ind. Eng. Chem., 49, 1297-1302

Dwivedi, P. N. and Upadhyay, S. N. (1977). Particle-Fluid Mass Transfer in Fixed and Fluidized Beds. Ind. Eng. Chem., Process Des. Dev., 16(2), 157-165

Dobbs, R. A., Uchida, S., Smith, L. M. and Cohen, J. M. (1975). Ammonia removal from wastewater by ligand exchange, AIChE. Symp. Ser., 152(71), 157-163

de la Vega M. R., Loureiro, J. M., and Rodrigues, A. E. (1996). Equivalence between Nernst-Planck and "corrected" Fick's law in modeling fixed-bed ion exchange process, The Chem. Eng. J., 61, 123-132

Divekar, S. V., Foutch, G. L., and Haub, C. E. (1987). Mixed-bed ion exchange at concentrations approaching the dissociation of water: Temperature effects. Ind. Eng. Chem. Res., 26(9), 1906-1909

Dorfner, K. (1972). Ion Exchangers: Properties and Applications, 3rd Ed., Ann Arbor Science, Michigan

Drucker, J. R. and Dale, J. (1988). Removal of colloids by the use of ion exchange resins. Ultrapure Water Journal, 5(9), 14-17

Faust, S. D. and Aly, O. M. (1981). Chemistry of Natural Waters, pp25, Ann Arbor

Science, Michigan

Fisher, S. and E. Burke (1993). The Chemical Impact of the Use of Low Crosslinked Resins in BWR Condensate Polishers, EPRI BWR Plant Chemists Meeting, Palo Alto, California, February 25-26

Foutch, G. L. (1991). Temperature effects on mixed-bed ion exchange performance, Ultrapure Water Journal, 8(4), 56-58

Franzreb, M., Holl, W. H., and Sontheimer, H. (1993). Liquid-side mass transfer in multicomponent ion exchange: I. System without chemical reactions in the film, Reactive Polymer, 21, 117-133

Frisch, N. W. and Kunin, R. (1960). Kinetics of mixed-bed deionization: I, AIChE. J. 6(4), 640-647

Gans, R. (1906). German Patent 174,097 and 197,111

Gaspard, M. and Martin, A. (1983). Clinoptilolite in drinking water treatment for $NH_4^+$ removal. Water Research, 17(3), 279-288

Gear, C. W. (1971). Numerical Initial Value Problems in Ordinary Differential Equations, Chapter 7, Prentice-Hall, Englewood Cliffs, NJ

Glaski, F. A. and J. S. Dranoff (1963). Ion exchange kinetics: A comparison of models, AIChE J., 9, 426-431

Glueckauf, E. and Coates, J. I (1947) Theory of chromatography, IV. The influence of incomplete equilibrium on the front boundary of chromatograms and on the effectiveness of separation. J. Chem. Soc., 1315-1321

Goto, K. (1955). J. Chem. Soc. Jpn, Pure Chem. Sect., 76, 1364-1371

Goto, K., Okura, T. and Kayama, I. (1953). Kagaku Tokyo, 23, 426-429

Grimshaw, R. W. (1975). Ion Exchange: Introduction to Theory and Practice, London, Chemical Society

Haralambous, A., Malliou, E. and Malamis, M. (1992). The use of zeolite for ammonia uptake, Water Sci. and Technol., 25, 139-146

Harfst, W. (1992) Treat methods differ for removing reactive and unreactive silica. Ultrapure Water Journal, 9(3), 59-62

Harries, R. R. (1987). Ion exchange kinetics in condensate purification, Chemistry and Industry, 16(2), 104-109

Haub, C. E. (1984). Model development for liquid resistance-controlled reactive ion exchange at low solution concentrations with applications to mixed-bed ion exchange, MS Thesis, Oklahoma State University, Stillwater, OK

Haub, C. E. and Foutch, G. L. (1986a). Mixed-bed ion exchange at concentrations approaching the dissociation of water: 1. Model development. Ind. Eng. Chem. Fund., 25, 373-381

Haub, C. E. and Foutch, G. L. (1986b). Mixed-bed ion exchange at concentrations approaching the dissociation of Water: 2. Column model applications. Ind. Eng. Chem. Fund., 25, 381-385

Helfferich, F. G. (1962). Ion Exchange, McGraw Hill, New York

Helfferich, F. G. (1965). Ion exchange kinetics: V. Ion exchange accompanied by reactions, J. Phys. Chem., 69, 1178-1187

Helfferich, F. G. and Bennett, B. J. (1984). Weak electrolytes, polybasic acids, and buffers in anion exchange columns I. Sodium acetate and Sodium carbonate systems, Reactive Polymers, 3, 51-66

Helfferich, F. G. and Y. L. Hwang (1985). Kinetics of acid uptake by weak-base anion exchangers. Mechanism of proton transfer. AIChE Symposium Series, Adsorption and Ion Exchange, 81, No. 242. 17-28

Helfferich, F. G. (1990). Ion-exchange equilibria of amino acids on strong-acid resins: Theory, Reactive Polymers, 12, 95-100

Helfferich, F. G. (1990). Models and physical reality in ion-exchange kinetics. Reactive Polymers, 13, 191-194

Henley, M. (1992). Colloidal and soluble silica removal remains challenge when producing high-purity water. Ultrapure Water Journal, 9(12), 13-16

Huang, H., Hsiang, C. C., Lee, S. C., and Ting, G. (1991). The sorption behavior of boric acid on weak-base anion exchange resin, Solvent Extraction and Ion Exchange, 9(2), 319-335

Hubner, P. and Kadlec, V. (1978). Kinetic behavior of weak base anion exchangers, AIChE. J., 24(1), 149-154

Hulett, G. (1926). Solubility and size of particles. In Jerome Alexander, Ed., Colloid Chemistry, Chemical Catalog Co., New York

Hussey, D. F. (1996). Development of a generalized multicomponent ion exchange reaction equilibrium model, MS Thesis, Oklahoma State University, Stillwater, OK

Höll, W. and Sontheimer H. (1977). Ion exchange kinetics of the protonation of weak acid ion exchange resins, Chem. Eng. Sci., 32, 755-762

Iler, R. K. (1955). Colloid Chemistry of Silica and Silicates. Cornell University Press, Ithaca, NY

Iler, R. K. (1979). The Chemistry of Silica, John Wiley & Sons, Inc, New York

Ingri, N. (1959). Equilibrium studies of polyanions, IV. Silicate ions in NaCl medium, Acta Chem. Scand., 13, 758-775

James, G. V. (1965). Water Treatment: A Guide to the Treatment of Water and Effluents Purification, 3rd Ed, Technical Press, London

Jansen, M. L., Straathof, A. J. J., van der Wielen, L. A. M., Luyben, K. Ch. A. M., and van den Tweel, W. J. J. (1996a). Rigorous model for ion exchange equilibria of strong and weak electrolytes, AIChE. J., 42(7), 1911-1924

Jansen, M. L., Hofland, G. W., Houwers J., Straathof, A. J. J., van der Wielen, L. A. M., Luyben, K. Ch. A. M., and van den Tweel. W. J. J. (1996b). Effect of pH and concentration on column dynamics of weak electrolyte ion exchange, AIChE. J., 42(7), 1925-1937

Jansen, M. L., Houwers, J., Straathof, A. J. J., van der Wielen, L. A. M., Luyben, K. Ch. A. M., and van den Tweel. W. J. J. (1997). Effect of dissociation equilibria on ion-exchange processes of weak electrolytes, AIChE. J., 43(1), 73-82

Kataoka, T., Sato, M., and Ueyama, K. (1968). Effective liquid phase diffusivity in ion exchange, J. Chem. Eng. Jpn., 1, 38-42

Kataoka, T., M. Sato, and K. Ueyama (1971) Influence of a noncounterion valence on liquid phase diffusion in ion exchange, Kogyo Kagaku Zasshi, 74, 1052-1058

Kataoka, T., H. Yoshida, and T. Yamada (1973) Liquid phase mass transfer in ion exchange based on the hydraulic radius model, J. Chem. Eng. Jpn., 6, 172-177

Kataoka, T., H. Yoshida, and Y. Shidahara (1976). Liquid phase mass transfer in exchange accompanied by chemical reactions, J. Chem. Eng. Jpn., 9, 130-136

Kataoka, T., Yoshida, H. and Uemura, T. (1987). Liquid-side ion exchange mass transfer in a ternary system. AIChE J., 33, 202-210

Koon, J. H. and Kaufman, W. J. (1975). Ammonia removal from municipal wastewaters by ion exchange. J. Water Pollut. Control Fed., 47(3), 448-465

Kun, K. A. and Kunin R. (1966). Removing colloidal silica by ion exchange, Ind. Water Eng., 3, 16-20

Kunin, R. and Myers R. J. (1950). Ion Exchange Resins, John Wiley & Sons, New York

Kunin, R. (1960). Elements of Ion Exchange, Reinhold Publishing Co., New York

Lagerstrom, G. (1959). Equilibrium studies of polyanions, III. Silicate ions in $NaClO_4$ medium, Acta Chem. Scand., 13, 722-736

Liberti, L. and Millar, J. R. (1985). Fundamentals and Applications of Ion Exchange, NATO ASI series, Martinus Nijhoff Publishers, Dordrecht

Lin, S. H. and Wu, C. L. (1996). Ammonia removal from aqueous solution by ion exchange. Ind. Eng. Chem. Res., 35, 553-558

Loewenthal, R. E. (1976). Carbonate chemistry of aquatic systems, An Arbor Science, An Arbor, Michigan

Lou, J (1997). Modeling of boron sorption equilibrium and kinetic studies of ion exchange with boron solution, Ph.D. Dissertation, Oklahoma State University, Stillwater, OK

Meichik, N. R., Lekkin, Y. A., Galitskaya, N. B., and Kosaeva, A. E. (1989). Kinetics of the sorption of boron by anion-exchange resins, Russ. J. Phys. Chem., 63(7), 1024-1025

Nakanishi, K (1978). Prediction of diffusion coefficient of nonelectrolytes in dilute solution based on generalized hammond-stokes plot, Ind. Eng. Chem. Fund., 17, 253-256

Newman, J. (1973). Electrochemical Systems, Prentice-Hall, Inc., Englewood Cliff, New Jersey

Noh, B. IL. (1992). Effects of step change in feed concentration and incomplete mixing of anion and cation resin on the performance of mixed-bed ion exchange. PhD dissertation, Oklahoma State University, Stillwater, OK

Omatete, O. O., Vermeulen, T. and Clazie, R. N. (1980a). Column dynamics of ternary ion exchange. Part I. Diffusion and mass transfer relations. Chem. Eng. J., 19, 229-240

Omatete, O. O., Vermeulen, T. and Clazie, R. N. (1980b). Column dynamics of ternary ion exchange. Part II. Diffusion and mass transfer relations. Chem. Eng. J., 19, 241-250

Pamarthy, S. (1995). Development of a column model to predict silica breakthrough in a mixed-bed ion exchanger, MS Thesis, Oklahoma State University, Stillwater, OK

Pate, K. T. (1991). Control of dissolved silica in High-purity systems. Ultrapure Water Journal, 8(4), 35-43

Pleijel, H. (1910). Die Potentialdiffernz zwishen zwei elektrolytischen Lösungen, Z. Phys, Chem., 72, 1-37

Pondugula, S. K. (1994). Mixed bed ion exchange modeling for divalent ions in a ternary system. MS Thesis, Oklahoma State University, Stillwater, OK

Porteous, A. (1975). Saline Water Distillation Process, Longman, London

Rahman, K. (1979). Film diffusion controlled kinetics in ternary ion exchange: Development of the basic equations. Chem. Eng. Res. Bull., 3, 27-30

Reid, R. C., Parusnitz, J. M. and Sherwook, T. K. (1987). The Properties of Gases and Liquids, 4th Ed, McGraw-Hill, New York

Rimstidt, J. D. and Barnes, H. L. (1980). The kinetics of silica-water reactions, Geochim. Cosmochim. Acta, 44, 1683-1699

Sadler, M. A. (1993). Developments in the Production and Control of Ultrapure Water, Ion Exchange Processes: Advances and Applications, A. Dyer, M. J. Hudson, and P. A. Williams, Editors, The Royal Society of Chemistry, Cambridge, UK, 15-28

Sahin, S. (1996). Mathematical model for adsorption of boric acid on a boron-specific ion exchanger. Bull. Chem. Soc. Jpn, 69, 1917-1920

Samuelson, O. (1963). Ion Exchange Separations in Analytical Chemistry, John Wiley And Sons, New York

Schlögl, R. (1954). Elektrodiffusion in freier Lösung und geladenen membranen, Z. Phys. Chem., 1, 305-339

Schlögl, R. and Helfferich , F. (1957). Comment on the significance of diffusion potentials in ion exchange kinetics, J. Chem. Phys., 26, 5-7

Siever, R. (1962). Silica solubility 0-200 °C and the diagenesis of siliceous sediments. Journal of Geology, 70, 127-150

Smith, T. G. and J. S. Dranoff (1964). Film-diffusion controlled kinetics of binary ion exchange. Ind. Eng. Chem. Fundam., 3, 195-200

Spiegler, K. S. and Laird, A. D. K. (1980). Principles of Desalination, Part B, pp577, Academic Press, New York

Streat, M. (1995). The Waters were made sweet: Advances in ion exchange technology, Ind. Eng. Chem., 34, 2841-2848

Stumm, W., Huper, H., and Champlin R. L. (1967). Formation of polysilicates as determined by coagulation effects, Environmental Sci. Tech., 1(3), 221-227

Sunkavalli, S. K. V. (1996). Development of generalized equilibrium and rate models to predict ion exchange column performance, MS Thesis, Oklahoma State University, Stillwater, OK

Thompson, H. S. (1850). On the adsorbent power of soils, Journal of the Royal Agricultural Society of England, 11, 68-74

Turner, J. C. R., and Snowdon, C. B. (1968). Liquid-side mass transfer in ion exchange: An examination of the Nernst-Planck model, Chem. Eng. Sci., 23, 221-230

Van Lier, J. A. (1960). The solubility of quartz, Drukkerji en Uitgeversmij v/h Kemink en Zoon n.v. Dom Plein, Ultrecht, Netherlands

Vermeulen, T. and Hiester, N. K. (1954). Ion exchange and adsorption column kinetics with uniform partial presaturation. J. Chem. Phys., 22, 96-101

Wachinski, A. M. (1997). Environmental Ion Exchange: principles and design, CRC/Lewis Publisher, Boca Raton, Florida

Wagner, J. D. and Dranoff, J. S. (1967). The kinetics of ion exchange accompanied by irreversible Reaction. III. Film diffusion controlled neutralization of a strong acid exchanger by a weak base. J. Phys. Chem., 71(13), 4551-4553

Way, J. T. (1850). Influence of time on the absorptive properties of soils, Journal of the Royal Agricultural Society of England, 25, 313-379

Wegmann, E. (1912). Ancient and modern water works, Eng. Rec., Vol. 65, June

Wildhagen, G. R. S., Qassim, R. Y., Rajagopal, K. and Rahman, K. (1985). Effective liquid-phase diffusivity in ion exchange, Ind. Eng. Chem. Fundam., 24, 423-432

Wong Y. and J. L. Niedawiecki (1982) A simplified model four multicomponent fixed bed adsorption, AIChE. Sym. Ser. 78 (219), 120-126

Yoon, T. K. (1990). The effect of cation to anion ratio on mixed-bed ion exchange performance at ultra-low concentrations, PhD Thesis, Oklahoma State University, Stillwater, OK

Yoshida, H. and Kataoka, T. (1987). Adsorption of amines on $H^+$-form ion exchanger, Chem. Eng. Sci., 42(7), 1805-1814

Yoshida, H., Shimizu, K. and Kataota, T. (1990). Adsorption of amine and paints on H-form resin from electrodeposition wastewater. AIChE. J. 36(12), 1815-1821

Yu, Q. and Wang, N.-H. L. (1989). Computer simulations of the dynamics of multicomponent ion exchange and adsorption in fixed beds: Gradient-directed moving finite element method. Computers Chem. Eng., 13(8), 915-926

Zecchini, E. J. (1990). Solutions to selected problems in multicomponent mixed bed ion exchange modeling, Ph.D. Dissertation, Oklahoma State University, Stillwater, OK

Zecchini, E. J. and Foutch, G. L (1991). Mixed bed ion exchange modeling with amine form cation resins, Ind. Eng. Chem. Res., 30(8), 1886-1892

Zoccolante, G. V. (1990). Produce ultrapure process water, Chem. Eng. Progress, 86(12), 69-72

# APPENDIXES

# APPENDIX A

# EQUILIBRIUM RELATIONSHIPS

## I. DISSOCIATION EQUILIBRIUM FOR TYPE I DISSOCIATIVE SPECIES

Type I dissociative species has the characteristic of one-step dissociation equilibrium. Monovalent amine is a typical type I dissociative species. Chowdiah (1996) and Sunkavalli (1996) reviewed the dissociation equilibrium for monovalent amines. Based on their derivations, a generalized dissociation equilibrium for type I species can be generated.

As we know, for monovalent amines,

$$RNH_2 + H_2O \xleftrightarrow{K_b} RNH_3^+ + OH^- \qquad \text{(A-1)}$$

$$K_b = \frac{[RNH_3^+][OH^-]}{[RNH_2]} \qquad \text{(A-2)}$$

and for monovalent organic acids,

$$RCOOH \xleftrightarrow{K_a} RCOO^- + H^+ \qquad \text{(A-3)}$$

$$K_a = \frac{[RCOO^-][H^+]}{[RCOOH]} \qquad \text{(A-4)}$$

Eq (A-2) and Eq (A-4) are the same form. Therefore, we can generalize the dissocaition equilibrium for the type I speces as

$$TypeI\_Mole \xleftrightarrow{K} TypeI\_Ion + Dum\_Ion \qquad \text{(A-5)}$$

$$K = \frac{[\mathrm{TypeI\_Ion}][\mathrm{Dum\_Ion}]}{[\mathrm{TypeI\_Mole}]} \tag{A-6}$$

where, the Dum_Ion could be $H^+$ or $OH^-$, depending the type I species is acid or base. If the total concentration of type I species is defined as

$$C_{TDI} = [\mathrm{TypeI\_Mole}] + [\mathrm{TypeI\_Ion}] \tag{A-7}$$

then from equations (A-6) and (A-7), we can derive the relationship between [TypeI_Ion] and [Dum_Ion], that is

$$[\mathrm{TypeI\_Ion}] = \frac{K\, C_{TDI}}{K + [\mathrm{Dum\_Ion}]} \tag{A-8}$$

This relation is used to eliminate [TypeI_Ion] from the charge balance equation, so that the charge balance equation contains only one unknown [Dum_Ion] for solving.

## II. DISSOCIATION EQUILIBRIUM FOR TYPE II DISSOCIATIVE SPECIES

Type II dissociative species has the characteristics of two-step dissociation. Carbonate is a typical type II dissociative species. Bulusu (1994) and Sunkavalli (1996) reviewed the water chemistry of carbonates, and the following dissociation equilibrium equations were presented:

$$CO_2 + H_2O \longleftrightarrow H_2CO_3 \tag{A-9}$$

$$H_2CO_3 \xleftrightarrow{K_1} H^+ + HCO_3^- \tag{A-10}$$

$$HCO_3^- \xleftrightarrow{K_2} H^+ + CO_3^{2-} \tag{A-11}$$

$$H_2O \xleftrightarrow{K_W} H^+ + OH^- \tag{A-12}$$

$$K_1 = \frac{[H^+][HCO_3^-]}{[H_2CO_3^*]} \tag{A-13}$$

$$K_2 = \frac{[H^+][CO_3^{2-}]}{[HCO_3^-]} \tag{A-14}$$

$$K_w = [H^+][OH^-] \tag{A-15}$$

where

$$[H_2CO_3^*] = [H_2CO_3] + [CO_2] \tag{A-16}$$

In general, for the type II species, we can generalized its dissociation equilibrium as

$$\text{TypeII\_Mole} \xleftrightarrow{K_1} \text{TypeII\_MonoIon} + \text{Dum\_Ion} \tag{A-17}$$

$$\text{TypeII\_MonoIon} \xleftrightarrow{K_2} \text{TypeII\_DiIon} + \text{Dum\_Ion} \tag{A-18}$$

$$K_1 = \frac{[\text{TypeII\_MonoIon}][\text{Dum\_Ion}]}{[\text{TypeII\_Mole}]} \tag{A-19}$$

$$K_2 = \frac{[\text{TypeII\_DiIon}][\text{Dum\_Ion}]}{[\text{TypeII\_MonoIon}]} \tag{A-20}$$

Where, Dum_Ion is a dummy ion, which could be $H^+$ or $OH^-$ depending the type II species is acid or base.

Given the total concentration $C_{TDII}$ for the type II dissociative species, that is

$$C_{TDII} = [\text{TypeII\_Mole}] + [\text{TypeII\_MonoIon}] + [\text{TypeII\_DiIon}] \tag{A-21}$$

the concentrations for monovalent ion [TypeII_MonoIon] and divalent Ion [TypeII_DiIon] can be determined.

From Eq (A-19) and Eq (A-20), we have

$$[\text{TypeII\_Mole}] = \frac{[\text{Dum\_Ion}][\text{TypeII\_MonoIon}]}{K_1} \tag{A-22}$$

$$[\mathrm{TypeII\_DiIon}] = \frac{K_2[\mathrm{TypeII\_MonoIon}]}{[\mathrm{Dum\_Ion}]} \tag{A-23}$$

Substituting Eq (A-22) and Eq (A-23) into Eq (A-21) would lead to

$$C_{TDII} = \frac{[\mathrm{Dum\_Ion}][\mathrm{TypeII\_MonoIon}]}{K_1} + [\mathrm{TypeII\_MonoIon}] + \frac{K_2[\mathrm{TypeII\_MonoIon}]}{[\mathrm{Dum\_Ion}]} \tag{A-24}$$

Then the concentration of monovalent ion of typeII species can be written as:

$$[\mathrm{TypeII\_MonoIon}] = \frac{C_{TDII}}{X} \tag{A-25}$$

where

$$X = \frac{[\mathrm{Dum\_Ion}]}{K_1} + \frac{K_2}{[\mathrm{Dum\_Ion}]} + 1 \tag{A-26}$$

The expressions for the concentrations of other type II species can be written as:

$$[\mathrm{TypeII\_Mole}] = \frac{[\mathrm{Dum\_Ion}]C_{TDII}}{K_1 X} \tag{A-27}$$

$$[\mathrm{TypeII\_DiIon}] = \frac{K_2 C_{TDII}}{[\mathrm{Dum\_Ion}]X} \tag{A-28}$$

The above relations Eq (A-25), (A-27) and (A-28) are used to express the concentrations of various forms of type II species in terms of the concentration of Dummy ion ($H^+$ or $OH^-$) in the charge balance equation, so that only one unknown [Dum_Ion] exists in the equation. Therefore, the charge balance equation can be solved.

## III. DISSOCIATION EQUILIBRIUM FOR TYPE III DISSOCIATIVE SPECIES

Type III species has the characteristics of three-step dissociations. For instance, triprotic acid – phosphate is a typical type III dissociation species. The dissociation equilibrium relationships for phosphate (Fast, 1984) are

$$H_3PO_4 \xleftrightarrow{K_1} H^+ + H_2PO_4^- \tag{A-29}$$

$$H_2PO_4^- \xleftrightarrow{K_2} H^+ + HPO_4^{2-} \tag{A-30}$$

$$HPO_4^{2-} \xleftrightarrow{K_3} H^+ + PO_4^{3-} \tag{A-31}$$

$$K_1 = \frac{[H^+][H_2PO_4^-]}{[H_3PO_4]} \tag{A-32}$$

$$K_2 = \frac{[H^+][HPO_4^{2-}]}{[H_2PO_4^-]} \tag{A-33}$$

$$K_3 = \frac{[H^+][PO_4^{3-}]}{[HPO_4^{2-}]} \tag{A-34}$$

Based on the above, we can generalize the dissociation equilibriums for type III species as

$$\mathrm{TypeIII\_Mole} \xleftrightarrow{K_1} \mathrm{Dum\_Ion} + \mathrm{TypeIII\_MonoIon} \tag{A-35}$$

$$\mathrm{TypeIII\_MonoIon} \xleftrightarrow{K_2} \mathrm{Dum\_Ion} + \mathrm{TypeIII\_DiIon} \tag{A-36}$$

$$\mathrm{TypeIII\_DiIon} \xleftrightarrow{K_3} \mathrm{Dum\_Ion} + \mathrm{TypeIII\_TriIon} \tag{A-37}$$

$$K_1 = \frac{[\mathrm{Dum\_Ion}][\mathrm{TypeIII\_MonoIon}]}{[\mathrm{TypeIII\_Mole}]} \tag{A-38}$$

$$\mathrm{K}_2=\frac{[\mathrm{Dum\_Ion}][\mathrm{TypeIII\_DiIon}]}{[\mathrm{TypeIII\_MonoIon}]} \tag{A-39}$$

$$\mathrm{K}_3=\frac{[\mathrm{Dum\_Ion}][\mathrm{TypeIII\_TriIon}]}{[\mathrm{TypeIII\_DiIon}]} \tag{A-40}$$

Where, Dum_Ion is a dummy ion, which could be $H^+$ or $OH^-$ depending the type III species is acid or base.

If the total concentration of type III species is given, that is

$$\mathrm{C}_{\mathrm{TDIII}}=[\mathrm{TypeIII\_Mole}]+[\mathrm{TypeIII\_MonoIon}]+[\mathrm{TypeIII\_DiIon}] + [\mathrm{TypeIII\_TriIon}] \tag{A-41}$$

then the concentration for each type III species can be determined.

From Eq (A-38), (A-39) and (A-40), we have

$$[\mathrm{TypeIII\_MonoIon}]=\frac{\mathrm{K}_1[\mathrm{TypeIII\_Mole}]}{[\mathrm{Dum\_Ion}]} \tag{A-42}$$

$$[\mathrm{TypeIII\_DiIon}]=\frac{\mathrm{K}_2[\mathrm{TypeIII\_MonoIon}]}{[\mathrm{Dum\_Ion}]}=\frac{\mathrm{K}_1\mathrm{K}_2[\mathrm{TypeIII\_Mole}]}{[\mathrm{Dum\_Ion}]^2} \tag{A-43}$$

$$[\mathrm{TypeIII\_TriIon}]=\frac{\mathrm{K}_3[\mathrm{TypeIII\_DiIon}]}{[\mathrm{Dum\_Ion}]}=\frac{\mathrm{K}_1\mathrm{K}_2\mathrm{K}_3[\mathrm{TypeIII\_Mole}]}{[\mathrm{Dum\_Ion}]^3} \tag{A-44}$$

Substituting Eq (A-42), (A-43) and (A-44) into Eq (A-41) leads to

$$\mathrm{C}_{\mathrm{TDIII}}=[\mathrm{TypeIII\_Mole}]+\frac{\mathrm{K}_1[\mathrm{TypeIII\_Mole}]}{[\mathrm{Dum\_Ion}]}+\frac{\mathrm{K}_1\mathrm{K}_2[\mathrm{TypeIII\_Mole}]}{[\mathrm{Dum\_Ion}]^2} + \frac{\mathrm{K}_1\mathrm{K}_2\mathrm{K}_3[\mathrm{TypeIII\_Mole}]}{[\mathrm{Dum\_Ion}]^3} \tag{A-45}$$

Solve for [TypeIII_Mole], we get

$$[\text{TypeIII\_Mole}] = \frac{C_{\text{TDIII}}}{X} \quad \text{(A-46)}$$

where

$$X = 1 + \frac{K_1}{[\text{Dum\_Ion}]} + \frac{K_1 K_2}{[\text{Dum\_Ion}]^2} + \frac{K_1 K_2 K_3}{[\text{Dum\_Ion}]^3} \quad \text{(A-47)}$$

Therefore, the expressions for other ionic forms of type III species are

$$[\text{TypeIII\_MonoIon}] = \frac{K_1}{[\text{Dum\_Ion}]} \frac{C_{\text{TDIII}}}{X} \quad \text{(A-48)}$$

$$[\text{TypeIII\_DiIon}] = \frac{K_1 K_2}{[\text{Dum\_Ion}]^2} \frac{C_{\text{TDIII}}}{X} \quad \text{(A-49)}$$

$$[\text{TypeIII\_TriIon}] = \frac{K_1 K_2 K_3}{[\text{Dum\_Ion}]^3} \frac{C_{\text{TDIII}}}{X} \quad \text{(A-50)}$$

Equations (A-48), (A-49) and (A-50) are used to express various forms of type III species in terms of the concentration of Dummy ion ($H^+$ or $OH^-$) in the charge balance equation, so that only one unknown exists in charge balance equation. This way the charge balance can be solved.

# APPENDIX B

# INTERFACIAL CONCENTRATIONS

## I. INTERFACIAL CONCENTRATIONS FOR IONS

Interfacial concentrations (solid-film interface) of ions are determined using ion exchange equilibria. Local equilibrium is assumed at the solid-film interface. The selectivity coefficient expression for a general case of ion B replacing ion A, can be written using mass action law, as

$$K_A^B = \left(\frac{q_B}{C_B^*}\right)^{Z_A} \left(\frac{C_A^*}{q_A}\right)^{Z_B} \tag{B-1}$$

where q is the resin phase concentration and $C^*$ is the interfacial concentration.

For any ion 'i' we can express the resin phase concentration and the interfacial concentration as

$$q_i = Y_i Q \tag{B-2}$$

$$C_i^* = X_i^* C_T^* \tag{B-3}$$

Eq (B-1) can be written in terms of equivalent fractions, total resin capacity and total interfacial concentration as follows:

$$K_A^B = \left(\frac{Y_B}{X_B^*}\right)^{Z_A} \left(\frac{X_A^*}{Y_A}\right)^{Z_B} Q^{(Z_A - Z_B)} C_T^{*(Z_B - Z_A)} \tag{B-4}$$

Rearranging the above equation, the interfacial fractional concentration of ion B can be expressed as

$$X_B^* = Y_B \left(K_A^B\right)^{-1/Z_A} \left(\frac{X_A^*}{Y_A}\right)^{Z_B/Z_A} \left(\frac{Q}{C_T^*}\right)^{1-Z_B/Z_A} \tag{B-5}$$

Generalizing the above expression, the interfacial fractional concentration of an ion 'i' exchanging for an ion 'A' can be written as

$$X_i^* = Y_i \left(K_A^i\right)^{-1/Z_A} \left(\frac{X_A^*}{Y_A}\right)^{Zi/Z_A} \left(\frac{Q}{C_T^*}\right)^{1-Zi/Z_A} \tag{B-6}$$

If we have 'n-1' ions exchanging for ion A we can write 'n-1' such equations. But we have 'n' unknowns. The extra equation needed to completely specify the system is obtained from material balance (summation relation) at the solid-film interface.

$$\sum_{i=1}^{n} X_i^* = 1.0 \tag{B-7}$$

Rewriting the interfacial fractional concentration (Eq B-6) as

$$X_i^* = \lambda_i \left(X_A^*\right)^{\frac{Z_i}{Z_A}} \tag{B-8}$$

where $\lambda_i = Y_i \left(K_A^i\right)^{-1/Z_A} \left(Y_A\right)^{-Z_i/Z_A} \left(\frac{Q}{C_T^*}\right)^{1-Z_i/Z_A}$ (B-9)

and substituting these in equation (B-7) we have

$$X_A^* + \sum_{i=1}^{n-1} \lambda_i \left(X_A^*\right)^{\frac{Z_i}{Z_A}} = 1.0 \tag{B-10}$$

This is a polynomial in $X_A^*$ and could be solved using an iteration technique. Regula-Falsi method was used in this work. In the above equations, the total interfacial concentration, $C_T^*$, is still unknown, but it can be determined by the try-and-error method. The expression for $C_T^*$ is derived in Appendix C.

## II. INTERFACIAL CONCENTRATIONS FOR NONIONIC FORMS

When weak electrolytes are involved in ion exchange systems, the interfacial concentrations of nonionic (molecular) forms need to be known to calculate the rate. In this work, the dissociation equilibrium relationship between ionic and nonionic (molecular) forms is used to determine interfacial concentrations of the nonionic forms, that is,

$$[AH]^* = \frac{[A^-]^*[H^+]^*}{K_a} = \frac{[A^-]^* K_w}{K_a[OH^-]^*} \qquad \text{(for acids)} \qquad \text{(B-11)}$$

$$[BOH]^* = \frac{[B^+]^*[OH^-]^*}{K_b} = \frac{[B^+]^* K_w}{K_b[H^+]^*} \qquad \text{(for bases)} \qquad \text{(B-12)}$$

where $K_a$ and $K_b$ are the dissociation constants for acid and base respectively; $[H^+]$ and $[OH^-]$ are the interfacial concentrations of hydrogen and hydroxyl respectively; $[A^-]^*$ and $[B^+]^*$ are the interfacial concentrations for ionic forms, which can be determined using the method mentioned above (selectivity expression).

# APPENDIX C

## FLUX EXPRESSIONS AND PARTICLE RATE

### I. IONIC FLUX EXPRESSION

Flux expressions for ions in multicomponent ion exchange are usually derived using Nernst-Planck model and basic principles of ion exchange. Haub and Foutch (1984), Zecchini and Foutch (1990) had successfully applied Nernst-Planck model to describe film diffusion-controlled mixed-bed ion exchange fluxes for monovlanet ions (binary or ternary systems). In addition, Pondugula (1994) further developed the flux expressions for divalent systems. But to handle multicomponent multivalent MBIE systems, generalized expression is required. In this work, the method proposed by Franzreb et al. (1993) is followed to derive the flux expressions for multicomponent, multivalent systems.

If neglecting the curvature of film, the Nernst-Planck equation can be expressed as

$$J_i = -D_i\left[\frac{\partial C_i}{\partial r} + \frac{C_i Z_i F}{RT}\frac{\partial \phi}{\partial r}\right] \qquad \text{(C-1)}$$

where $\phi$ is the electric potential and $Z_i$ is the ion valence. Assuming pseudo steady state allows us to replace the partial derivatives by ordinary derivatives. The flux expressions derived in this model are based on bulk-phase neutralization.

The conditions that must be satisfied within the film surrounding the resin are:

$$\sum Z_i C_i = \sum Z_j C_j \qquad \text{(Electroneutrality, charge balance)} \qquad \text{(C-2)}$$

where 'i' stands for counterions and 'j' represents coions.

$$Z_j J_j = 0 \qquad \text{(No coion flux)} \qquad \text{(C-3)}$$

$$\sum Z_i J_i = \sum Z_j J_j \qquad \text{(No net current flow)} \qquad \text{(C-4)}$$

From Eq (C-3) and Eq (C-4) we have

$$\sum Z_i J_i = 0 \qquad \text{(No net current flow)} \qquad \text{(C-5)}$$

The total equivalent ion concentration can be defined as:

$$C_T = \omega \sum_{i=1}^{n} Z_i C_i = \omega_j \sum_{j=1}^{m} Z_j C_j \qquad \text{(C-6)}$$

where, *n* is the number of counterions, *m* is the number of coions. And ω =+1 for cations and -1 for anions.

Using the no coion flux condition – Eq (C-3), we have

$$\frac{d\phi}{dr} = -\frac{RT}{F} \frac{Z_j \dfrac{dC_j}{dr}}{Z_j^2 C_j} \qquad \text{(C-7)}$$

From the no coion flux condition we have that the sum of the coion fluxes in the film is also zero. Now the electric potential term in the Nernst-Planck equation can be eliminated in terms of the total equivalent concentration as:

$$\frac{d\phi}{dr} = -\frac{RT}{F}\frac{\sum_{j=1}^{m} Z_j \frac{dC_j}{dr}}{\sum_{j=1}^{m} Z_j^2 C_j} \tag{C-8}$$

Introducing a mean coion valence defined as

$$Z_Y = \frac{\sum_{j=1}^{m} Z_j^2 C_j}{\sum_{j=1}^{m} Z_j C_j} \tag{C-9}$$

and combining with the definition for total concentration – Eq (C-6), the Eq (C-8) can be reduced to

$$\frac{d\phi}{dr} = \frac{-RT}{Z_Y F}\frac{1}{C_T}\frac{dC_T}{dr} \tag{C-10}$$

Now the Nernst-Planck expression for counterions can be written as:

$$J_i = -D_i\left(\frac{dC_i}{dr} - \frac{C_i Z_i}{C_T Z_Y}\frac{dC_T}{dr}\right) \tag{C-11}$$

Applying the no net current flow condition - Eq (C-5) to Eq (C-11), we get

$$\sum_{i=1}^{n} Z_i D_i \frac{dC_i}{dr} + \sum_{i=1}^{n} Z_i D_i N_i \frac{C_i}{C_T}\frac{dC_T}{dr} = 0 \tag{C-12}$$

where, $N_i = -\frac{Z_i}{Z_Y}$. (C-13)

For monovalent system of ions or equal valence system of ions, the above equation could be easily integrated to obtain a relation between $C_i$ and $C_T$. This is not possible in the case of arbitrary valences. At this point the method proposed by Franzreb (1993) is used to proceed further. In this method, Eq (C-11) is differentiated to eliminate the unknown $J_i$. This leads to a homogeneous second order differential equation:

$$\frac{d^2C_i}{dr^2}+\frac{N_i}{C_T}\frac{dC_i}{dr}\frac{dC_T}{dr}+N_i\frac{C_i}{C_T}\left(\frac{d^2C_T}{dr^2}-\frac{1}{C_T}\left(\frac{dC_T}{dr}\right)^2\right)=0 \qquad \text{(C-14)}$$

This method leads to an exact solution for the case of equal valences and only an approximation for the case of arbitrary valences. For counterions of equal valences, summation of Eq (C-14) for all the ions leads to

$$\sum_{i=1}^{n}\frac{d^2C_i}{dr^2}+\frac{1}{C_T}\frac{dC_T}{dr}\sum_{i=1}^{n}N_i\frac{dC_i}{dr}+\frac{1}{C_T}\frac{d^2C_T}{dr^2}\sum_{i=1}^{n}N_iC_i-\frac{1}{C_T^2}\left(\frac{dC_T}{dr}\right)^2\sum_{i=1}^{n}N_iC_i=0$$

(C-15)

Substituting Eq (C-6) and its derivatives in the above leads to

$$\frac{d^2C_T}{dr^2}=0 \qquad \text{(C-16)}$$

From the above equation it can be understood that for the case of counterions of equal valences, the profile of the total concentration in the film is linear. Zecchini and Foutch (1990) arrived at the same conclusion in their model for univalent ternary ions. The above equation combined with Eq (C-6) can be used to obtain relationships between the derivatives of $C_i$ and $C_T$. Substitution of all these derivatives in Eq (C-15) leads to

$$\frac{d^2C_i}{dC_T^2}+\frac{N_i}{C_T}\frac{dC_i}{dC_T}-\frac{N_iC_i}{C_T^2}=0 \qquad \text{(C-17)}$$

This is the Euler's differential equation the solution of which is

$$Z_iC_i = A_iC_T + B_iC_T^{-P} \qquad \text{(C-18)}$$

For the case of equal valences, we have $P = N_i$. Using the boundary conditions

$$r = 0, C_T = C_T^* \qquad \text{(C-19)}$$

$$r = \delta, C_T = C_T^o \qquad \text{(C-20)}$$

the values of the parameters $A_i$ and $B_i$ can be determined as follows:

$$A_i = \frac{1}{C_T^o}\left(Z_iC_i^o - B_i\left(C_T^o\right)^{-P}\right) \qquad \text{(C-21)}$$

and

$$B_i = \omega\frac{X_i^* - X_i^o}{\left(C_T^*\right)^{-P-1}-\left(C_T^o\right)^{-P-1}} \qquad \text{(C-22)}$$

Eq (C-18) gives us a relation between the individual ion concentrations, $C_i$ and the total equivalent concentration, $C_T$. Substituting for $C_i$ and its derivative in the modified Nernst-Planck Equation (C-11), we get the following flux expression:

$$J_i = -\frac{D_i}{Z_i}\frac{dC_T}{dr}\left[\left(A_i - PB_iC_T^{-P-1}\right)+N_i\left(A_i + B_iC_T^{-P-1}\right)\right] \qquad \text{(C-23)}$$

For the case of arbitrary valences, Eq (C-18) is only an approximation. In this case however, $N_i$ is not the same for all the counterions and hence, P cannot be equal to $N_i$. Combining the above equation with the condition of no net current flow (Eq C-5) results in

$$\left(\sum_{i=1}^{n} D_i A_i + \sum_{i=1}^{n} N_i D_i A_i\right) + \left(\sum_{i=1}^{n} N_i D_i B_i - P\sum_{i=1}^{n} D_i B_i\right) C_T^{-P-1} = 0 \qquad \text{(C-24)}$$

The only way the above equation can hold true is when both the terms are equal to zero. That leads to

$$\sum_{i=1}^{n} (1 + N_i) D_i A_i = 0 \qquad \text{(C-25)}$$

Substitution of $A_i$ (Eq C-21) into the above and with some mathematical manipulations give the desired expression for total interfacial concentration, $C_T^*$:

$$C_T^* = \left(\frac{\sum_{i=1}^{n} (1 + N_i) D_i X_i^o}{\sum_{i=1}^{n} (1 + N_i) D_i X_i^*}\right)^{1/P+1} C_T^o \qquad \text{(C-26)}$$

Equating the second parentheses term to zero and substitution of $B_i$ would give us the expression for the exponent P as:

$$P = \frac{\sum_{i=1}^{n} N_i D_i \left(X_i^* - X_i^o\right)}{\sum_{i=1}^{n} D_i \left(X_i^* - X_i^o\right)} \qquad \text{(C-27)}$$

In an equal valance case, P is equal to Ni and in an arbitrary case, it would be in the neighborhood of Ni. The concentrations involved in this work are very low and lead to lot of numerical errors and instability. Because of numerical discrepancies, sometimes the value of P computed in the code is unusually high and leads to problems in further computations. Hence the expression for P (Eq C-27) is modified as follows:

$$P = \frac{\sum_{i=1}^{n} N_i D_i \left| X_i^* - X_i^o \right|}{\sum_{i=1}^{n} D_i \left| X_i^* - X_i^o \right|} \qquad \text{(C-27A)}$$

Once again, substituting the above equations into Eq (C-11) and integrating between the boundary conditions given earlier, we would get the final desired form of the ionic flux expression:

$$J_i = \frac{D_i}{\delta}\left( (1 - \frac{N_i}{P})(C_i^* - C_i^o) + N_i A_i (1 + \frac{1}{P})(C_T^* - C_T^o) \right) \qquad \text{(C-28)}$$

Particle Rates

The rate of exchange is related to the flux of the species by

$$\frac{dq_i}{dt} = -J_i a_s \qquad \text{(C-29)}$$

The resin phase concentration $q_i$ can be represented as:

$$q_i = y_i Q \qquad \text{(C-30)}$$

Now Eq (C-29) can be written as

$$\frac{dy_i}{dt} = \frac{-J_i a_s}{Q} \tag{C-31}$$

The rate of ion loadings in to the resin can be determined using the above equation once the individual ionic fluxes are known.

The effective diffusivity is defined as:

$$D_e = \frac{\sum_{i=1}^{n} |J_i \delta|}{\sum_{i=1}^{n} |C_i^* - C_i^o|} \tag{C-32}$$

The film thickness in Eq (C-28) is eliminated using the relation

$$\delta = D_e / K \tag{C-33}$$

where K is a mass transfer coefficient found from Dwivedi and Upadhyay's correlation (1979)

$$K = \frac{D_i}{d_p} Sc^{1/3} Re \left[ \frac{0.765}{(\varepsilon Re)^{0.82}} + \frac{0.365}{(\varepsilon Re)^{0.386}} \right] \tag{C-34}$$

In the above Equation, $D_i$ is the diffusivity of ion 'i', $d_p$ is the particle diameter, Re is Reynolds number, Sc is Schmidt number and $\varepsilon$ is void fraction.

Schmidt number (Sc) is defined using the effective diffusivity as

$$Sc = \frac{\mu}{\rho D_e} \tag{C-35}$$

Substituting Eq (C-33) in the flux expression (Eq C-28), we get

$$\frac{J_i}{K} = \frac{D_i}{D_e}\left( (1 - \frac{N_i}{P})(C_i^* - C_i^o) + N_i A_i (1 + \frac{1}{P})(C_T^* - C_T^o) \right) \tag{C-36}$$

This $J_i/K$ is computed for each of the ions in the subroutines and returned to the main program in the computer code developed for this model.

## II. FLUXES EXPRESSION FOR NONIONIC SPECIES

The fluxes of nonionic (molecular) form species are usually described by Fick's second law, that is

$$J_{Mol} = -D_{Mol} \frac{dC_{Mol}}{dz} \tag{C-37}$$

Integrating Eq (C-37) yields

$$J_{Mol,i} = \frac{D_{Mol,i}}{\delta}(C^o_{Mol,i} - C^*_{Mo,il}) = k_{Mol,i}(C^o_{Mol,i} - C^*_{Mol,i}) \tag{C-38}$$

where, $k_{Mol,i} = \frac{D_{Mol,i}}{\delta}$ is molecular mass transfer coefficient.

Frequently in a multicomponent system with dissociative species, the concentration of molecular form is much greater than that of ionic form, so that the effect of molecular concentration on the total mass transfer rate has to be taken into account

# APPENDIX D

## COLUMN MATERIAL BALANCE

To determine the effluent concentration profile for a MBIE column, the column material balance equation is required. If neglecting the axial dispersion, the overall column material balance for species 'i' in an ion exchange column can be expressed as

$$\frac{u_s}{\varepsilon}\frac{\partial C_i}{\partial Z}+\frac{\partial C_i}{\partial t}+\frac{(1-\varepsilon)}{\varepsilon}\frac{\partial q_i}{\partial t}=0 \tag{D-1}$$

where $u_s$= superficial velocity, and $\varepsilon$ = void fraction.

For easy solving, the column material equation can be simplified by using dimension-less variables in time and distance. The dimension-less expressions are defined as:

$$\tau=\frac{K_i C_T^f}{d_p Q}\left(t-\frac{\varepsilon Z}{u_s}\right) \tag{D-2}$$

and

$$\xi=\frac{K_i(1-\varepsilon)}{u_s}\frac{Z}{d_p} \tag{D-3}$$

$K_i$ is the non-ionic mass transfer coefficient for species i, $d_p$ is the particle diameter, Q is the resin capacity and $C_T^f$ is the total cationic feed concentration. The above expressions are differentiated with respect to time and distance respectively to yield:

$$\frac{\partial\tau}{\partial t}=\frac{K_i C_T^f}{d_p Q}, \qquad \frac{\partial\tau}{\partial Z}=\frac{K_i C_T^f \varepsilon}{d_p Q u_s},$$

$$\frac{\partial \xi}{\partial t} = 0 \quad \text{and} \qquad \frac{\partial \xi}{\partial Z} = \frac{K_i C_f^T \varepsilon}{d_P Q u_s}$$

Using the chain rule, the original derivatives are expressed as:

$$\frac{\partial C_i}{\partial Z} = \frac{\partial C_i}{\partial \xi}\left(\frac{\partial \xi}{\partial Z}\right) + \frac{\partial C_i}{\partial \tau}\left(\frac{\partial \tau}{\partial Z}\right) = \frac{K_i(1-\varepsilon)}{u_s d_p}\left(\frac{\partial C_i}{\partial \xi}\right) - \frac{K_i C_T^f \varepsilon}{d_p u_s Q}\left(\frac{\partial C_i}{\partial \tau}\right) \tag{D-4}$$

$$\frac{\partial q_i}{\partial t} = \frac{\partial q_i}{\partial \tau}\left(\frac{\partial \tau}{\partial t}\right) + \frac{\partial q_i}{\partial \xi}\left(\frac{\partial \xi}{\partial t}\right) = \frac{K_i C_T^f}{d_p Q}\frac{\partial q_i}{\partial \tau} + 0\,\frac{\partial q_i}{\partial \xi} \tag{D-5}$$

Replacing these into the material balance yields:

$$\frac{\partial C_i}{\partial \xi} + \frac{C_T^f}{Q}\frac{\partial q_i}{\partial \tau} = 0 \tag{D-6}$$

This expression is easier to handle. Introducing the fractions in liquid phase and resin phase, that is, $x_i = C_i / C_T^f$ and $q_i = y_i Q$, and then substituting into the material balance equation yields:

$$\frac{\partial x_i}{\partial \xi} + \frac{\partial y_i}{\partial \tau} = 0 \tag{D-7}$$

In the current code, chloride is selected as the reference species. Since all the material balance is to be solved using same steps in $\tau$ and $\xi$, expressions for the base species result as:

$$\tau = \tau_c = \frac{K_c C_T^f}{d_{pa} Q_a}\left(t - \frac{\varepsilon Z}{u_s}\right) \tag{D-8}$$

and

$$\xi = \xi_c = \frac{K_c(1-\varepsilon)}{u_s d_{pa}} Z \tag{D-9}$$

The partial derivatives of all the species can be written in terms of the reference ion as follows: Cations:

$$\frac{\partial x_i}{\partial \xi_c} = \frac{\partial x_i}{\partial \xi_i}\left(\frac{\partial \xi_i}{\partial \xi_c}\right) = \frac{K_i}{K_c}\frac{d_{pa}}{d_{pc}}\frac{\partial x_i}{\partial \xi_i} \tag{D-10}$$

$$\frac{\partial y_i}{\partial \tau_c} = \frac{\partial y_i}{\partial \tau_i}\left(\frac{\partial \tau_i}{\partial \tau_c}\right) = \frac{K_i}{K_c}\frac{d_{pa}}{d_{pc}}\frac{Q_a}{Q_c}\frac{\partial y_i}{\partial \tau_i} \tag{D-11}$$

and anions:

$$\frac{\partial x_i}{\partial \xi_c} = \frac{\partial x_i}{\partial \xi_i}\left(\frac{\partial \xi_i}{\partial \xi_c}\right) = \frac{K_i}{K_c}\frac{\partial x_i}{\partial \xi_i} \tag{D-12}$$

$$\frac{\partial y_i}{\partial \tau_c} = \frac{\partial y_i}{\partial \tau_i}\left(\frac{\partial \tau_i}{\partial \tau_c}\right) = \frac{K_i}{K_c}\frac{\partial y_i}{\partial \tau_i} \tag{D-13}$$

Replacing these partial derivatives into the general material balance equation and introducing the cation and anion resin volume fractions (FCR, FAR) within the bed, we get

$$\frac{\partial x_i}{\partial \xi_c} + \mathrm{FCR}\frac{\partial y_i}{\partial \tau_c} = 0 \qquad \text{(cations)} \tag{D-14}$$

$$\frac{\partial x_i}{\partial \xi_c} + \mathrm{FAR}\frac{\partial y_i}{\partial \tau_c} = 0 \qquad \text{(anions)} \tag{D-15}$$

Now the rate expressions developed earlier have to be modified to incorporate the dimensionless variables that have been introduced. This involves changing t to $\tau_C$ as the basis for each of the individual ions.

$$\frac{dy_i}{dt} = \frac{-J_i a_s}{Q} \tag{D-16}$$

Changing from t to $\tau_i$ results:

$$\frac{dy_i}{d\tau} = \left(\frac{-J_i a_s}{Q}\right)\left(\frac{d_p Q}{K_i C_f}\right) \tag{D-17}$$

Now changing from $\tau$ to $\tau_C$ basis and noting that $a_S d_p = 6$ (for spherical particle), we get

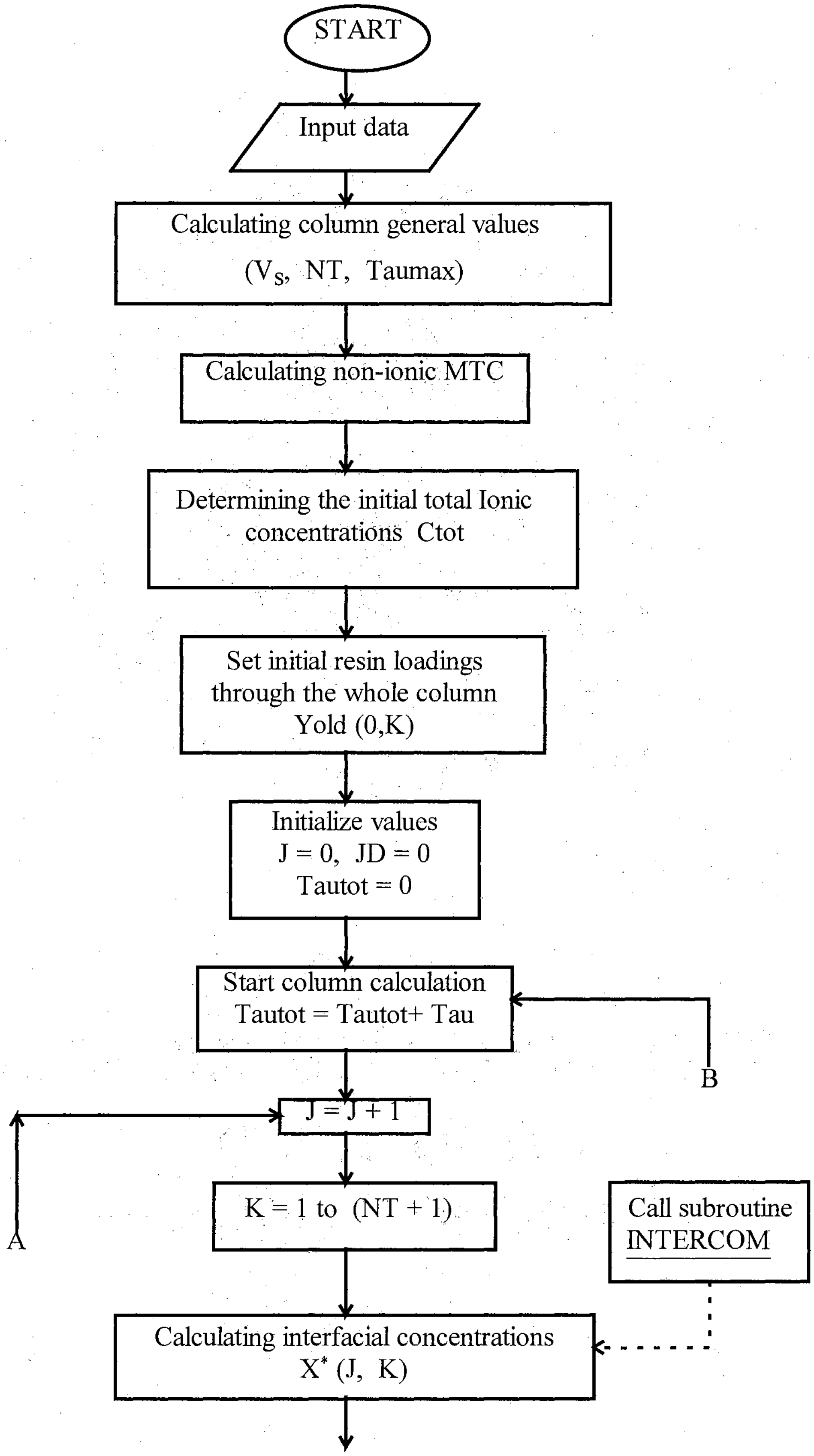
$$\frac{\partial y_i}{\partial \tau_c} = \left(-\frac{6J_i}{K_i C_f}\right)\frac{K_i}{K_c}\frac{d_{pa}}{d_{pc}}\frac{Q_a}{Q_c} \qquad \text{(cations)} \tag{D-18}$$

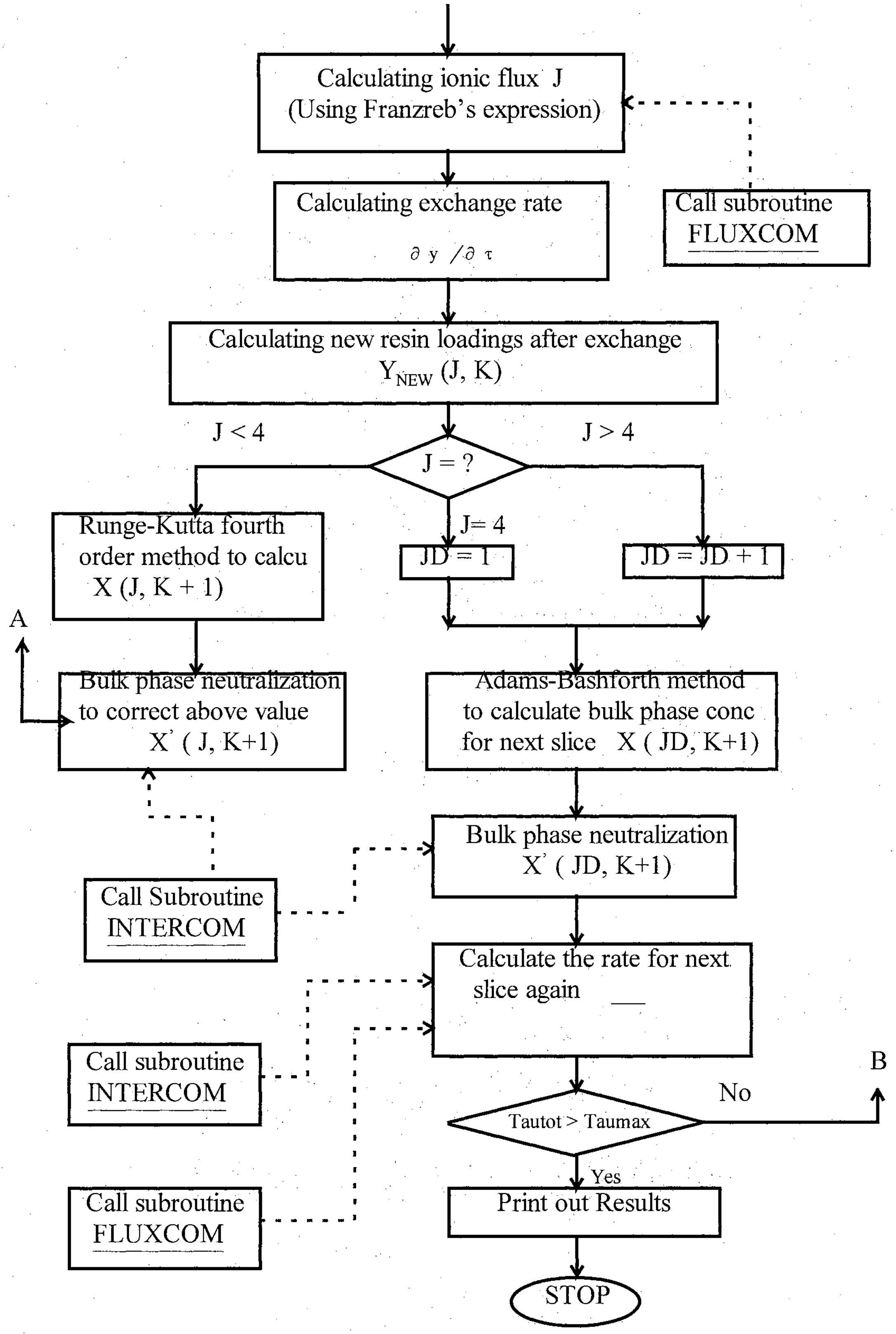
$$\frac{\partial y_i}{\partial \tau_c} = \left(-\frac{6J_i}{K_i C_f}\right)\frac{K_i}{K_c} \qquad \text{(anions)} \tag{D-19}$$

This is the final form of the dimensionless rate equations that describe the exchange process. These are combined with the material balances to predict the column effluent concentrations.

# APPENDIX E

# SCHEMATIC DIAGRAM OF MODEL ALGORITHM


START
Input data
Calculating column general values
(V_S, NT, Taumax)
Calculating non-ionic MTC
Determining the initial total Ionic concentrations Ctot
Set initial resin loadings through the whole column Yold (0,K)
Initialize values
J = 0, JD = 0
Tautot = 0
Start column calculation
Tautot = Tautot+ Tau
B
J = J + 1
A
K = 1 to (NT + 1)
Call subroutine INTERCOM
Calculating interfacial concentrations
X* (J, K)


Calculating ionic flux J
(Using Franzreb's expression)
Call subroutine
FLUXCOM
Calculating exchange rate
$\partial y / \partial \tau$
Calculating new resin loadings after exchange
$Y_{NEW}$ (J, K)
J < 4
J > 4
J = ?
J= 4
Runge-Kutta fourth
order method to calcu
X (J, K + 1)
JD = 1
JD = JD + 1
A
Bulk phase neutralization
to correct above value
X' ( J, K+1)
Adams-Bashforth method
to calculate bulk phase conc
for next slice X ( JD, K+1)
Call Subroutine
INTERCOM
Bulk phase neutralization
X' ( JD, K+1)
Calculate the rate for next
slice again
Call subroutine
INTERCOM
B
No
Tautot > Taumax
Yes
Call subroutine
FLUXCOM
Print out Results
STOP

# APPENDIX F

# COMPUTER SOURCE CODE

```
**************************************************************************
*     MULTICOMPONENT MIXED-BED ION-EXCHANGE CODE FOR "EPRI"              *
*        (This is the generalized version that can handle different      *
*     C/A ratios, variable inlet concentrations and variable inlet       *
*     flowrate)                                                          *
*                                                                        *
*     This program predicts the effluent concentrations of a             *
*     multicomponent system in which there could be some cationic        *
*     and anionic dissociative species.                                  *
*                                                                        *
*     In this system,                                                    *
*     The number of non-dissociative cations may be:    8                *
*     The number of non-dissociative anions may be:     8                *
*     The number of dissociative species may be                          *
*                type 1 dissociative cations:           5                *
*                   (like monovalent amines)                             *
*                type 1 dissociative anions:            5                *
*                   (like formate/acetate acids)                         *
*                type 2 dissociative cations:           2                *
*                   (like divalent amines)                               *
*                type 2 dissociative anions:            2                *
*                   (like carbonates)                                    *
*                type 3 dissociative cations:           3                *
*                   (like trivalent amines)                              *
*                type 3 dissociative cations:           3                *
*                   (like phosphate acids)                               *
*                No of silica ions:                     3                *
*                    (H3SiO4-, H2SiO4=, PS=)                             *
*                                                                        *
*            Coded by:    Liu, Vinay, Hussey & Dr. Foutch                *
*                         Oklahoma State University                      *
*            Last updated by LIU, Aug 12, 1998                           *
*                                                                        *
*------------------------------------------------------------------------*
*     DISCLAIMER:                                                        *
*         This computer code is the intellectual property of OSU and     *
*     cannot be used without the explicit permission of Oklahoma State   *
*     University.                                                        *
*========================================================================*
*     NOTATION:                                                          *
*                                                                        *
*     CAT     - Suffix for cations                                       *
*     ANI     - Suffix for anions                                        *
*     D1C     - Suffix for amines(or type 1 dissociative cations)        *
*     D1A     - Suffix for carboxylic acids (or type 1 disso anion)      *
*     D2C     - Suffix for divalent amines  (type 2 disso cations)       *
*     D2A     - Suffix for carbonates  (type 2 disso anions)             *
*     D1CSTA  - Suffix for molecular amines (or type 1 disso cations)    *
*     D1ASTA  - Suffix for molecular carboxylic acids (type 1 anions)    *
*     D2CSTA  - Suffix for molecular divalent amines                     *
*     D2ASTA  - Suffix for molecular carbonate                           *
*                                                                        *
*     INDICES:                                                           *
*                                                                        *
*     II      - CATIONS                                                  *
*     JJ      - ANIONS                                                   *
*     KK      - AMINES (or type 1 dissociative cation)                   *
*     LL      - CARBONATES (type 2 disso anions)                         *
*     MM      - CARBOXYLIC ACIDS (or type 1 disso anions)                *
*     NN      - DIVALENT AMINES (type 2 disso cations)                   *
*     KT      - TRIVALENT AMINES (type 3 disso cations)                  *
*     KP      - PHOSPHATES (type 3 disso anions)                         *
*                                                                        *
*========================================================================*
*     NOMENCLATURE OF VARIABLES                                          *
*                                                                        *
*     ANI_DIA           Anion diameter (cm)                              *
*     ANI_CAP           Anion capacity (meq/ml)                          *
*     BED_DIA           Bed diameter (cm)                                *
*     CAT_DIA           Cation diameter (cm)                             *
*     CAT_CAP           Cation capacity (meq/ml)                         *
```

```
*     CFCAT (I)         Feed concentration of cation i                  *
*     CFANI (I)         Feed concentration of anion i                   *
*     CFA               Total feed concentration of anions              *
*     CFC               Total feed concentration of cations             *
*     CF                Total feed concentration                        *
*     CBCAT (I)         Bulk phase concentration of cation i            *
*     CBANI (I)         Bulk phase concentration of anion i             *
*     CBD1C (I)         Bulk phase concentration of amine i             *
*     CBD1CSTA (I)      Bulk phase concentration of molecular amine     *
*     CD1CSTAI (I)      Interfacial concentration of molecular amine    *
*     CBD2A (I)         Bulk phase concentration of carbonate i         *
*     CBD2ASTA          Bulk phase concentration of molecular carbonate *
*     CD2ASTAI          Interfacial concentration of molecular carb     *
*     CBD1A (I)         Bulk phase concentration of RCOOH               *
*     CBD1ASTA (I)      Bulk phase concentration of molecular RCOOH     *
*     CD1ASTAI (I)      Interfacial concentration of molecular RCOOH    *
*     CBC (I)           Combined array of CBCAT (I) and CBCOOH (I)      *
*     CBA (I)           Combined array of CBANI (I) and CBD2A (I)       *
*     CFEED_D1C(I)      Feed concentration of amine i                   *
*     CFEED_D2A         Feed concentration of carbonate                 *
*     CFEED_D1A (I)     Feed concentration of RCOOH                     *
*     CTO_D1C (I)       Total concentration of amine i                  *
*     CTO_D2A           Total concentraton of carbonate                 *
*     CTO_D1A (I)       Total concentration of RCOOH i                  *
*     CTCI              Total interfacial concentration of all cations  *
*     CTAI              Total interfacial concentration of all anions   *
*     CHTD              Dimensionless column height                     *
*     DEC               Effective diffusivity of cations                *
*     DEA               Effective diffusivity of anions                 *
*     DEN               Density of water (g/cm3)                        *
*     DISSW             Dissociation constant of water                  *
*     DIFU_CAT (I)      Diffusivity of cation i (cm2/s)                 *
*     DIFU_ANI (I)      Diffusivity of anion i (cm2/s)                  *
*     DIFU_D1C (I)      Diffusivity of amine i (cm2/s)                  *
*     DIFU_D1CSTA(I)    Diffusivity of molecular amine (cm2/s)          *
*     DIFU_D2A (I)      Diffusivity of carbonates (cm2/s)               *
*     DIFU_D2ASTA       Diffusivity of molecular carbonate(cm2/s)       *
*     DIFU_D1A (I)      Diffusivity of RCOOH i  (cm2/s)                 *
*     DIFU_D1ASTA (I)   Diffusivity of molecular RCOOH i (cm2/s)        *
*     DIFUCOMC (I)      Combined array of cationic diffusivities        *
*     DIFUCOMA (I)      Combined array of anionic diffusivities         *
*     DISSD1C (I)       Dissociation constant of amines                 *
*     DISSD2A (I)       Dissociation constant of carbonates             *
*     DISSD1A (I)       Dissociation constant of RCOOH                  *
*     EWANI(I)          Equivalent weight of cation i                   *
*     EWCAT(I)          Equivalent weight of anion i                    *
*     EWD1C(I)          Equivalent weight of amine i                    *
*     EWD1CSTA(I)       Equivalent weight of molecular amine            *
*     EWD2A(I)          Equivalent weight of carbonate i                *
*     EWD2ASTA          Equivalent weight of molecular carbonate        *
*     EWD1A (I)         Equivalent weight of RCOOH i                    *
*     EWD1ASTA (I)      Equivalent weight of molecualr RCOOH            *
*     FCR               Volume fraction of cationic resin               *
*     FAR               Volume fraction of anionic resin                *
*     INDEXC(I)         Index of cation i                               *
*     INDEXA(I)         Index of anion i                                *
*     MTC_ANI (I)       Mass transfer coefficient of anion i (cm/s)     *
*     MTC_CAT (I)       Mass transfer coefficient of cation i           *
*     MTC_D1C (I)       Mass transfer coefficient of amine i            *
*     MTC_D1CSTA (I)    Mass transfer coefficient of molecular amine    *
*     MTC_D2A (I)       Mass transfer coefficient of carbonate  i       *
*     MTC_D2ASTA        Mass transfer coefficient of molecular carb     *
*     MTC_D1A (I)       Mass transfer coefficient of RCOOH              *
*     MTC_D1ASTA (I)    Mass transfer coefficient of RCOOH              *
*     MTC_REF           Mass transfer coefficient of reference ion      *
*     NOCAT             Number of non-dissociative cations              *
*     NOANI             Number of non-dissociative anions               *
*     NOD1C             Number of amines                                *
*     NOD2A             Number of carbonates                            *
*     NOD1A             Number of carboxylic acids                      *
*     NC                Number of all cations (NOCAT+NOD1C+NOD2C)       *
```

```
*     NA                Number of all anions (NOANI+NOD1A+NOD2A)         *
*     NOSLICE           Number of slices                                 *
*     NEXC(I)           Number of exchanging cations in slice i          *
*     NEXA(I)           Number of exchanging anions in slice i           *
*     OUT_CAT (I)       Outlet concentration of cation i    (ppb)        *
*     OUT_ANI (I)       Outlet concentration of anion i     (ppb)        *
*     OUT_D1C (I)       Outlet concentration of amine i     (ppb)        *
*     OUT_D1CSTA (I)    Outlet concentration of molecular amine          *
*     OUT_D1CT (I)      Total outlet concentration of amines             *
*     OUT_D2A (I)       Outlet concentration of carbonate i (ppb)        *
*     OUT_D2ASTA        Outlet concentration of molecular carbonate      *
*     OUT_D2AT          Total outlet concentration of carbonates         *
*     OUT_D1A (I)       Outlet concentration of RCOOH i    (ppb)         *
*     OUT_D1ASTA (I)    Outlet concentration of molecular RCOOH          *
*     OUT_D1AT (I)      Total outlet concentration of RCOOH              *
*     PRIORITY_C(I)     Priority of cation i                             *
*     PRIORITY_A(I)     Priority of anion i                              *
*     RATE_COMC (I)     Mass transfer rate of cation i                   *
*     RATE_COMA (I)     Mass transfer rate of anion i                    *
*     RATE_D1CSTA (I)   Mass transfer rate of molecular amine            *
*     RATE_D2ASTA       Mass transfer rate of molecular carbonate i      *
*     RE_ANI            Reynolds number of anions                        *
*     RE_CAT            Reynolds number of cations                       *
*     RES_HT            Resin (column) height (cm)                       *
*     SCH_ANI (I)       Schmidt number of anion i                        *
*     SCH_CAT (I)       Schmidt number of cation  i                      *
*     SCH_D1CSTA (I)    Schmidt number of molecular amine                *
*     SCH_D2ASTA        Schmidt number of molecular carbonate            *
*     SCH_D1ASTA (I)    Schmidt number of molecular RCOOH                *
*     SEL_ANI (I)       Selectivity of anion i                           *
*     SEL_CAT (I)       Selectivity of cation i                          *
*     SELH              Selectivity of hydrogen                          *
*     SELCOMC (I)       Combined array of SEL_CAT (I) and SEL_D1C (I)    *
*     SELCOMA (I)       Combined array of SEL_ANI (I) and SEL_D2A (I)    *
*     TAU               Step size for the time                           *
*     XI                Step size for the distance                       *
*     XI1               Distance step for the first few slices           *
*     VAANI (I)         Valency of cation i                              *
*     VACAT (I)         Valency of anion i                               *
*     VAD1C (I)         Valency of amine i                               *
*     VAD2A (I)         Valency of carbonate i                           *
*     VAD1A (I)         Valency of carboxylic acid (RCOOH) i             *
*     VACOMC (I)        Combined array of VACAT (I) and VAD1C (I)        *
*     VACOMA (I)        Combined array of VAANI (I) and VAD2A (I)        *
*     VAH               Valency of hydrogen                              *
*     VAOH              Valency of hydroxide                             *
*     VOID_FRAC         Void fraction of the bed                         *
*     VOL_FLOW          Volume flow rate (cm3/s)                         *
*     VS                Superfacial velocity (cm/s)                      *
*     VISCO             Viscosity of water  (centipoise)                 *
*     TMPC              Inlet water temperature  (deg. celsius)          *
*     XBC(I,K)          Fraction of cation i in bulk phase               *
*     XBCI (I)          Fraction of cation i in the interface            *
*     XBA(I,K)          Fraction of anion i in bulk phase                *
*     XBAI (I)          Fraction of anion i in the interface             *
*     XBD1CSTA(I,K)     Fraction of molecular amine i in bulk phase      *
*     XBD2ASTA          Fraction of molecular carbonate in bulk phase    *
*     XBD1ASTA (I,K)    Fraction of molecular RCOOH i in bulk phase      *
*     XANICUR(I)        Current fraction of anion i in bulk phase        *
*     XCATCUR(I)        Current fraction of cation i in bulk phase       *
*     YANICUR(I)        Current fraction of anion i in resin phase       *
*     YCATCUR(I)        Current fraction of cation i in resin phase      *
*     YAINIT (I)        Initial loading of anion i in resin phase        *
*     YCINIT (I)        Initial loading of cation i in resin phase       *
*     YD1C_INIT (I)     Initial loading of amine i in resin phase        *
*     YCRAB_INIT (I)    Initial loading of carbonate i                   *
*     YD1A_INIT (I)     Initial loading of RCOOH i                       *
*     YRC(I,J,K )       Fraction of cation i in resin phase              *
*     YRA(I,J,K)        Fraction of anion i in resin phase               *
*                                                                        *
**************************************************************************
```

```
      IMPLICIT  REAL*8 (A-H,O-Z)
      COMMON /S/SEL_CAT(8),SEL_ANI(8),SELD1C(5),SELD1A(5),SELD2C(2),
     1          SELD2A(2),SELD3C(3),SELD3A(3),SELslca(3),
     1          SELCOMC(21),SELCOMA(21),SELH,SELOH
      COMMON /D/DIFU_CAT(8),DIFU_ANI(8),DIFU_D1C(5),DIFU_D1CSTA(5),
     1          DIFU_D1A(5),DIFU_D1ASTA(5),DIFU_D2C(2),DIFU_D2CSTA,
     1          DIFU_D2A(2),DIFU_D2ASTA,DIFU_D3C(3),DIFU_D3CSTA,
     1          DIFU_D3A(3),DIFU_D3ASTA,DIFU_slca(3),DIFU_slcaSH,
     1          DIFUH,DIFUOH,DIFUCOMC(21),DIFUCOMA(21)
      COMMON /V/VACAT(8),VAANI(8),VAD1C(5),VAD1A(5),VAD2C(2),
     1          VAD2A(2),VAD3C(3),VAD3A(3),VAslca(3),VAH,VAOH,
     1          VACOMC(21),VACOMA(21)
      COMMON /E/EWCAT(8),EWANI(8),EWD1C(5),EWD1CSTA(5),EWD1A(5),
     1          EWD1ASTA(5),EWD2C(2),EWD2CSTA,EWD2A(2),EWD2ASTA,
     1          EWD3C(3),EWD3CSTA,EWD3A(3),EWD3ASTA,EWslca(3),
     1          EWslcaSH,EWC(21),EWA(21)
      COMMON /N/NOCAT,NOANI,NOD1C,NOD1A,NOD2C,NOD2A,NOD3C,NOD3A,
     1          NOSLCA,NC,NA
      COMMON /R/TMPC,DISSW,DISSD1C(5),DISSD1A(5),DISSD2C(2),
     1          DISSD2A(2),DISSD3C(3),DISSD3A(3),DISSslca(3)
      COMMON /I/INDEXC(21),INDEXA(21)
      COMMON /Q/BED_DIA,RES_HT,VOID_FRAC,FCR,FAR,DEN,VISCO,
     1            CAT_DIA,ANI_DIA,SFLAG,FFLAG
      COMMON /C/SCH_CAT(8),SCH_ANI(8),SCH_D1C(5),SCH_D2C(2),
     1      SCH_D3C(3),SCH_D1A(5),SCH_D2A(2),SCH_D3A(3),SCH_SLCA(3),
     1      SCH_D1CSTA(5),SCH_D1ASTA(5),SCH_D2CSTA,SCH_D2ASTA,
     1      SCH_D3CSTA,SCH_D3ASTA,SCH_SLCASH,SCH_SLCASC,SCH_SLCAPT,
     1      SCH_OH

     REAL*8 RATE_COMC(21,5000),RATE_COMA(21,5000),RATE_D1CSTA(5,5000)
     1 ,RATE_D1ASTA(5,5000),RATE_D2CSTA(5000),RATE_D2ASTA(5000),
     1 RATE_D3CSTA(5000),RATE_D3ASTA(5000),RATE_SHPROT(5000),
     1 CFCAT(8),CFANI(8),CFD1C(5),CFEED_D1C(5),CFD1CSTA(5),CFD1A(5),
     1 CFEED_D1A(5),CFD1ASTA(5),CFD2C(2),CFEED_D2C,CFD2CSTA,CFD2A(2),
     1 CFEED_D2A,CFD2ASTA,CFD3C(3),CFEED_D3C,CFD3CSTA,CFD3A(3),
     1 CFEED_D3A,CFD3ASTA,CFslca(3),CFEED_slca,CFslcaSH,
     1 YCINIT(8),YAINIT(8),YD1C_INIT(5),YD1A_INIT(5),YD2C_INIT(2),
     1 YD2A_INIT(2),YD3C_INIT(3),YD3A_INIT(3),Yslca_INIT(3),
     1 COEC(21),COEA(21),COED1CSTA(5),CD1CSTAI(5),COED1ASTA(5),
     1 CD1ASTAI(5),COED2CSTA,COED2ASTA,COED3CSTA,COED3ASTA,
     1 OUT_CAT(8),OUT_ANI(8),OUT_D1C(5),OUT_D1CSTA(5),OUT_D1CT(5),
     1 OUT_D1A(5),OUT_D1ASTA(5),OUT_D1AT(5),OUT_D2C(2),OUT_D2CT,
     1 OUT_D2A(2),OUT_D2AT,OUT_D3C(3),OUT_D3CT,OUT_D3A(3),OUT_D3AT,
     1 OUT_slca(3),YCATCUR(21),YANICUR(21),XCATCUR(21),XANICUR(21),
     1 YRC(21,4,5000),YRA(21,4,5000),XBCI(21),XBC(21,5000),XBAI(21),
     1 XBA(21,5000),XBCAT(10,5000),XBANI(10,5000),XBD1C(5,5000),
     1 XBD1CSTA(5,5000),XBD1A(5,5000),XBD1ASTA(5,5000),XBD2C(2,5000),
     1 XBD2CSTA(5000),XBD2A(2,5000),XBD2ASTA(5000),XBD3C(3,5000),
     1 XBD3CSTA(5000),XBD3A(3,5000),XBD3ASTA(5000),XBslca(3,5000),
     1 XBslcaSH(5000),XBslcaSC(5000),XBslcaPT(5000),
     1 MTC_CAT(8),MTC_ANI(8),MTC_D1C(5),MTC_D1CSTA(5),MTC_D1A(5),
     1 MTC_D1ASTA(5),MTC_D2C(2),MTC_D2A(2),MTC_D3C(3),MTC_D3A(3),
     1 MTC_D2CSTA,MTC_D2ASTA,MTC_D3CSTA,MTC_D3ASTA,MTC_slca(3),
     1 MTC_slcaSH,MTC_slcaSC,MTC_slcaPT,
     1 MTC_COMC(21),MTC_COMA(21),MTC_C,MTC_A,MTC_REF,FUNC_MTC,
     1 CFCAT_NEW1(8),CFANI_NEW1(8),CFEED_D1CNEW1(5),CFEED_D1ANEW1(5),
     1 CFD1C_NEW(5),CFD1CSTA_NEW(5),CFD1A_NEW(5),CFD1ASTA_NEW(5),
     1 CFD2C_NEW(2),CFD2A_NEW(2),CFD3C_NEW(3),CFD3A_NEW(3),
     1 CFslca_NEW(3),DK(3), MTC_CATNEW(8),MTC_ANINEW(8),
     1 MTC_D1CNEW(5),MTC_D1ANEW(5),MTC_D2CNEW(2),MTC_D2ANEW(2),
     1 MTC_D3CNEW(3),MTC_D3ANEW(3),MTC_slcaNEW(3),MTC_D1CSTANEW(5),
     1 MTC_D1ASTANEW(5),MTC_D2CSTANEW,MTC_D2ASTANEW,MTC_D3CSTANEW,
     1 MTC_D3ASTANEW,MTC_slcaSHNEW,MTC_slcaSCNEW,MTC_slcaPTNEW,
     1 MTC_COMCNEW(21),MTC_COMANEW(21),MTC_REFNEW

     REAL*8 CBCAT(8),CBANI(8),CBD1C(5),CBD1CSTA(5),CTO_D1C(5),
     1 CBD1A(5),CBD1ASTA(5),CTO_D1A(5),CBD2C(2),CTO_D2C,CBD2A(2),
     1 CTO_D2A,CBD3C(3),CTO_D3C,CBD3A(3),CTO_D3A,CBslca(3),CTO_slca,
     1 CBC(21),CBA(21),FLUXC(21),FLUXA(21),
     1 CBH,CBOH,CFC,CFA,CF,CBD2CSTA,CBD2ASTA,CBD3CSTA,CBD3ASTA,
```

```
     1 SUMYC,SUMYA,DEC,DEA,CATLOADING(21),ANILOADING(21),
     1 ALOGDISSD1C(5)

       REAL*8  YRC_SORP_D1C(5,4,5000),YRC_PROT_D1C(5,4,5000),
     1  YRA_SORP_D1A(5,4,5000), YRA_PROT_D1A(5,4,5000),
     1  YRC_SORP_D2C(1,4,5000),YRC_PROT_D2C(1,4,5000),
     1  YRA_SORP_D2A(1,4,5000),YRA_PROT_D2A(1,4,5000),
     1  YRC_SORP_D3C(1,4,5000),YRC_PROT_D3C(1,4,5000),
     1  YRA_SORP_D3A(1,4,5000),YRA_PROT_D3A(1,4,5000),
     1  YRA_Sorp_slca(1,4,5000),YRA_PROT_slca(1,4,5000),
     1  RATE_D1CTOT(5,5000),RATE_D1CSORP(5,5000),RATE_D1ATOT(5,5000),
     1  RATE_D1ASORP(5,5000),RATE_D2CTOT(5000),RATE_D2CSORP(5000),
     1  RATE_D2ATOT(5000),RATE_D2ASORP(5000),RATE_D3CTOT(5000),
     1  RATE_D3CSORP(5000),RATE_D3ASORP(5000),RATE_D3ATOT(5000),
     1  RATE_SHSORP(5000),RATE_SHTOT(5000),RATE_slcaSC(5000),
     1  RATE_slcaPT(5000),SUMsorpd1c(5000), SUMsorpd1a(5000),
     1  SUMsorpd2c(5000), SUMsorpd2a(5000),SUMsorpd3c(5000),
     1  SUMsorpd3a(5000), SUMsorpslca(5000),
     1  XNC(5000),XNA(5000),ZD1CSTA(5000),ZD1ASTA(5000),S(5000),
     1  U(5000),V(5000),Z(5000),W(5000),B(5000),T(5000),
     1  YNC(5000),YNA(5000),YsorpD1C(5000),YsorpD1A(5000),
     1  YsorpD2C(5000),YsorpD2A(5000),YsorpD3C(5000),
     1  YsorpD3A(5000),Ysorpslca(5000)

       DIMENSION NEXC(5000),NEXA(5000),PRIORITY_C(21),PRIORITY_A(21)

       CHARACTER*1 SFLAG,TFLAG,CFLAG,RFLAG,FFLAG,VFLAG,BFLAG
       CHARACTER*20 INPFILE
*    ______________________
*     Reading the data
*    ______________________

      WRITE(*,*)"Enter the input file name"
      READ(*,500)INPFILE
 500  FORMAT(20A)

      OPEN(UNIT = 9, FILE = INPFILE, STATUS = 'UNKNOWN')
*    __________________________________________________________________
*     Flag to account for desulphonation effects if SO4-2 is present in
*     the system. If this flag is "Y/y" the desulphonation effects will
*     be included for SO4-2. If it is "N/n" it will not be included.
*    __________________________________________________________________

      READ(9,501) SFLAG
 501  FORMAT(1A)
*    __________________________________________________________________
*     Reading the time and distance steps.The second distance step is
*     for the first few steps of the numerical integration. Usually
*     XI1 is set to be 1/10th of XI.
*    __________________________________________________________________

       READ(9,*)TAU
       READ(9,*)XI
       READ(9,*)XI1
*    __________________________________________________________________
*     Reading the number of non-dissociative cations,non-dissociative
*     anions,amines and the carbonates respectively. A maximum of five
*     monovalent amines can be handled. If carbonate is present"NOD2A"
*     should be "2" to account for both HCO3- and CO3-2, otherwise it
*     should be zero.
*    __________________________________________________________________

       READ(9,*) NOCAT
       READ(9,*) NOANI
       READ(9,*) NOD1C
       READ(9,*) NOD1A
       READ(9,*) NOD2C
       READ(9,*) NOD2A
       READ(9,*) NOD3C
       READ(9,*) NOD3A
       READ(9,*) NOSLCA
```

```
      IF(NOCAT.GT.8.OR.NOCAT.LT.0)THEN
          WRITE(*,*)"Number of cations should be between 0 and 8"
      ENDIF

      IF(NOANI.GT.8.OR.NOANI.LT.0)THEN
          WRITE(*,*)"Number of anions should be between 0 and 8"
      ENDIF

      IF(NOD1C.GT.5.OR.NOD1C.LT.0)THEN
          WRITE(*,*)"No. of amines should be between 0 and 5"
      ENDIF

      IF(NOD1A.LT.0.OR.NOD1A.GT.5) THEN
         WRITE(*,*) "No of carboxylic acid should be beween 0 and 5"
      ENDIF

      IF(NOD2C.LT.0.OR.NOD2C.GT.2.OR.NOD2C.EQ.1)THEN
        WRITE(*,*)"No of divalent amines must be 0 or 2, can not be 1"
      ENDIF

      IF(NOD2A.LT.0.OR.NOD2A.GT.2.OR.NOD2A.EQ.1)THEN
         WRITE(*,*)"Number of carbonates must be 0 or 2, can not be 1"
      ENDIF

      IF(NOD3C.LT.0.OR.NOD3C.GT.3.OR.NOD3C.EQ.1.OR.NOD3C.EQ.2)THEN
        WRITE(*,*)"No of tri-amines must be 0 or 3, cannot be other"
      ENDIF

      IF(NOD3A.LT.0.OR.NOD3A.GT.3.OR.NOD3A.EQ.1.OR.NOD3A.EQ.2)THEN
         WRITE(*,*)"No of phosphates must be 0 or 3, cannot be other"
      ENDIF

      IF(NOslca.LT.0.OR.NOslca.GT.3.OR.NOslca.EQ.1.OR.NOslca.EQ.2)THEN
         WRITE(*,*)"Number of silica must be 0 or 3, cannot be other"
      ENDIF
*
*     Reading the resin diameters and capacities.
*

        READ(9,*) CAT_DIA
        READ(9,*) ANI_DIA
        READ(9,*) CAT_CAP
        READ(9,*) ANI_CAP
*
*     Reading the bed diameter, resin height, void fraction, flowrate
*

        READ(9,*) BED_DIA
        READ(9,*) RES_HT
        READ(9,*) VOID_FRAC
        READ(9,*) VOL_FLOW
*
*     Reading the cationic and anionic resin fractions,feed temperature
*     and pH.
*

        READ(9,*) FCR
        READ(9,*) FAR
        READ(9,*) TMPC
        READ(9,*) PH
*
*     Reading the data for cations(Initial loading, feed concentration,
*     valency, selectivity, equivalent weight, diffusivity)
*

      DO 2 II = 1,NOCAT
         READ(9,*)YCINIT(II)
         READ(9,*)CFCAT(II)
         READ(9,*)VACAT(II)
         READ(9,*)SEL_CAT(II)
```

```
            READ(9,*)EWCAT(II)
            READ(9,*)DIFU_CAT(II)
  2      CONTINUE
*
*        Reading the data for anions(Initial loading, feed concentration,
*        valency, selectivity, equivalent weight, diffusivity)
*

         DO 4 JJ = 1,NOANI
            READ(9,*)YAINIT(JJ)
            READ(9,*)CFANI(JJ)
            READ(9,*)VAANI(JJ)
            READ(9,*)SEL_ANI(JJ)
            READ(9,*)EWANI(JJ)
            READ(9,*)DIFU_ANI(JJ)
  4      CONTINUE
*
*       Reading the data for amines. Input for each amine is read in two
*       lines. First line consists of Initial loading,valency,selectivity
*       equivalent weight,dissociation constant and diffusivity of the
*       ionic form amine and total feed concentration. Second line consists
*       of equivalent weight and diffusivity of the amine in molecular form.
*

         IF (NOD1C.GE.1) THEN
            DO 6 KK=1,NOD1C
               READ(9,*)YD1C_INIT(KK)
               READ(9,*)VAD1C(KK)
               READ(9,*)SELD1C(KK)
               READ(9,*)EWD1C(KK)
               READ(9,*)DISSD1C(KK)
               READ(9,*)DIFU_D1C(KK)
               READ(9,*)CFEED_D1C(KK)
               READ(9,*)EWD1CSTA(KK)
               READ(9,*)DIFU_D1CSTA(KK)
  6         CONTINUE
            WRITE(*,*) "Input adsorption ratio for monovalent amines ="
            READ(*,*) RATIO_D1C
         ENDIF
*
*        Reading the data for acetic/formic acid (type 1 disso anions)
*
         IF (NOD1A.GE.1) THEN
           DO 7 MM=1, NOD1A
              READ(9,*)YD1A_INIT(MM)
              READ(9,*)VAD1A(MM)
              READ(9,*)SELD1A(MM)
              READ(9,*)EWD1A(MM)
              READ(9,*)DISSD1A(MM)
              READ(9,*)DIFU_D1A(MM)
              READ(9,*)CFEED_D1A(MM)
              READ(9,*)EWD1ASTA(MM)
              READ(9,*)DIFU_D1ASTA(MM)
  7        CONTINUE
           WRITE(*,*) "Input adsorption ratio for Organic Acids="
           READ(*,*) RATIO_D1A
         ENDIF
*
*        Reading the data for divalent amines (type 2 disso cations)
*
         IF (NOD2C.GE.1) THEN
            READ(9,*) CFEED_D2C
            DO 8 NN=1,NOD2C
               READ(9,*)YD2C_INIT(NN)
               READ(9,*)VAD2C(NN)
               READ(9,*)SELD2C(NN)
               READ(9,*)EWD2C(NN)
               READ(9,*)DISSD2C(NN)
               READ(9,*)DIFU_D2C(NN)
  8         CONTINUE
               READ(9,*)EWD2CSTA
```

```
         READ(9,*)EWCAT(II)
         READ(9,*)DIFU_CAT(II)
 2    CONTINUE
*     ________________________________________________________________
*     Reading the data for anions(Initial loading, feed concentration,
*     valency, selectivity, equivalent weight, diffusivity)
*     ________________________________________________________________

      DO 4 JJ = 1,NOANI
         READ(9,*)YAINIT(JJ)
         READ(9,*)CFANI(JJ)
         READ(9,*)VAANI(JJ)
         READ(9,*)SEL_ANI(JJ)
         READ(9,*)EWANI(JJ)
         READ(9,*)DIFU_ANI(JJ)
 4    CONTINUE
*     ________________________________________________________________
*    Reading the data for amines. Input for each amine is read in two
*    lines. First line consists of Initial loading,valency,selectivity
*    equivalent weight,dissociation constant and diffusivity of the
*    ionic form amine and total feed concentration. Second line consists
*    of equivalent weight and diffusivity of the amine in molecular form.
*     ________________________________________________________________

      IF (NOD1C.GE.1) THEN
         DO 6 KK=1,NOD1C
            READ(9,*)YD1C_INIT(KK)
            READ(9,*)VAD1C(KK)
            READ(9,*)SELD1C(KK)
            READ(9,*)EWD1C(KK)
            READ(9,*)DISSD1C(KK)
            READ(9,*)DIFU_D1C(KK)
            READ(9,*)CFEED_D1C(KK)
            READ(9,*)EWD1CSTA(KK)
            READ(9,*)DIFU_D1CSTA(KK)
 6       CONTINUE
         WRITE(*,*) "Input adsorption ratio for monovalent amines ="
         READ(*,*) RATIO_D1C
      ENDIF
*     ________________________________________________________
*     Reading the data for acetic/formic acid (type 1 disso anions)
*     ________________________________________________________
      IF (NOD1A.GE.1) THEN
        DO 7 MM=1, NOD1A
           READ(9,*)YD1A_INIT(MM)
           READ(9,*)VAD1A(MM)
           READ(9,*)SELD1A(MM)
           READ(9,*)EWD1A(MM)
           READ(9,*)DISSD1A(MM)
           READ(9,*)DIFU_D1A(MM)
           READ(9,*)CFEED_D1A(MM)
           READ(9,*)EWD1ASTA(MM)
           READ(9,*)DIFU_D1ASTA(MM)
 7      CONTINUE
        WRITE(*,*) "Input adsorption ratio for Organic Acids="
        READ(*,*) RATIO_D1A
      ENDIF
*     ________________________________________________________
*     Reading the data for divalent amines (type 2 disso cations)
*     ________________________________________________________
      IF (NOD2C.GE.1) THEN
         READ(9,*) CFEED_D2C
         DO 8 NN=1,NOD2C
            READ(9,*)YD2C_INIT(NN)
            READ(9,*)VAD2C(NN)
            READ(9,*)SELD2C(NN)
            READ(9,*)EWD2C(NN)
            READ(9,*)DISSD2C(NN)
            READ(9,*)DIFU_D2C(NN)
 8       CONTINUE
            READ(9,*)EWD2CSTA
```

```
         READ(9,*)EWCAT(II)
         READ(9,*)DIFU_CAT(II)
  2   CONTINUE
*
*     Reading the data for anions(Initial loading, feed concentration,
*     valency, selectivity, equivalent weight, diffusivity)
*

      DO 4 JJ = 1,NOANI
         READ(9,*)YAINIT(JJ)
         READ(9,*)CFANI(JJ)
         READ(9,*)VAANI(JJ)
         READ(9,*)SEL_ANI(JJ)
         READ(9,*)EWANI(JJ)
         READ(9,*)DIFU_ANI(JJ)
  4   CONTINUE
*
*    Reading the data for amines. Input for each amine is read in two
*    lines. First line consists of Initial loading,valency,selectivity
*    equivalent weight,dissociation constant and diffusivity of the
*    ionic form amine and total feed concentration. Second line consists
*    of equivalent weight and diffusivity of the amine in molecular form.
*

      IF (NOD1C.GE.1) THEN
         DO 6 KK=1,NOD1C
            READ(9,*)YD1C_INIT(KK)
            READ(9,*)VAD1C(KK)
            READ(9,*)SELD1C(KK)
            READ(9,*)EWD1C(KK)
            READ(9,*)DISSD1C(KK)
            READ(9,*)DIFU_D1C(KK)
            READ(9,*)CFEED_D1C(KK)
            READ(9,*)EWD1CSTA(KK)
            READ(9,*)DIFU_D1CSTA(KK)
  6      CONTINUE
         WRITE(*,*) "Input adsorption ratio for monovalent amines ="
         READ(*,*) RATIO_D1C
      ENDIF
*
*     Reading the data for acetic/formic acid (type 1 disso anions)
*
      IF (NOD1A.GE.1) THEN
        DO 7 MM=1, NOD1A
           READ(9,*)YD1A_INIT(MM)
           READ(9,*)VAD1A(MM)
           READ(9,*)SELD1A(MM)
           READ(9,*)EWD1A(MM)
           READ(9,*)DISSD1A(MM)
           READ(9,*)DIFU_D1A(MM)
           READ(9,*)CFEED_D1A(MM)
           READ(9,*)EWD1ASTA(MM)
           READ(9,*)DIFU_D1ASTA(MM)
  7     CONTINUE
        WRITE(*,*) "Input adsorption ratio for Organic Acids="
        READ(*,*) RATIO_D1A
      ENDIF
*
*     Reading the data for divalent amines (type 2 disso cations)
*
      IF (NOD2C.GE.1) THEN
         READ(9,*) CFEED_D2C
         DO 8 NN=1,NOD2C
            READ(9,*)YD2C_INIT(NN)
            READ(9,*)VAD2C(NN)
            READ(9,*)SELD2C(NN)
            READ(9,*)EWD2C(NN)
            READ(9,*)DISSD2C(NN)
            READ(9,*)DIFU_D2C(NN)
  8      CONTINUE
            READ(9,*)EWD2CSTA
```

```
            READ(9,*)Yslca_INIT(KS)  ! Data for soluble silica
            READ(9,*)VAslca(KS)
            READ(9,*)SELslca(KS)
            READ(9,*)EWslca(KS)
            READ(9,*)DK(KS)          ! dissociation constants
            READ(9,*)DIFU_slca(KS)
            DISSslca(KS)=10**(-DK(KS))
612      CONTINUE
         READ(9,*)EWslcaSH        ! molecular silica
         READ(9,*)DIFU_slcaSH
         READ(9,*)EWslcaSC        ! colloidal silica
         READ(9,*)DIFU_slcaSC
         READ(9,*)EWslcaPT        ! particulate silica
         READ(9,*)DIFU_slcaPT


         WRITE(*,*)"Input adsorption ratio for molecular silica ="
         READ(*,*) RATIO_Slca
         WRITE(*,*)"Give the ratio of colloidal silica to the"
         WRITE(*,*)"total soluble silica (usually 0.01 -- 0.20)"
         READ(*,*) RATIO_SC
        endif

*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      write(*,*)"Do you want to handle the fouled resin effects(y/n)?"
      read(*,501) FFLAG

      if (FFLAG.eq.'Y'.or.FFLAG.eq.'y') then
         write(*,*) "input the fouling percentage for cationic resin"
         write(*,*) "p (0=<p<1)="
         read(*,*) p

         write(*,*) "input the fouling percentage for anionic resin"
         write(*,*) "q (0=<q<1)="
         read(*,*) q
      endif
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

C#######################################################################
C     Ask user if he/she wants to handle the case for different cation/
C     anion ratio(layered-bed) case.  If this g is 'Y/y', the user
C     needs to input the the different 'cation/anion ratio' values for
C     different length (deepth) of the bed. If it is 'N/n', the program
C     will skip this part, and execute as usual.
C#######################################################################
        WRITE(*,*)
        WRITE(*,*)'Do you want to handle the case of different'
        WRITE(*,*)'cation/anion ratio in the bed (y/n)?'
        READ(*,501) RFLAG

      IF (RFLAG.EQ.'Y'.OR.RFLAG.EQ.'y') THEN
         WRITE(*,*)
         WRITE(*,*)'input the C/A ratio for the inlet part of column'
         READ(*,*) RATIO1
         WRITE(*,*)'input the length of the inlet part of column (cm)'
         READ(*,*)BED_HT1

         WRITE(*,*)
         WRITE(*,*)'input the C/A ratio for the middle part of column'
         READ(*,*)RATIO2
         WRITE(*,*)'input the length of the middle part of column (cm)'
         READ(*,*)BED_HT2

         WRITE(*,*)
         WRITE(*,*) 'input the C/A ratio for the outlet part of column'
         READ(*,*)RATIO3
         WRITE(*,*) 'input the length of the outlet part of column (cm)'
         READ(*,*)BED_HT3

       FAR1=1./(1+RATIO1)
       FCR1=1.-FAR1
```

```
       FAR2=1./(1+RATIO2)
       FCR2=1.-FAR2

       FAR3=1./(1+RATIO3)
       FCR3=1.-FAR3
       ENDIF
*
*    Initializing the indices
*
      DO 10 II=1,NOCAT
          INDEXC(II) = II
 10   CONTINUE

      IF(NOD1C.GE.1)THEN
          DO 12 KK=1,NOD1C
             INDEXC(NOCAT+KK) = NOCAT+KK
 12       CONTINUE
      ENDIF

      IF(NOD2C.GE.1)THEN
          DO 13 NN=1,NOD2C
             INDEXC(NOCAT+NOD1C+NN) = NOCAT+NOD1C+NN
 13       CONTINUE
      ENDIF

      IF(NOD3C.GE.1)THEN
          DO 19 KT=1,NOD3C
             INDEXC(NOCAT+NOD1C+NOD2C+KT) = NOCAT+NOD1C+NOD2C+KT
 19       CONTINUE
      ENDIF

      DO 14 JJ=1,NOANI
          INDEXA(JJ) = JJ
 14   CONTINUE

      IF(NOD1A.GE.1) THEN
          DO 16 MM = 1,NOD1A
             INDEXA(NOANI+MM)=NOANI+MM
 16       CONTINUE
      ENDIF

      IF(NOD2A.GE.1)THEN
          DO 17 LL=1,NOD2A
             INDEXA(NOANI+NOD1A+LL) = NOANI+NOD1A+LL
 17       CONTINUE
      ENDIF

      IF(NOD3A.GE.1)THEN
          DO 21 KP=1,NOD3A
             INDEXA(NOANI+NOD1A+NOD2A+KP) = NOANI+NOD1A+NOD2A+KP
 21       CONTINUE
      ENDIF

      IF(NOSLCA.GE.1)THEN
       DO 321 KS=1,NOSLCA
       INDEXA(NOANI+NOD1A+NOD2A+NOD3A+KS) = NOANI+NOD1A+NOD2A+NOD3A+KS
 321   CONTINUE
      ENDIF
*
*    Setting the valences and selectivities of hydrogen and hydroxide.
*
*     VAH = 1.0
      VAOH = -1.0
      SELH = 1.0
      SELOH = 1.0
*
*    Calculating the viscosity and dissociation constant of water.
*    (Divekar et al, Ind. Eng. Chem. Res., P1906, Vol.26, 1987)
*
```

```
      VISCO = 1.43123+TMPC*(0.000127065*TMPC-0.0241537)
      ALOGDISSW = 4470.99/(TMPC+273.15)-6.0875+0.01706*(TMPC+273.15)
      DISSW = 10**(-ALOGDISSW)
*    ___________________________________________________________
*    Calculating the density of water. (Reid et al. 1987,
*    "The properties of Gases & Liquids", 4th Ed.)
*    ___________________________________________________________

      SS1 = 374.11 - TMPC
      SS2 = SS1**0.33333
      DEN = SQRT((1+0.1342489*SS2-3.946263E-3*SS1)/
     1      (3.1975-0.3151548*SS2-1.203374E-3*SS1+7.48908E-13
     1       *(SS1**4)))
*    _________________________________________________________
*    Calculating the diffusivities of hydrogen & hydroxide
*    at the practical operating temperature T (C)
*    (Robinson and Stokes, 1959, "Electrolyte Solutions")
*    _________________________________________________________

      RTF = (8.931D-10)*(TMPC+273.15)
      DIFUH=RTF*(221.7134+5.5294*TMPC-0.014445*TMPC*TMPC)
      DIFUOH=RTF*(104.74113+3.807544*TMPC)
*    _______________________________________________________________
*     Incorporate the temperature effect on diffusivities for
*     other ions: T---- oC.  Call subroutines to calculate them.
*    _______________________________________________________________
      IF (ABS(TMPC-25.0).GE.0.5) THEN
         VISCO_25 = 0.9068  ! viscosity of water at 25 C
         if (NOCAT.GE.1) then
            DO 921 I=1, NOCAT
                CALL DIFUTEMP_C(DIFU_CAT,EWCAT,TMPC,RTF,NOCAT)
 921        CONTINUE
         endif

          if (NOD1C.GE.1) then
            DO 922 I=1, NOD1C
                CALL DIFUTEMP_C(DIFU_D1C,EWD1C,TMPC,RTF,NOD1C)
               CALL DIFUTEMP_C(DIFU_D1CSTA,EWD1C,TMPC,RTF,NOD1C)
 922        CONTINUE
         endif

          if (NOD2C.GE.1) then
            DO 923 I=1, NOD2C
              CALL DIFUTEMP_C(DIFU_D2C,EWD2C,TMPC,RTF,NOD2C)
              DIFU_D2CSTA=DIFU_D2CSTA*(TMPC+273.15)/334./VISCO_25
 923        CONTINUE
         endif

          if (NOD3C.GE.1) then
            DO 924 I=1, NOD3C
               CALL DIFUTEMP_C(DIFU_D3C,EWD3C,TMPC,RTF,NOD3C)
               DIFU_D3CSTA=DIFU_D3CSTA*(TMPC+273.15)/334./VISCO_25
 924        CONTINUE
         endif

         if (NOANI.GE.1) then
            DO 925 J=1, NOANI
               CALL DIFUTEMP_A(DIFU_ANI,EWANI,TMPC,RTF,NOANI)
 925        CONTINUE
         endif

         if (NOD1A.GE.1) then
            DO 926 J=1, NOD1A
               CALL DIFUTEMP_A(DIFU_D1A,EWD1A,TMPC,RTF,NOD1A)
               CALL DIFUTEMP_A(DIFU_D1ASTA,EWD1A,TMPC,RTF,NOD1A)
 926        CONTINUE
         endif

          if (NOD2A.GE.1) then
            DO 927 J=1, NOD2A
               CALL DIFUTEMP_A(DIFU_D2A,EWD2A,TMPC,RTF,NOD2A)
```

```
                    DIFU_D2ASTA=DIFU_D2ASTA*(TMPC+273.15)/334./VISCO_25
 927            CONTINUE
             endif

              if (NOD3A.GE.1) then
                DO 928 J=1, NOD3A
                   CALL DIFUTEMP_A(DIFU_D3A,EWD3A,TMPC,RTF,NOD3A)
                   DIFU_D3ASTA=DIFU_D3ASTA*(TMPC+273.15)/334./VISCO_25
 928            CONTINUE
             endif

               if (NOSLCA.GE.1) then
                DO 929 J=1, NOSLCA
                   CALL DIFUTEMP_A(DIFU_SLCA,EWSLCA,TMPC,RTF,NOSLCA)
 929            CONTINUE
                DIFU_slcaSH=DIFU_slcaSH*(TMPC+273.15)/334./VISCO_25
                DIFU_slcaSC=DIFU_slcaSC*(TMPC+273.15)/334./VISCO_25
                DIFU_slcaPT=DIFU_slcaPT*(TMPC+273.15)/334./VISCO_25
               endif
           ENDIF
*    _______________________________________________________
*    Calculating the column area and superficial velocity.
*    _______________________________________________________

      AREA = 3.1415927*(BED_DIA**2.)/4.
      VS = VOL_FLOW/AREA

*    cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*    Calculate the solubility of silica.
*    Rimstidt J. D. and Barnes, H. L. (1980) Geochim. Cosmochim.
*        Acta, vol 44, pp1683
*    cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

        PKs=0.3380-(7.889E-4)*(TMPC+273.15)-840.1/(TMPC+273.15)
        Csoluslca=10**(PKs)       ! solubility of silica at inlet temp
*
*    ______________________________________________________________
*     Incorporate the temperature effect on dissciation constants
*     for ions whose disso_temp relations are known. Currently, only
*     ammonia, morphline, carbonates, phosphates are availabe
*    ______________________________________________________________
      IF (ABS(TMPC-25.0).GE.0.5) THEN
         TMPK=TMPC+273.15
           if (NOD1C.GE.1) then
             do 1335 KK=1,NOD1C
              if (ABS(EWD1CSTA(KK)-17.0).LE.0.1) then  ! if it is NH3
               ALOGDISSD1C(KK)=4.8601+(6.31E-5)*TMPK-(5.98E-3)/(TMPK*TMPK)
               DISSD1C(KK)=10**(-ALOGDISSD1C(KK))
              endif

              if (ABS(EWD1CSTA(KK)-87.0).LE.0.1) then  ! if it is morphline
            ALOGDISSD1C(KK)=5.7461+(8.095E-5)*TMPK-(1.3881E-2)/(TMPK*TMPK)
               DISSD1C(KK)=10**(-ALOGDISSD1C(KK))
              endif
1335       continue
         endif

           if (NOD2A.GE.1) then
              if (ABS(EWD2ASTA-62.0).LE.0.1) then   ! if it is carbonate
          ALOGCK1=17052./TMPK+215.21*LOG10(TMPK)-0.12675*TMPK-545.56
          ALOGCK2=2902.39/TMPK+0.02379*TMPK-6.489
              DISSD2A(1)=10**(-ALOGCK1)
              DISSD2A(2)=10**(-ALOGCK2)
              endif
         endif

           if (NOD3A.GE.1) then
              if (ABS(EWD3ASTA-98.0).LE.0.1) then  ! if it is phosphate
            ALOGDISSD3A1=799.31/TMPK-4.5535+0.013486*TMPK
            ALOGDISSD3A2=2076.0/TMPK-5.9884+0.020912*TMPK
            ALOGDISSD3A3=12.1152-0.00908*TMPK+0.000127*TMPK*TMPK
            DISSD3A(1)=10**(-ALOGDISSD3A1)
```

```
          DISSD3A(2)=10**(-ALOGDISSD3A2)
          DISSD3A(3)=10**(-ALOGDISSD3A3)
           endif
         endif
       ENDIF
*
*   ____________________________________________________________
*   Assume the bulk concentrations in the first slice to be equal
*   to the inlet concentrations, and then call the equilibrium
*   subroutine 'DISSOEQ' to calculate the concentrations of
*   dissociative species in this slice.
*   ____________________________________________________________

      DO 18 II=1,NOCAT
         CBCAT(II) = CFCAT(II)
 18   CONTINUE
      DO 20 JJ=1,NOANI
         CBANI(JJ) = CFANI(JJ)
 20   CONTINUE

         IF(NOD1C.GE.1)THEN
           DO 22 KK = 1,NOD1C
              CTO_D1C(KK) = CFEED_D1C(KK)
 22        CONTINUE
         ENDIF

         IF(NOD2C.GE.1)THEN
           CTO_D2C = CFEED_D2C
         ENDIF

         IF(NOD3C.GE.1)THEN
           CTO_D3C = CFEED_D3C
         ENDIF

        IF(NOD1A.GE.1) THEN
           DO 23 MM = 1, NOD1A
             CTO_D1A(MM) = CFEED_D1A(MM)
 23        CONTINUE
        ENDIF

         IF(NOD2A.GE.1)THEN
           CTO_D2A = CFEED_D2A
         ENDIF

         IF(NOD3A.GE.1)THEN
           CTO_D3A = CFEED_D3A
         ENDIF

        IF(NOSLCA.GE.1) THEN
            CTO_slca=CFEED_slca
          If (CFEED_slca.GT.Csoluslca) then
           Csolu=Csoluslca
           CBslcaSC=Ratio_SC*Csolu       ! colloidal silica
            CBslcaPT=CTO_slca-Csolu-CBslcaSC ! the conc. of particulate silica
        else
            Csolu=CTO_slca/(1+Ratio_SC)
              CBslcaSC=Ratio_SC*Csolu
            CBslcaPT=0                        ! no particulate silica exist
          endif
         ENDIF
*
*     ____________________________________________________________
*     Call "DISSOEQ" first time to get the ionic concentrations and
*     molecular concentration for dissociative species
*     ____________________________________________________________

      PH_OLD=PH
      CALL DISSOEQ(CBCAT,CBANI,CBD1C,CBD1CSTA,CTO_D1C,CBD1A,CBD1ASTA,
     1         CTO_D1A,CBD2C,CBD2CSTA,CTO_D2C,CBD2A,CBD2ASTA,CTO_D2A,
     1         CBD3C,CBD3CSTA,CTO_D3C,CBD3A,CBD3ASTA,CTO_D3A,
     1         Csolu,CBslcaSH,CBslca,CBH,CBOH,PH_OLD,PH_NEW,IFLAG)

      IF(IFLAG.EQ.1) THEN
```

```
         WRITE(*,*)"Abnormal exit from 'DISSOEQ' subroutine"
         WRITE(*,*)"(outside the loop)"
         GO TO 538
      ENDIF
*     _______________________________________________________________
*     Storing the feed concentrations of H+ and OH-.
*     _______________________________________________________________
      CFH = CBH
      CFOH = CBOH
      PH_FEED = PH_NEW
*    _________________________________________________________________
*    Setting the feed concentrations of the ionic and molecular forms
*    of the amines equal to the bulk concentrations in the first slice
*    which are calculated in the equilibrium subroutine.
*    _________________________________________________________________

      IF(NOD1C.GE.1)THEN
        DO 24 KK = 1,NOD1C
           CFD1C(KK) = CBD1C(KK)
           CFD1CSTA(KK) = CBD1CSTA(KK)
 24     CONTINUE
      ENDIF

      IF(NOD2C.GE.1)THEN
        DO 29 NN = 1, NOD2C
           CFD2C(NN) = CBD2C(NN)
 29     CONTINUE
           CFD2CSTA = CBD2CSTA
      ENDIF

      IF(NOD3C.GE.1)THEN
        DO 629 KT = 1, NOD3C
           CFD3C(KT) = CBD3C(KT)
629     CONTINUE
           CFD3CSTA = CBD3CSTA
      ENDIF

*    _________________________________________________________________
*    Setting the feed concentrations of the ionic and molecular forms
*    of the carbonate equal to the bulk concentrations in the first slice
*    which are calculated in the equilibrium subroutine.
*    _________________________________________________________________

       IF(NOD1A.GE.1) THEN
          DO 25 MM = 1,NOD1A
             CFD1A(MM) = CBD1A(MM)
                CFD1ASTA(MM) = CBD1ASTA(MM)
 25       CONTINUE
       ENDIF

      IF(NOD2A.GE.1)THEN
        DO 619 LL = 1, NOD2A
           CFD2A(LL) = CBD2A(LL)
619     CONTINUE
           CFD2ASTA = CBD2ASTA
      ENDIF

      IF(NOD3A.GE.1)THEN
        DO 719 KP = 1, NOD3A
           CFD3A(KP) = CBD3A(KP)
719     CONTINUE
           CFD3ASTA = CBD3ASTA
      ENDIF

      IF(NOSLCA.GE.1)THEN
        DO 819 KS = 1, NOSLCA
           CFslca(KS) = CBslca(KS)
819     CONTINUE
           CFslcaSH = CBslcaSH
           CFslcaSC = CBslcaSC
           CFslcaPT = CBslcaPT
```

```
      ENDIF
*
*    Printing the concentrations after the equilibrium calculation
*
      WRITE(*,*)"********************************************"
      WRITE(*,502)PH
      WRITE(*,503)PH_NEW
 502  FORMAT(' Input value of the feed pH      = ',F5.2)
 503  FORMAT(' Calculated value of the feed pH = ',F5.2)
      WRITE(*,*)
      WRITE(*,*)"Feed concentrations at equilibrium:"

      WRITE(*,*)"cations:"
      WRITE(*,504)(CBCAT(II),II=1,NOCAT)
      WRITE(*,*)

        IF(NOD1C.GE.1)THEN
          WRITE(*,*)"Amines(CTotal,amine+,amine*):"
          DO 26 KK=1,NOD1C
             WRITE(*,504)CTO_D1C(KK),CBD1C(KK),CBD1CSTA(KK)
 26       CONTINUE
          WRITE(*,*)
        ENDIF

        IF(NOD2C.GE.1)THEN
         WRITE(*,*)"Di_amines(Total,Di_amine+,Di_amine2+,Di_amine*):"
         WRITE(*,504)CTO_D2C,(CBD2C(NN),NN=1,NOD2C),CBD2CSTA
         WRITE(*,*)
        ENDIF

        IF(NOD3C.GE.1)THEN
       WRITE(*,*)"Tri_amines(Total,Tri_amine+,Tri_amine2+,Tri_amine3+,
     1            Tri_amine*):"
       WRITE(*,504)CTO_D3C,(CBD3C(NN),KT=1,NOD3C),CBD3CSTA
       WRITE(*,*)
        ENDIF

      WRITE(*,*)"Anions:"
      WRITE(*,504)(CBANI(JJ),JJ=1,NOANI)
      WRITE(*,*)

      IF(NOD1A.GE.1) THEN
        WRITE(*,*)"RCOOH (CTotal, COOH-,RCOOH*):"
        DO 27 MM=1,NOD1A
             WRITE(*,504)CTO_D1A(MM),CBD1A(MM),CBD1ASTA(MM)
 27     CONTINUE
      ENDIF

        IF(NOD2A.GE.1)THEN
          WRITE(*,*)"Carbonates(Total,HCO3-,CO3-2,H2CO3*):"
          WRITE(*,504)CTO_D2A,(CBD2A(LL),LL=1,NOD2A),CBD2ASTA
          WRITE(*,*)
        ENDIF

        IF(NOD3A.GE.1)THEN
          WRITE(*,*)"Phosphates(Total,H2PO4-,HPO4-2,PO4-3,H3PO4):"
          WRITE(*,504)CTO_D3A,(CBD3A(KP),KP=1,NOD3A),CBD3ASTA
          WRITE(*,*)
        ENDIF

      IF(NOSLCA.GE.1)THEN
      WRITE(*,*)"Silica{Total,Soluble,H3SiO4-,H2SiO4-2,PS-2"
      WRITE(*,*)"Si(OH)4,SiO2_col,SiO2(particulate)}: (meq/ml)"
      WRITE(*,504)CTO_slca,Csolu,(CBslca(KS),KS=1, NOSLCA),
     1           CBslcaSH,CBslcaSC,CBslcaPT
      WRITE(*,*)
      ENDIF

 504  FORMAT(8(E12.4,4X))
      WRITE(*,*)"**********************************************"
```

```
*
*    ________________________________________________________________
*    Calculating the total feed concentrations of the cationic and anionic
*    species.
*    ________________________________________________________________

      CFC = 0.0
      DO 28 II = 1,NOCAT
         CFC = CFC + CFCAT(II)
 28   CONTINUE

      IF (NOD1C.GE.1) THEN
          DO 30 KK=1,NOD1C
             CFC = CFC + CFD1C(KK)
 30       CONTINUE
      END IF

      IF (NOD2C.GE.1) THEN
          DO 34 NN = 1, NOD2C
             CFC = CFC + CFD2C(NN)
 34       CONTINUE
      END IF

      IF (NOD3C.GE.1) THEN
          DO 634 KT = 1, NOD3C
             CFC = CFC + CFD3C(KT)
634       CONTINUE
      END IF

      CFC = CFC + CFH

      CFA = 0.0
      DO 32 JJ = 1, NOANI
         CFA = CFA + CFANI(JJ)
 32   CONTINUE

      IF (NOD1A.GE.1) THEN
          DO 35 MM = 1, NOD1A
             CFA = CFA + CFD1A(MM)
 35       CONTINUE
      END IF

      IF (NOD2A.GE.1) THEN
          DO 36 LL = 1, NOD2A
             CFA = CFA + CFD2A(LL)
 36       CONTINUE
      END IF

      IF (NOD3A.GE.1) THEN
          DO 636 KP = 1, NOD3A
             CFA = CFA + CFD3A(KP)
636       CONTINUE
      END IF

      IF (NOSLCA.GE.1) THEN
          DO 736 Ks = 1, NOSLCA
             CFA = CFA + CFslca(Ks)
736       CONTINUE
      END IF

        CFA = CFA + CFOH
*    ______________________________________________________________
*    Setting the total feed concentration equal to the total conc.
*    of cations which is also equal to the total concentration of
*    anions because of charge balance.
*    ______________________________________________________________
        CF = (CFC+CFA)/2.       ! taking the average, changed by Liu
*    ______________________________________________________________
*    Calculating Reynolds Numbers.
*    ______________________________________________________________
      IF (VOID_FRAC.GE.1.0) THEN
         WRITE (*,*) " VOID_FRAC = 1, Abnormal exit"
```

```
         GOTO 538
      END IF

      RE_CAT = CAT_DIA*100.*VS*DEN/((1.-VOID_FRAC)*VISCO)
      RE_ANI = ANI_DIA*100.*VS*DEN/((1.-VOID_FRAC)*VISCO)
*
*    Calculating Schmidt Numbers.
*
      CALL SCHMIDT(VISCO,DEN,DIFU_CAT,NOCAT,SCH_CAT) ! for ionic
      CALL SCHMIDT(VISCO,DEN,DIFU_ANI,NOANI,SCH_ANI) ! for molecular

      IF (NOD1C.GE.1) THEN
         CALL SCHMIDT(VISCO,DEN,DIFU_D1C,NOD1C,SCH_D1C)
         CALL SCHMIDT(VISCO,DEN,DIFU_D1CSTA,NOD1C,SCH_D1CSTA)
      END IF

      IF (NOD1A.GE.1) THEN
         CALL SCHMIDT(VISCO,DEN,DIFU_D1A,NOD1A,SCH_D1A)
         CALL SCHMIDT(VISCO,DEN,DIFU_D1ASTA,NOD1A,SCH_D1ASTA)
      ENDIF

      IF (NOD2C.GE.1) THEN
         CALL SCHMIDT(VISCO,DEN,DIFU_D2C,NOD2C,SCH_D2C)
         SCH_D2CSTA = (VISCO/100.)/DEN/DIFU_D2CSTA
      END IF

      IF (NOD2A.GE.1) THEN
         CALL SCHMIDT(VISCO,DEN,DIFU_D2A,NOD2A,SCH_D2A)
         SCH_D2ASTA = (VISCO/100.)/DEN/DIFU_D2ASTA
      END IF

      IF (NOD3C.GE.1) THEN
         CALL SCHMIDT(VISCO,DEN,DIFU_D3C,NOD3C,SCH_D3C)
         SCH_D3CSTA = (VISCO/100.)/DEN/DIFU_D3CSTA
      END IF

      IF (NOD3A.GE.1) THEN
         CALL SCHMIDT(VISCO,DEN,DIFU_D3A,NOD3A,SCH_D3A)
         SCH_D3ASTA = (VISCO/100.)/DEN/DIFU_D3ASTA
      END IF

      IF (NOSLCA.GE.1) THEN
         CALL SCHMIDT(VISCO,DEN,DIFU_slca,NOSLCA,SCH_slca)
         SCH_slcaSH = (VISCO/100.)/DEN/DIFU_slcaSH
         SCH_slcaSC = (VISCO/100.)/DEN/DIFU_slcaSC
         SCH_slcaPT = (VISCO/100.)/DEN/DIFU_slcaPT
      END IF
*
*    Calculating the mass transfer coefficients.
*

      DO 44 II=1,NOCAT
         MTC_CAT(II) = FUNC_MTC(RE_CAT,SCH_CAT(II),DIFU_CAT(II)
     1                          ,CAT_DIA)
 44   CONTINUE

      IF (NOD1C.GE.1) THEN
      DO 46 KK=1,NOD1C
              MTC_D1C(KK) = FUNC_MTC(RE_CAT,SCH_D1C(KK),DIFU_D1C(KK)
     1                          ,CAT_DIA)
         IF (CBD1CSTA(KK).GT.CBD1C(KK)) THEN
              MTC_D1CSTA(KK) = (1.+LOG(DIFU_D1CSTA(KK)*CBD1CSTA(KK)/
     1            DIFU_D1C(KK)/CBD1C(KK)))*
     1        FUNC_MTC(RE_CAT,SCH_D1CSTA(KK),DIFU_D1CSTA(KK),CAT_DIA)
         ELSE
              MTC_D1CSTA(KK) = FUNC_MTC(RE_CAT,SCH_D1CSTA(KK),
     1            DIFU_D1CSTA(KK),CAT_DIA)
         END IF
 46   CONTINUE
      END IF
```

```
      IF (NOD2C.GE.1) THEN
         DO 47 NN=1,NOD2C
            MTC_D2C(NN) = FUNC_MTC(RE_CAT,SCH_D2C(NN),DIFU_D2C(NN)
     1                          ,CAT_DIA)
  47     CONTINUE
         IF (CBD2CSTA.GT.CBD2C(1)) THEN
            MTC_D2CSTA = (1.+LOG(DIFU_D2CSTA*CBD2CSTA/
     1              DIFU_D2C(1)/CBD2C(1)))*
     1              FUNC_MTC(RE_CAT,SCH_D2CSTA,DIFU_D2CSTA,CAT_DIA)
         ELSE
            MTC_D2CSTA = FUNC_MTC(RE_CAT,SCH_D2CSTA,DIFU_D2CSTA,
     1              CAT_DIA)
         ENDIF
      END IF

      IF (NOD3C.GE.1) THEN
         DO 647 KT=1,NOD3C
            MTC_D3C(KT) = FUNC_MTC(RE_CAT,SCH_D3C(KT),DIFU_D3C(KT)
     1                          ,CAT_DIA)
 647     CONTINUE
         IF (CBD3CSTA.GT.CBD3C(1)) THEN
            MTC_D3CSTA = (1.+LOG(DIFU_D3CSTA*CBD3CSTA/
     1              DIFU_D3C(1)/CBD3C(1)))*
     1              FUNC_MTC(RE_CAT,SCH_D3CSTA,DIFU_D3CSTA,CAT_DIA)
         ELSE
            MTC_D3CSTA = FUNC_MTC(RE_CAT,SCH_D3CSTA,DIFU_D3CSTA,
     1              CAT_DIA)
         ENDIF
      END IF

      DO 48 JJ=1,NOANI
           MTC_ANI(JJ) = FUNC_MTC(RE_ANI,SCH_ANI(JJ),DIFU_ANI(JJ)
     1                          ,ANI_DIA)
  48  CONTINUE

      IF (NOD1A.GE.1) THEN
       DO 49 MM=1,NOD1A
          MTC_D1A(MM) = FUNC_MTC(RE_ANI,SCH_D1A(MM),DIFU_D1A(MM)
     1                          ,ANI_DIA)
          IF (CBD1ASTA(MM).GT.CBD1A(MM)) THEN
              MTC_D1ASTA(MM) = (1.+LOG(DIFU_D1ASTA(MM)*CBD1ASTA(MM)/
     1            DIFU_D1A(MM)/CBD1A(MM)))*
     1        FUNC_MTC(RE_ANI,SCH_D1ASTA(MM),DIFU_D1ASTA(MM),ANI_DIA)
          ELSE
              MTC_D1ASTA(MM) = FUNC_MTC(RE_ANI,SCH_D1ASTA(MM),
     1            DIFU_D1ASTA(MM),ANI_DIA)
          END IF
  49   CONTINUE
      ENDIF

      IF (NOD2A.GE.1) THEN
         DO 50 LL=1,NOD2A
            MTC_D2A(LL) = FUNC_MTC(RE_ANI,SCH_D2A(LL),DIFU_D2A(LL)
     1                          ,ANI_DIA)
  50     CONTINUE
         IF (CBD2ASTA.GT.CBD2A(1)) THEN
            MTC_D2ASTA = (1.+LOG(DIFU_D2ASTA*CBD2ASTA/
     1              DIFU_D2A(1)/CBD2A(1)))*
     1              FUNC_MTC(RE_ANI,SCH_D2ASTA,DIFU_D2ASTA,ANI_DIA)
         ELSE
            MTC_D2ASTA = FUNC_MTC(RE_ANI,SCH_D2ASTA,DIFU_D2ASTA,
     1              ANI_DIA)
         ENDIF
      END IF

      IF (NOD3A.GE.1) THEN
         DO 650 KP=1,NOD3A
            MTC_D3A(KP) = FUNC_MTC(RE_ANI,SCH_D3A(KP),DIFU_D3A(KP)
     1                          ,ANI_DIA)
 650     CONTINUE
         IF (CBD3ASTA.GT.CBD3A(1)) THEN
```

```
            MTC_D3ASTA = (1.+LOG(DIFU_D3ASTA*CBD3ASTA/
     1              DIFU_D3A(1)/CBD3A(1)))*
     1              FUNC_MTC(RE_ANI,SCH_D3ASTA,DIFU_D3ASTA,ANI_DIA)
         ELSE
            MTC_D3ASTA = FUNC_MTC(RE_ANI,SCH_D3ASTA,DIFU_D3ASTA,
     1              ANI_DIA)
         ENDIF
      END IF

      IF (NOSLCA.GE.1) THEN
         DO 750 KS=1,NOSLCA
            MTC_slca(KS) = FUNC_MTC(RE_ANI,SCH_slca(KS),DIFU_slca(KS)
     1                            ,ANI_DIA)
750      CONTINUE
         IF (CBslcaSH.GT.CBslca(1)) THEN
            MTC_slcaSH = (1.+LOG(DIFU_slcaSH*CBslcaSH/
     1           DIFU_slca(1)/CBslca(1)))*
     1         FUNC_MTC(RE_ANI,SCH_slcaSH,DIFU_slcaSH,ANI_DIA)
         ELSE
            MTC_slcaSH = FUNC_MTC(RE_ANI,SCH_slcaSH,DIFU_slcaSH,
     1              ANI_DIA)    ! for molecular silica Si(OH)4
         ENDIF
            MTC_slcaSC = FUNC_MTC(RE_ANI,SCH_slcaSC,DIFU_slcaSC,
     1              ANI_DIA)    ! for colloidal silica

            MTC_slcaPT = FUNC_MTC(RE_ANI,SCH_slcaPT,DIFU_slcaPT,
     1              ANI_DIA)    ! for particular silica
      END IF
*     ---------------------------------------------------------
*     Call the subroutine "COMBNO" to combine the total number
*     of cations and total number of anions in the solution
*     ---------------------------------------------------------

        IF (NOCAT.GE.1) THEN
           CALL COMBNO(NOCAT,NC,0)
        ENDIF

        IF (NOD1C.GE.1) THEN
           CALL COMBNO(NOD1C,NC,NC)
        ENDIF

        IF (NOD2C.GE.1) THEN
           CALL COMBNO(NOD2C,NC,NC)
        ENDIF

        IF (NOD3C.GE.1) THEN
           CALL COMBNO(NOD3C,NC,NC)
        ENDIF

        IF (NOANI.GE.1) THEN
           CALL COMBNO(NOANI,NA,0)
        ENDIF

        IF (NOD1A.GE.1) THEN
           CALL COMBNO(NOD1A,NA,NA)
        ENDIF

        IF (NOD2A.GE.1) THEN
           CALL COMBNO(NOD2A,NA,NA)
        ENDIF

        IF (NOD3A.GE.1) THEN
           CALL COMBNO(NOD3A,NA,NA)
        ENDIF

        IF (NOSLCA.GE.1) THEN
           CALL COMBNO(NOSLCA,NA,NA)
        ENDIF
*     ------------------------------------------------------------------
*     Call subroutine "COMBPROP" to combine physlcal properties like
*     selectivities, valences, diffusivities and mass transfer
```

```
*    coefficients for the interfacial and flux calculations.
*    ______________________________________________________________

        IF (NOCAT.GE.1) THEN
           CALL COMBPROP(SEL_CAT,SELCOMC,VACAT,VACOMC,DIFU_CAT,DIFUCOMC,
     1          MTC_CAT,MTC_COMC,EWCAT,EWC,NOCAT,0)
       ENDIF

       IF (NOD1C.GE.1) THEN
           CALL COMBPROP(SELD1C,SELCOMC,VAD1C,VACOMC,DIFU_D1C,
     1          DIFUCOMC,MTC_D1C,MTC_COMC,EWD1C,EWC,NOD1C,NOCAT)
       ENDIF

       IF (NOD2C.GE.1) THEN
           CALL COMBPROP(SELD2C,SELCOMC,VAD2C,VACOMC,DIFU_D2C,
     1      DIFUCOMC,MTC_D2C,MTC_COMC,EWD2C,EWC,NOD2C,NOCAT+NOD1C)
       ENDIF

       IF (NOD3C.GE.1) THEN
           CALL COMBPROP(SELD3C,SELCOMC,VAD3C,VACOMC,DIFU_D3C,DIFUCOMC,
     1      MTC_D3C,MTC_COMC,EWD3C,EWC,NOD3C,NOCAT+NOD1C+NOD2C)
       ENDIF

        IF (NOANI.GE.1) THEN
           CALL COMBPROP(SEL_ANI,SELCOMA,VAANI,VACOMA,DIFU_ANI,
     1        DIFUCOMA,MTC_ANI,MTC_COMA,EWANI,EWA,NOANI,0)
       ENDIF

        IF (NOD1A.GE.1) THEN
           CALL COMBPROP(SELD1A,SELCOMA,VAD1A,VACOMA,DIFU_D1A,
     1      DIFUCOMA,MTC_D1A,MTC_COMA,EWD1A,EWA,NOD1A,NOANI)
       ENDIF

        IF (NOD2A.GE.1) THEN
           CALL COMBPROP(SELD2A,SELCOMA,VAD2A,VACOMA,DIFU_D2A,
     1      DIFUCOMA,MTC_D2A,MTC_COMA,EWD2A,EWA,NOD2A,NOANI+NOD1A)
       ENDIF

        IF (NOD3A.GE.1) THEN
           CALL COMBPROP(SELD3A,SELCOMA,VAD3A,VACOMA,DIFU_D3A,DIFUCOMA,
     1      MTC_D3A,MTC_COMA,EWD3A,EWA,NOD3A,NOANI+NOD1A+NOD2A)
       ENDIF

       IF (NOSLCA.GE.1) THEN
           CALL COMBPROP(SELslca,SELCOMA,VAslca,VACOMA,DIFU_slca,
     1                  DIFUCOMA,MTC_slca,MTC_COMA,EWslca,EWA,NOSLCA,
     1                  NOANI+NOD1A+NOD2A+NOD3A)
       ENDIF

*---- Adding the properties of H+ as the (NC+1) element of array----*

      SELCOMC(NC+1) = SELH
      VACOMC(NC+1) = VAH
      DIFUCOMC(NC+1) = DIFUH

C--------implementing the effects of cationic resin fouling ------
      IF (FFLAG.eq.'Y'.or.FFLAG.eq.'y') THEN
      DO 55 I=1,NC
         MTC_COMC(I)=(1.-p)*MTC_COMC(I)
 55   CONTINUE
      ENDIF

*----- Adding the properties of OH- as the (NA+1) element of array -----*

      SELCOMA(NA+1) = SELOH
      VACOMA(NA+1) = VAOH
      DIFUCOMA(NA+1) = DIFUOH

C ------- Implementing the effects of anionic resin fouling
      IF (FFLAG.eq.'Y'.or.FFLAG.eq.'y') THEN
      DO 59 I = 1, NA
```

```
         MTC_COMA(I)=(1.-q)*MTC_COMA(I)
 59   CONTINUE
      ENDIF
*    _______________________________________________________________
*     Calling the subroutine to sort the valences, selectivities,
*     diffusivities and indices.
*    _______________________________________________________________

      CALL SORT_IONS(MTC_COMC,MTC_COMA)  !added two parameters 5/26/98
*    _______________________________________________________________
*    Setting the priorities of the ions.
*    _______________________________________________________________
      DO 60 II = 1,NC
         PRIORITY_C(INDEXC(II)) = II
 60   CONTINUE

      DO 62 JJ = 1,NA
         PRIORITY_A(INDEXA(JJ)) = JJ
 62   CONTINUE

*    ________________________________________________________________
*    Choosing the reference ion for calculations. The anion with
*    largest mass transfer coefficient is chosen as reference
*    ________________________________________________________________
      IF (NA.EQ.1) THEN               ! if only one anion, itself is reference
         MTC_REF = MTC_COMA(NA)
         GOTO 65
      END IF

*---- Pick hydroxide as the reference, if no other anion is present.

      IF(NA.EQ.0)THEN
         SCH_OH = (VISCO/100.)/DEN/DIFUOH
         MTC_OH = FUNC_MTC(RE_ANI,SCH_OH,DIFUOH,ANI_DIA)
         MTC_REF = MTC_OH
         GOTO 65
      ENDIF
*-------------

      NN = 1
      DO 64 JJ = 2, NA
         IF (MTC_COMA(NN).LT.MTC_COMA(JJ)) THEN
             NN = JJ
         END IF
 64   CONTINUE
      MTC_REF = MTC_COMA(NN)
*    ________________________________________________________________
*    Calculating the dimensionless column height(CHTD) and the total
*    number of distance steps (NOSLICE) down the column.
*    ________________________________________________________________

 65   IF (VS.LE.0.0 .OR. ANI_DIA .LE. 0.0) THEN
          WRITE(*,*)"ERROR: VS <= 0.0 and/or ANI_DIA <= 0.0"
          GOTO 538
      ENDIF

      CHTD = MTC_REF*(1.-VOID_FRAC)*RES_HT/(VS*ANI_DIA)
      NOSLICE = CHTD/XI-3+3*XI/XI1

C######################################################################
      IF (RFLAG.EQ.'Y'.OR.RFLAG.EQ.'y') THEN
       GRPNUM = MTC_REF*(1.-VOID_FRAC)/(VS*ANI_DIA)
       CHTD1 = GRPNUM*BED_HT1
       CHTD2 = GRPNUM*BED_HT2
       CHTD3 = GRPNUM*BED_HT3

       NT1=CHTD1/XI-3+3*XI/XI1
       NT2=CHTD2/XI
       NT3=CHTD3/XI

       NOSLICE = NT1+NT2+NT3
```

```
      ENDIF
C######################################################################

      WRITE(*,*)
      WRITE(*,*)"Number of slices =",NOSLICE
      WRITE(*,*)

      IF(NOSLICE.GE.5000)THEN
         WRITE(*,*)"ERROR: Insufficient array sizes"
         GO TO 538
      ENDIF
*    ___________________________________________________________________
*    Set the initial resin loadings throughout the column. Resin loadings
*    are assumed to be uniform throughout the column.
*    ___________________________________________________________________

      MT = NOSLICE + 1
      DO 74 M = 1, MT
         DO 66 II = 1,NOCAT
            YRC(PRIORITY_C(II),1,M) = YCINIT(II)
 66      CONTINUE

         IF (NOD1C.GE.1) THEN
            DO 68 KK = 1, NOD1C
               YRC(PRIORITY_C(NOCAT+KK),1,M) = YD1C_INIT(KK)
 68         CONTINUE
         END IF

         IF (NOD2C.GE.1) THEN
            DO 69 NN = 1,NOD2C
               YRC(PRIORITY_C(NOCAT+NOD1C+NN),1,M) = YD2C_INIT(NN)
 69         CONTINUE
         END IF

         IF (NOD3C.GE.1) THEN
            DO 669 KT = 1,NOD3C
            YRC(PRIORITY_C(NOCAT+NOD1C+NOD2C+KT),1,M) = YD3C_INIT(KT)
669         CONTINUE
         END IF

         DO 70 JJ = 1, NOANI
            YRA(PRIORITY_A(JJ),1,M) = YAINIT(JJ)
 70      CONTINUE

         IF (NOD1A.GE.1) THEN
            DO 73 MM = 1, NOD1A
                YRA(PRIORITY_A(NOANI+MM),1,M) = YD1A_INIT(MM)
 73         CONTINUE
         ENDIF

         IF (NOD2A.GE.1) THEN
            DO 72 LL = 1,NOD2A
               YRA(PRIORITY_A(NOANI+NOD1A+LL),1,M) = YD2A_INIT(LL)
 72         CONTINUE
         END IF

         IF (NOD3A.GE.1) THEN
            DO 672 KP = 1,NOD3A
            YRA(PRIORITY_A(NOANI+NOD1A+NOD2A+KP),1,M) = YD3A_INIT(KP)
672         CONTINUE
         END IF

         IF (NOSLCA.GE.1) THEN
            DO 674 KS = 1,NOSLCA
            YRA(PRIORITY_A(NOANI+NOD1A+NOD2A+NOD3A+KS),1,M)
     1                     = Yslca_INIT(KS)
674         CONTINUE
         END IF
 74   CONTINUE
*    ___________________________________________________________________
*    Calculating dimensionless program time limit based on inlet
```

```
*    conditions (at Z = 0)
*    __________________________________________________________________

      IF (VOL_FLOW .LE.0.0 .OR. CF.LE.0.0) THEN
         WRITE(*,*) "ERROR: VOL_FLOW <= 0.0 and/or CF <= 0.0"
         GOTO 538
      ENDIF

      TMAXC = CAT_CAP*3.142*(BED_DIA/2.)**2.*RES_HT*FCR
     1          /(VOL_FLOW*CF*60.)
      TMAXA = ANI_CAP*3.142*(BED_DIA/2.)**2.*RES_HT*FAR
     1          /(VOL_FLOW*CF*60.)

C#####################################################################
C     Recalculate the 'TMAXC'and 'TMAXA' for the case when different
C     cation/anion ratio encounted at different position of column
C#####################################################################
      IF (RFLAG.EQ.'Y'.OR.RFLAG.EQ.'y') THEN
        GRPCAT = CAT_CAP*3.142*(BED_DIA/2.)**2/(VOL_FLOW*CF*60.)
        TMAXC1 = GRPCAT*BED_HT1*FCR1
        TMAXC2 = GRPCAT*BED_HT2*FCR2
        TMAXC3 = GRPCAT*BED_HT3*FCR3
        TMAXC=TMAXC1+TMAXC2+TMAXC3

        GRPANI = ANI_CAP*3.142*(BED_DIA/2.)**2/(VOL_FLOW*CF*60.)
        TMAXA1 = GRPANI*BED_HT1*FAR1
        TMAXA2 = GRPANI*BED_HT2*FAR2
        TMAXA3 = GRPANI*BED_HT3*FAR3
        TMAXA=TMAXA1+TMAXA2+TMAXA3
      ENDIF
C####################################################################

      IF(TMAXC.GE.TMAXA) THEN
          TMAX = TMAXC
      ELSE
          TMAX = TMAXA
      ENDIF

      IF (ANI_DIA .LE.0.0 .OR. ANI_CAP .LE.0.0) THEN
          WRITE(*,*)"ERROR: ANI_DIA <= 0.0 OR ANI_CAP <= 0.0"
          GOTO 538
      END IF

      TAUMAX = MTC_REF*CF*(TMAX*60.)/(ANI_DIA*ANI_CAP)
      DMAX=TMAX/1440.

      WRITE(*,505)
      WRITE(*,506) DMAX
      WRITE(*,507)
  505 FORMAT(' Program run time is based on total resin capacity and')
  506 FORMAT(' flow conditions. The program shows results for',F12.1)
  507 FORMAT(' days of column operation for the current conditions.')

c@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
c    Ask user if he/she wants to handle different inlet concentrations
c    at different time
c@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
      WRITE(*,*)
      WRITE(*,*) 'Do you want to handle variable inlet concentrations'
      WRITE(*,*) 'at different time (y/n)'
      READ(*,501) CFLAG

      IF (CFLAG.eq.'Y'.or.CFLAG.eq.'y') THEN
         WRITE(*,*) 'Input the new inlet concentrations (meq/ml):'
         DO 441 II=1, NOCAT
            WRITE(*,*)'    cation',II,'  concentration'
            READ(*,*)CFCAT_NEW1(II)
 441     CONTINUE

         DO 442 JJ=1, NOANI
            WRITE(*,*)'    anion',JJ,'  concentration'
```

```
            READ(*,*)CFANI_NEW1(JJ)
 442     CONTINUE

         IF(NOD1C.GE.1) THEN
            DO 443 KK=1, NOD1C
              WRITE(*,*) '  new total feed concentrations of amine',KK
              READ(*,*)CFEED_D1CNEW1(KK)
 443        CONTINUE
         ENDIF

         IF (NOD1A.GE.1) THEN
            DO 444 MM=1, NOD1A
            WRITE(*,*) '  new total feed conc.of carboxylic acid',MM
            READ(*,*)CFEED_D1ANEW1(MM)
 444        CONTINUE
         ENDIF

          IF (NOD2C.GE.1) THEN
           WRITE(*,*)'  new total feed concentration of divalent amines'
           READ(*,*)CFEED_D2CNEW1
         ENDIF

          IF(NOD2A.GE.1) THEN
             WRITE(*,*) '  new total feed concentration of carbonates'
             READ(*,*)CFEED_D2ANEW1
         ENDIF

          IF (NOD3C.GE.1) THEN
          WRITE(*,*)'  new total feed concentration of Trivalent amines'
          READ(*,*)CFEED_D3CNEW1
         ENDIF

          IF(NOD3A.GE.1) THEN
             WRITE(*,*) '  new total feed concentration of Phosphates'
             READ(*,*)CFEED_D3ANEW1
         ENDIF

          IF(NOSLCA.GE.1) THEN
             WRITE(*,*) '  new total feed concentration of Silica'
             READ(*,*)CFEED_slcaNEW1
         ENDIF

          WRITE(*,*)'Input the time (days) corresponding to these conc.'
          READ(*,*)DAY_NEW1
          if (DAY_NEW1.GE.DMAX) then
            write(*,*) "abnormal, your input data is wrong:"
            write(*,*) "day_new1 should be less than Dmax"
          endif
       ENDIF
c@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
C     Ask user if he/she wants to handle the case of variable inlet
C     flowrate.  If it is 'Y/y', the user needs to input the new
C     flowrate and the time when the flowrate changes. If it is 'N/n',
C     the program will skip this part, and execute as normal.
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       WRITE(*,*)
       WRITE(*,*)'Do you want to handle the case of variable'
       WRITE(*,*)'inlet flowrate at different time (y/n)?'
       READ(*,501) VFLAG

       IF (VFLAG.EQ.'Y'.OR.VFLAG.EQ.'y') THEN
           WRITE(*,*) 'Input the new flowrate (cm3/s)'
           READ(*,*)VOL_FLOWNEW
           WRITE(*,*)'Input the time(days) corresponding to the new flowrate'
          READ(*,*)DAY_NEW2
          if (DAY_NEW2.GE.DMAX) then
            write(*,*) "abnormal, your input data wrong:"
            write(*,*) "day_new2 should be less than Dmax"
          endif
```

```
       ENDIF
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*
*    Initialize values prior to iterative loops
*
        J = 1
        JK = 1
        TAUTOT = 0.
        JFLAG = 0
        KPRINT = 2

        DO 76 K = 1, NOSLICE
           NEXC(K) = (NC+1)   ! Initially all the ions exchange in all
           NEXA(K) = (NA+1)   ! the slices
 76     CONTINUE

*
*    Defining the desulphonation term (Fisher's data)
*

      IF (SFLAG.EQ.'Y'.OR.SFLAG.EQ.'y') THEN
         S1 = (7.5E+6*EXP(-10278.6/(TMPC+273.15))*CHTD
     1        *3.1415927*(BED_DIA**2.)*CAT_CAP)*(VS*ANI_DIA)*FCR
     1        /(NOSLICE*3600.*4.0*VOL_FLOW*MTC_REF*(1.-VOID_FRAC))

        DD = S1/CF
      ELSEIF (SFLAG.EQ.'N'.OR.SFLAG.EQ.'n') THEN
        DD = 0.0
      END IF

*
*     Calculating the constants outside the loops to avoid repititive
*     calculations inside the loops. These constants are used inside
*     the loops during rate calculations.
*
      IF (MTC_REF .LE. 0.0) THEN
          WRITE(*,*) "ERROR: MTC_REF <= 0.0"
          GOTO 538
      END IF

      CONP = -6.*ANI_DIA/(MTC_REF*CAT_DIA*CF)
      CONS = -6./(MTC_REF*CF)
      CONY = ANI_CAP/CAT_CAP

c@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
c     Calculate the dimensionless time corresponding to the days
c     when the new inlet concentrations start based on the old
c     CF and ANI_CAP,CAT_CAP
c@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
        IF(CFLAG.eq.'Y'.or.CFLAG.eq.'y') THEN
        TAUMAX_OLD1=DAY_NEW1/DMAX*TAUMAX

        ANI_CAPNEW1=(1-DAY_NEW1/DMAX)*ANI_CAP
        CAT_CAPNEW1=(1-DAY_NEW1/DMAX)*CAT_CAP
          CALL NEWCONCEN(PH,CFCAT_NEW1,CFANI_NEW1,CFEED_D1CNEW1,
     1      CFEED_D1ANEW1,CFEED_D2CNEW1,CFEED_D2ANEW1,
     1      CFEED_D3CNEW1,CFEED_D3ANEW1,CFEED_slcaNEW1,ANI_CAPNEW1,
     1      CAT_CAPNEW1,VOL_FLOW,MTC_REF,S1,CFH_NEW,CFOH_NEW,
     1      PH_FEEDNEW,CFD1C_NEW,CFD1CSTA_NEW,CFD1A_NEW,CFD1ASTA_NEW,
     1      CFD2C_NEW,CFD2CSTA_NEW,CFD2A_NEW,CFD2ASTA_NEW,CFD3C_NEW,
     1      CFD3CSTA_NEW,CFD3A_NEW,CFD3ASTA_NEW,CFslca_NEW,
     1      CFSLCASH_NEW,CF_NEW,TAUMAX_NEW,DMAX_NEW,DD_NEW,
     1      CONP_NEW,CONS_NEW)

         DMAX_PRAC=DMAX_NEW+DAY_NEW1

      WRITE(*,*)
      WRITE(*,709)
      WRITE(*,710) DMAX_NEW            ! Changed by LIU 1/8/98
      WRITE(*,*)
      WRITE(*,711) DMAX_PRAC
```

```
 709  format(' After the change of inlet concs -- shift to new conc')
 710  format(' the column will operate for another ',F12.1,'days')
 711  format(' So the actual total operating time is',F12.1, 'days')

        ENDIF
*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*     Calculate the dimensionless time corresponding to the days
*     when the inlet flowrate changes based on the old values of
*     VOL_FLOW and ANI_CAP,CAT_CAP
*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        IF(VFLAG.eq.'Y'.or.VFLAG.eq.'y') THEN
        TAUMAX_OLD2=DAY_NEW2/DMAX*TAUMAX      ! ratio them
        ENDIF
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        IF (VFLAG.eq.'Y'.or.VFLAG.eq.'y') THEN
                              !(deduct the capacity consumed)
        ANI_CAPNEW2=(1-DAY_NEW2/DMAX)*ANI_CAP
        CAT_CAPNEW2=(1-DAY_NEW2/DMAX)*CAT_CAP
        CALL NEWFLOW(XI,XI1,p,q,CAT_CAPNEW2,ANI_CAPNEW2,VOL_FLOWNEW,
     1     VS_NEW,CHTDNEW,CBD1C,CBD1A,CBD2C,CBD2A,CBD3C,CBD3A,CBSLCA,
     1     CBD1CSTA,CBD1ASTA,CBD2CSTA,CBD2ASTA,CBD3CSTA,CBD3ASTA,
     1     CBSLCASH,CF,MTC_CATNEW,MTC_ANINEW,
     1     MTC_D1CNEW,MTC_D1CSTANEW,MTC_D1ANEW,MTC_D1ASTANEW,
     1     MTC_D2CNEW,MTC_D2CSTANEW,MTC_D2ANEW,MTC_D2ASTANEW,
     1     MTC_D3CNEW,MTC_D3CSTANEW,MTC_D3ANEW,MTC_D3ASTANEW,
     1     MTC_slcaNEW,MTC_slcaSHNEW,MTC_slcaSCNEW,MTC_slcaPTNEW,
     1     MTC_COMCNEW,MTC_COMANEW,MTC_REFNEW,TAUMAX_NEW,DMAX_NEW,
     1     DD_NEW,CONP_NEW,CONS_NEW,NOSLICE_NEW)

        DMAX_PRAC=DMAX_NEW+DAY_NEW2

      WRITE(*,*)
      WRITE(*,909)
      WRITE(*,910) DMAX_NEW             ! Added by LIU 7/8/98
      WRITE(*,*)
      WRITE(*,911) DMAX_PRAC
 909  format(' After the change of flowrate - shift to new flowrate')
 910  format(' the column will operate for another ',F12.1,'days')
 911  format(' So the actual total operating time is',F12.1, 'days')

      WRITE(*,*)
      WRITE(*,*)"Number of slices =",NOSLICE_NEW
      WRITE(*,*)

        IF(NOSLICE.GE.5000)THEN
           WRITE(*,*)"ERROR: Insufficient array sizes"
           GO TO 538
        ENDIF

        ENDIF

C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C    Start handling the bed cleaning effect
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      WRITE(*,*)
      WRITE(*,*) 'Do you want to handle the bed cleaning case (y/n)?'
      READ(*,501) BFLAG
      if (BFLAG.eq.'Y'.OR. BFLAG.eq.'y') THEN
      write(*,*)
      write(*,*) "input the time interval of bed cleaning"
      read(*,*) Day_clean

      Tclean_user=Day_clean*1440
      tau_clean=MTC_REF*CF*(Tclean_user*60.)/(ANI_DIA*ANI_CAP)
      endif
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
*-------------------------------------------------------------------
      WRITE(*,*)
      WRITE(*,*)"Do you wish to see the results for a different"
      WRITE(*,*)"number of days? (y/n)"
      READ(*,501)TFLAG

      IF(TFLAG.EQ.'Y'.OR.TFLAG.EQ.'y')THEN
         WRITE(*,*)"Enter the number of days"
         READ(*,*)DMAX_USER

         TMAX_USER = DMAX_USER*1440.
         TAUMAX = MTC_REF*CF*(TMAX_USER*60.)/(ANI_DIA*ANI_CAP)
      ENDIF
*
*     Printing the headings for concentration profiles
*
      WRITE(*,*)"*****************************************************"
      WRITE(*,*)"Printing breakthrough curves."
      WRITE(*,*)"The output columns are in the following order:"
      WRITE(*,*)"Time-pH-Cat-Amines(I,II,III)-Ani-COOH-Carb-Phos-Silica"
      WRITE(*,*)"(Days)        (ppb)     (ppb)      (ppb)    (ppb)"
      WRITE(*,*)"*****************************************************"
*
*     Beginning of time step loop within which all column calculations are
*     implemented, time is incremented and outlet concentrations checked.
*
 111  CONTINUE

c@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
        IF (TAUTOT.GT.TAUMAX_OLD1) THEN
           IF(CFLAG.EQ.'Y'.OR.CFLAG.EQ.'y') THEN

           TAUMAX=TAUMAX_NEW+TAUMAX_OLD1

           IF(TFLAG.EQ.'Y'.OR.TFLAG.EQ.'y')THEN
             TMAX_USER = DMAX_USER*1440.
             TAUMAX = MTC_REF*CF*(TMAX_USER*60.)/(ANI_DIA*ANI_CAP)
           ENDIF
               CF_DUM=CF
               CF=CF_NEW  ! data from calling subroutine 'newconcen'
               CFC=CF
               CFA=CF

             CFH=CFH_NEW
             CFOH=CFOH_NEW
             PH_FEED=PH_FEEDNEW

             CONP=CONP_NEW
             CONS=CONS_NEW

             DO 551 II = 1, NOCAT
                  CFCAT(II) = CFCAT_NEW1(II)
 551         CONTINUE

             DO 552 JJ=1,NOANI
                  CFANI(JJ) = CFANI_NEW1(JJ)
 552         CONTINUE

             DO 553 KK = 1, NOD1C
                  CFD1C(KK)=CFD1C_NEW(KK)
                  CFEED_D1C(KK) = CFEED_D1CNEW1(KK)
 553         CONTINUE

              DO 554 MM = 1, NOD1A
                  CFD1A(MM) = CFD1A_NEW(MM)
                  CFEED_D1A(MM) = CFEED_D1ANEW1(MM)
 554         CONTINUE

              DO 555 NN = 1, NOD2C
                  CFD2C(NN) = CFD2C_NEW(NN)
```

```
 555         CONTINUE
                CFEED_D2C = CFEED_D2CNEW1

             DO 556 LL = 1, NOD2A
                 CFD2A(LL)=CFD2A_NEW(LL)
 556         CONTINUE
                CFEED_D2A = CFEED_D2ANEW1

              DO 655 KT = 1, NOD3C
                 CFD3C(KT)=CFD3C_NEW(KT)
 655         CONTINUE
                CFEED_D3C = CFEED_D3CNEW1

             DO 656 KP = 1, NOD3A
                 CFD3A(KP) = CFD3A_NEW(KP)
 656         CONTINUE
                CFEED_D3A = CFEED_D3ANEW1

              DO 657 KS = 1, NOSLCA
                     CFslca(KS) = CFslca_NEW(KS)
 657         CONTINUE
                CFEED_slca = CFEED_slcaNEW1
          ENDIF
       ENDIF
c@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       IF (TAUTOT.GT.TAUMAX_OLD2) THEN
          IF(VFLAG.EQ.'Y'.OR.VFLAG.EQ.'y') THEN
             !using data calculated from calling subroutine 'newflow'
              TAUMAX=TAUMAX_NEW+TAUMAX_OLD2

            IF(TFLAG.EQ.'Y'.OR.TFLAG.EQ.'y')THEN
             TMAX_USER = DMAX_USER*1440.
             TAUMAX = MTC_REF*CF*(TMAX_USER*60.)/(ANI_DIA*ANI_CAP)
            ENDIF
              VS=VS_NEW
               CHTD=CHTDNEW
              NOSLICE=NOSLICE_NEW
              MTC_REF=MTC_REFNEW

               CONP=CONP_NEW
               CONS=CONS_NEW

              DO 769 I = 1, NC
                 MTC_COMC(I)=MTC_COMCNEW(I)
 769         CONTINUE

             DO 770 I=1,NA
                 MTC_COMA(I)=MTC_COMANEW(I)
 770         CONTINUE

             DO 751 II = 1, NOCAT
                 MTC_CAT(II) = MTC_CATNEW(II)
 751         CONTINUE

             DO 752 JJ=1,NOANI
                 MTC_ANI(JJ) = MTC_ANINEW(JJ)
 752         CONTINUE

             DO 753 KK = 1, NOD1C
                 MTC_D1C(KK)=MTC_D1CNEW(KK)
                 MTC_D1CSTA(KK)=MTC_D1CSTANEW(KK)
 753         CONTINUE

              DO 754 MM = 1, NOD1A
                 MTC_D1A(MM) = MTC_D1ANEW(MM)
                 MTC_D1ASTA(MM) = MTC_D1ASTANEW(MM)
 754         CONTINUE

              DO 755 NN = 1, NOD2C
```

```
               MTC_D2C(NN) = MTC_D2CNEW(NN)
 755        CONTINUE
              MTC_D2CSTA = MTC_D2CSTANEW

            DO 756 LL = 1, NOD2A
               MTC_D2A(LL) = MTC_D2ANEW(LL)
 756        CONTINUE
              MTC_D2ASTA = MTC_D2ASTANEW

             DO 955 KT = 1, NOD3C
               MTC_D3C(KT)=MTC_D3CNEW(KT)
 955        CONTINUE
              MTC_D3CSTA = MTC_D3CSTANEW

            DO 956 KP = 1, NOD3A
               MTC_D3A(KP) = MTC_D3ANEW(KP)
 956        CONTINUE
              MTC_D3ASTA = MTC_D3ASTANEW

             DO 957 KS = 1, NOSLCA
                  MTC_slca(KS) = MTC_slcaNEW(KS)
 957        CONTINUE
              MTC_slcaSH = MTC_slcaSHNEW
              MTC_slcaSC = MTC_slcaSCNEW
              MTC_slcaPT = MTC_slcaPTNEW

*    ==================================================================
*    Interpolate resin loadings and bulk phase concentrations at new
*    slices using the natural cubic spline
*    ==================================================================
        NMT=NOSLICE_NEW+1          ! new number of slices

         IF(NMT.GE.5000)THEN
          WRITE(*,*)"ERROR: Insufficient array sizes"
          GO TO 538
        ENDIF
*     ______________________________________________________________
*     Converting two or three dimensional arrays into one dimensional
*     and then calling subroutine to interpolating values
*     ______________________________________________________________

          IF ((TAUTOT-TAUMAX_OLD2).LE.TAU) THEN
*    ------ Strong cations ------
        Do 1115 I=1, NC
           do 1116 K=1, MT
               XNC(K)=XBC(I,K)       ! for bulk phase fraction
             YNC(K)=YRC(I,JD,K)      ! for resin loadings
1116       continue
        CALL cubspline(CHTD,XI1,XI,MT,NMT,XNC,U,V,Z,W,B,S,T)
        CALL cubspline(CHTD,XI1,XI,MT,NMT,YNC,U,V,Z,W,B,S,T)
           do 1117 K=1, NMT
               XBC(I,K)=S(K)
               YRC(I,JD,K)=S(K)
1117       continue
1115    CONTINUE

*    ------ Strong anions ------
        Do 1118 I=1, NA
           do 1119 K=1, MT
               XNA(K)=XBA(I,K)
             YNA(K)=YRA(I,JD,K)
1119       continue
        CALL cubspline(CHTD,XI1,XI,MT,NMT,XNA,U,V,Z,W,B,S,T)
        CALL cubspline(CHTD,XI1,XI,MT,NMT,YNA,U,V,Z,W,B,S,T)
           do 1120 K=1, NMT
               XBA(I,K)=S(K)
             YRA(I,JD,K)=S(K)
1120       continue
1118    CONTINUE

*    ------ Type I disso cations ------
```

```
          DO 1121 I=1,NOD1C
             do 1122 K=1,MT
                 ZD1CSTA(K)=XBD1CSTA(I,K)
                 YsorpD1C(K)=YRC_sorp_D1C(I,JD,K)
1122        continue
         CALL cubspline(CHTD,XI1,XI,MT,NMT,ZD1CSTA,U,V,Z,W,B,S,T)
         CALL cubspline(CHTD,XI1,XI,MT,NMT,YsorpD1C,U,V,Z,W,B,S,T)
         CALL cubspline(CHTD,XI1,XI,MT,NMT,SUMsorpD1C,U,V,Z,W,B,S,T)
             do 1123 K=1,NMT
                 XBD1CSTA(I,K)=S(K)
               YRC_sorp_D1C(I,JD,K)=S(K)
               SUMsorpD1C(K)=S(K)
1123        continue
1121     CONTINUE

*    ------ Type I disso anions ------
         DO 1124 I=1,NOD1A
             do 1125 K=1,MT
                 ZD1ASTA(K)=XBD1ASTA(I,K)
               YsorpD1A(K)=YRA_sorp_D1A(I,JD,K)
1125        continue
         CALL cubspline(CHTD,XI1,XI,MT,NMT,ZD1ASTA,U,V,Z,W,B,S,T)
         CALL cubspline(CHTD,XI1,XI,MT,NMT,YsorpD1A,U,V,Z,W,B,S,T)
         CALL cubspline(CHTD,XI1,XI,MT,NMT,SUMsorpD1A,U,V,Z,W,B,S,T)
             do 1126 K=1,NMT
                 XBD1ASTA(I,K)=S(K)
               YRA_sorp_D1A(I,JD,K)=S(K)
               SUMsorpD1A(K)=S(K)
1126        continue
1124     CONTINUE

*    ------ Type II disso cation (molecular form) ------
            do 1150 K=1,MT
                    YsorpD2C(K)=YRC_sorp_D2C(1,JD,K)
1150        continue
         CALL cubspline(CHTD,XI1,XI,MT,NMT,XBD2CSTA,U,V,Z,W,B,S,T)
          CALL cubspline(CHTD,XI1,XI,MT,NMT,YsorpD2C,U,V,Z,W,B,S,T)
         CALL cubspline(CHTD,XI1,XI,MT,NMT,SUMsorpD2C,U,V,Z,W,B,S,T)
             do 1127 K=1,NMT
                 XBD2CSTA(K)=S(K)
                 YRC_sorp_D2C(1,JD,K)=S(K)
               SUMsorpD2C(K)=S(K)
1127        continue

*    ------ Type II disso anion (molecular form) ------
            do 1152 K=1,MT
                    YsorpD2A(K)=YRA_sorp_D2A(1,JD,K)
1152        continue
         CALL cubspline(CHTD,XI1,XI,MT,NMT,XBD2ASTA,U,V,Z,W,B,S,T)
         CALL cubspline(CHTD,XI1,XI,MT,NMT,YsorpD2A,U,V,Z,W,B,S,T)
         CALL cubspline(CHTD,XI1,XI,MT,NMT,SUMsorpD2A,U,V,Z,W,B,S,T)
              do 1128 K=1,NMT
                 XBD2ASTA(K)=S(K)
                 YRA_sorp_D2A(1,JD,K)=S(K)
               SUMsorpD2A(K)=S(K)
1128        continue

*    ------ Type III disso cation (molecular form) ------
            do 1154 K=1,MT
                    YsorpD3C(K)=YRC_sorp_D3C(1,JD,K)
1154        continue
         CALL cubspline(CHTD,XI1,XI,MT,NMT,XBD3CSTA,U,V,Z,W,B,S,T)
         CALL cubspline(CHTD,XI1,XI,MT,NMT,YsorpD3C,U,V,Z,W,B,S,T)
         CALL cubspline(CHTD,XI1,XI,MT,NMT,SUMsorpD3C,U,V,Z,W,B,S,T)
              do 1129 K=1,NMT
                 XBD3CSTA(K)=S(K)
                 YRC_sorp_D3C(1,JD,K)=S(K)
               SUMsorpD3C(K)=S(K)
1129        continue

*    ------ Type III disso anion (molecular form) ------
```

```
            do 1156 K=1,MT
                    YsorpD3A(K)=YRA_sorp_D3A(1,JD,K)
1156        continue
         CALL cubspline(CHTD,XI1,XI,MT,NMT,XBD3ASTA,U,V,Z,W,B,S,T)
          CALL cubspline(CHTD,XI1,XI,MT,NMT,YsorpD3A,U,V,Z,W,B,S,T)
         CALL cubspline(CHTD,XI1,XI,MT,NMT,SUMsorpD3A,U,V,Z,W,B,S,T)
           do 1130 K=1,NMT
                 XBD3ASTA(K)=S(K)
                 YRA_sorp_D3A(1,JD,K)=S(K)
               SUMsorpD3A(K)=S(K)
1130        continue

*    ------ Silica (molecular form) ------
            do 1158 K=1,MT
                    YsorpSLCA(K)=YRA_sorp_slca(1,JD,K)
1158        continue
         CALL cubspline(CHTD,XI1,XI,MT,NMT,XBslcaSH,U,V,Z,W,B,S,T)
         CALL cubspline(CHTD,XI1,XI,MT,NMT,YsorpSLCA,U,V,Z,W,B,S,T)
             do 1131 K=1,NMT
                 XBslcaSH(K)=S(K)
                 YRA_sorp_slca(1,JD,K)=S(K)
1131        continue

         CALL cubspline(CHTD,XI1,XI,MT,NMT,XBslcaSC,U,V,Z,W,B,S,T)
             do 1132 K=1,NMT
                 XBslcaSC(K)=S(K)
1132        continue

         CALL cubspline(CHTD,XI1,XI,MT,NMT,XBslcaPT,U,V,Z,W,B,S,T)
             do 1133 K=1,NMT
                 XBslcaPT(K)=S(K)
1133        continue

          ENDIF
        ENDIF
      ENDIF
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

      IF (BFLAG.eq.'Y'.OR. BFLAG.eq.'y') THEN
        IF (MOD(tautot,tau_clean).lt.tau) THEN

        DO 1202 II = 1,NC
          catloading(II)=0
          Do 1203 M = 1, NOSLICE+1
             catloading(II)=catloading(II)+YRC(II,JD,M)
1203      Continue
          catloading(II)=catloading(II)/(NOSLICE+1)
1202    CONTINUE

        DO 1208 JJ = 1,NA
          aniloading(JJ)=0
          Do 1209 M=1,NOSLICE+1
             aniloading(JJ)=aniloading(JJ)+YRA(JJ,JD,M)
1209      Continue
          aniloading(JJ)=aniloading(JJ)/(NOSLICE+1)
1208    CONTINUE

        DO 974 M = 1, NOSLICE+1
          DO 966 II = 1,NC
            YRC(II,JD,M) = catloading(II)
 966      CONTINUE

          DO 970 JJ = 1, NA
            YRA(JJ,JD,M) = aniloading(JJ)
 970      CONTINUE
 974    CONTINUE

        ENDIF
      ENDIF
```

```
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

         IF (TAUTOT.GT.TAUMAX) GOTO 538

        IF (J.EQ.4) THEN
                JD = 1
         ELSE
                JD = J + 1
         ENDIF
*
*     ________________________________________________________________
*     Setting liquid phase fractional concentrations for each species
*     in the first slice of the matrix.
*     ________________________________________________________________

      IF (CF .LE. 0.0) THEN
         WRITE(*,*) "ERROR: CF <= 0.0"
         GOTO 538
      ENDIF

      DO 78 II = 1,NOCAT
         XBCAT(II,1) = CFCAT(II)/CF
 78   CONTINUE

      IF (NOD1C.GE.1) THEN
         DO 80 KK = 1,NOD1C
            XBD1C(KK,1) = CFD1C(KK)/CF
            CBD1CSTA(KK) = CFD1CSTA(KK)
            XBD1CSTA(KK,1) = CFD1CSTA(KK)/CFEED_D1C(KK)
 80      CONTINUE
      END IF

        IF (NOD2C.GE.1) THEN
         DO 81 NN = 1, NOD2C
            XBD2C(NN,1) = CFD2C(NN)/CF
 81      CONTINUE
         CBD2CSTA = CFD2CSTA
         XBD2CSTA(1) = CFD2CSTA/CFEED_D2C
      END IF

        IF (NOD3C.GE.1) THEN
         DO 681 KT = 1, NOD3C
            XBD3C(KT,1) = CFD3C(KT)/CF
681      CONTINUE
         CBD3CSTA = CFD3CSTA
         XBD3CSTA(1) = CFD3CSTA/CFEED_D3C
      END IF

      DO 82 JJ = 1,NOANI
          XBANI(JJ,1) = CFANI(JJ)/CF
 82   CONTINUE

      IF (NOD1A.GE.1) THEN
         DO 83 MM=1, NOD1A
             XBD1A(MM,1) = CFD1A(MM)/CF
             XBD1ASTA(MM,1) = CFD1ASTA(MM)/CFEED_D1A(MM)
             CBD1ASTA(MM) = CFD1ASTA(MM)
 83      CONTINUE
      END IF

        IF (NOD2A.GE.1) THEN
         DO 84 LL = 1, NOD2A
            XBD2A(LL,1) = CFD2A(LL)/CF
 84      CONTINUE
         CBD2ASTA = CFD2ASTA
         XBD2ASTA(1) = CFD2ASTA/CFEED_D2A
      END IF

        IF (NOD3A.GE.1) THEN
         DO 884 KP = 1, NOD3A
            XBD3A(KP,1) = CFD3A(KP)/CF
884      CONTINUE
```

```
         CBD3ASTA = CFD3ASTA
         XBD3ASTA(1) = CFD3ASTA/CFEED_D3A
      END IF

        IF (NOSLCA.GE.1) THEN
         DO 886 KS = 1, NOSLCA
            XBslca(KS,1) = CFslca(KS)/CF
886      CONTINUE
         CBslcaSH = CFslcaSH
         CBslcaSC = CFslcaSC
         CBslcaPT = CFslcaPT

         XBslcaSH(1) = CFslcaSH/CFEED_slca
         XBslcaSC(1) = CFslcaSC/CFEED_slca   ! colloidal silica
         XBslcaPT(1) = CFslcaPT/CFEED_slca   ! particulate silica
      END IF
*     _________________________________________________________________
*     Setting the total dissociative species concentrations in the first
*     slice equal to the feed concentrations.
*     _________________________________________________________________

      IF(NOD1C.GE.1)THEN
         DO 86 KK = 1,NOD1C
            CTO_D1C(KK) = CFEED_D1C(KK)
 86      CONTINUE
      ENDIF

      IF(NOD2C.GE.1)THEN
         CTO_D2C = CFEED_D2C
      ENDIF

      IF(NOD3C.GE.1)THEN
         CTO_D3C = CFEED_D3C
      ENDIF

      IF (NOD1A.GE.1) THEN
         DO 87 MM = 1, NOD1A
            CTO_D1A(MM) = CFEED_D1A(MM)
 87      CONTINUE
      END IF

        IF(NOD2A.GE.1)THEN
         CTO_D2A = CFEED_D2A
      ENDIF

        IF(NOD3A.GE.1)THEN
         CTO_D3A = CFEED_D3A
      ENDIF

        IF(NOSLCA.GE.1)THEN
         CTO_slca = CFEED_slca
      ENDIF
*     _____________________________________________________
*     Combining the arrays of strong and weak electrolytes
*     _____________________________________________________

      DO 88 II = 1, NOCAT
         XBC(PRIORITY_C(II),1) = XBCAT(II,1)
 88   CONTINUE

      IF (NOD1C.GE.1) THEN
         DO 90 KK = 1, NOD1C
            XBC(PRIORITY_C(NOCAT+KK),1) = XBD1C(KK,1)
 90      CONTINUE
      END IF

        IF (NOD2C.GE.1) THEN
         DO 91 NN = 1, NOD2C
            XBC(PRIORITY_C(NOCAT+NOD1C+NN),1) = XBD2C(NN,1)
 91      CONTINUE
      ENDIF
```

```
        IF (NOD3C.GE.1) THEN
         DO 691 KT = 1, NOD3C
         XBC(PRIORITY_C(NOCAT+NOD1C+NOD2C+KT),1) = XBD3C(KT,1)
691      CONTINUE
      ENDIF

      DO 92 JJ = 1, NOANI
         XBA(PRIORITY_A(JJ),1) = XBANI(JJ,1)
 92   CONTINUE

      IF (NOD1A.GE.1) THEN
           DO 95 MM = 1, NOD1A
              XBA(PRIORITY_A(NOANI+MM),1) = XBD1A(MM,1)
 95      CONTINUE
      ENDIF

        IF (NOD2A.GE.1) THEN
         DO 94 LL = 1, NOD2A
            XBA(PRIORITY_A(NOANI+NOD1A+LL),1) = XBD2A(LL,1)
 94      CONTINUE
      ENDIF

        IF (NOD3A.GE.1) THEN
         DO 694 KP = 1, NOD3A
            XBA(PRIORITY_A(NOANI+NOD1A+NOD2A+KP),1) = XBD3A(KP,1)
694      CONTINUE
      ENDIF

        IF (NOSLCA.GE.1) THEN
         DO 696 KS = 1, NOSLCA
         XBA(PRIORITY_A(NOANI+NOD1A+NOD2A+NOD3A+KS),1) = XBslca(KS,1)
696      CONTINUE
      ENDIF
*    ______________________________________________________________
*    Loop to increment distance (along bed depth) at a fixed time
*    ______________________________________________________________

      DO 400 K = 1, NOSLICE

C########################################################
        IF (RFLAG.EQ.'Y'.OR.RFLAG.EQ.'y') THEN
          IF (K.LE.NT1) THEN
            FAR=FAR1
            FCR=FCR1

            if (Sflag.eq.'Y'.or. Sflag.eq.'y') then
               S1 = (7.5E+6*EXP(-10278.6/(TMPC+273.15))*CHTD1
     1         *3.1415927*(BED_DIA**2.)*CAT_CAP)*(VS*ANI_DIA)*FCR1
     1         /(NT1*3600.*4.0*VOL_FLOW*MTC_REF*(1.-VOID_FRAC))

             DD = S1/CF
          elseif (SFLAG.EQ.'N'.OR.SFLAG.EQ.'n') THEN
             DD = 0.0
          endif

          ELSEIF (K.GT.NT1.AND.K.LE.(NT1+NT2)) THEN
            FAR=FAR2
            FCR=FCR2
            if (Sflag.eq.'Y'.or. Sflag.eq.'y') then
               S1 = (7.5E+6*EXP(-10278.6/(TMPC+273.15))*CHTD2
     1         *3.1415927*(BED_DIA**2.)*CAT_CAP)*(VS*ANI_DIA)*FCR2
     1         /(NT2*3600.*4.0*VOL_FLOW*MTC_REF*(1.-VOID_FRAC))

             DD = S1/CF
          elseif (SFLAG.EQ.'N'.OR.SFLAG.EQ.'n') THEN
             DD = 0.0
          endif

          ELSEIF (K.GT.(NT1+NT2).AND.K.LE.(NT1+NT2+NT3)) THEN
            FAR=FAR3
```

```
           FCR=FCR3
           if (Sflag.eq.'Y'.or. Sflag.eq.'y') then
              S1 = (7.5E+6*EXP(-10278.6/(TMPC+273.15))*CHTD3
     1          *3.1415927*(BED_DIA**2.)*CAT_CAP)*(VS*ANI_DIA)*FCR3
     1          /(NT3*3600.*4.0*VOL_FLOW*MTC_REF*(1.-VOID_FRAC))

            DD = S1/CF
          elseif (SFLAG.EQ.'N'.OR.SFLAG.EQ.'n') THEN
             DD = 0.0
          endif

        ENDIF
        ENDIF
C###########################################################

        IF (K .EQ. 1) THEN
           DO 96 II = 1, NC
              CBC(II) = XBC(II,K)*CF
 96        CONTINUE

           DO 98 JJ = 1, NA
              CBA(JJ) = XBA(JJ,K)*CF
 98        CONTINUE

*          __________________________________________________________
*          Call subroutines to calculate interfacial concentrations
*          and fluxes for ionic species
*          __________________________________________________________

*          __________________________________________
*          Calculating the sum of loading fractions
*          __________________________________________
           SUMYC = 0.0
           DO 100 II = 1,NEXC(K)-1
              SUMYC = SUMYC + YRC(II,J,K)
 100       CONTINUE
           YRC(NEXC(K),J,K) = 1.0 - SUMYC

           SUMYA = 0.0
           DO 102 JJ = 1,NEXA(K)-1
              SUMYA = SUMYA + YRA(JJ,J,K)
 102       CONTINUE
           YRA(NEXA(K),J,K) = 1.0 - SUMYA

*          __________________________________________
*          Calculating the number of exchanging ions
*          __________________________________________

           IF(SUMYC.GE.0.999 .AND. NEXC(K).GE.2) THEN
               NEXC(K) = NEXC(K)-1
           END IF

           IF (SUMYA.GE.0.999 .AND. NEXA(K).GE.2) THEN
               NEXA(K) = NEXA(K)-1
           END IF

           IF (NEXC(K).EQ.NC+1) THEN
               CBC(NC+1) = CFH               ! add the conc of H+ to array
           ENDIF

           IF (NEXA(K).EQ.NA+1) THEN
               CBA(NA+1) = CFOH       ! add the conc of OH- to array
           ENDIF
*          __________________________________________________________
*          Copying the current resin and liquid phase fractions to
*          single dimension arrays to pass to the "CATION" subroutine
*          __________________________________________________________
           DO 104 II = 1,NEXC(K)-1
              YCATCUR(II) = YRC(II,J,K)
              XCATCUR(II) = XBC(II,K)
 104       CONTINUE
```

```
           YCATCUR(NEXC(K)) = YRC(NEXC(K),J,K)

           DO 106 JJ = 1,NEXA(K)-1
              YANICUR(JJ) = YRA(JJ,J,K)
              XANICUR(JJ) = XBA(JJ,K)
 106       CONTINUE
           YANICUR(NEXA(K)) = YRA(NEXA(K),J,K)

           IF (NEXC(K).GE.2)THEN
              NTEMPC = NEXC(K)
              NTEMPA = NEXA(K)
              CALL FLUXION(YCATCUR,XCATCUR,NTEMPC,NTEMPA,XBCI,CTCI,
     1          CAT_CAP,ANI_CAP,IFLAG,CBC,CBA,CFC,CFA,FLUXC,DEC,1)
                       ! call subroutine to calculate cation flux
              IF(IFLAG.EQ.1)THEN
              WRITE(*,*)"ERROR: Abnormal exit from 'cation' subroutine"
              WRITE(*,*)'(inside loop)'
                GO TO 538
              ENDIF

              IF (DEC.EQ.0.) THEN
                WRITE(*,*) "ERROR: DEC = 0 "
                GOTO 538
              END IF
*             ________________________________________________________
*             Calculating the effective mass transfer coefficient
*             for cations (Based on the effective diffusivities DEC)
*             ________________________________________________________
              SCH_C = (VISCO/100.0)/DEN/DEC
              MTC_C = FUNC_MTC(RE_CAT,SCH_C,DEC,CAT_DIA)

*        ----- Implementing the effects of cationic resin fouling ----
              if (FFLAG.eq.'Y'.or.FFLAG.eq.'y') then
              MTC_C = (1.-p)*MTC_C
              endif
*        --------------------------------
*             _________________________________________________________
*             Setting the fluxes of non-exchanging cations to zero
*             _________________________________________________________
              IF(NEXC(K).LT.NC)THEN
                 DO 108 II = NEXC(K)+1,NC
                     FLUXC(II) = 0.0
 108             CONTINUE
              ENDIF
           ELSE

              DO 110 II = 1, NC
                  FLUXC(II) = 0.0
 110          CONTINUE
           ENDIF

           IF(NEXA(K).GE.2)THEN
              NTEMPC = NEXC(K)
              NTEMPA = NEXA(K)
              CALL FLUXION(YANICUR,XANICUR,NTEMPC,NTEMPA,XBAI,CTAI,
     1           CAT_CAP,ANI_CAP,IFLAG,CBC,CBA,CFC,CFA,FLUXA,DEA,2)
                        ! call subroutine to calculate anion flux
              IF(IFLAG.EQ.1)THEN
                WRITE(*,*)"ERROR: Abnormal exit from 'anion' subroutine"
                WRITE(*,*)'(inside loop)'
                GO TO 538
              ENDIF

              IF (DEA.EQ.0.) THEN
                  WRITE(*,*) "ERROR: DEA = 0 "
                  GOTO 538
              END IF
*             ________________________________________________________
*             Calculating the effective mass transfer coefficients
*             for anions (Based on the effective diffusivities DEA)
*             ________________________________________________________
```

```
              SCH_A = (VISCO/100.0)/DEN/DEA
              MTC_A = FUNC_MTC(RE_ANI,SCH_A,DEA,ANI_DIA)

C------ Implementing the effects of resin fouling ------

              if (FFLAG.eq.'Y'.or.FFLAG.eq.'y') then
              MTC_A = (1.-q)*MTC_A   !##newly added line
              endif
C----------------------
*             __________________________________________________________
*             Setting the fluxes of non-exchanging anions to zero
*             __________________________________________________________

              IF( NEXA(K).LT.NA)THEN
                  DO 112 JJ = NEXA(K)+1,NA
                     FLUXA(JJ) = 0.0
 112              CONTINUE
              ENDIF

           ELSE
              DO 114 JJ = 1, NA
                  FLUXA(JJ) = 0.0
 114          CONTINUE
           END IF

*
*         ____________________________________________________________
*         Calculating the (mass transfer) rates (bulk--->resin)
*         ____________________________________________________________

*    ------- rate for the ionic species (cations) -------
          DO 116 II = 1,NC
             RATE_COMC(II,1) = FLUXC(II)*MTC_C*CONP*CONY
 116      CONTINUE

*     ------- rate for the ionic form species (anions) -------
          DO 120 JJ = 1,NA
             RATE_COMA(JJ,1) = FLUXA(JJ)*MTC_A*CONS
 120      CONTINUE

*    ------- protonation rate for the type 1 disso cations -------

          IF(NOD1C.GE.1)THEN
            DO 118 KK = 1,NOD1C
              ITEMP = PRIORITY_C(NOCAT+KK)
              IF(NEXC(K).GE.ITEMP.AND.YRC(ITEMP,J,1).LT.0.999)THEN

                IF(NEXA(K).EQ.(NA+1)) THEN
                  CD1CSTAI(KK) = XBCI(ITEMP)*CTCI*XBAI(NA+1)*CTAI
     1                   /DISSD1C(KK)    ! ratio the interfacial conc.
                ELSE
                  CD1CSTAI(KK) = 0.0
                END IF

              IF(CD1CSTAI(KK).GE.CBD1CSTA(KK)) CD1CSTAI(KK)=
     1                                                CBD1CSTA(KK)

                RATE_D1CSTA(KK,1) = MTC_D1CSTA(KK)*(CBD1CSTA(KK)-
     1                CD1CSTAI(KK))*(-CONP)*CONY*CF/CFEED_D1C(KK)
              ELSE
                RATE_D1CSTA(KK,1) = 0.0
              ENDIF
 118        CONTINUE

          ENDIF

*     ------- protonation rate for type 2 disso cations (Div_amines) ------
          IF(NOD2C.GE.1)THEN
             ITEMP = PRIORITY_C(NOCAT+NOD1C+1)
             IF(NEXC(K).GE.ITEMP.AND.YRC(ITEMP,J,1).LT.0.999)THEN
               IF(NEXA(K).EQ.(NA+1)) THEN
```

```
                  CD2CSTAI=XBCI(ITEMP)*CTCI*XBAI(NA+1)*CTAI
     1                   /DISSD2C(1)
               ELSE
                  CD2CSTAI = 0.0
               END IF

               IF(CD2CSTAI.GE.CBD2CSTA) CD2CSTAI=CBD2CSTA
                  RATE_D2CSTA(1) = MTC_D2CSTA*(CBD2CSTA-CD2CSTAI)
     1                   *(-CONP)*CONY*CF/CFEED_D2C
             ELSE
                  RATE_D2CSTA(1) = 0.0
             ENDIF

        ENDIF

*    ------- protonation rate for type 3 disso cations (Tri_amines) -------
          IF(NOD3C.GE.1)THEN
             ITEMP = PRIORITY_C(NOCAT+NOD1C+NOD2C+1)
             IF(NEXC(K).GE.ITEMP.AND.YRC(ITEMP,J,1).LT.0.999)THEN
               IF(NEXA(K).EQ.(NA+1)) THEN
                  CD3CSTAI=XBCI(ITEMP)*CTCI*XBAI(NA+1)*CTAI
     1                   /DISSD3C(1)
               ELSE
                  CD3CSTAI = 0.0
               END IF

               IF(CD3CSTAI.GE.CBD3CSTA) CD3CSTAI=CBD3CSTA
                  RATE_D3CSTA(1) = MTC_D3CSTA*(CBD3CSTA-CD3CSTAI)
     1                   *(-CONP)*CONY*CF/CFEED_D3C
             ELSE
                  RATE_D3CSTA(1) = 0.0
             ENDIF
          ENDIF

*    ------- protonation rate for the type 1 disso anions -------
          IF(NOD1A.GE.1)THEN
            DO 121 MM = 1,NOD1A
                ITEMP = PRIORITY_A(NOANI+MM)
              IF(NEXA(K).GE.ITEMP.AND.YRA(ITEMP,J,1).LT.0.999)THEN

                IF(NEXC(K).EQ.(NC+1)) THEN
                  CD1ASTAI(MM) = XBCI(NC+1)*CTCI*XBAI(ITEMP)*CTAI
     1                   /DISSD1A(MM)
                ELSE
                  CD1ASTAI(MM) = 0.0
                END IF

                IF(CD1ASTAI(MM).GE.CBD1ASTA(MM)) CD1ASTAI(MM)=
     1                                CBD1ASTA(MM)

                RATE_D1ASTA(MM,1) = MTC_D1ASTA(MM)*(CBD1ASTA(MM)-
     1                CD1ASTAI(MM))*(-CONS)*CF/CFEED_D1A(MM)
              ELSE
                 RATE_D1ASTA(MM,1) = 0.0
              ENDIF
 121        CONTINUE
         ENDIF

*    ------ protonation rate for carbonates -----------
         IF(NOD2A.GE.1)THEN
        ITEMP = PRIORITY_A(NOANI+NOD1A+1)
        IF(NEXA(K).GE.ITEMP.AND.YRA(ITEMP,J,1).LT.0.999)THEN
            IF(NEXC(K).EQ.(NC+1)) THEN
                CD2ASTAI=XBCI(NC+1)*CTCI*XBAI(ITEMP)*CTAI
     1                   /DISSD2A(1)
            ELSE
                CD2ASTAI = 0.0
            END IF

            IF(CD2ASTAI.GE.CBD2ASTA)   CD2ASTAI=CBD2ASTA
```

```
                 RATE_D2ASTA(1) = MTC_D2ASTA*(CBD2ASTA-CD2ASTAI)*
     1                     (-CONS)*CF/CFEED_D2A

        ELSE
            RATE_D2ASTA(1) = 0.0
        END IF

          ENDIF

*      ----- protonation rate for type 3 disso anions (Phosphates) -----
          IF(NOD3A.GE.1)THEN
             ITEMP = PRIORITY_A(NOANI+NOD1A+NOD2A+1)
             IF(NEXA(K).GE.ITEMP.AND.YRA(ITEMP,J,1).LT.0.999)THEN
               IF(NEXC(K).EQ.(NC+1)) THEN
                  CD3ASTAI=XBCI(NC+1)*CTCI*XBAI(ITEMP)*CTAI
     1                   /DISSD3A(1)
               ELSE
                  CD3ASTAI = 0.0
               END IF

               IF(CD3ASTAI.GE.CBD3ASTA) CD3ASTAI=CBD3ASTA
                  RATE_D3ASTA(1) = MTC_D3ASTA*(CBD3ASTA-CD3ASTAI)
     1                   *(-CONS)*CF/CFEED_D3A
             ELSE
                  RATE_D3ASTA(1) = 0.0
             ENDIF
          ENDIF

*    -----/// 'protonation' rate for silica acid (molecular) ///-----
          IF(NOSLCA.GE.1)THEN
             ITEMP = PRIORITY_A(NOANI+NOD1A+NOD2A+NOD3A+1)
             IF(NEXA(K).GE.ITEMP.AND.YRA(ITEMP,J,1).LT.0.999)THEN
               IF(NEXC(K).EQ.(NC+1)) THEN
                  CslcaprotSHI=XBCI(NC+1)*CTCI*XBAI(ITEMP)*CTAI
     1                   /DISSslca(1)  ! interfacial conc of SH
               ELSE
                  CslcaprotSHI = 0.0
               END IF

               IF(CslcaprotSHI.GE.CBslcaSH) CslcaprotSHI=CBslcaSH
                  RATE_SHPROT(1) = MTC_slcaSH*(CBslcaSH-CslcaprotSHI)
     1                   *(-CONS)*CF/CFEED_slca      ! protonation rate
             ELSE
                  RATE_SHPROT(1) = 0.0
             ENDIF

          ENDIF
**********************************************************************
*     -------- adsorption rate for dissociative cations ------
**********************************************************************
       if(NOD1C.GE.1) then
          DO 119 KK = 1,NOD1C
             if (SUMsorpd1c(1).LT.0.15) then
            RATE_D1CSORP(KK,1)=RATIO_D1C*RATE_COMC(NOCAT+KK,1)
            else
             RATE_D1CSORP(KK,1)=0
            endif
            RATE_D1CTOT(KK,1)=RATE_D1CSTA(KK,1)+RATE_D1CSORP(KK,1)
 119     CONTINUE
      endif

      if(NOD2C.GE.1) then
          if (SUMsorpd2c(1).LT.0.15) then
             RATE_D2CSORP(1)=RATIO_D2C*RATE_COMC(NOCAT+NOD1C+1,1)
           else
             RATE_D2CSORP(1)=0
           endif
            RATE_D2CTOT(1)=RATE_D2CSTA(1)+RATE_D2CSORP(1)
       endif
```

```
       if(NOD3C.GE.1) then
           if (SUMsorpd3c(1).LT.0.15) then
             RATE_D3CSORP(1)=RATIO_D3C*RATE_COMC(NOCAT+NOD1C+NOD2C+1,1)
           else
             RATE_D3CSORP(1)=0
           endif
             RATE_D3CTOT(1)=RATE_D3CSTA(1)+RATE_D3CSORP(1)
       endif

**********************************************************************
*     -------- adsorption rate for type 1 disso anions ------
**********************************************************************
      IF (NOD1A.GE.1) THEN
            DO 123 MM = 1,NOD1A
             if (SUMsorpd1a(1).LT.0.15) then
              RATE_D1ASORP(MM,1)=RATIO_D1A*RATE_COMA(NOANI+MM,1)
               else
              RATE_D1ASORP(MM,1)=0
               endif  ! RATIO THE ADSORPTION RATE TO IONIC TRANSFER RATE

             RATE_D1ATOT(MM,1)=RATE_D1ASTA(MM,1)+RATE_D1ASORP(MM,1)
 123       CONTINUE
      ENDIF

*       ------ adsorption rate for molecular carbonates ------

       IF (NOD2A.GE.1) THEN
          if (SUMsorpd2a(1).LT.0.15) then
          RATE_D2ASORP(1)=RATIO_D2A*RATE_COMA(NOANI+NOD1A+1,1)
           else
           RATE_D2ASORP(1)=0
           endif
          RATE_D2ATOT(1)=RATE_D2ASTA(1)+RATE_D2ASORP(1)
       ENDIF

*      ----- adsorption rate for type 3 disso anions (Phosphates) ----

      IF (NOD3A.GE.1) THEN
            if ( SUMsorpd3a(1).LT.0.15) then
            RATE_D3ASORP(1)=RATIO_D3A*RATE_COMA(NOANI+NOD1A+NOD2A+1,1)
             else
             RATE_D3ASORP(1)=0
             endif   ! ratio the adsorption rate to ionic transfer rate
            RATE_D3ATOT(1)=RATE_D3ASTA(1)+RATE_D3ASORP(1)
       ENDIF

*      ----- physical adsorption rates for various silica ----

       IF (NOslca.GE.1) THEN
           if (SUMsorpslca(1).LT.0.05) then
            AC= 1.0       ! Frundlich adsorption coefficient & index
            BI= 3.0
            CslcaorpSHI=AC*CBslcaSH**BI
            RATE_SHSORP(1)= MTC_slcaSH*(CBslcaSH-CslcasorpSHI)
     1                 *(-CONS)*CF/CFEED_slca
                           ! adsorption rate for SH
                 a1=1.0        ! Frundlich adsorption coefficient & index
                 e1=0.5
                 a2=5.0
                 e2=0.8
             CslcaSCI=((1.0/a1)*CBslcaSC)**(1.0/e1)
             CslcaPTI=((1.0/a2)*CBslcaPT)**(1.0/e2)

             IF(CslcaSCI.GE.CBslcaSC)    CslcaSCI=CBslcaSC
             IF(CslcaPTI.GE.CBslcaPT)    CslcaPTI=CBslcaPT

                 RATE_slcaSC(1)=MTC_slcaSC*(CBslcaSC-CslcaSCI)
     1                  *(-CONS)*CF/CFEED_slca   ! for colloidal
                 RATE_slcaPT(1)=MTC_slcaPT*(CBslcaPT-CslcaPTI)
     1                  *(-CONS)*CF/CFEED_slca   ! for particulate
            else
```

```
                RATE_SHSORP(1)=0
                RATE_slcaSC(1)=0
                RATE_slcaPT(1)=0
             endif
             RATE_SHTOT(1)=RATE_SHPROT(1)+RATE_SHSORP(1)
                          ! total rate for molecular silica (SH)
      ENDIF
*
*         ___________________________________________________________
*         Calculating the resin loadings for the next time step.
*
          ___________________________________________________________
           DO 122 II = 1,NC
              YRC(II,JD,1) = YRC(II,J,1)+TAU*RATE_COMC(II,1)
              IF (YRC(II,JD,1).LE.0.0)THEN
                  YRC(II,JD,1) = 0.0
              ENDIF
 122       CONTINUE

           DO 124 JJ = 1,NA
              YRA(JJ,JD,1) = YRA(JJ,J,1)+TAU*RATE_COMA(JJ,1)
              IF(YRA(JJ,JD,1).LE.0.0)THEN
                  YRA(JJ,JD,1) = 0.0
              ENDIF
 124       CONTINUE

*
*          ___________________________________________________________
*          Calculating the amount of "molecular form" of disso species
*          transfer to the resin phase by protonation and adsorption
*          ___________________________________________________________

*     -----molecular monovalent amines to the resin phase -----
           IF(NOD1C.GE.1) THEN
              DO 126 KK = 1, NOD1C
         YRC_PROT_D1C(KK,JD,1) = TAU*RATE_D1CSTA(KK,1)*CFEED_D1C(KK)/CF
         YRC_SORP_D1C(KK,JD,1)= TAU*RATE_D1CSORP(KK,1)*CFEED_D1C(KK)/CF
                       ! 6/16/98  added cfeed/cf for sorption term
          SUMsorpd1c(1) = SUMsorpd1c(1) + YRC_SORP_D1C(KK,JD,1)
                                        ! add the protonation term to loadings
                YRC(nocat+kk,JD,1) = YRC(nocat+kk,JD,1)+
     1                                YRC_PROT_D1C(KK,JD,1)
 126          CONTINUE
           ENDIF

*     ---- molecular acetic/formic acids to the resin phase -----

           IF (NOD1A.GE.1) THEN
              DO 127 MM = 1, NOD1A
            YRA_PROT_D1A(MM,JD,1)= TAU*RATE_D1ASTA(MM,1)*CFEED_D1A(MM)/CF
           YRA_SORP_D1A(MM,JD,1)=TAU*RATE_D1ASORP(MM,1)*CFEED_D1A(MM)/CF

            SUMsorpd1a(1)= SUMsorpd1a(1) + YRA_SORP_D1A(MM,JD,1)
                       ! add the protonation term to loadings
               YRA(noani+mm,JD,1) = YRA(noani+mm,JD,1)+
     1                                YRA_PROT_D1A(mm,JD,1)
 127          CONTINUE
           ENDIF

*     ----- divalent amines into the resin phase ------
           IF(NOD2C.GE.1)THEN
              YRC_PROT_D2C(1,JD,1)=TAU*RATE_D2CSTA(1)*CFEED_D2C/CF
              YRC_SORP_D2C(1,JD,1)=TAU*RATE_D2CSORP(1)*CFEED_D2C/CF
     1
               SUMsorpd2c(1) = SUMsorpd2c(1) + YRC_SORP_D2C(1,JD,1)
                              !   add the protonation term to loadings
              YRC(nocat+nod1c+1,JD,1) = YRC(nocat+nod1c+1,JD,1)+
     1                         YRC_PROT_D2C(1,JD,1)
           ENDIF

*     ----- carbonates into the resin phase ------
           IF(NOD2A.GE.1)THEN
         YRA_PROT_D2A(1,JD,1)=TAU*RATE_D2ASTA(1)*CFEED_D2A/CF
         YRA_SORP_D2A(1,JD,1)= TAU*RATE_D2ASORP(1)*CFEED_D2A/CF
```

```
     1
                SUMsorpd2a(1) = SUMsorpd2a(1) + YRA_SORP_D2A(1,JD,1)
                               !   add the protonation term to loadings
               YRA(noani+nod1a+1,JD,1) = YRA(noani+nod1a+1,JD,1)+
     1                         YRA_PROT_D2A(1,JD,1)

            ENDIF

*      ----- Trivalent amines into the resin phase ------
            IF(NOD3C.GE.1)THEN
            YRC_PROT_D3C(1,JD,1)=TAU*RATE_D3CSTA(1)*CFEED_D3C/CF
            YRC_SORP_D3C(1,JD,1)=TAU*RATE_D3CSORP(1)*CFEED_D3C/CF

                 SUMsorpd3c(1) = SUMsorpd3c(1)+YRC_SORP_D3C(1,JD,1)
                             !  add the protonation term to loadings
             YRC(nocat+nod1c+nod2c+1,JD,1) = YRC(nocat+nod1c+nod2c+1,JD,1)+
     1                         YRC_PROT_D3C(1,JD,1)
            ENDIF

*      ----- Phosphates into the resin phase ------
            IF(NOD3A.GE.1)THEN
               YRA_PROT_D3A(1,JD,1)=TAU*RATE_D3ASTA(1)*CFEED_D3A/CF
               YRA_SORP_D3A(1,JD,1)= TAU*RATE_D3ASORP(1)*CFEED_D3A/CF

                SUMsorpd3a(1) = SUMsorpd3a(1)+YRA_SORP_D3A(1,JD,1)
                         !   add the protonation term to loadings
             YRA(noani+nod1a+nod2a+1,JD,1) = YRA(noani+nod1a+nod2a+1,JD,1)+
     1                         YRA_PROT_D3A(1,JD,1)
            ENDIF

*      ----- Silica into the resin phase ------
            IF(NOslca.GE.1)THEN
            YRA_PROT_slca(1,JD,1) = TAU*RATE_SHPROT(1)*CFEED_slca/CF
            YRA(NA-KS+1,JD,1) = YRA(NA-KS+1,JD,1)+YRA_PROT_slca(1,JD,1)

            YRA_SORP_slca(1,JD,1) = CFEED_slca/CF*
     1          TAU*(RATE_SHSORP(1)+Rate_slcaSc(1)+Rate_slcaPT(1))

              SUMsorpslca(1)= SUMsorpslca(1)+YRA_SORP_slca(1,JD,1)

            ENDIF
        END IF

*
*     ______________________________________________________________
*     Implement implicit portion of the "gears backward method" to
*     calculate bulk phase fraction for the next distance step based
*     on the previous values, for the first three steps Euler's
*     first-order method is used.
*     ______________________________________________________________

      ISTEPS = 3*XI/XI1 ! NO of steps to be taken with a small step size

      NCNA = 21
      ND1 = 5

        IF(K.LE.3)THEN

           CALL EULER(XBC,XI1,FCR,RATE_COMC,K,NC,NCNA)
          CALL EULERANI(XBA,XI1,FAR,RATE_COMA,K,NA,NCNA,EWA,DD)
*         ___________________________________________________________
*         Calculate fraction of undissociated species (molecular form)
*         ___________________________________________________________
          IF(NOD1C.GE.1)THEN
              CALL EULER(XBD1CSTA,XI1,FCR,RATE_D1CTOT,K,
     1                 NOD1C,ND1)
          ENDIF

          IF(NOD1A.GE.1) THEN
              CALL EULER(XBD1ASTA,XI1,FAR,RATE_D1ATOT,K,
     1                 NOD1A,ND1)
          ENDIF
```

```
          IF(NOD2C.GE.1)THEN
            XBD2CSTA(K+1)=XBD2CSTA(K)-XI1*FCR*RATE_D2CTOT(K)
         ENDIF

          IF(NOD2A.GE.1)THEN
            XBD2ASTA(K+1)=XBD2ASTA(K)-XI1*FAR*RATE_D2ATOT(K)
         ENDIF

          IF(NOD3C.GE.1)THEN
            XBD3CSTA(K+1)=XBD3CSTA(K)-XI1*FCR*RATE_D3CTOT(K)
         ENDIF

          IF(NOD3A.GE.1)THEN
            XBD3ASTA(K+1)=XBD3ASTA(K)-XI1*FAR*RATE_D3ATOT(K)
         ENDIF

          IF(NOSLCA.GE.1)THEN
            XBslcaSH(K+1)=XBslcaSH(K)-XI1*FAR*RATE_SHTOT(K)
            XBslcaSC(K+1)=XBslcaSC(K)-XI1*FAR*RATE_slcaSC(K)
            XBslcaPT(K+1)=XBslcaPT(K)-XI1*FAR*RATE_slcaPT(K)
         ENDIF

       ELSEIF (K.GT.3.AND.K.LE.ISTEPS)THEN
         CALL GEAR(XBC,COEC,FCR,RATE_COMC,XI1,NC,K,NCNA)
         CALL GEARANI(XBA,COEA,FAR,RATE_COMA,XI1,NA,EWA,DD,K)

         IF(NOD1C.GE.1)THEN
          CALL GEAR(XBD1CSTA,COED1CSTA,FCR,RATE_D1CTOT,
     1                 XI1,NOD1C,K,ND1)
         ENDIF

          IF(NOD1A.GE.1) THEN
             CALL GEAR(XBD1ASTA,COED1ASTA,FAR,RATE_D1ATOT,
     1           XI1,NOD1A,K,ND1)
         ENDIF

          IF(NOD2C.GE.1)THEN
             COED2CSTA = 3.*XBD2CSTA(K-3)-16.*XBD2CSTA(K-2)+
     1                    36.*XBD2CSTA(K-1)-48.*XBD2CSTA(K)
             XBD2CSTA(K+1) =  -XI1*12.*FCR*RATE_D2CTOT(K)/25.
     1                   -COED2CSTA/25.
         ENDIF

         IF(NOD2A.GE.1)THEN
             COED2ASTA = 3.*XBD2ASTA(K-3)-16.*XBD2ASTA(K-2)+
     1                    36.*XBD2ASTA(K-1)-48.*XBD2ASTA(K)
             XBD2ASTA(K+1) =  -XI1*12.*FAR*RATE_D2ATOT(K)/25.
     1                   -COED2ASTA/25.
         ENDIF

          IF(NOD3C.GE.1)THEN
             COED3CSTA = 3.*XBD3CSTA(K-3)-16.*XBD3CSTA(K-2)+
     1                    36.*XBD3CSTA(K-1)-48.*XBD3CSTA(K)
             XBD3CSTA(K+1) =  -XI1*12.*FCR*RATE_D3CTOT(K)/25.
     1                   -COED3CSTA/25.
         ENDIF

         IF(NOD3A.GE.1)THEN
             COED3ASTA = 3.*XBD3ASTA(K-3)-16.*XBD3ASTA(K-2)+
     1                    36.*XBD3ASTA(K-1)-48.*XBD3ASTA(K)
             XBD3ASTA(K+1) =  -XI1*12.*FAR*RATE_D3ATOT(K)/25.
     1                   -COED3ASTA/25.
         ENDIF

          IF(NOSLCA.GE.1)THEN
             COEslcaSH = 3.*XBslcaSH(K-3)-16.*XBslcaSH(K-2)+
     1                    36.*XBslcaSH(K-1)-48.*XBslcaSH(K)
             XBslcaSH(K+1) =  -XI1*12.*FAR*RATE_SHTOT(K)/25.
     1                  -COEslcaSH/25. ! for molecular silica
```

```
             COEslcaSC = 3.*XBslcaSC(K-3)-16.*XBslcaSC(K-2)+
     1                    36.*XBslcaSC(K-1)-48.*XBslcaSC(K)
             XBslcaSC(K+1) =  -XI1*12.*FAR*RATE_slcaSC(K)/25.
     1                   -COEslcaSC/25. ! for colloidal silica

              COEslcaPT = 3.*XBslcaPT(K-3)-16.*XBslcaPT(K-2)+
     1                    36.*XBslcaPT(K-1)-48.*XBslcaPT(K)
             XBslcaPT(K+1) =  -XI1*12.*FAR*RATE_slcaPT(K)/25.
     1                   -COEslcaPT/25. ! for particulate silica
         ENDIF

      ELSE IF(K.GE.(ISTEPS+1).AND.K.LE.(ISTEPS+3))THEN

           IF(K.EQ.ISTEPS+1) THEN
                K1 = 1
                K2 = ISTEPS/3+1
                K3 = ISTEPS*2/3+1
                K4 = K
           ELSE IF(K.EQ.ISTEPS+2)THEN
                K1 = ISTEPS/3+1
                K2 = ISTEPS*2/3+1
                K3 = ISTEPS+1
                K4 = K
           ELSE IF(K.EQ.ISTEPS+3)THEN
                K1 = ISTEPS*2/3+1
                K2 = ISTEPS+1
                K3 = K-1
                K4 = K
           ENDIF

      DO 410 II=1,NC
         COEC(II)=3.*XBC(II,K1)-16.*XBC(II,K2)+36.*XBC(II,K3)
     1           -48.*XBC(II,K4)
         XBC(II,K+1)=-XI*12.*FCR*RATE_COMC(II,K)/25.-COEC(II)/25.
 410  CONTINUE

      DO 712 JJ=1,NA
         COEA(JJ)=3.*XBA(JJ,K1)-16.*XBA(JJ,K2)+36.*XBA(JJ,K3)
     1           -48.*XBA(JJ,K4)
         XBA(JJ,K+1)=-XI*12.*FAR*RATE_COMA(JJ,K)/25.-COEA(JJ)/25.

*
*             Adding the desulphonation term to the sulfate. Sulfate is
*             identified by its equivalent weight.
*
              IF (ABS(EWA(JJ)-48.0) .LE. 0.1)  THEN
                  XBA(JJ,K+1) = XBA(JJ,K+1) + DD
              END IF
 712  CONTINUE

*  ------- AMINES ------------*
      IF(NOD1C.GE.1)THEN
      DO 714 KK=1,NOD1C
         COED1CSTA(KK)=3.*XBD1CSTA(KK,K1)-16.*XBD1CSTA(KK,K2)+36.
     1           *XBD1CSTA(KK,K3)-48.*XBD1CSTA(KK,K4)
         XBD1CSTA(KK,K+1)=-XI*12.*FCR*RATE_D1CTOT(KK,K)/25.
     1   -COED1CSTA(KK)/25.
 714  CONTINUE
      ENDIF

*  -------- ORGANIC ACIDS ----*

       IF(NOD1A.GE.1) THEN
       DO 716 MM=1,NOD1A
          COED1ASTA(MM)=3.*XBD1ASTA(MM,K1)-16.*XBD1ASTA(MM,K2)+36.
     1           *XBD1ASTA(MM,K3)-48.*XBD1ASTA(MM,K4)
          XBD1ASTA(MM,K+1)=-XI*12.*FAR*RATE_D1ATOT(MM,K)/25.
     1           -COED1ASTA(MM)/25.
 716   CONTINUE
       ENDIF
* ------- Diamine ---------*
```

```
            IF(NOD2C.GE.1)THEN
             COED2CSTA = 3.*XBD2CSTA(K1)-16.*XBD2CSTA(K2)+
     1                    36.*XBD2CSTA(K3)-48.*XBD2CSTA(K4)
             XBD2CSTA(K+1) =  -XI*12.*FCR*RATE_D2CTOT(K)/25.
     1               -COED2CSTA/25.

          ENDIF

          IF(NOD2A.GE.1)THEN
             COED2ASTA = 3.*XBD2ASTA(K1)-16.*XBD2ASTA(K2)+
     1                    36.*XBD2ASTA(K3)-48.*XBD2ASTA(K4)
             XBD2ASTA(K+1) =  -XI*12.*FAR*RATE_D2ATOT(K)/25.
     1               -COED2ASTA/25.
          ENDIF

            IF(NOD3C.GE.1)THEN
             COED3CSTA = 3.*XBD3CSTA(K1)-16.*XBD3CSTA(K2)+
     1                    36.*XBD3CSTA(K3)-48.*XBD3CSTA(K4)
             XBD3CSTA(K+1) =  -XI*12.*FCR*RATE_D3CTOT(K)/25.
     1               -COED3CSTA/25.

          ENDIF

          IF(NOD3A.GE.1)THEN
             COED3ASTA = 3.*XBD3ASTA(K1)-16.*XBD3ASTA(K2)+
     1                    36.*XBD3ASTA(K3)-48.*XBD3ASTA(K4)
             XBD3ASTA(K+1) =  -XI*12.*FAR*RATE_D3ATOT(K)/25.
     1               -COED3ASTA/25.

          ENDIF
* -------- molecular silica -------*
               IF(NOSLCA.GE.1)THEN
             COEslcaSH = 3.*XBslcaSH(K1)-16.*XBslcaSH(K2)+
     1                    36.*XBslcaSH(K3)-48.*XBslcaSH(K4)
             XBslcaSH(K+1) =  -XI*12.*FAR*RATE_SHTOT(K)/25.
     1           -COEslcaSH/25.

              COEslcaSC = 3.*XBslcaSC(K1)-16.*XBslcaSC(K2)+
     1                    36.*XBslcaSC(K3)-48.*XBslcaSC(K4)
             XBslcaSC(K+1) =  -XI*12.*FAR*RATE_slcaSC(K)/25.
     1               -COEslcaSC/25.

                  COEslcaPT = 3.*XBslcaPT(K1)-16.*XBslcaPT(K2)+
     1                    36.*XBslcaPT(K3)-48.*XBslcaPT(K4)
             XBslcaPT(K+1) =  -XI*12.*FAR*RATE_slcaPT(K)/25.
     1               -COEslcaPT/25.

          ENDIF

      ELSE IF(K.GT.ISTEPS+3)THEN
           CALL GEAR(XBC,COEC,FCR,RATE_COMC,XI,NC,K,NCNA) ! cations
         CALL GEARANI(XBA,COEA,FAR,RATE_COMA,XI,NA,EWA,DD,K) ! anions

        IF (NOD1C.GE.1) THEN
          CALL GEAR(XBD1CSTA,COED1CSTA,FCR,RATE_D1CTOT,
     1            XI,NOD1C,K,ND1)
        ENDIF

         IF(NOD1A.GE.1) THEN
           CALL GEAR(XBD1ASTA,COED1ASTA,FAR,RATE_D1ATOT,
     1         XI,NOD1A,K,ND1)
        ENDIF

            IF(NOD2C.GE.1)THEN
             COED2CSTA = 3.*XBD2CSTA(K-3)-16.*XBD2CSTA(K-2)+
     1                    36.*XBD2CSTA(K-1)-48.*XBD2CSTA(K)
             XBD2CSTA(K+1) =  -XI1*12.*FCR*RATE_D2CTOT(K)/25.
     1              -COED2CSTA/25.
          ENDIF

          IF(NOD2A.GE.1)THEN
```

```
             COED2ASTA = 3.*XBD2ASTA(K-3)-16.*XBD2ASTA(K-2)+
     1                    36.*XBD2ASTA(K-1)-48.*XBD2ASTA(K)
             XBD2ASTA(K+1) =  -XI1*12.*FAR*RATE_D2ATOT(K)/25.
     1              -COED2ASTA/25.
          ENDIF

          IF(NOD3C.GE.1)THEN
             COED3CSTA = 3.*XBD3CSTA(K-3)-16.*XBD3CSTA(K-2)+
     1                    36.*XBD3CSTA(K-1)-48.*XBD3CSTA(K)
             XBD3CSTA(K+1) =  -XI1*12.*FCR*RATE_D3CTOT(K)/25.
     1              -COED3CSTA/25.
          ENDIF

          IF(NOD3A.GE.1)THEN
             COED3ASTA = 3.*XBD3ASTA(K-3)-16.*XBD3ASTA(K-2)+
     1                    36.*XBD3ASTA(K-1)-48.*XBD3ASTA(K)
             XBD3ASTA(K+1) =  -XI1*12.*FAR*RATE_D3ATOT(K)/25.
     1              -COED3ASTA/25.
          ENDIF

          IF(NOSLCA.GE.1)THEN                               ! silica
             COEslcaSH = 3.*XBslcaSH(K-3)-16.*XBslcaSH(K-2)+
     1                    36.*XBslcaSH(K-1)-48.*XBslcaSH(K)
             XBslcaSH(K+1) =  -XI1*12.*FAR*RATE_SHTOT(K)/25.
     1           -COEslcaSH/25.

                  COEslcaSC = 3.*XBslcaSC(K-3)-16.*XBslcaSC(K-2)+
     1                    36.*XBslcaSC(K-1)-48.*XBslcaSC(K)
             XBslcaSC(K+1) =  -XI1*12.*FAR*RATE_slcaSC(K)/25.
     1              -COEslcaSC/25.

             COEslcaPT = 3.*XBslcaPT(K-3)-16.*XBslcaPT(K-2)+
     1                    36.*XBslcaPT(K-1)-48.*XBslcaPT(K)
             XBslcaPT(K+1) =  -XI1*12.*FAR*RATE_slcaPT(K)/25.
     1              -COEslcaPT/25.
          ENDIF
      END IF
*    _________________________________________________________________
*    Determine concentrations for this distance step and recalculate
*    bulk phase equilibria.
*    _________________________________________________________________

      DO 170 II = 1, NC
         CBC(II)=XBC(II,K+1)*CF
 170  CONTINUE

      DO 172 JJ = 1, NA
         CBA(JJ)=XBA(JJ,K+1)*CF
 172  CONTINUE

*---- Splitting of the arrays to calculate specific species -----

      DO 174 II = 1, NOCAT
         CBCAT(II) = CBC(PRIORITY_C(II))
 174  CONTINUE

      DO 176 JJ = 1, NOANI
         CBANI(JJ) = CBA(PRIORITY_A(JJ))
 176  CONTINUE

      IF (NOD1C.GE.1) THEN
         DO 178 KK = 1, NOD1C
            CBD1C(KK) = CBC(PRIORITY_C(NOCAT+KK))
 178     CONTINUE
      END IF

      IF (NOD1A.GE.1) THEN
         DO 180 MM=1, NOD1A
             CBD1A(MM) = CBA(PRIORITY_A(NOANI+MM))
 180     CONTINUE
      ENDIF
```

```
        IF (NOD2C.GE.1) THEN
          DO 179 NN = 1, NOD2C
              CBD2C(NN) = CBC(PRIORITY_C(NOANI+NOD1C+NN))
 179      CONTINUE
      END IF

        IF (NOD2A.GE.1) THEN
          DO 181 LL = 1, NOD2A
              CBD2A(LL) = CBA(PRIORITY_A(NOANI+NOD1A+LL))
 181      CONTINUE
      END IF

        IF (NOD3C.GE.1) THEN
          DO 779 KT = 1, NOD3C
              CBD3C(KT) = CBC(PRIORITY_C(NOANI+NOD1C+NOD2C+KT))
 779      CONTINUE
      END IF

        IF (NOD3A.GE.1) THEN
          DO 781 KP = 1, NOD3A
              CBD3A(KP) = CBA(PRIORITY_A(NOANI+NOD1A+NOD2A+KP))
 781      CONTINUE
      END IF

      IF (NOSLCA.GE.1) THEN
          DO 782 KS = 1, NOSLCA
              CBslca(KS) = CBA(PRIORITY_A(NOANI+NOD1A+NOD2A+NOD3A+KS))
 782      CONTINUE
      END IF

*-------------------------------------------

*     _________________________________________________________
*     Calculate the concentrations of the molecular amines
*     in the next slice
*     _________________________________________________________

      IF(NOD1C.GE.1)THEN
          DO 182 KK = 1,NOD1C
            IF(XBD1CSTA(KK,K+1).LE.0.0)THEN
                XBD1CSTA(KK,K+1) = 0.0
            ENDIF
            CBD1CSTA(KK) = XBD1CSTA(KK,K+1)*CFEED_D1C(KK)
 182      CONTINUE
      ENDIF
*     ___________________________________________________________
*     Calculate the concentration of molecular acetic /formic acid
*     in the next slice
*     ___________________________________________________________

        IF (NOD1A.GE.1) THEN
           DO 183 MM = 1, NOD1A
              IF (XBD1ASTA(MM,K+1) .LE.0.0) THEN
                  XBD1ASTA(MM,K+1) = 0.0
              ENDIF
                  CBD1ASTA(MM)=XBD1ASTA(MM,K+1)*CFEED_D1A(MM)

183       CONTINUE
        ENDIF
*     ____________________________________________________
*     Calculate the molecular divalent amines concentration
*     ____________________________________________________

      IF(NOD2C.GE.1)THEN
          IF(XBD2CSTA(K+1).LE.0.0)THEN
             XBD2CSTA(K+1) = 0.0
          ENDIF
             CBD2CSTA = XBD2CSTA(K+1)*CFEED_D2C
      ENDIF
*     ____________________________________________________
```

```
*     Calculate the molecular carbonate concentration
*     ____________________________________________________

      IF(NOD2A.GE.1)THEN
          IF(XBD2ASTA(K+1).LE.0.0)THEN
             XBD2ASTA(K+1) = 0.0
          ENDIF
             CBD2ASTA = XBD2ASTA(K+1)*CFEED_D2A
      ENDIF
*     ____________________________________________________
*     Calculate the molecular Trivalent amines concentration
*     ____________________________________________________

      IF(NOD3C.GE.1)THEN
          IF(XBD3CSTA(K+1).LE.0.0)THEN
             XBD3CSTA(K+1) = 0.0
          ENDIF
             CBD3CSTA = XBD3CSTA(K+1)*CFEED_D3C
      ENDIF
*     ____________________________________________________
*     Calculate the molecular Phosphate concentration
*     ____________________________________________________

      IF(NOD3A.GE.1)THEN
          IF(XBD3ASTA(K+1).LE.0.0)THEN
             XBD3ASTA(K+1) = 0.0
          ENDIF
             CBD3ASTA = XBD3ASTA(K+1)*CFEED_D3A
      ENDIF
*     _____________________________________________________________
*     Calculate the molecular & other kinds of silica concentrations
*     for the next slice
*     _____________________________________________________________

        IF(NOSLCA.GE.1)THEN
          IF( XBslcaSH(K+1).LE.0.0)THEN
              XBslcaSH(K+1) = 0.0
          ENDIF

          IF( XBslcaSC(K+1).LE.0.0)THEN
              XBslcaSC(K+1) = 0.0
          ENDIF

          IF( XBslcaPT(K+1).LE.0.0)THEN
              XBslcaPT(K+1) = 0.0
          ENDIF

            CBslcaSH = XBslcaSH(K+1)*CFEED_slca
             CBslcaSC = XBslcaSC(K+1)*CFEED_slca
             CBslcaPT = XBslcaPT(K+1)*CFEED_slca
      ENDIF
*     ___________________________________________________
*     Calculating the new total mono_amine concentration
*     ___________________________________________________
      IF(NOD1C.GE.1)THEN
         DO 184 KK = 1,NOD1C
           CTO_D1C(KK) = CBD1C(KK)+CBD1CSTA(KK)
 184     CONTINUE
      ENDIF
*     ________________________________________________________
*     Calculate the new total carboxylic acid concentration
*     ________________________________________________________
      IF (NOD1A.GE.1) THEN
         DO 185 MM = 1, NOD1A
             CTO_D1A(MM)= CBD1A(MM)+CBD1ASTA(MM)
 185     CONTINUE
      ENDIF
*     ___________________________________________________
*     Calculating the new total Di_amine concentration
*     ___________________________________________________
      IF(NOD2C.GE.1)THEN
```

```
         CTO_D2C = CBD2C(1)+CBD2C(2)+CBD2CSTA
      ENDIF
*     ____________________________________________
*     Calculating the new total carbonate concentration
*     ____________________________________________
      IF(NOD2A.GE.1)THEN
         CTO_D2A = CBD2A(1)+CBD2A(2)+CBD2ASTA
      ENDIF
*     ____________________________________________
*     Calculating the new total Tri_amine concentration
*     ____________________________________________
      IF(NOD3C.GE.1)THEN
         CTO_D3C = CBD3C(1)+CBD3C(2)+CBD3C(3)+CBD3CSTA
      ENDIF
*     ____________________________________________
*     Calculating the new total Phosphate concentration
*     ____________________________________________
      IF(NOD3A.GE.1)THEN
         CTO_D3A = CBD3A(1)+CBD3A(2)+CBD3A(3)+CBD3ASTA
      ENDIF
*     ________________________________________________
*     Calculating the new soluble & total silica concentrations
*     ________________________________________________
      IF(NOSLCA.GE.1)THEN
         Csolu=CBslca(1)+CBslca(2)+CBslca(3)+CBslcaSH
         CBslcaSC=RATIO_SC*Csolu
          if (CFEED_slca.GT.Csoluslca) then
         CTO_slca = Csolu +CBslcaPT + CBslcaSC
          else
         CTO_slca = Csolu
          endif
      ENDIF

* ------------------------------------------

      IF(K.EQ.1)THEN              ! check if it is the first slice
          PH_OLD = PH_FEED

      DO 1518 II=1,NOCAT
         CBCAT(II) = CFCAT(II)
1518   CONTINUE

      DO 1520 JJ=1,NOANI
         CBANI(JJ) = CFANI(JJ)
1520   CONTINUE

          IF(NOD1C.GE.1)THEN
         DO 1522 KK = 1,NOD1C
              CTO_D1C(KK) = CFEED_D1C(KK)
1522        CONTINUE
          ENDIF

          IF(NOD2C.GE.1)THEN
         CTO_D2C = CFEED_D2C
          ENDIF

          IF(NOD3C.GE.1)THEN
         CTO_D3C = CFEED_D3C
          ENDIF

        IF(NOD1A.GE.1) THEN
         DO 1523 MM = 1, NOD1A
            CTO_D1A(MM) = CFEED_D1A(MM)
1523        CONTINUE
        ENDIF

          IF(NOD2A.GE.1)THEN
            CTO_D2A = CFEED_D2A
          ENDIF

          IF(NOD3A.GE.1)THEN
```

```
           CTO_D3A = CFEED_D3A
         ENDIF

         IF(NOSLCA.GE.1)THEN
           CTO_slca = CFEED_slca
         ENDIF

      ELSE
          PH_OLD = PH_NEW
      ENDIF

       CALL DISSOEQ(CBCAT,CBANI,CBD1C,CBD1CSTA,CTO_D1C,CBD1A,CBD1ASTA,
     1        CTO_D1A,CBD2C,CBD2CSTA,CTO_D2C,CBD2A,CBD2ASTA,CTO_D2A,
     1        CBD3C,CBD3CSTA,CTO_D3C,CBD3A,CBD3ASTA,CTO_D3A,
     1        Csolu, CBslcaSH, CBslca,
     1        CBH,CBOH,PH_OLD,PH_NEW,IFLAG)

      IF(IFLAG.EQ.1)THEN
          WRITE(*,*)"ERROR: Abnormal exit from 'dissoeq' subroutine"
               WRITE(*,*)"(inside loop)"
               GO TO 538
      ENDIF
*    ___________________________________________________________
*    Determine rates at constant XI for solutions of the TAU
*    material balance
*    ___________________________________________________________

*    ________________________________________________________
*     Calculating the sum of initial loadings
*    ________________________________________________________
      SUMYC = 0.0
      DO 186 II = 1,NEXC(K+1)-1
         SUMYC = SUMYC + YRC(II,J,K+1)
 186  CONTINUE
      YRC(NEXC(K+1),J,K+1) = 1.0 - SUMYC

      SUMYA = 0.0
      DO 188 JJ = 1,NEXA(K+1)-1
         SUMYA = SUMYA + YRA(JJ,J,K+1)
 188  CONTINUE
      YRA(NEXA(K+1),J,K+1) = 1.0 - SUMYA
*     __________________________________________________________
*     Calculating the number of exchanging ions by determining
*     the satuaration.
*     __________________________________________________________

      IF(SUMYC.GE.0.999 .AND. NEXC(K+1).GE.2) THEN
          NEXC(K+1) = NEXC(K+1)-1
      END IF

      IF (SUMYA.GE.0.999 .AND. NEXA(K+1).GE.2) THEN
          NEXA(K+1) = NEXA(K+1)-1
      END IF

      IF(NEXC(K+1).EQ.NC+1)THEN
          CBC(NC+1) = CBH
      ENDIF

      IF(NEXA(K+1).EQ.NA+1)THEN
          CBA(NA+1) = CBOH
      ENDIF
*     ___________________________________________________________
*     Copying the current resin and liquid phase fractions to
*     single dimension arrays to pass to the "CATION" subroutine
*     ___________________________________________________________

      DO 190 II = 1, NEXC(K+1)-1
         YCATCUR(II) = YRC(II,J,K+1)
         XCATCUR(II) = XBC(II,K+1)
 190  CONTINUE
      YCATCUR(NEXC(K+1)) = YRC(NEXC(K+1),J,K+1)
```

```
      DO 192 JJ = 1, NEXA(K+1)-1
         YANICUR(JJ) = YRA(JJ,J,K+1)
         XANICUR(JJ) = XBA(JJ,K+1)
 192  CONTINUE
      YANICUR(NEXA(K+1)) = YRA(NEXA(K+1),J,K+1)

      IF(NEXC(K+1).GE.2)THEN
          NTEMPC = NEXC(K+1)
          NTEMPA = NEXA(K+1)
          CALL FLUXION(YCATCUR,XCATCUR,NTEMPC,NTEMPA,XBCI,CTCI,
     1        CAT_CAP,ANI_CAP,IFLAG,CBC,CBA,CFC,CFA,FLUXC,DEC,1)
          IF(IFLAG.EQ.1)THEN
            WRITE(*,*)"ERROR: Abnormal exit from 'cation' subroutine"
            GO TO 538
          ENDIF

*         _______________________________________________________
*         Recalculate the mass transfer coefficients based on the
*         effective diffusivities.
*         _______________________________________________________

          IF (DEC.EQ.0.) THEN
            WRITE(*,*) "ERROR: DEC = 0 "
            GOTO 538
          END IF

          SCH_C = (VISCO/100)/DEN/DEC
          MTC_C = FUNC_MTC(RE_CAT,SCH_C,DEC,CAT_DIA)

C ----- Implementing the effects of cationic resin fouling ----
          if (FFLAG.eq.'Y'.or.FFLAG.eq.'y') then
          MTC_C = (1.-p)*MTC_C
          endif
C---------------------------
*         __________________________________________________________
*         Setting the fluxes of non-exchanging ions to zero
*         __________________________________________________________

          IF(NEXC(K+1).LT.NC)THEN
             DO 194 II = NEXC(K+1)+1,NC
               FLUXC(II) = 0.0
 194         CONTINUE
          ENDIF

      ELSE
          DO 196 II = 1,NC
             FLUXC(II) = 0.0
 196      CONTINUE
      ENDIF

      IF(NEXA(K+1).GE.2)THEN
          NTEMPC = NEXC(K+1)
          NTEMPA = NEXA(K+1)
          CALL FLUXION(YANICUR,XANICUR,NTEMPC,NTEMPA,XBAI,CTAI,
     1        CAT_CAP,ANI_CAP,IFLAG,CBC,CBA,CFC,CFA,FLUXA,DEA,2)
          IF(IFLAG.EQ.1)THEN
            WRITE(*,*)"ERROR: Abnormal exit from 'anion' subroutine"
            GO TO 538
          ENDIF
*         _______________________________________________________
*         Recalculate the mass transfer coefficients based on the
*         effective diffusivities.
*         _______________________________________________________

          IF (DEA.EQ.0.) THEN
             WRITE(*,*) "ERROR: DEA = 0 "
             GOTO 538
          END IF

          SCH_A = (VISCO/100)/DEN/DEA
```

```
          MTC_A = FUNC_MTC(RE_ANI,SCH_A,DEA,ANI_DIA)

C ----- implementing the effects of resin fouling on MTC -----
          if (FFLAG.eq.'Y'.or.FFLAG.eq.'y') then
              MTC_A = (1.-q)*MTC_A    !##newly added line
          endif
C ------------------
*         __________________________________________________________
*         Setting the fluxes of non-exchanging ions to zero
*         __________________________________________________________

          IF(NEXA(K+1).LT.NA)THEN
             DO 198 JJ = NEXA(K+1)+1,NA
               FLUXA(JJ) = 0.0
 198         CONTINUE
          ENDIF
      ELSE
          DO 200 JJ = 1, NA
             FLUXA(JJ) = 0.0
 200      CONTINUE

      ENDIF
*     _________________________________________________________
*     Calculating the rates and determining the ionic loadings
*     for the next distance step (k+1)
*     _________________________________________________________

*     ------ rate and loadings for ionic form (cations) -----
      DO 202 II = 1,NC
          RATE_COMC(II,K+1) = FLUXC(II)*MTC_C*CONP*CONY
          YRC(II,JD,K+1) = YRC(II,J,K+1)+TAU*RATE_COMC(II,K+1)
          IF (YRC(II,JD,K+1).LE.0.0)THEN
              YRC(II,JD,K+1) = 0.0
          ENDIF       ! maybe needs to add the protonaton loadings
 202  CONTINUE

*    ------ rate and loadings for ionic form (anions) ------
      DO 204 JJ = 1,NA
         RATE_COMA(JJ,K+1) = FLUXA(JJ)*MTC_A*CONS
         YRA(JJ,JD,K+1) = YRA(JJ,J,K+1)+TAU*RATE_COMA(JJ,K+1)
         IF (YRA(JJ,JD,K+1).LE.0.0)THEN
             YRA(JJ,JD,K+1) = 0.0
         ENDIF        ! maybe needs to add the protonaton loadings
 204  CONTINUE

*    ---- protonation rate for molecular mono_amines (type 1) -----
      IF(NOD1C.GE.1)THEN
         DO 206 KK = 1,NOD1C
           ITEMP = PRIORITY_C(NOCAT+KK)
           IF(NEXC(K+1).GE.ITEMP.AND.YRC(ITEMP,J,K+1).LT.0.999)THEN

            IF(NEXA(K).EQ.(NA+1)) THEN
                  CD1CSTAI(KK) = XBCI(ITEMP)*CTCI*XBAI(NA+1)*CTAI
     1                   /DISSD1C(KK)
             ELSE
                  CD1CSTAI(KK) = 0.0
             END IF

            IF(CD1CSTAI(KK).GE.CBD1CSTA(KK)) CD1CSTAI(KK)=CBD1CSTA(KK)
               RATE_D1CSTA(KK,K+1) = MTC_D1CSTA(KK)*(CBD1CSTA(KK)-
     1                CD1CSTAI(KK))*(-CONP)*CONY*CF/CFEED_D1C(KK)
            ELSE
               RATE_D1CSTA(KK,K+1) = 0.0
            ENDIF
 206     CONTINUE
      ENDIF

*     ---- protonation rate for molecular organic acids (type 1) -----
      IF(NOD1A.GE.1) THEN
         DO 209 MM = 1, NOD1A
              ITEMP = PRIORITY_A(NOANI+MM)
```

```
          IF(NEXA(K+1).GE.ITEMP.AND.YRA(ITEMP,J,K+1).LT.0.999) THEN
               IF(NEXC(K).EQ.(NC+1)) THEN
                  CD1ASTAI(MM) = XBCI(NC+1)*CTCI*XBAI(ITEMP)*CTAI
     1                   /DISSD1A(MM)
               ELSE
                  CD1ASTAI(MM) = 0.0
               END IF
          IF(CD1ASTAI(MM).GE.CBD1ASTA(MM)) CD1ASTAI(MM)=CBD1ASTA(MM)
             RATE_D1ASTA(MM,K+1) = MTC_D1ASTA(MM)*(CBD1ASTA(MM)-
     1                CD1ASTAI(MM))*(-CONS)*CF/CFEED_D1A(MM)
          ELSE
             RATE_D1ASTA(MM,K+1) = 0.0
          ENDIF
 209    CONTINUE

      ENDIF

*    ------- protonation rate for divalent amines -------
          IF(NOD2C.GE.1)THEN
             ITEMP = PRIORITY_C(NOCAT+NOD1C+1)
             IF(NEXC(K+1).GE.ITEMP.AND.YRC(ITEMP,J,K+1).LT.0.999)THEN

               IF(NEXA(K).EQ.(NA+1)) THEN
                  CD2CSTAI=XBCI(ITEMP)*CTCI*XBAI(NA+1)*CTAI
     1                   /DISSD2C(1)
               ELSE
                  CD2CSTAI = 0.0
               END IF

               IF(CD2CSTAI.GE.CBD2CSTA) CD2CSTAI=CBD2CSTA

              RATE_D2CSTA(K+1) = MTC_D2CSTA*(CBD2CSTA-CD2CSTAI)
     1                   *(-CONP)*CONY*CF/CFEED_D2C
             ELSE
                  RATE_D2CSTA(K+1) = 0.0
             ENDIF
          ENDIF

*    ------ protonation rate for carbonates -----------
         IF(NOD2A.GE.1)THEN
        ITEMP = PRIORITY_A(NOANI+NOD1A+1)
        IF(NEXA(K+1).GE.ITEMP.AND.YRA(ITEMP,J,K+1).LT.0.999)THEN
            IF(NEXC(K).EQ.(NC+1)) THEN
                CD2ASTAI=XBCI(NC+1)*CTCI*XBAI(ITEMP)*CTAI
     1                   /DISSD2A(1)
            ELSE
                CD2ASTAI = 0.0
            END IF

            IF(CD2ASTAI.GE.CBD2ASTA)    CD2ASTAI=CBD2ASTA

          RATE_D2ASTA(K+1) = MTC_D2ASTA*(CBD2ASTA-CD2ASTAI)*(-CONS)
     1                     *CF/CFEED_D2A

        ELSE
                RATE_D2ASTA(K+1) = 0.0
        END IF

         ENDIF

*    ----- protonation rate for Trivalent amines (molecular) -----
          IF(NOD3C.GE.1)THEN
             ITEMP = PRIORITY_C(NOCAT+NOD1C+NOD2C+1)
             IF(NEXC(K+1).GE.ITEMP.AND.YRC(ITEMP,J,K+1).LT.0.999)THEN

               IF(NEXA(K).EQ.(NA+1)) THEN
                  CD3CSTAI=XBCI(ITEMP)*CTCI*XBAI(NA+1)*CTAI
     1                   /DISSD3C(1)
               ELSE
                  CD3CSTAI = 0.0
               END IF
```

```
               IF(CD3CSTAI.GE.CBD3CSTA) CD3CSTAI=CBD3CSTA

                RATE_D3CSTA(K+1) = MTC_D3CSTA*(CBD3CSTA-CD3CSTAI)
     1                   *(-CONP)*CONY*CF/CFEED_D3C
             ELSE
                  RATE_D3CSTA(K+1) = 0.0
             ENDIF

          ENDIF

*    ------ protonation rate for Phosphates -----------
         IF(NOD3A.GE.1)THEN
        ITEMP = PRIORITY_A(NOANI+NOD1A+NOD2A+1)
        IF(NEXA(K+1).GE.ITEMP.AND.YRA(ITEMP,J,K+1).LT.0.999)THEN
            IF(NEXC(K).EQ.(NC+1)) THEN
                CD3ASTAI=XBCI(NC+1)*CTCI*XBAI(ITEMP)*CTAI
     1                   /DISSD3A(1)
            ELSE
                CD3ASTAI = 0.0
            END IF

            IF(CD3ASTAI.GE.CBD3ASTA)   CD3ASTAI=CBD3ASTA

          RATE_D3ASTA(K+1) = MTC_D3ASTA*(CBD3ASTA-CD3ASTAI)
     1                   *(-CONS)*CF/CFEED_D3A
        ELSE
                RATE_D3ASTA(K+1) = 0.0
        END IF

         ENDIF

*    ------ protonation rate for Silica -----------
         IF(NOSLCA.GE.1)THEN
        ITEMP = PRIORITY_A(NOANI+NOD1A+NOD2A+NOD3A+1)
        IF(NEXA(K+1).GE.ITEMP.AND.YRA(ITEMP,J,K+1).LT.0.999)THEN
            IF(NEXC(K).EQ.(NC+1)) THEN
                CslcaprotSHI=XBCI(NC+1)*CTCI*XBAI(ITEMP)*CTAI
     1                   /DISSslca(1)
            ELSE
                CslcaprotSHI = 0.0
            END IF

            IF(CslcaprotSHI.GE.CBslcaSH)   CslcaprotSHI=CBslcaSH

            RATE_SHPROT(K+1) = MTC_slcaSH*(CBslcaSH-CslcaprotSHI)
     1                   *(-CONS)*CF/CFEED_slca
        ELSE
            RATE_SHPROT(K+1) = 0.0
        END IF

         ENDIF

**********************************************************************
*     -------- adsorption rate for amines ------
**********************************************************************
      if(NOD1C.GE.1) then
          DO 208 KK = 1,NOD1C
            if (SUMsorpd1c(K+1).LT.0.15) then
           RATE_D1CSORP(KK,K+1)=RATIO_D1C*RATE_COMC(NOCAT+KK,K+1)
           else
            RATE_D1CSORP(KK,K+1)=0
           endif
           RATE_D1CTOT(KK,K+1)=RATE_D1CSTA(KK,K+1)+RATE_D1CSORP(KK,K+1)
 208     CONTINUE
      endif

      if(NOD2C.GE.1) then
          if (SUMsorpd2c(K+1).LT.0.15) then
             RATE_D2CSORP(K+1)=RATIO_D2C*RATE_COMC(NOCAT+NOD1C+1,K+1)
           else
```

```
             RATE_D2CSORP(K+1)=0
           endif
            RATE_D2CTOT(K+1)=RATE_D2CSTA(K+1)+RATE_D2CSORP(K+1)
       endif

       if(NOD3C.GE.1) then
         if (SUMsorpd3c(K+1).LT.0.15) then
          RATE_D3CSORP(K+1)=RATIO_D3C*RATE_COMC(NOCAT+NOD1C+NOD2C+1,K+1)
         else
          RATE_D3CSORP(K+1)=0
         endif
          RATE_D3CTOT(K+1)=RATE_D3CSTA(K+1)+RATE_D3CSORP(K+1)
       endif


**********************************************************************
*     -------- adsorption rate for type 1 disso anions ------
**********************************************************************
      IF (NOD1A.GE.1) THEN
          DO 214 MM = 1,NOD1A
           if (SUMsorpd1a(K+1).LT.0.15) then
             RATE_D1ASORP(MM,K+1)=RATIO_D1A*RATE_COMA(NOANI+MM,K+1)
            else
             RATE_D1ASORP(MM,K+1)=0
            endif  ! RATIO THE ADSORPTION RATE TO IONIC TRANSFER RATE

         RATE_D1ATOT(MM,K+1)=RATE_D1ASTA(MM,K+1)+RATE_D1ASORP(MM,K+1)
 214     CONTINUE
      ENDIF

*      ------ adsorption rate for molecular carbonates ------

       IF (NOD2A.GE.1) THEN
          if (SUMsorpd2a(K+1).LT.0.15) then
          RATE_D2ASORP(K+1)=RATIO_D2A*RATE_COMA(NOANI+NOD1A+1,K+1)
           else
           RATE_D2ASORP(K+1)=0
           endif
          RATE_D2ATOT(K+1)=RATE_D2ASTA(K+1)+RATE_D2ASORP(K+1)
       ENDIF

*     ----- adsorption rate for type 3 disso anions (Phosphates) ----

      IF (NOD3A.GE.1) THEN
            if ( SUMsorpd3a(K+1).LT.0.15) then
            RATE_D3ASORP(K+1)=RATIO_D3A*RATE_COMA(NOANI+NOD1A+NOD2A+1,1)
             else
             RATE_D3ASORP(K+1)=0
             endif   ! ratio the adsorption rate to ionic transfer rate
            RATE_D3ATOT(1)=RATE_D3ASTA(K+1)+RATE_D3ASORP(K+1)
       ENDIF

*     ----- physical adsorption rates for various silica ----

       IF (NOslca.GE.1) THEN
             if (SUMsorpslca(K+1).LT.0.05) then
             AC= 1.0
             BI= 3.0
                 CslcasorpSHI=AC*CBslcaSH**BI
            RATE_SHSORP(K+1)= MTC_slcaSH*(CBslcaSH-CslcasorpSHI)
     1                 *(-CONS)*CF/CFEED_slca
                               ! adsorption rate for SH

                 a1=1.0
                 e1=0.5
                 a2=5.0
                 e2=0.8
             CslcaSCI=((1.0/a1)*Yslca_INIT(1))**(1.0/e1)
             CslcaPTI=((1.0/a2)*Yslca_INIT(1))**(1.0/e2)

             IF(CslcaSCI.GE.CBslcaSC)    CslcaSCI=CBslcaSC
```

```
             IF(CslcaPTI.GE.CBslcaPT)    CslcaPTI=CBslcaPT

         RATE_slcaSC(K+1)=MTC_slcaSC*(CBslcaSC-CslcaSCI)
     1                   *(-CONS)*CF/CFEED_slca    ! for colloidal
         RATE_slcaPT(K+1)=MTC_slcaPT*(CBslcaPT-CslcaPTI)
     1                   *(-CONS)*CF/CFEED_slca    ! for particulate! for particulate
           else
             RATE_SHSORP(K+1)=0
             RATE_slcaSC(K+1)=0
             RATE_slcaPT(K+1)=0
           endif

                 RATE_SHTOT(K+1)=RATE_SHPROT(K+1)+RATE_SHSORP(K+1)
                                  ! total rate for molecular silica (SH)
      ENDIF
*    _________________________________________________________
*     Calculate the protonation and adsorption amount to resin
*    _________________________________________________________

*     -----molecualr mono_amines -----
        IF(NOD1C.GE.1) THEN
        DO 207 KK=1,NOD1C
           YRC_PROT_D1C(KK,JD,K+1) = TAU*RATE_D1CSTA(KK,K+1)
     1                                    *CFEED_D1C(KK)/CF
           YRC_SORP_D1C(KK,JD,K+1) = TAU*RATE_D1CSORP(KK,K+1)
     1                                  *CFEED_D1C(KK)/CF

            SUMsorpd1c(K+1) = SUMsorpd1c(K+1) + YRC_SORP_D1C(KK,JD,K+1)
                 !   add the protonation term to loadings
           YRC(nocat+kk,JD,K+1) = YRC(nocat+kk,JD,K+1)+
     1                                 YRC_PROT_D1C(KK,JD,K+1)
 207    CONTINUE
        ENDIF

*    ----- molecular organic acids ------*
        IF(NOD1A.GE.1) THEN
        DO 213 MM=1,NOD1A
           YRA_PROT_D1A(MM,JD,K+1) = TAU*RATE_D1ASTA(MM,K+1)
     1                                    *CFEED_D1A(MM)/CF
           YRA_SORP_D1A(MM,JD,K+1) = TAU*RATE_D1ASORP(MM,K+1)
     1                                    *CFEED_D1A(MM)/CF

            SUMsorpd1a(K+1)= SUMsorpd1a(K+1) + YRA_SORP_D1A(MM,JD,K+1)
                 !   add the protonation term to loadings
           YRA(noani+mm,JD,K+1) = YRA(noani+mm,JD,K+1)+
     1                                 YRA_PROT_D1A(mm,JD,K+1)
 213    CONTINUE
        ENDIF

*    ---- molecular divalent amines -----
        IF(NOD2C.GE.1) THEN
          YRC_PROT_D2C(1,JD,K+1) = TAU*RATE_D2CSTA(K+1)*CFEED_D2C/CF
          YRC_SORP_D2C(1,JD,K+1)= TAU*RATE_D2CSORP(K+1)*CFEED_D2C/CF

           SUMsorpd2c(K+1) = SUMsorpd2c(K+1) + YRC_SORP_D2C(1,JD,K+1)
              !   add the protonation term to loadings
          YRC(nocat+nod1c+1,JD,K+1) = YRC(nocat+nod1c+1,JD,K+1)+
     1                          YRC_PROT_D2C(1,JD,K+1)
       ENDIF

*    ---- molecular carbonates -----
        IF(NOD2A.GE.1) THEN
          YRA_PROT_D2A(1,JD,K+1) = TAU*RATE_D2ASTA(K+1)*CFEED_D2A/CF
          YRA_SORP_D2A(1,JD,K+1)= TAU*RATE_D2ASORP(K+1)*CFEED_D2A/CF

          SUMsorpd2a(K+1) = SUMsorpd2a(K+1) + YRA_SORP_D2A(1,JD,K+1)

               !   add the protonation term to loadings
               YRA(noani+nod1a+1,JD,K+1) = YRA(noani+nod1a+1,JD,K+1)+
     1                          YRA_PROT_D2A(1,JD,K+1)
        ENDIF
```

```
*    ---- molecular Trivalent amines -----
       IF(NOD3C.GE.1) THEN
         YRC_PROT_D3C(1,JD,K+1) = TAU*RATE_D3CSTA(K+1)*CFEED_D3C/CF
         YRC_SORP_D3C(1,JD,K+1)= TAU*RATE_D3CSORP(K+1)*CFEED_D3C/CF

            SUMsorpd3c(K+1) = SUMsorpd3c(K+1)+YRC_SORP_D3C(1,JD,K+1)
             !   add the protonation term to loadings
            YRC(nocat+nod1c+nod2c+1,JD,K+1) = YRC(nocat+nod1c+nod2c+1,
     1                   JD,K+1)+YRC_PROT_D3C(1,JD,K+1)
       ENDIF

*    ---- molecular Phosphates -----
       IF(NOD3A.GE.1) THEN
         YRA_PROT_D3A(1,JD,K+1) = TAU*RATE_D3ASTA(K+1)*CFEED_D3A/CF
         YRA_SORP_D3A(1,JD,K+1) = TAU*RATE_D3ASORP(K+1)*CFEED_D3A/CF

            SUMsorpd3a(K+1) = SUMsorpd3a(K+1)+YRA_SORP_D3A(1,JD,K+1)
                      !   add the protonation term to loadings
            YRA(noani+nod1a+nod2a+1,JD,K+1) = YRA(noani+nod1a+nod2a+1,
     1                    JD,K+1)+YRA_PROT_D3A(1,JD,K+1)
       ENDIF

*    ---- molecular Silica -----
       IF(NOslca.GE.1) THEN
         YRA_PROT_slca(1,JD,K+1) = TAU*RATE_SHPROT(K+1)*CFEED_slca/CF
            YRA(NA-KS+1,JD,K+1) = YRA(NA-KS+1,JD,K+1)
     1                                   +YRA_PROT_slca(1,JD,K+1)
         YRA_SORP_slca(1,JD,K+1) = TAU*(RATE_SHSORP(K+1)
     1           +Rate_slcaSc(K+1)+Rate_slcaPT(K+1))*CFEED_slca/CF
            SUMsorpslca(K+1)= SUMsorpslca(K+1)+YRA_SORP_slca(1,JD,K+1)
                                  ! add the both to the protonation

       ENDIF

 400  CONTINUE
*     ___________________________
*     // End of distance loop //
*     ___________________________

*    ___________________________
*    Print breakthrough curves
*    ___________________________

*---- Converting outlet concentrations into " ppb"

      DO 810 II = 1,NOCAT
         OUT_CAT(II) = CBCAT(II)*EWCAT(II)/1.E-6
 810  CONTINUE

      IF (NOD1C.GE.1) THEN
          DO 812 KK =1, NOD1C
             OUT_D1C(KK) = CBD1C(KK)*EWD1C(KK)/1.E-6
             OUT_D1CSTA(KK) = CBD1CSTA(KK)*EWD1CSTA(KK)/1.E-6
             OUT_D1CT(KK) = OUT_D1C(KK) + OUT_D1CSTA(KK)
 812      CONTINUE
      END IF

      IF (NOD2C.GE.1) THEN
         DO 215 NN = 1, NOD2C
            OUT_D2C(NN) = CBD2C(NN)*EWD2C(NN)/1.E-6
 215     CONTINUE
            OUT_D2CSTA = CBD2CSTA*EWD2CSTA/1.E-06
            OUT_D2CT = OUT_D2C(1)+OUT_D2C(2)+OUT_D2CSTA
      END IF

        IF (NOD3C.GE.1) THEN
         DO 715 KT = 1, NOD3C
            OUT_D3C(KT) = CBD3C(KT)*EWD3C(KT)/1.E-6
 715     CONTINUE
            OUT_D3CSTA = CBD3CSTA*EWD3CSTA/1.E-06
```

```
            OUT_D3CT = OUT_D3C(1)+OUT_D3C(2)+OUT_D3C(3)+OUT_D3CSTA
      END IF

      DO 814 JJ = 1,NOANI
         OUT_ANI(JJ) = CBANI(JJ)*EWANI(JJ)/1.E-6
 814  CONTINUE

      IF (NOD1A.GE.1) THEN
          DO 217 MM = 1, NOD1A
             OUT_D1A(MM) = CBD1A(MM)*EWD1A(MM)/1.E-6
             OUT_D1ASTA(MM)= CBD1ASTA(MM)*EWD1ASTA(MM)/1.E-6
            OUT_D1AT(MM) = OUT_D1A(MM) + OUT_D1ASTA(MM)
 217   CONTINUE
      END IF

       IF (NOD2A.GE.1) THEN
         DO 216 LL = 1, NOD2A
            OUT_D2A(LL) = CBD2A(LL)*EWD2A(LL)/1.E-6
 216     CONTINUE
            OUT_D2ASTA = CBD2ASTA*EWD2ASTA/1.E-06   ! for molecular form
            OUT_D2AT = OUT_D2A(1)+OUT_D2A(2)+OUT_D2ASTA
      END IF

       IF (NOD3A.GE.1) THEN
         DO 816 KP = 1, NOD3A
            OUT_D3A(KP) = CBD3A(KP)*EWD3A(KP)/1.E-6
 816     CONTINUE
            OUT_D3ASTA = CBD3ASTA*EWD3ASTA/1.E-06
            OUT_D3AT = OUT_D3A(1)+OUT_D3A(2)+OUT_D3A(3)+OUT_D3ASTA
      END IF

      IF (NOSLCA.GE.1) THEN
         DO 818 KS = 1, NOSLCA
            OUT_slca(KS) = CBslca(KS)*EWslca(KS)/1.E-06
 818     CONTINUE
            OUT_slcaSC = CBslcaSC*EWslcaSC/1.E-06
             OUT_slcaSH = CBslcaSH*EWslcaSH/1.E-06
             OUT_slcaPT = CBslcaPT*EWslcaPT/1.E-06
             if (CFEED_slca.GT.Csoluslca) then
            OUT_slcaT=OUT_slca(1)+OUT_slca(2)+OUT_slca(3)+OUT_slcaSH
     1                        +OUT_slcaSC+OUT_slcaPT
             else
            OUT_slcaT=OUT_slca(1)+OUT_slca(2)+OUT_slca(3)+OUT_slcaSH
             endif
      END IF

*-------

      TAUTIM = TAUTOT*ANI_DIA*ANI_CAP/(MTC_REF*CF*60.)/1440.
      PH = ALOGDISSW + LOG10(CBOH)

C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

       IF (CFLAG.EQ.'Y'.OR.CFLAG.EQ.'y') THEN
          IF (TAUTOT.GT.TAUMAX_OLD1)THEN
            DALTTAU1=(TAUTOT-TAUMAX_OLD1)
            DALTIME1=DALTTAU1*ANI_DIA*ANI_CAP/(MTC_REF*CF*60.)/1440.
            TAUTIM=DAY_NEW1+DALTIME1
          ENDIF
       ENDIF
C@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

       IF (VFLAG.EQ.'Y'.OR.VFLAG.EQ.'y') THEN
          IF (TAUTOT.GT.TAUMAX_OLD2)THEN
            DALTTAU2=(TAUTOT-TAUMAX_OLD2)
            DALTIME2=DALTTAU2*ANI_DIA*ANI_CAP/(MTC_REF*CF*60.)/1440.
            TAUTIM=DAY_NEW2+DALTIME2
          ENDIF
       ENDIF
```

```
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*    ______________________________________________
*    Store every tenth iteration to the print file
*    ______________________________________________

      IF (KPRINT.NE.2) GOTO 218

          IF (NOSLCA.GE.1) THEN
            WRITE(*,508)TAUTIM,PH,(OUT_CAT(II),II=1,NOCAT),
     1                           (OUT_D1C(KK),KK=1,NOD1C),
c     1                           OUT_D2CT,
c     1                           (OUT_D2C(NN),NN=1,NOD2C),
c     1                           OUT_D3CT,
c     1                           (OUT_D3C(KT),KT=1,NOD3C),
     1                           (OUT_ANI(JJ),JJ=1,NOANI),
     1                           (OUT_D1A(MM),MM=1,NOD1A),
     1                           OUT_D2AT,
     1                           (OUT_D2A(LL),LL=1,NOD2A),
     1                           OUT_D3AT,
     1                           (OUT_D3A(KP),KP=1,NOD3A),
     1                           OUT_slcaT,
     1                           (OUT_slca(KS),KS=1,NOSLCA),
     1                   OUT_slcaSH,OUT_slcaSC,OUT_slcaPT

          ELSE IF(NOD3A.GE.1) THEN
            WRITE(*,508)TAUTIM,PH,(OUT_CAT(II),II=1,NOCAT),
     1                           (OUT_D1C(KK),KK=1,NOD1C),
     1                           OUT_D2CT,
     1                           (OUT_D2C(NN),NN=1,NOD2C),
     1                           OUT_D3CT,
     1                           (OUT_D3C(KT),KT=1,NOD3C),
     1                           (OUT_ANI(JJ),JJ=1,NOANI),
     1                           (OUT_D1A(MM),MM=1,NOD1A),
     1                           OUT_D2AT,
     1                           (OUT_D2A(LL),LL=1,NOD2A),
     1                           OUT_D3AT,
     1                           (OUT_D3A(KP),KP=1,NOD3A)
          ELSE IF(NOD2A.GE.1) THEN
                 WRITE(*,508)TAUTIM,PH,(OUT_CAT(II),II=1,NOCAT),
     1                           (OUT_D1C(KK),KK = 1,NOD1C),
     1                           OUT_D2CT,
     1                           (OUT_D2C(NN),NN=1,NOD2C),
     1                           OUT_D3CT,
     1                           (OUT_D3C(KT),KT=1,NOD3C),
     1                           (OUT_ANI(JJ),JJ=1,NOANI),
     1                           (OUT_D1A(MM),MM=1,NOD1A),
     1                           OUT_D2AT,
     1                           (OUT_D2A(LL),LL=1,NOD2A)
            ELSE IF(NOD1A.GE.1) THEN
                  WRITE(*,508)TAUTIM,PH,(OUT_CAT(II),II=1,NOCAT),
     1                           (OUT_D1C(KK),KK = 1,NOD1C),
     1                           OUT_D2CT,
     1                           (OUT_D2C(NN),NN=1,NOD2C),
     1                           OUT_D3CT,
     1                           (OUT_D3C(KT),KT=1,NOD3C),
     1                           (OUT_ANI(JJ),JJ=1,NOANI),
     1                           (OUT_D1A(MM),MM=1,NOD1A)

            ELSE IF(NOD3C.GE.1) THEN
                  WRITE(*,508)TAUTIM,PH,(OUT_CAT(II),II=1,NOCAT),
     1                           (OUT_D1C(KK),KK = 1,NOD1C),
     1                           OUT_D2CT,
     1                           (OUT_D2C(NN),NN=1,NOD2C),
     1                           OUT_D3CT,
     1                           (OUT_D3C(KT),KT=1,NOD3C),
     1                           (OUT_ANI(JJ),JJ=1,NOANI)
               ELSE IF(NOD2C.GE.1) THEN
              WRITE(*,508)TAUTIM,PH,(OUT_CAT(II),II=1,NOCAT),
     1                           (OUT_D1C(KK),KK = 1,NOD1C),
     1                           OUT_D2CT,
     1                           (OUT_D2C(NN),NN=1,NOD2C),
```

```
     1                           (OUT_ANI(JJ),JJ=1,NOANI)

           ELSE IF(NOD1C.GE.1) THEN
                      WRITE(*,508)TAUTIM,PH,(OUT_CAT(II),II=1,NOCAT),
     1                           (OUT_D1C(KK),KK = 1,NOD1C),
     1                           (OUT_ANI(JJ),JJ=1,NOANI)
           ELSE
                   WRITE(*,508)TAUTIM,PH,(OUT_CAT(II),II=1,NOCAT),
     1                           (OUT_ANI(JJ),JJ=1,NOANI)

        ENDIF
 508  FORMAT(1X,F10.4,4X,F8.4,16(4X,E12.4))

       KPRINT = 0

 218  CONTINUE

       KPRINT = KPRINT+1
       JK = J
       IF (J.EQ.4) THEN
          J = 1
       ELSE
         J = J+1
       ENDIF
*    ______________________________
*    End of time loop
*    ______________________________

       IF (JFLAG.EQ.1) STOP
       TAUTOT = TAUTOT + TAU

       GOTO 111

 538  STOP
      END

*   ____________________________
*   End of main program  !
*   ____________________________

***********************************************************************
*    Subroutine "DISSOEQ" to calculate the pH to satisfy the charge
*    balance. It also calculates the new concentrations of the disso-
*    ciative species. It calls the function "SOLVER" and "FUNC"
*    during the calculations. It finds the root for the charge balance
*    equation using bisection method. If more than one root are found
*    it selects the root nearest to the value in the previous slice.
*    PHOLD - pH in the previous slice.
*    PHNEW - pH in the current slice, this is calculated here.
***********************************************************************

        SUBROUTINE DISSOEQ(CBCAT,CBANI,CBD1C,CBD1CSTA,CTO_D1C,CBD1A,
     1         CBD1ASTA,CTO_D1A,CBD2C,CBD2CSTA,CTO_D2C,CBD2A,CBD2ASTA,
     1         CTO_D2A,CBD3C,CBD3CSTA,CTO_D3C,CBD3A,CBD3ASTA,CTO_D3A,
     1         Csolu,CBslcaSH, CBslca,
     1         CBH,CBOH,PHOLD,PHNEW,IFLAG)

        IMPLICIT  REAL*8 (A-H,O-Z)
        COMMON /S/SEL_CAT(8),SEL_ANI(8),SELD1C(5),SELD1A(5),SELD2C(2),
     1            SELD2A(2),SELD3C(3),SELD3A(3),SELslca(3),
     1            SELCOMC(21),SELCOMA(21),SELH,SELOH
        COMMON /D/DIFU_CAT(8),DIFU_ANI(8),DIFU_D1C(5),DIFU_D1CSTA(5),
     1            DIFU_D1A(5),DIFU_D1ASTA(5),DIFU_D2C(2),DIFU_D2CSTA,
     1            DIFU_D2A(2),DIFU_D2ASTA,DIFU_D3C(3),DIFU_D3CSTA,
     1            DIFU_D3A(3),DIFU_D3ASTA,DIFU_slca(3),Difu_slcaSH,
     1            DIFUH,DIFUOH,DIFUCOMC(21),DIFUCOMA(21)
        COMMON /V/VACAT(8),VAANI(8),VAD1C(5),VAD1A(5),VAD2C(2),
     1            VAD2A(2),VAD3C(3),VAD3A(3),VAslca(3),VAH,VAOH,
     1            VACOMC(21),VACOMA(21)
        COMMON /E/EWCAT(8),EWANI(8),EWD1C(5),EWD1CSTA(5),EWD1A(5),
```

```
     1           EWD1ASTA(5),EWD2C(2),EWD2CSTA,EWD2A(2),EWD2ASTA,
     1           EWD3C(3),EWD3CSTA,EWD3A(3),EWD3ASTA,EWslca(3),
     1           EWslcaSH,EWC(21),EWA(21)
       COMMON /N/NOCAT,NOANI,NOD1C,NOD1A,NOD2C,NOD2A,NOD3C,NOD3A,
     1           NOSLCA,NC,NA
       COMMON /R/TMPC,DISSW,DISSD1C(5),DISSD1A(5),DISSD2C(2),
     1           DISSD2A(2),DISSD3C(3),DISSD3A(3),DISSslca(3)
       COMMON /I/INDEXC(21),INDEXA(21)


      REAL*8 LEFT, RIGHT, LEFTVAL,RIGHTVAL,ROOTS(10)
      REAL*8 DIFF, NEWDIFF,OLDROOT,NEWROOT

      REAL*8 CBCAT(8),CBANI(8),CBD1C(5),CBD1CSTA(5),CTO_D1C(5),
     1       CBD1A(5),CBD1ASTA(5),CTO_D1A(5),CBD2C(2),CBD2CSTA,
     1       CTO_D2C,CBD2A(2),CBD2ASTA,CTO_D2A,CBD3C(3),CBD3CSTA,
     1       CTO_D3C,CBD3A(3),CBD3ASTA,CTO_D3A,CBslca(3)


*______________________________________________________________________
*     This section of the code tries to identify the sections of the
*     pH scale from 1-14 which has the roots and finds the roots.
*______________________________________________________________________
      I = 1
      LEFT = 1.0E-14     !Start from pH = 14(left end of the section)
      RIGHT = 1.0E-13  !Right end of the section
      DO WHILE( RIGHT.LE.1.0E-01 )
              LEFTVAL = FUNC(CBCAT,CBANI,CBD1C,CTO_D1C,CBD1A,CTO_D1A,CBD2C,
     1        CTO_D2C,CBD2A,CTO_D2A,CBD3C,CTO_D3C,CBD3A,CTO_D3A,
     1        csolu,LEFT,IFLAG )
              IF(IFLAG.EQ.1)THEN
                     WRITE(*,*)"Abnormal exit from 'func'"
                     GO TO 999
              ENDIF

              RIGHTVAL = FUNC(CBCAT,CBANI,CBD1C,CTO_D1C,CBD1A,CTO_D1A,
     1        CBD2C,CTO_D2C,CBD2A,CTO_D2A,CBD3C,CTO_D3C,CBD3A,CTO_D3A,
     1        csolu,RIGHT,IFLAG )
              IF(IFLAG.EQ.1)THEN
                     WRITE(*,*)"Abnormal exit from 'func'"
                     GO TO 999
              ENDIF

              IF(LEFTVAL*RIGHTVAL.LT.0.0)THEN       !Check if there is a root
                     ROOTS(I) = SOLVER(CBCAT,CBANI,CBD1C,CTO_D1C,CBD1A,CTO_D1A
     1       ,CBD2C,CTO_D2C,CBD2A,CTO_D2A,CBD3C,CTO_D3C,CBD3A,CTO_D3A,
     1       csolu,LEFT,RIGHT,IFLAG)
                     IF(IFLAG.EQ.1)THEN
                            WRITE(*,*)"Abnormal exit from 'solver'"
                            GO TO 999
                     ENDIF

                     I = I+1
                     LEFT = RIGHT
                     RIGHT = RIGHT*10
              ELSE   !if there is no root increase the section length
                     RIGHT = RIGHT*10
              ENDIF
      END DO

*--------------- End of finding the roots ----------------------

*____________________________________________________________
*    Selecting an appropriate root in case of more than one root
*____________________________________________________________
      IF(I.EQ.1)THEN
              WRITE(*,*)"No roots between pH 0-14"
              IFLAG = 1
      ELSE
              OLDROOT = 10**(-PHOLD)
              NEWROOT = ROOTS(1)
```

```
               DIFF = ABS( OLDROOT - NEWROOT )

               K=2
               DO WHILE( K.LE.I-1 )
                      NEWDIFF = ABS( OLDROOT - ROOTS(K) )
                      IF( NEWDIFF.LT.DIFF )THEN
                             DIFF = NEWDIFF
                             NEWROOT = ROOTS(K)
                      ENDIF
                      K=K+1
               END DO

               CBH = NEWROOT
               IF(NEWROOT.LE.0)THEN
                      IFLAG = 1
                      GO TO 999
               ENDIF

*_____________________________________________________________
*     Calculating the concentrations based on the new pH.
*_____________________________________________________________
                 ! CBD1 and CBD1STA has been combined by Liu, 1/10/98
               CBOH = DISSW/NEWROOT
               PHNEW = -LOG10( NEWROOT )

           IF(NOD1C.GE.1)THEN
                 CALL CBD1(CTO_D1C,CBD1C,CBD1CSTA,DISSD1C,CBOH,NOD1C,IFLAG)
                 IF (IFLAG.EQ.1)GOTO 999
               ENDIF

               IF(NOD1A.GE.1)THEN
             CALL CBD1(CTO_D1A,CBD1A,CBD1ASTA,DISSD1A,CBH,NOD1A,IFLAG)
                 IF (IFLAG.EQ.1)GOTO 999
           ENDIF

               IF(NOD2C.GE.1)THEN
             CALL CBD2(CTO_D2C,CBD2C,CBD2CSTA,DISSD2C,CBOH,IFLAG)
            IF (IFLAG.EQ.1)GOTO 999
               ENDIF

               IF(NOD2A.GE.1)THEN
             CALL CBD2(CTO_D2A,CBD2A,CBD2ASTA,DISSD2A,CBH,IFLAG)
             IF (IFLAG.EQ.1)GOTO 999
               ENDIF

           IF(NOD3C.GE.1)THEN
             CALL CBD3(CTO_D3C,CBD3C,CBD3CSTA,DISSD3C,CBOH,IFLAG)
            IF (IFLAG.EQ.1)GOTO 999
               ENDIF

               IF(NOD3A.GE.1)THEN
             CALL CBD3(CTO_D3A,CBD3A,CBD3ASTA,DISSD3A,CBH,IFLAG)
             IF (IFLAG.EQ.1)GOTO 999
               ENDIF

          if(NOSLCA.ge.1) then
                 Call NewtonSH(Csolu,CBH,DISSSLCA,CBslcaSH,CBslca,IFLAG)
            IF (IFLAG.EQ.1)GOTO 999
           endif

        ENDIF
 999   RETURN
       END

**********************************************************************
*   Function which solves for a root bounded between two values.
*   Uses bisection method to find the root.
**********************************************************************
      FUNCTION SOLVER(CBCAT,CBANI,CBD1C,CTO_D1C,CBD1A,CTO_D1A,
     1 CBD2C,CTO_D2C,CBD2A,CTO_D2A,CBD3C,CTO_D3C,CBD3A,CTO_D3A,
     1    csolu,LEFT,RIGHT,IFLAG)
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LEFT, MID, RIGHT, LEFTVAL, RIGHTVAL, MIDVAL
       REAL*8 CBCAT(8),CBANI(8),CBD1C(5),CTO_D1C(5),CBD1A(5),
     1      CTO_D1A(5),CBD2C(2),CTO_D2C,CBD2A(2),CTO_D2A,CBD3C(3),
     1      CTO_D3C,CBD3A(3),CTO_D3A

      DO WHILE( ABS(1.0 - LEFT/RIGHT).GT.1E-04 )
               LEFTVAL = FUNC( CBCAT,CBANI,CBD1C,CTO_D1C,CBD1A,CTO_D1A,
     1      CBD2C,CTO_D2C,CBD2A,CTO_D2A,CBD3C,CTO_D3C,CBD3A,CTO_D3A,
     1      csolu,LEFT,IFLAG)
               IF(IFLAG.EQ.1)THEN
                       WRITE(*,*)"Abnormal exit from 'func'"
                       GO TO 999
               ENDIF

               RIGHTVAL = FUNC( CBCAT,CBANI,CBD1C,CTO_D1C,CBD1A,CTO_D1A,
     1      CBD2C,CTO_D2C,CBD2A,CTO_D2A,CBD3C,CTO_D3C,CBD3A,CTO_D3A,
     1      csolu,RIGHT,IFLAG)
               IF(IFLAG.EQ.1)THEN
                       WRITE(*,*)"Abnormal exit from 'func'"
                       GO TO 999
               ENDIF

               MID = (LEFT + RIGHT)/2.0
               MIDVAL = FUNC( CBCAT,CBANI,CBD1C,CTO_D1C,CBD1A,CTO_D1A,
     1      CBD2C,CTO_D2C,CBD2A,CTO_D2A,CBD3C,CTO_D3C,CBD3A,CTO_D3A,
     1      csolu,MID,IFLAG)
               IF(IFLAG.EQ.1)THEN
                       WRITE(*,*)"Abnormal exit from 'func'"
                       GO TO 999
               ENDIF

               IF( LEFTVAL*MIDVAL.LT.0.0 )THEN
                       RIGHT = MID
               ELSE
                       LEFT = MID
               ENDIF
      END DO

      SOLVER = (LEFT+RIGHT)/2
 999  RETURN
      END

*****************************************************************
*     Function to evaluate the "charge balance" function
*****************************************************************

        FUNCTION FUNC(CBCAT,CBANI,CBD1C,CTO_D1C,CBD1A,CTO_D1A,
     1         CBD2C,CTO_D2C,CBD2A,CTO_D2A, CBD3C,CTO_D3C,CBD3A,
     1         CTO_D3A,csolu,HPLUS,IFLAG)
        IMPLICIT  REAL*8 (A-H,O-Z)

        COMMON /S/SEL_CAT(8),SEL_ANI(8),SELD1C(5),SELD1A(5),SELD2C(2),
     1            SELD2A(2),SELD3C(3),SELD3A(3),SELslca(3),
     1            SELCOMC(21),SELCOMA(21),SELH,SELOH
        COMMON /D/DIFU_CAT(8),DIFU_ANI(8),DIFU_D1C(5),DIFU_D1CSTA(5),
     1            DIFU_D1A(5),DIFU_D1ASTA(5),DIFU_D2C(2),DIFU_D2CSTA,
     1            DIFU_D2A(2),DIFU_D2ASTA,DIFU_D3C(3),DIFU_D3CSTA,
     1            DIFU_D3A(3),DIFU_D3ASTA,DIFU_slca(3),Difu_slcaSH,
     1            DIFUH,DIFUOH,DIFUCOMC(21),DIFUCOMA(21)
        COMMON /V/VACAT(8),VAANI(8),VAD1C(5),VAD1A(5),VAD2C(2),
     1            VAD2A(2),VAD3C(3),VAD3A(3),VAslca(3),VAH,VAOH,
     1            VACOMC(21),VACOMA(21)
        COMMON /E/EWCAT(8),EWANI(8),EWD1C(5),EWD1CSTA(5),EWD1A(5),
     1            EWD1ASTA(5),EWD2C(2),EWD2CSTA,EWD2A(2),EWD2ASTA,
     1            EWD3C(3),EWD3CSTA,EWD3A(3),EWD3ASTA,EWslca(3),
     1            EWslcaSH,EWC(21),EWA(21)
        COMMON /N/NOCAT,NOANI,NOD1C,NOD1A,NOD2C,NOD2A,NOD3C,NOD3A,
     1            NOSLCA,NC,NA
        COMMON /R/TMPC,DISSW,DISSD1C(5),DISSD1A(5),DISSD2C(2),
```

```
     1            DISSD2A(2),DISSD3C(3),DISSD3A(3),DISSslca(3)
        COMMON /I/INDEXC(21),INDEXA(21)

      REAL*8 FUNC,CTOTANI,CTOTCAT
      REAL*8 CBCAT(8),CBANI(8),CBD1C(5),CBD1CSTA(5),CTO_D1C(5),
     1       CBD1A(5),CBD1ASTA(5),CTO_D1A(5),CBD2C(2),CTO_D2C,
     1       CBD2A(2),CTO_D2A,CBD3C(3),CTO_D3C,CBD3A(3),CTO_D3A,
     1       CBslca(3)

      CTOTCAT = 0
      CTOTANI = 0
       OHMINUS = DISSW/HPLUS

      DO 11 I=1,NOCAT
        IF(CBCAT(I).LT.0.0)THEN
          WRITE(*,*)"Negative bulk concentrations encounted in func."
          IFLAG = 1
          GO TO 999
        ENDIF
        CTOTCAT = CTOTCAT + CBCAT(I)
  11  CONTINUE

       IF(NOD1C.GE.1)THEN
        CALL CBD1(CTO_D1C,CBD1C,CBD1CSTA,DISSD1C,OHMINUS,NOD1C,IFLAG)
           IF (IFLAG.EQ.1)GOTO 999
        DO 12 I=1,NOD1C
          CTOTCAT = CTOTCAT + CBD1C(I)
  12    CONTINUE
      ENDIF

       IF(NOD2C.GE.1)THEN
         CALL CBD2(CTO_D2C,CBD2C,CBD2CSTA,DISSD2C,OHMINUS,IFLAG)
            IF (IFLAG.EQ.1)GOTO 999
      DO 15 I=1,NOD2C
          CTOTCAT = CTOTCAT + CBD2C(I)
  15  CONTINUE
      ENDIF

       IF(NOD3C.GE.1)THEN
         CALL CBD3(CTO_D3C,CBD3C,CBD3CSTA,DISSD3C,OHMINUS,IFLAG)
            IF (IFLAG.EQ.1)GOTO 999
      DO 415 I=1,NOD3C
          CTOTCAT = CTOTCAT + CBD3C(I)
415   CONTINUE
      ENDIF

      CTOTCAT = CTOTCAT + HPLUS

      DO 13 I=1,NOANI
        IF(CBANI(I).LT.0.0)THEN
          WRITE(*,*)"Negative bulk concentrations encounted in func."
          IFLAG = 1
          GO TO 999
        ENDIF
               CTOTANI = CTOTANI +  CBANI(I)
 13   CONTINUE

       IF(NOD1A.GE.1)THEN
         CALL CBD1(CTO_D1A,CBD1A,CBD1ASTA,DISSD1A,HPLUS,NOD1A,IFLAG)
            IF (IFLAG.EQ.1)GOTO 999
         DO 14 LL = 1,NOD1A
           CTOTANI = CTOTANI + CBD1A(LL)
 14      CONTINUE
       ENDIF

         IF(NOD2A.GE.1)THEN
           CALL CBD2(CTO_D2A,CBD2A,CBD2ASTA,DISSD2A,HPLUS,IFLAG)
               IF (IFLAG.EQ.1)GOTO 999
          DO 16 I=1,NOD2A
              CTOTANI = CTOTANI + CBD2A(I)
 16       CONTINUE
```

```
        ENDIF

        IF(NOD3A.GE.1)THEN
           CALL CBD3(CTO_D3A,CBD3A,CBD3ASTA,DISSD3A,HPLUS,IFLAG)
               IF (IFLAG.EQ.1)GOTO 999
          DO 416 I=1,NOD3A
              CTOTANI = CTOTANI + CBD3A(I)
416       CONTINUE
        ENDIF

******** silica case***********  Liu 4/2/98
       if(NOSLCA.ge.1) then
           Call NewtonSH(Csolu,HPLUS,DISSSLCA,CBslcaSH,CBslca,IFLAG)
          IF (IFLAG.EQ.1)GOTO 999
          CTOTANI = CTOTANI+CBslca(1)+CBslca(2)+CBslca(3)
       endif
*******************************+

      CTOTANI = CTOTANI+ OHMINUS

      FUNC = CTOTCAT - CTOTANI
999   RETURN
      END

**********************************************************************
*       Subroutine "CBD1"

*       This subroutine is a generalized calculation of the ionic conc.
*       of type 1 dissociative species.  The calculations apply for
*       both disso cations and anions. Note: variable DUM is CH for acids,
*       COH for bases.

**********************************************************************

      SUBROUTINE CBD1(CTO_D1,CD1,CD1STA,DISSD1,DUM,NOD1,IFLAG)
         INTEGER IFLAG,NOD1
         REAL*8 DUM,CD1(5),DISSD1(5),CD1STA(5),CTO_D1(5)
         DO 112 I = 1,NOD1
         IF((DUM+DISSD1(I)).EQ.0.OR.DISSD1(I).EQ.0.)THEN
           IFLAG = 1
           RETURN
         ENDIF
        CD1(I) = DISSD1(I)*CTO_D1(I)/(DISSD1(I)+DUM)
         CD1STA(I) = CD1(I)*DUM/DISSD1(I)
 112  CONTINUE
         RETURN
         END

**********************************************************************
*       Subroutine "CBD2"

*       This subroutine is a generalized calculation of the ionic conc.
*       of type 2 dissociative species.  The calculations apply for
*       both cations and anions, note variable DUM is CH for acids,
*       COH for bases.

**********************************************************************

      SUBROUTINE CBD2(CTO_D2,CD2,CD2STA,DISSD2,DUM,IFLAG)
         INTEGER IFLAG
         REAL*8 DUM,CD2(2),DISSD2(2),CTO_D2,CD2STA

         IF(DUM.EQ.0.)THEN
           IFLAG = 1
           RETURN
         ENDIF
         CD2(1) = CTO_D2/(DUM/DISSD2(1)+
     1                    DISSD2(2)/DUM+1)     !--HCO3-
        CD2(2) = DISSD2(2)*CD2(1)/DUM           !--CO3-2
        CD2STA = DUM*CD2(1)/DISSD2(1)               ! Molecular CO2
```

```
         RETURN
         END

*********************************************************************
*       Subroutine "CBD3"

*       This subroutine is a generalized calculation of the ionic conc.
*       of type 3 dissociative species.  The calculations apply for
*       both cations and anions, note variable DUM is CH for acids,
*       COH for bases.

*********************************************************************

      SUBROUTINE CBD3(CTO_D3,CD3,CD3STA,DISSD3,DUM,IFLAG)
         INTEGER IFLAG
         REAL*8 DUM,DENOM,CD3(3),DISSD3(3),CTO_D3,CD3STA

         IF(DUM.EQ.0.)THEN
           IFLAG = 1
           RETURN
         ENDIF

         DENOM = DUM**3+DISSD3(1)*DUM**2+DISSD3(1)*DISSD3(2)*DUM+
     1          DISSD3(1)*DISSD3(2)*DISSD3(3)
         CD3(1)= DISSD3(1)*DUM**2/DENOM*CTO_D3         ! H2PO4-
         CD3(2)= DISSD3(1)*DISSD3(2)*DUM/DENOM*CTO_D3         ! HPO4-2
         CD3(3)= DISSD3(1)*DISSD3(2)*DISSD3(3)/DENOM*CTO_D3 !PO4-3
         CD3STA = DUM**3/DENOM*CTO_D3                            ! H3PO4 (molecular)
         RETURN
         END
*********************************************************************
*       Subroutine "NewtonSH"

*       This subroutine uses Newton-Raphson  method to calculate the
*       concentration of silica acid Si(OH)4
*       initial guess of SH concentration is Csolubility

*********************************************************************

      SUBROUTINE NewtonSH(Csolu,CBH,DISSSLCA,X,CBslca,IFLAG)
       REAL*8 Csolu,CBH,DISSSLCA(3),X,CBslca(3)

       EPS=1.E-6
       X0=Csolu         ! silica solubility as initial guess value

       IF(CBH.EQ.0.)THEN
           IFLAG = 1
           RETURN
      ENDIF

       F=(1.+DISSslca(1)/CBH+DISSslca(1)*DISSslca(2)/CBH**2.)*X0
     1      +DISSslca(3)/CBH**2.*X0**4.-Csolu
       DF=(1.+DISSslca(1)/CBH+DISSslca(1)*DISSslca(2)/CBH**2.)
     1      +4.*DISSslca(3)/CBH**2.*X0**3.
      X=X0-F/DF

      DO WHILE ((ABS(X-X0)/X).GT.EPS)
       X0=X
       F=(1.+DISSslca(1)/CBH+DISSslca(1)*DISSslca(2)/CBH**2.)*X0
     1      +DISSslca(3)/CBH**2.*X0**4.-Csolu
      DF=(1.+DISSslca(1)/CBH+DISSslca(1)*DISSslca(2)/CBH**2.)
     1      +4.*DISSslca(3)/CBH**2.*X0**3.
       X=X0-F/DF
       ENDDO

       CBslcaSH=X
       CBslca(1)=CBslcaSH*DISSslca(1)/CBH        ! S- concentration
       CBslca(2)=CBslcaSH*DISSslca(1)*DISSslca(2)/CBH**2. ! S=
       CBslca(3)=(CBslcaSH**4.)*DISSslca(3)/CBH**2. ! polysilicates
       RETURN
       END
```

```
*  ///// END of "DISSOEQ"  /////*

************************************************************************
*    Subroutine "FLUXION" to calculate the interfacial concentrations
*    and fluxes of ions using Franzreb's flux expressions.
*    (Franzreb et al. 1993, Reactive Polymer, 21, P 117-133)
*    ICRIT --- criterion, 1 -- cation side; 2 -- anion side
************************************************************************

      SUBROUTINE FLUXION(YY,XX,NCATION,NANION,XXI,CTI,CAT_CAP,
     1           ANI_CAP,IFLAG,CBC,CBA,CFC,CFA,FLUX,DE,ICRIT)
        IMPLICIT  REAL*8 (A-H,O-Z)

         COMMON /S/SEL_CAT(8),SEL_ANI(8),SELD1C(5),SELD1A(5),SELD2C(2),
     1           SELD2A(2),SELD3C(3),SELD3A(3),SELslca(3),
     1           SELCOMC(21),SELCOMA(21),SELH,SELOH
        COMMON /D/DIFU_CAT(8),DIFU_ANI(8),DIFU_D1C(5),DIFU_D1CSTA(5),
     1           DIFU_D1A(5),DIFU_D1ASTA(5),DIFU_D2C(2),DIFU_D2CSTA,
     1           DIFU_D2A(2),DIFU_D2ASTA,DIFU_D3C(3),DIFU_D3CSTA,
     1           DIFU_D3A(3),DIFU_D3ASTA,DIFU_slca(3),Difu_slcaSH,
     1           DIFUH,DIFUOH,DIFUCOMC(21),DIFUCOMA(21)
        COMMON /V/VACAT(8),VAANI(8),VAD1C(5),VAD1A(5),VAD2C(2),
     1           VAD2A(2),VAD3C(3),VAD3A(3),VAslca(3),VAH,VAOH,
     1           VACOMC(21),VACOMA(21)
        COMMON /E/EWCAT(8),EWANI(8),EWD1C(5),EWD1CSTA(5),EWD1A(5),
     1           EWD1ASTA(5),EWD2C(2),EWD2CSTA,EWD2A(2),EWD2ASTA,
     1           EWD3C(3),EWD3CSTA,EWD3A(3),EWD3ASTA,EWslca(3),
     1           EWslcaSH,EWC(21),EWA(21)
        COMMON /N/NOCAT,NOANI,NOD1C,NOD1A,NOD2C,NOD2A,NOD3C,NOD3A,
     1           NOSLCA,NC,NA
        COMMON /R/TMPC,DISSW,DISSD1C(5),DISSD1A(5),DISSD2C(2),
     1           DISSD2A(2),DISSD3C(3),DISSD3A(3),DISSslca(3)
        COMMON /I/INDEXC(21),INDEXA(21)

        REAL*8 YY(21),XX(21),XXN(21),XXI(21),CCO(21),N(21),AA(21),
     1  BB(21),CBN(21),CI(21),R1(21),CBC(21),CBA(21),CB(21),
     1  FLUX(21),VA(21),DIFU(21),SEL(21)

*-----------------------------------------------------------------
*    Finding the total concentration of cations in bulk phase
*----------------------------------------------------------------

       CTO = 0.0
          IF (ICRIT.EQ.1) THEN
              NO = NCATION
                  C=CFC
                  DO 2 I = 1, NO
                     CB(I) = CBC (I)
  2          CONTINUE
          ELSE
              NO = NANION
                  C=CFA
                  DO 3 I=1,NO
  3                     CB(I)=CBA(I)
             CONTINUE
          ENDIF

       DO 5 I = 1, NO    !(Total number of cations or anions)
          CTO = CTO + CB(I)
  5    CONTINUE
*--------------------------------------------------------------------
*   Finding bulk equivalent fraction relative to feed concentration
*--------------------------------------------------------------------
       IF (CTO.EQ.0.) THEN
          WRITE(*,*) "CTO = 0, Divided by zero, exit"
          IFLAG = 1
          GOTO 65
       END IF

       DO 7 II = 1, NO-1
```

```
          XXN(II) = XX(II)*C/CTO
  7    CONTINUE

          SUMXB = 0.0
       DO 8 II = 1,NO-1
          SUMXB = SUMXB + XXN(II)
  8    CONTINUE
       XXN(NO) = 1. - SUMXB
*____________________________________________________________________
*  Converting the equivalent concentration to molar concentration
*____________________________________________________________________

          IF (ICRIT.EQ.1) THEN            ! ICRIT =1, calculation for cation side.
              NO = NANION         !here the Ncations and Nanions are oposite
                  DO 32 J = 1, NO
                     CB(J) = CBA(J)
                        VA(J) = VACOMA(J)
 32          CONTINUE
          ELSE                ! ICRIT=2, for anion side
              NO = NCATION
                  DO 34 J = 1, NO
                     CB(J) = CBC(J)
                        VA(J) = VACOMC(J)
 34          CONTINUE
          ENDIF

       DO 9 JJ = 1, NO
          CCO(JJ) = CB(JJ)/ABS(VA(JJ))
  9    CONTINUE

*------------------------------------------------------------
*    Calculate the summation of (Zj**2)*Cj or (Zj*Cj)
*------------------------------------------------------------

       SUMZN = 0.0
          SUMZD = 0.0
       DO 10 JJ = 1, NO
          SUMZN = SUMZN + (VA(JJ)**2.)*CCO(JJ)
                 SUMZD = SUMZD + (VA(JJ)*CCO(JJ))
  10   CONTINUE

       IF (SUMZD.EQ.0.) THEN
          WRITE(*,*) "SUMZD = 0 (in 'cation' subroutine), exit"
          IFLAG = 1
          GOTO 65
       ENDIF
*------------

       ZY = SUMZN/SUMZD     ! Mean valency of coions

          IF (ICRIT.EQ.1) THEN
              NO = NCATION
                  DO 36 I = 1, NO
                     VA(I) = VACOMC(I)
36           CONTINUE
          ELSE
              NO = NANION
                  DO 38 I = 1, NO
                     VA(I) = VACOMA(I)
38           CONTINUE
          ENDIF

          IF (ZY .EQ. 0.) THEN
             WRITE(*,*) "ZY = 0 (in 'cation' subroutine), exit"
             GOTO 65
       END IF

       DO 12 II = 1, NO
          N(II) = - VA(II)/ZY    ! relative valency to pseudo ion
  12   CONTINUE
```

```
*-----------------------------------------------------------
*  Start the iteration to find the total interfacial conc
*-----------------------------------------------------------

      CTI = CTO
      EPS = 1.0E-15
      ITER = 0
  15  CONTINUE
      ITER = ITER+1

*----- subroutine to calcualte the interfacial conc for each species----

               IF (ICRIT.EQ.1) THEN
              NO = NCATION
                  CAP = CAT_CAP
                  DO 97 I = 1, NO
                        CB(I) = CBC(I)
                        DIFU(I)=DIFUCOMC(I)
                        SEL(I)= SELCOMC(I)
                        VA(I) = VACOMC(I)
 97          CONTINUE
           ELSE
              NO = NANION
                  CAP = ANI_CAP
                  DO 98 I = 1, NO
                        CB(I) = CBA(I)
                        DIFU(I)=DIFUCOMA(I)
                        SEL(I)= SELCOMA(I)
                        VA(I) = VACOMA(I)
 98          CONTINUE
           ENDIF

       CALL INTERCOM(YY,SEL,VA,CTI,CAP,NO,XXI,IFLAG)
       IF (IFLAG.EQ.1) THEN
         WRITE(*,*)"Abnormal exit from 'intercom' subroutine"
         GO TO 65
       ENDIF

*__________________________________________________________________
*    Calculation of the total interfacial concentration CTI
*__________________________________________________________________

       SUMPN = 0.0
              SUMPD = 0.0
       DO 20 II = 1, NO
         SUMPN = SUMPN + ABS(N(II)*DIFU(II)*(XXI(II)-XXN(II)))
                SUMPD = SUMPD + ABS(DIFU(II)*(XXI(II) - XXN(II)))
  20   CONTINUE

       P = SUMPN/SUMPD          ! The P value in Franzreb's expression

       SUMTN = 0.0
              SUMTD = 0.0
       DO 35 II = 1, NO
          SUMTN = SUMTN + (1.+ N(II))*DIFU(II)*XXN(II)
                 SUMTD = SUMTD + (1.+ N(II))*DIFU(II)*XXI(II)
  35   CONTINUE

       IF ((P+1.) .EQ. 0. .OR. SUMTD .EQ. 0.) THEN
           IFLAG = 1
           WRITE(*,*) " P = -1, OR SUMTD = 0, Divided by zero"
           GOTO 65
       ENDIF

*------ The CTI expression based on Franzreb's algorithm -----

       CTIN = (SUMTN/SUMTD)**(1./(P+1.))*CTO

       IF(ITER.GE.1.E+6)THEN
           WRITE(*,*)"Abnormal exit from 'cation or anion' subroutine"
           IFLAG =1
```

```
            GO TO 65
        ENDIF

* ----- The CTI expression based on Franzreb's algorithm -----

           IF ((ABS(CTIN - CTI)/CTIN) .GT. EPS) THEN
              CTI = CTIN
              GO TO 15
            ELSE
              CTI = CTIN
            END IF

*_____________________________________________________________
*   Calculation of flux and effective diffusivities using
*   Franzreb's flux expressions
*_____________________________________________________________

*________________________________________________________________
*   If total interfacial concentration is equal to the total bulk
*   concentration, then individual interfacial concentrations will
*   be equal to their bulk concentrations according to the relation
*   between CTI and CTO in franzreb's expressions
*________________________________________________________________

      IF( CTI.EQ.CTO ) THEN
          DO 46  II = 1, NO
             FLUX(II) = 0.0
 46       CONTINUE
          GO TO 65
      ENDIF

          IF (ICRIT.EQ.1) THEN
              W = 1.0    ! The charge coefficient, for cations w =1.0
          ELSE
              W = - 1.0   ! For anions, w = -1.0
          ENDIF

*----- Calculate the Bi value in Franzreb's algorithm ------

      DO 50 II = 1, NO
          IF ((CTI**(-P-1.)-CTO**(-P-1.)) .EQ. 0) THEN
             IFLAG = 1
             WRITE(*,*)"Divide by 0 in calculating BB(II), cation"
             GOTO 65
          ENDIF
          BB(II) = W*(XXI(II)-XXN(II))/(CTI**(-P-1.)-CTO**(-P-1.))
 50   CONTINUE

*------ Converting concentrations from equivalents to molar -----

      DO 51 II = 1, NO
          CBN(II) = W*CB(II)/VA(II)
 51   CONTINUE

*------ Calculate the Ai value in Franzreb's algorithm -----*

      DO 53 II = 1, NO
          IF (CTO.EQ.0) THEN
            IFLAG = 1
            WRITE(*,*) "CTO = 0, Divided by zero"
            GOTO 65
          ENDIF
          AA(II) = (VA(II)*CBN(II)-BB(II)*CTO**(-P))/CTO
 53   CONTINUE

*------------------------------------------------------------------
*    Calculate the interfacial concentration for each species from
*    equivalent fractions at the interface
*-----------------------------------------------------------------

      DO 55 II = 1, NO
```

```
            CI(II) = W*XXI(II)*CTI/VA(II)
 55     CONTINUE

*--------------------------------------------------------------------
*    Calculating the flux Ji for cations using Franzreb's expression
*--------------------------------------------------------------------
           IF (P.EQ.0) THEN
              WRITE(*,*) " P = 0, Divided by 0 in 'cation' subroutine"
              IFLAG = 0
              GOTO 65
            END IF

        DO 57 II = 1, NO
            R1(II) = DIFU(II)*((1.-N(II)/P)*(CI(II)-CBN(II))
     1              +N(II)*(AA(II)/VA(II))*(1.+1./P)*(CTI-CTO))
 57     CONTINUE

        SIGR = 0.0
        DO 59 II = 1, NO
           SIGR = SIGR + ABS(R1(II))
 59     CONTINUE

        SIGD = 0.0
        DO 61 II = 1, NO
           SIGD = SIGD + ABS(CI(II)-CBN(II))
 61     CONTINUE

        IF (SIGD .EQ.0) THEN
           IFLAG = 1
           WRITE(*,*) " SIGD = 0, Divided by zero in 'cation' subrou."
           GOTO 65
        END IF
        DE = SIGR/SIGD    !Calculate the effective diffusivity for cations

        DO 63 II = 1, NO
           FLUX(II) = W*VA(II)*R1(II)/DE   ! The ratio of Ji/K
 63     CONTINUE

 65     RETURN
        END


************************************************************************
*  Subroutine "INTERCOM" to calculate the interfacial concentrations
************************************************************************

        SUBROUTINE INTERCOM(Y,K,ZT,CTOT,Q,N,X,IFLAG)
        IMPLICIT REAL*8(A-H,O-Z)
        EXTERNAL POL
        REAL*8 Y(21),X(21),KT(21),K(21),Z(21),CTOT,Q,ORD(21),KREF,LAM(21)
        REAL*8 XSUM,ZT(21)
        INTEGER N,IFLAG

        DO 2 I = 1, N
           Z(I) = ABS(ZT(I))
 2      CONTINUE

*_______________________________________________________________________
*       Calculate the values of LAM for each component, and calculate
*       the order of the concentration as determined by the ratio of
*       the ion valence to the reference ion valence.
*_______________________________________________________________________
        YTEST = 0.0
        IFLAG = 0

        DO 5 I=1,N
          IF(Y(I).LT.0.0) THEN
c            WRITE(*,*)"Negative loadings encountered in 'intercom'"
c             write(*,*) Y(I)
c            IFLAG = 1
c            RETURN
```

```
           Y(I)=0.
         ENDIF
         YTEST = YTEST + Y(I)
 5     CONTINUE

*    __________________________________________________
*    Normalize the initial loading
*    __________________________________________________

         DO 6 I=1,N
           Y(I)=Y(I)/YTEST
 6       CONTINUE

       CALL SELMEAN(K,KREF,Y,YREF,N,IFLAG)
       CALL CHARGEMEAN(Z,ZREF,Y,Q,N,IFLAG)
       CALL SELREF(K,KT,Z,N,KREF,ZREF,IFLAG)

       DO 10 I = 1,N
         LAM(I) = Y(I)*(KT(I)**(-1/ZREF))*(YREF**(-Z(I)/ZREF))*(Q/CTOT)
      1           **(1-(Z(I)/ZREF))
         ORD(I) = Z(I)/ZREF      ! exponent
 10     CONTINUE

*___________________________________________________________________
*     Construct the polynomial and solve for the root using the
*     regula falsi (false position) search algorithm.
*___________________________________________________________________
       X1=0.0
       XACC=1e-10
       X2=1.0
       XROOT = REGFALS(POL,X1,X2,XACC,LAM,ORD,N,IFLAG)

       IF(IFLAG.eq.1)THEN
         WRITE(*,*)"LOADINGS:",(Y(I),I=1,N)
         WRITE(*,*)"SELECTIVITIES:",(k(I),I=1,N)
         WRITE(*,*)"VALENCIES:",(Z(I),I=1,N)
         WRITE(*,*) CTOT,Q,N
         RETURN
       ENDIF

       DO 22 I = 1,N
       X(I)=LAM(I)*XROOT**(ORD(I))
 22     CONTINUE
       XSUM = 0.0
       DO 23 I = 1,N
           XSUM = XSUM + X(I)
 23    CONTINUE
       DO 24 I=1,N
           X(I) = X(I)/XSUM

 24    CONTINUE
 26    RETURN
       END

*****************************************************************
*     Function which calculates the value of the molar fraction
*     polynomial expression.
******************************************************************

       FUNCTION POL(X,LAM,ORD,N)
       IMPLICIT REAL*8(A-H,O-Z)
       REAL*8 X,LAM(21),ORD(21)
       INTEGER N
       POL = -1.
       DO 25 I = 1,N
           POL = POL + LAM(I)*X**(ORD(I))
 25    CONTINUE

 26    RETURN
       END
```

```
*******************************************************************
*   Subroutine SELMEAN
*   This subroutine calculates the mean selectivity coefficient
*   given the current resin loading.  See the documentation
*   enclosed with this code for the derivation of the mean sel-
*   ectivity coefficient
*
*      K(I)= Selectivity array with respect to refer
*      K(I)= Selectivity array with respect to reference ion
*      Y(I)= Current resin loading (meq/ml)
*      YREF = Equivalent ionic fraction of new reference ion,
*                by definition it has a value of 1.0
*      N = Number of cations or anions (includes dissociative
*            species)
*      KREF = the selectivity of the new reference ion with
*            respect to the old one.
********************************************************************

      SUBROUTINE SELMEAN(K,KREF,Y,YREF,N,IFLAG)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 K(21),KREF,Y(21),YREF
      INTEGER N,IFLAG
      KREF = 1.
      YREF = 1.

      DO 10 I = 1,N
      IF (K(I).LT.0.0) THEN
            WRITE(*,*) 'Negative selectivity, abnormal exit'
            IFLAG = 1
            RETURN
      ENDIF
            KREF = KREF*(K(I)**Y(I))
 10   CONTINUE
      RETURN
      END

*******************************************************************
*   Subroutine CHARGEMEAN
*   This subroutine calculates the mean ionic valence as
*   described by equation 6 in Franzreb's flux expressions
*   The variables used are:
*       Q = the total capacity of the resin (meq/ml)
*       Z(I) = the charge array of either cations or anions
*       ZN = the summation term in the numerator of Franzreb's
*           expression
*       ZD = the summation term in the denominator of Franzreb's
*           expression
*       ZREF = the mean ionic valence
*       Y(I) = the equivalent fraction array of  cations or
*              anions (meq/ml)
*       YT(I) = mole fraction resin loading array (mol/l)
*       N = the number of either cations or anions
********************************************************************

      SUBROUTINE CHARGEMEAN(Z,ZREF,Y,Q,N,IFLAG)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 Z(21),Y(21),Qt(21),Q,ZN,ZD,ZREF
      INTEGER N,IFLAG
      ZN = 0.
      ZD = 0.
      ZREF = 0.

*----- Convert equivalent ionic capacities to molar capacities

      DO 5 I = 1,N
        Qt(I) = Q*Y(I)/ABS(Z(I))
 5    CONTINUE

*----- Apply mean valence relationship to calculate the mean valence

      DO 10 I = 1,N
```

```
        ZN = ZN + (Z(I)**2)*Qt(I)
        ZD = ZD + Z(I)*Qt(I)
 10    CONTINUE
        IF (ZD.EQ.0.0) THEN
          IFLAG = 1
          WRITE(*,*) 'Mean valency denominator equals zero'
          RETURN
        ELSE
        ENDIF

      ZREF = ZN/ZD
      RETURN
      END

********************************************************************
*   Subroutine SELREF
*   This subroutine changes the reference ion in the selectivity
*   coefficient array for either cations or anions of arbitrary
*   valence
*   The variables used are:
*       K = Selectivity array with common reference ion
*       KREF = Selectivity of new reference with respect to the
*           old reference
*       N = number of cations or anions in system, also size of
*           arrays
*       Z = ion charge array
*       ZREF = reference ion charge
*********************************************************************

      SUBROUTINE SELREF(K,KT,Z,N,KREF,ZREF,IFLAG)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 K(21),KT(21),KREF,Z(21),ZREF
      INTEGER N,IFLAG
      DO 10 I = 1,N
         IF (KREF.EQ.0.0) THEN
            WRITE (*,*) 'Refefence selectivity equals zero'
            IFLAG = 1
            RETURN
          ENDIF
          KT(I)=(K(I)**ZREF)/(KREF**Z(I))
 10   CONTINUE
      RETURN
      END

***********************************************************************
*     Function to solve the polynomial using regula-falsi method.
***********************************************************************
      FUNCTION REGFALS(POL,X1,X2,XACC,LAM,ORD,N,IFLAG)
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER MAXIT
      REAL*8 REGFALS,X1,X2,XACC,POL,LAM(21),ORD(21)
      INTEGER N,IFLAG
      EXTERNAL POL
      PARAMETER (MAXIT=50)
      INTEGER J
      REAL*8 DEL,DX,F,FH,FL,SWAP,XH,XL

      FL=POL(X1,LAM,ORD,N)
      FH=POL(X2,LAM,ORD,N)

      IF(FL*FH.GT.0.) THEN
          WRITE(*,*) 'Root must be bracked in regfls'
          IFLAG=1
          RETURN
      ENDIF

      IF(FL.LT.0.)THEN
          XL=X1
          XH=X2
      ELSE
          XL=X2
```

```
          XH=X1
          SWAP=FL
          FL=FH
          FH=SWAP
      ENDIF
      DX=XH-XL
      DO 11 J=1,MAXIT
          REGFALS=XL+DX*FL/(FL-FH)
          F=POL(REGFALS,LAM,ORD,N)
          IF(F.LT.0.) THEN
            DEL=XL-REGFALS
            XL=REGFALS
            FL=F
          ELSE
            DEL=XH-REGFALS
            XH=REGFALS
            FH=f
          ENDIF
          DX=XH-XL
        IF(ABS(DEL).LT.XACC.OR.F.EQ.0.) RETURN
 11    CONTINUE

       IF(J.GE.MAXIT)THEN
         IFLAG = 1
         WRITE(*,*)'Regfals exceeds maximun iterations'
         RETURN

       ENDIF
       END
*///// End of calculation of interfacial concen routine /////*

***********************************************************************
*     Subroutine "SORT_IONS" to sort the selectivities, diffusivities
*     and thevalences. They are sorted in the decreasing order of their
*     valences. The ions of same valency are sorted in the descending
*     order of their selectivities. Bubble sort algorithm is used to
*     sort the arrays.
************************************************************************

        SUBROUTINE SORT_IONS(MTC_COMC,MTC_COMA)
        IMPLICIT  REAL*8 (A-H,O-Z)
         COMMON /S/SEL_CAT(8),SEL_ANI(8),SELD1C(5),SELD1A(5),SELD2C(2),
     1            SELD2A(2),SELD3C(3),SELD3A(3),SELslca(3),
     1            SELCOMC(21),SELCOMA(21),SELH,SELOH
        COMMON /D/DIFU_CAT(8),DIFU_ANI(8),DIFU_D1C(5),DIFU_D1CSTA(5),
     1            DIFU_D1A(5),DIFU_D1ASTA(5),DIFU_D2C(2),DIFU_D2CSTA,
     1            DIFU_D2A(2),DIFU_D2ASTA,DIFU_D3C(3),DIFU_D3CSTA,
     1            DIFU_D3A(3),DIFU_D3ASTA,DIFU_slca(3),Difu_slcaSH,
     1            DIFUH,DIFUOH,DIFUCOMC(21),DIFUCOMA(21)
        COMMON /V/VACAT(8),VAANI(8),VAD1C(5),VAD1A(5),VAD2C(2),
     1            VAD2A(2),VAD3C(3),VAD3A(3),VAslca(3),VAH,VAOH,
     1            VACOMC(21),VACOMA(21)
        COMMON /E/EWCAT(8),EWANI(8),EWD1C(5),EWD1CSTA(5),EWD1A(5),
     1            EWD1ASTA(5),EWD2C(2),EWD2CSTA,EWD2A(2),EWD2ASTA,
     1            EWD3C(3),EWD3CSTA,EWD3A(3),EWD3ASTA,EWslca(3),
     1            EWslcaSH,EWC(21),EWA(21)
        COMMON /N/NOCAT,NOANI,NOD1C,NOD1A,NOD2C,NOD2A,NOD3C,NOD3A,
     1            NOSLCA,NC,NA
        COMMON /R/TMPC,DISSW,DISSD1C(5),DISSD1A(5),DISSD2C(2),
     1            DISSD2A(2),DISSD3C(3),DISSD3A(3),DISSslca(3)
        COMMON /I/INDEXC(21),INDEXA(21)

        REAL*8 MTC_COMC(21),MTC_COMA(21)       ! added 5/26/98
*       __________________________________________________________
*       Sorting the cation data
*       __________________________________________________________
        IF (NC.EQ.1) THEN
           GOTO 555
        END IF
           CALL BUBBLE(NC,VACOMC,SELCOMC,DIFUCOMC,MTC_COMC,EWC,INDEXC)
```

```
*
*        ____________________________________________________________
*        Sorting the anion data
*
         ____________________________________________________________
         IF (NA.EQ.1) THEN
            GOTO 555
         END IF

         CALL BUBBLE(NA,VACOMA,SELCOMA,DIFUCOMA,MTC_COMA,EWA,INDEXA)

 555     RETURN
         END

***********************************************************************
*       Subroutine "SCHMIDT" to calculate the Schmidt numbers for all
*       kinds of ions
***********************************************************************
        SUBROUTINE SCHMIDT(VISCO,DEN,DIFU_GEN,NOGEN,SCH_GEN)
        REAL*8 SCH_GEN(NOGEN),VISCO,DEN,DIFU_GEN(NOGEN)
         DO 2 II=1, NOGEN
            SCH_GEN(II) = (VISCO/100.)/DEN/DIFU_GEN(II)
 2       CONTINUE
         RETURN
        END

***********************************************************************
*       Subroutine 'EULER' applies the Euler numerical method for the
*       first three distance steps
***********************************************************************

        SUBROUTINE EULER(XB,XI1,FR,RATE_GEN,K,NE,N)
        REAL*8 XB(N,5000),XI1,RATE_GEN(N,5000),FR
        INTEGER NE,K,N
         DO 128 II = 1, NE
            XB(II,K+1)=XB(II,K)-XI1*FR*RATE_GEN(II,K)
 128     CONTINUE
        RETURN
        END

        SUBROUTINE EULERANI(XB,XI1,FR,RATE_GEN,K,NE,N,EW,DD)
        REAL*8 XB(N,5000),XI1,RATE_GEN(N,5000),FR,DD,EW(N)
        INTEGER K,NE,N
        DO 132 JJ = 1, NE
           XB(JJ,K+1)=XB(JJ,K)-XI1*FR*RATE_GEN(JJ,K)
*          -------------------------------------------------------
*          Adding the desulphonation term to the sulfate. Sulfate
*          identified by its equivalent weight.
*
           _______________________________________________________
           IF ((EW(JJ)-48.0).LE.0.1)  THEN
              XB(JJ,K+1) = XB(JJ,K+1) + DD
           END IF
 132  CONTINUE
        RETURN
        END


**********************************************************************
*       Subroutine "GEAR"--calculates solution fraction using Gear's
*       backward difference method.  "GEARANI" is used for adding
*       Fischer's desulphonation term to the anion array.  Sulfate
*       is recognized by the equivalent weight
**********************************************************************

        SUBROUTINE GEAR(XB,COE,FR,RATE_GEN,XI1,NE,K,N)
        REAL*8 XB(N,5000),COE(N),FR,RATE_GEN(N,5000),XI1
        INTEGER K,NE,N
        DO 134 II = 1,NE
         COE(II) = 3.*XB(II,K-3)-16.*XB(II,K-2)+ 36.*
     1              XB(II,K-1) - 48.*XB(II,K)
          XB(II,K+1) = -XI1*12.*FR*RATE_GEN(II,K)/25.-COE(II)/25.
134     CONTINUE
        RETURN
```

```
       END

       SUBROUTINE GEARANI(XB,COE,FR,RATE_GEN,XI1,NE,EW,DD,K)
       REAL*8 XB(21,5000),COE(21),FR,RATE_GEN(21,5000),XI1,EW(21),DD
       INTEGER NE
       DO 136 II = 1,NE
        COE(II) = 3.*XB(II,K-3)-16.*XB(II,K-2)+ 36.*
     1             XB(II,K-1) - 48.*XB(II,K)
        XB(II,K+1) = -XI1*12.*FR*RATE_GEN(II,K)/25.-COE(II)/25.

*     __________________________________________________________
*        Adding the desulphonation term to the sulfate. Sulfate
*        identified by its equivalent weight (48.0).
*     __________________________________________________________
        IF ((EW(II)-48.0).LE.0.1)   THEN
          XB(II,K+1) = XB(II,K+1) + DD
        ENDIF
 136  CONTINUE
       RETURN
       END

*********************************************************
*     Subroutine to combine the number of total cations
*     or total anions (including dissociative species)
*********************************************************
      SUBROUTINE COMBNO(NI,NT,INIT)
      INTEGER NI,NT,INIT
      NT = INIT
      NT = NT + NI
      RETURN
      END

****************************************************************
*     Subroutine "COMBPROP" to combine the arrays of physlcal
*     properties, like diffusivity, valency, selectivity etc.
****************************************************************
      SUBROUTINE COMBPROP(PROPSEL,PROPSELCOM,PROPVA,PROPVACOM,
     1   PROPDIFU,PROPDIFUCOM,PROPMTC,PROPMTCCOM,PROPEW,
     1   PROPEWCOM,N,INIT)
      REAL*8 PROPSEL(N),PROPSELCOM(21),PROPVA(N),PROPVACOM(21),
     1   PROPDIFU(N),PROPDIFUCOM(21),PROPMTC(N),PROPMTCCOM(21),
     1   PROPEW(N),PROPEWCOM(21)
      INTEGER N,INIT
      DO 52 II = 1, N
         PROPSELCOM(INIT+II) = PROPSEL(II)
         PROPVACOM(INIT+II) = PROPVA(II)
         PROPDIFUCOM(INIT+II) = PROPDIFU(II)
         PROPMTCCOM(INIT+II) = PROPMTC(II)
         PROPEWCOM(INIT+II) = PROPEW(II)
 52   CONTINUE
      RETURN
      END

****************************************************************
*     Subroutine to sort ions using bubble sorting algorithm
****************************************************************
      SUBROUTINE BUBBLE(N,VA,SEL,DIFU,MTC,EW,IDEX)
      REAL*8 VA(21),SEL(21),DIFU(21),MTC(21),EW(21)
      INTEGER IDEX(21)

         DO 44 II = 1,N-1
            DO 33 JJ = II+1,N

               IF(VA(II).LT.VA(JJ))THEN
*                    ________________________________
*                    Interchange valences
*                    ________________________________
                     TEMP = VA(II)
                     VA(II) = VA(JJ)
                     VA(JJ) = TEMP
*                    ________________________________
```

```
*                   Interchange selectivities
*                   ________________________________
                    TEMP = SEL(II)
                    SEL(II) = SEL(JJ)
                    SEL(JJ) = TEMP
*                   ________________________________
*                   Interchange diffusivities
*                   ________________________________
                    TEMP = DIFU(II)
                    DIFU(II) = DIFU(JJ)
                    DIFU(JJ) = TEMP
*                   ______________________________________
*                   Interchange mass transer coefficients
*                   ______________________________________
                    TEMP = MTC(II)
                    MTC(II) = MTC(JJ)
                    MTC(JJ) = TEMP
*                   ________________________________
*                   Interchange equivalent weight
*                   ________________________________
                    TEMP = EW(II)
                    EW(II) = EW(JJ)
                    EW(JJ) = TEMP
*                   _______________________________
*                   Interchange indices
*                   _______________________________
                    TEMP = IDEX(II)
                    IDEX(II) = IDEX(JJ)
                    IDEX(JJ) = TEMP

              ELSE IF(VA(II).EQ.VA(JJ).AND.
     1                SEL(II).LT.SEL(JJ)) THEN
*                   ________________________________
*                   Interchange valences
*                   ________________________________
                    TEMP = VA(II)
                    VA(II) = VA(JJ)
                    VA(JJ) = TEMP
*                   ________________________________
*                   Interchange selectivities
*                   ________________________________
                    TEMP = SEL(II)
                    SEL(II) = SEL(JJ)
                    SEL(JJ) = TEMP
*                   ________________________________
*                   Interchange diffusivities
*                   ________________________________
                    TEMP = DIFU(II)
                    DIFU(II) = DIFU(JJ)
                    DIFU(JJ) = TEMP
*                   ______________________________________
*                   Interchange mass transer coefficients
*                   ______________________________________
                    TEMP = MTC(II)
                    MTC(II) = MTC(JJ)
                    MTC(JJ) = TEMP
*                   ________________________________
*                   Interchange equivalent weight
*                   ________________________________
                    TEMP = EW(II)
                    EW(II) = EW(JJ)
                    EW(JJ) = TEMP
*                   _______________________________
*                   Interchange indices
*                   _______________________________
                    TEMP = IDEX(II)
                    IDEX(II) = IDEX(JJ)
                    IDEX(JJ) = TEMP

              ENDIF
```

```
  33        CONTINUE
  44    CONTINUE
      RETURN
      END
***********************************************************************
*     SUBROUTINE 'DIFUTEMP_C' and 'DIFUTEMP_A' to calculate the
*     diffusivities at new temperature T for other ions: T---- oC.
*     using the Equation (generalized) from the book:
*     Reid et al, "The Properties of Gases and Liquids," 4th Ed.
*     P 622, McGraw Hill (1987)
*     BUT for the ions whose D - T correlations are already known,
*     use correlations instead.   -- ### changed by LIU  6/20/98
***********************************************************************
      SUBROUTINE DIFUTEMP_C(DIFU_C,EWC,TMPC,RTF,N)
       IMPLICIT  REAL*8 (A-H,O-Z)
       REAL*8 DIFU_C(N),EWC(N)

      VISCO = 0.9068   ! viscosity of water at 25 oC

      DO 5 I=1, N
          DIFU_C(I)=DIFU_C(I)*(TMPC+273.15)/334./VISCO

          if (ABS(EWC(I)-23.0).LE.0.1) THEN   ! to see if it is Na+ ion
           DIFU_C(I)=RTF*(23.00498+1.06416*TMPC+(3.3196E-3)*TMPC**2.)
          endif

          if (ABS(EWC(I)-39.55).LE.0.1) THEN  ! if it is K+ ion
           DIFU_C(I)=RTF*(40.22+1.278*TMPC+0.00271*TMPC**2.)
          end if

          if (ABS(EWC(I)-20.04).LE.0.1) THEN  ! if it is Ca++ ion
           DIFU_C(I)=RTF*(1.575*TMPC+23.27)/2.
          end if

          if (ABS(EWC(I)-12.15).LE.0.1) THEN  ! if it is Mg++ ion
           DIFU_C(I)=RTF*(28.81+0.8192*TMPC+0.005419*TMPC**2.)
          end if

          if (ABS(EWC(I)-17.03).LE.0.1) THEN  ! if it is NH3
           DIFU_C(I)=RTF*(39.87+1.356*TMPC+4.554E-4*TMPC**2.)
          end if

          if (ABS(EWC(I)-107.87).LE.0.1) THEN  ! if it is Ag+
              DIFU_C(I)=RTF*(33.12+1.065*TMPC+3.538E-3*TMPC**2.)
          end if

          if (ABS(EWC(I)-68.66).LE.0.1) THEN  ! if it is Ba+2
           DIFU_C(I)=RTF*(34.0+1.042*TMPC+5.68E-3*TMPC**2.)
          end if

 5    CONTINUE
      RETURN
       END
**************************************************************************

      SUBROUTINE DIFUTEMP_A(DIFU_A,EWA,TMPC,RTF,N)
      IMPLICIT  REAL*8 (A-H,O-Z)
       REAL*8 DIFU_A(N),EWA(N)

      VISCO = 0.9068   ! viscosity of water at 25 oC

      DO 10 J=1, N
          DIFU_A(J)=DIFU_A(J)*(TMPC+273.15)/334./VISCO

          if (ABS(EWA(J)-35.45).LE.0.1) THEN  ! if it is Cl-
            DIFU_A(J)=RTF*(39.6493+1.3976*TMPC+0.0033196*TMPC**2.)
          end if

          if (ABS(EWA(J)-48.0).LE.0.1) THEN   ! if it is SO4-2
            DIFU_A(J)=RTF*(2.079*TMPC+35.76)/2.
          end if
```

```
          if (ABS(EWA(J)-61.0).LE.0.1) THEN  ! if it is HCO3-
            DIFU_A(J)=RTF*44.5
          end if

          if (ABS(EWA(J)-30.0).LE.0.1) THEN  ! if it is CO3-2
            DIFU_A(J)=RTF*(1.44*TMPC+36.)/2.
          end if

          if (ABS(EWA(J)-62.0).LE.0.1) THEN  ! if it is NO3-
            DIFU_A(J)=RTF*(40.03+1.161*TMPC+0.003882*TMPC**2.)
          end if

         if (ABS(EWA(J)-19.0).LE.0.1) THEN  ! if it is F-
                DIFU_A(J)=RTF*(107.7-6.61*TMPC+0.1804*TMPC**2.)
          end if

          if (ABS(EWA(J)-79.9).LE.0.1) THEN  ! if it is Br-
            DIFU_A(J)=RTF*(42.52+1.326*TMPC+0.004104*TMPC**2.)
          end if

          if (ABS(EWA(J)-99.45).LE.0.1) THEN  ! if it is ClO4-
            DIFU_A(J)=RTF*(36.96+1.136*TMPC+0.003443*TMPC**2.)
          end if

          if (ABS(EWA(J)-59.0).LE.0.1) THEN  ! if it is NO3-
            DIFU_A(J)=RTF*(40.03+1.161*TMPC+0.003882*TMPC**2.)
          end if
 10   CONTINUE
      RETURN
       END

*******************************************************************
*    Subroutine "NEWCONCEN" to calculate the new operating time
*    when changed to new inlet concentrations
*******************************************************************
        SUBROUTINE NEWCONCEN(PH,CFCAT,CFANI,CFEED_D1C,CFEED_D1A,
     1     CFEED_D2C,CFEED_D2A,CFEED_D3C,CFEED_D3A,CFEED_SLCA,
     1     ANI_CAP,CAT_CAP,VOL_FLOW,MTC_REF,S1,CFH,CFOH,PH_FEED,
     1     CFD1C,CFD1CSTA,CFD1A,CFD1ASTA,CFD2C,CFD2CSTA,CFD2A,
     1     CFD2ASTA,CFD3C,CFD3CSTA,CFD3A,CFD3ASTA,CFslca,CFslcaSH,
     1     CF,TAUMAX,DMAX,DD,CONP,CONS)


        IMPLICIT  REAL*8 (A-H,O-Z)
        COMMON /S/SEL_CAT(8),SEL_ANI(8),SELD1C(5),SELD1A(5),SELD2C(2),
     1            SELD2A(2),SELD3C(3),SELD3A(3),SELslca(3),
     1            SELCOMC(21),SELCOMA(21),SELH,SELOH
        COMMON /D/DIFU_CAT(8),DIFU_ANI(8),DIFU_D1C(5),DIFU_D1CSTA(5),
     1            DIFU_D1A(5),DIFU_D1ASTA(5),DIFU_D2C(2),DIFU_D2CSTA,
     1            DIFU_D2A(2),DIFU_D2ASTA,DIFU_D3C(3),DIFU_D3CSTA,
     1            DIFU_D3A(3),DIFU_D3ASTA,DIFU_slca(3),Difu_slcaSH,
     1            DIFUH,DIFUOH,DIFUCOMC(21),DIFUCOMA(21)
        COMMON /V/VACAT(8),VAANI(8),VAD1C(5),VAD1A(5),VAD2C(2),
     1            VAD2A(2),VAD3C(3),VAD3A(3),VAslca(3),VAH,VAOH,
     1            VACOMC(21),VACOMA(21)
        COMMON /E/EWCAT(8),EWANI(8),EWD1C(5),EWD1CSTA(5),EWD1A(5),
     1            EWD1ASTA(5),EWD2C(2),EWD2CSTA,EWD2A(2),EWD2ASTA,
     1            EWD3C(3),EWD3CSTA,EWD3A(3),EWD3ASTA,EWslca(3),
     1            EWslcaSH,EWC(21),EWA(21)
        COMMON /N/NOCAT,NOANI,NOD1C,NOD1A,NOD2C,NOD2A,NOD3C,NOD3A,
     1            NOSLCA,NC,NA
        COMMON /R/TMPC,DISSW,DISSD1C(5),DISSD1A(5),DISSD2C(2),
     1            DISSD2A(2),DISSD3C(3),DISSD3A(3),DISSslca(3)
        COMMON /I/INDEXC(21),INDEXA(21)
        COMMON /Q/BED_DIA,RES_HT,VOID_FRAC,FCR,FAR,DEN,VISCO,
     1              CAT_DIA,ANI_DIA,SFLAG,FFLAG

        REAL*8  CFCAT(8),CFANI(8),CFD1C(5),CFEED_D1C(5),CFD1CSTA(5)
        REAL*8  CFD1A(5),CFEED_D1A(5),CFD1ASTA(5),CFD2C(2),CFD2A(2)
        REAL*8  CFD3C(3),CFD3A(3),CFslca(3),CBCAT(8),CBANI(8),CBD1C(5)
```

```
        REAL*8  CTO_D1C(5),CBD1CSTA(5),CBD1A(5),CTO_D1A(5),CBD1ASTA(5)
        REAL*8  CBD2C(2),CBD2CSTA,CBD2A(2),CBD2ASTA,CBD3C(3),CBD3CSTA
        REAL*8  CBD3A(3),CBD3ASTA,CFslcaSH,CBslca(3),MTC_REF

        CHARACTER*1 SFLAG,FFLAG

*    ________________________________________________________________
*    Assume the bulk concentrations in the first slice to be equal
*    to the inlet concentrations, and then call the equilibrium
*    subroutine 'DISSOEQ' to calculate the concentrations of
*    dissociation species in this slice.
*    ________________________________________________________________
      DO 18 II=1,NOCAT
         CBCAT(II) = CFCAT(II)
 18   CONTINUE

      DO 20 JJ=1,NOANI
         CBANI(JJ) = CFANI(JJ)
 20   CONTINUE

          IF(NOD1C.GE.1)THEN
         DO 22 KK = 1,NOD1C
              CTO_D1C(KK) = CFEED_D1C(KK)
 22      CONTINUE
          ENDIF

          IF(NOD2C.GE.1)THEN
         CTO_D2C = CFEED_D2C
          ENDIF

          IF(NOD3C.GE.1)THEN
         CTO_D3C = CFEED_D3C
          ENDIF

        IF(NOD1A.GE.1) THEN
         DO 23 MM = 1, NOD1A
            CTO_D1A(MM) = CFEED_D1A(MM)
 23      CONTINUE
        ENDIF

          IF(NOD2A.GE.1)THEN
           CTO_D2A = CFEED_D2A
          ENDIF

          IF(NOD3A.GE.1)THEN
           CTO_D3A = CFEED_D3A
          ENDIF

          IF(NOSLCA.GE.1)THEN
           CTO_slca = CFEED_slca
          ENDIF

      PH_OLD=PH

        CALL DISSOEQ(CBCAT,CBANI,CBD1C,CBD1CSTA,CTO_D1C,CBD1A,CBD1ASTA,
     1         CTO_D1A,CBD2C,CBD2CSTA,CTO_D2C,CBD2A,CBD2ASTA,CTO_D2A,
     1         CBD3C,CBD3CSTA,CTO_D3C,CBD3A,CBD3ASTA,CTO_D3A,
     1         Csolu, CBslcaSH, CBslca,
     1         CBH,CBOH,PH_OLD,PH_NEW,IFLAG)


      IF(IFLAG.EQ.1) THEN
         WRITE(*,*)"Abnormal exit from 'DISSOEQ' subroutine"
         WRITE(*,*)"(outside the loop)"
         GO TO 938
      ENDIF

*
*     Storing the feed concentrations of H+ and OH-.
*     _________________________________________________________________
      CFH = CBH
```

```
      CFOH = CBOH
      PH_FEED = PH_NEW
*
*    Setting the feed concentrations of the ionic and molecular forms
*    of the disso cations equal to the bulk concentrations in the first
*    slice which are calculated in the equilibrium subroutine.
*

      IF(NOD1C.GE.1)THEN
         DO 24 KK = 1,NOD1C
            CFD1C(KK) = CBD1C(KK)
            CFD1CSTA(KK) = CBD1CSTA(KK)
 24      CONTINUE
      ENDIF

        IF(NOD2C.GE.1)THEN
        DO 19 LL = 1, NOD2C
           CFD2C(LL) = CBD2C(LL)
 19     CONTINUE
           CFD2CSTA = CBD2CSTA
      ENDIF

        IF(NOD3C.GE.1)THEN
        DO 219 LL = 1, NOD3C
           CFD3C(LL) = CBD3C(LL)
219     CONTINUE
           CFD3CSTA = CBD3CSTA
      ENDIF
*
*    Setting the feed concentrations of the ionic and molecular forms
*    of the disso anions equal to the bulk concentrations in the first
*    slice which are calculated in the equilibrium subroutine.
*

         IF(NOD1A.GE.1) THEN
           DO 25 MM = 1,NOD1A
             CFD1A(MM) = CBD1A(MM)
                CFD1ASTA(MM) = CBD1ASTA(MM)
 25        CONTINUE
         ENDIF

        IF(NOD2A.GE.1)THEN
        DO 21 LL = 1, NOD2A
           CFD2A(LL) = CBD2A(LL)
 21     CONTINUE
           CFD2ASTA = CBD2ASTA
      ENDIF

        IF(NOD3A.GE.1)THEN
        DO 221 LL = 1, NOD3A
           CFD3A(LL) = CBD3A(LL)
221     CONTINUE
           CFD3ASTA = CBD3ASTA
      ENDIF

      IF(NOSLCA.GE.1)THEN
            CFslca(1) = CBslca(1)
            CFslca(2) = CBslca(2)
            CFslca(3) = CBslca(3)
           CFslcaSH = CBslcaSH
      ENDIF

*
*    Calculating the total feed concentrations of the cationic and anionic
*    species.
*

      CFC = 0.0
      DO 28 II = 1,NOCAT
         CFC = CFC + CFCAT(II)
 28   CONTINUE
```

```
      IF (NOD1C.GE.1) THEN
         DO 30 KK=1,NOD1C
            CFC = CFC + CFD1C(KK)
 30      CONTINUE
      END IF

        IF (NOD2C.GE.1) THEN
           DO 31 LL = 1, NOD2C
              CFC = CFC + CFD2C(LL)
 31        CONTINUE
      END IF

        IF (NOD3C.GE.1) THEN
           DO 231 LL = 1, NOD3C
              CFC = CFC + CFD3C(LL)
231        CONTINUE
      END IF

      CFC = CFC + CFH

      CFA = 0.0
      DO 32 JJ = 1, NOANI
         CFA = CFA + CFANI(JJ)
 32   CONTINUE

      IF (NOD1A.GE.1) THEN
          DO 35 MM = 1, NOD1A
             CFA = CFA + CFD1A(MM)
 35       CONTINUE
      END IF

        IF (NOD2A.GE.1) THEN
           DO 34 LL = 1, NOD2A
              CFA = CFA + CFD2A(LL)
 34        CONTINUE
      END IF

        IF (NOD3A.GE.1) THEN
           DO 234 LL = 1, NOD3A
              CFA = CFA + CFD3A(LL)
234        CONTINUE
      END IF

        IF (NOSLCA.GE.1) THEN
           DO 236 KS = 1, NOSLCA
              CFA = CFA + CFslca(KS)
236        CONTINUE
      END IF

         CFA = CFA + CFOH
*    _________________________________________________________________
*    Setting the total feed concentration equal to the total conc.
*    of cations which is also equal to the total concentration of
*    anions because of charge balance.
*    _________________________________________________________________
         CF = CFC
*    _________________________________________________________________
*    Calculating dimensionless program time limit based on inlet
*    conditions (at Z = 0)
*    _________________________________________________________________

      TMAXC = CAT_CAP*3.142*(BED_DIA/2.)**2.*RES_HT*FCR
     1           /(VOL_FLOW*CF*60.)
      TMAXA = ANI_CAP*3.142*(BED_DIA/2.)**2.*RES_HT*FAR
     1           /(VOL_FLOW*CF*60.)

      IF(TMAXC.GE.TMAXA) THEN
          TMAX = TMAXC
      ELSE
          TMAX = TMAXA
```

```
      ENDIF

      TAUMAX = MTC_REF*CF*(TMAX*60.)/(ANI_DIA*ANI_CAP)
      DMAX=TMAX/1440.
*    _____________________________________________________
*    Defining the desulphonation term (Fisher's data)
*    _____________________________________________________

      IF (SFLAG.EQ.'Y'.OR.SFLAG.EQ.'y') THEN
        DD = S1/CF
      ELSEIF (SFLAG.EQ.'N'.OR.SFLAG.EQ.'n') THEN
        DD = 0.0
      END IF
*    ____________________________________________________________________
*     Calculating the constants outside the loops to avoid repititive
*     calculations inside the loops. These constants are used inside
*     the loops during rate calculations.
*    ____________________________________________________________________

      CONP = -6.*ANI_DIA/(MTC_REF*CAT_DIA*CF)
      CONS = -6./(MTC_REF*CF)

 938  RETURN
      END

**********************************************************************
*     Subroutine"NEWFLOW" to recalculate those variables which depend
*     on the inlet flow rate 'VOL_FLOW', or superficial velocity 'Vs'.
*     After calculation, these varialbes return to the main program
*     for use in the time loop when TAUTOT greater than TAUMAX_OLD2
*********************************************************************
      SUBROUTINE NEWFLOW(XI,XI1,p,q,CAT_CAP,ANI_CAP,VOL_FLOW,
     1    VS,CHTD,CBD1C,CBD1A,CBD2C,CBD2A,CBD3C,CBD3A,CBSLCA,
     1       CBD1CSTA,CBD1ASTA,CBD2CSTA,CBD2ASTA,CBD3CSTA,CBD3ASTA,
     1       CBSLCASH,CF,MTC_CAT,MTC_ANI,MTC_D1C,MTC_D1CSTA,MTC_D1A,
     1       MTC_D1ASTA,MTC_D2C,MTC_D2CSTA,MTC_D2A,MTC_D2ASTA,
     1       MTC_D3C,MTC_D3CSTA,MTC_D3A,MTC_D3ASTA,MTC_slca,
     1       MTC_slcaSH,MTC_slcaSC,MTC_slcaPT,MTC_COMC,MTC_COMA,
     1       MTC_REF,TAUMAX,DMAX,DD,CONP,CONS,NOSLICE)

        IMPLICIT  REAL*8 (A-H,O-Z)
        COMMON /S/SEL_CAT(8),SEL_ANI(8),SELD1C(5),SELD1A(5),SELD2C(2),
     1            SELD2A(2),SELD3C(3),SELD3A(3),SELslca(3),
     1            SELCOMC(21),SELCOMA(21),SELH,SELOH
        COMMON /D/DIFU_CAT(8),DIFU_ANI(8),DIFU_D1C(5),DIFU_D1CSTA(5),
     1            DIFU_D1A(5),DIFU_D1ASTA(5),DIFU_D2C(2),DIFU_D2CSTA,
     1            DIFU_D2A(2),DIFU_D2ASTA,DIFU_D3C(3),DIFU_D3CSTA,
     1            DIFU_D3A(3),DIFU_D3ASTA,DIFU_slca(3),DIFU_slcaSH,
     1            DIFUH,DIFUOH,DIFUCOMC(21),DIFUCOMA(21)
        COMMON /V/VACAT(8),VAANI(8),VAD1C(5),VAD1A(5),VAD2C(2),
     1            VAD2A(2),VAD3C(3),VAD3A(3),VAslca(3),VAH,VAOH,
     1            VACOMC(21),VACOMA(21)
        COMMON /E/EWCAT(8),EWANI(8),EWD1C(5),EWD1CSTA(5),EWD1A(5),
     1            EWD1ASTA(5),EWD2C(2),EWD2CSTA,EWD2A(2),EWD2ASTA,
     1            EWD3C(3),EWD3CSTA,EWD3A(3),EWD3ASTA,EWslca(3),
     1            EWslcaSH,EWC(21),EWA(21)
        COMMON /N/NOCAT,NOANI,NOD1C,NOD1A,NOD2C,NOD2A,NOD3C,NOD3A,
     1            NOSLCA,NC,NA
        COMMON /R/TMPC,DISSW,DISSD1C(5),DISSD1A(5),DISSD2C(2),
     1            DISSD2A(2),DISSD3C(3),DISSD3A(3),DISSslca(3)
        COMMON /I/INDEXC(21),INDEXA(21)
        COMMON /Q/BED_DIA,RES_HT,VOID_FRAC,FCR,FAR,DEN,VISCO,
     1              CAT_DIA,ANI_DIA,SFLAG,FFLAG
        COMMON /C/SCH_CAT(8),SCH_ANI(8),SCH_D1C(5),SCH_D2C(2),
     1       SCH_D3C(3),SCH_D1A(5),SCH_D2A(2),SCH_D3A(3),SCH_SLCA(3),
     1       SCH_D1CSTA(5),SCH_D1ASTA(5),SCH_D2CSTA,SCH_D2ASTA,
     1       SCH_D3CSTA,SCH_D3ASTA,SCH_SLCASH,SCH_SLCASC,SCH_SLCAPT,
     1       SCH_OH

      REAL*8 MTC_CAT(8),MTC_ANI(8),MTC_D1C(5),MTC_D1CSTA(5),MTC_D1A(5)
     1 ,MTC_D1ASTA(5),MTC_D2C(2),MTC_D2A(2),MTC_D3C(3),MTC_D3A(3),
```

```
     1 MTC_D2CSTA,MTC_D2ASTA,MTC_D3CSTA,MTC_D3ASTA,MTC_slca(3),
     1 MTC_slcaSH,MTC_slcaSC,MTC_slcaPT,FUNC_MTC,MTC_COMC(21),
     1 MTC_COMA(21),MTC_REF,CBD1CSTA(5),CBD1ASTA(5),CBD1C(5),
     1 CBD1A(5),CBD2C(2),CBD2A(2),CBD3C(3),CBD3A(3),CBSLCA(3),
     1 CBD2CSTA,CBD2ASTA,CBD3CSTA,CBD3ASTA,CBSLCASH,CF,VOL_FLOW,
     1 p,q, XI,XI1,VS,CAT_CAP,ANI_CAP,CHTD

       CHARACTER*1 SFLAG,FFLAG
*    ______________________________________________________
*    Recalculating the column superficial velocity
*    ______________________________________________________
      AREA = 3.1415927*(BED_DIA**2.)/4.
        VS = VOL_FLOW/AREA
*    __________________________________________________________________
*    Calculating Reynolds Numbers.
*    __________________________________________________________________
      IF (VOID_FRAC.GE.1.0) THEN
         WRITE (*,*) " VOID_FRAC = 1, Abnormal exit"
         GOTO 838
      END IF

      RE_CAT = CAT_DIA*100.*VS*DEN/((1.-VOID_FRAC)*VISCO)
      RE_ANI = ANI_DIA*100.*VS*DEN/((1.-VOID_FRAC)*VISCO)
*    __________________________________________________________________
*    Calculating the mass transfer coefficients.
•    __________________________________________________________________
•

     □
      DO 44 II=1,NOCAT
□
         MTC_CAT(II) = FUNC_MTC(RE_CAT,SCH_CAT(II),DIFU_CAT(II)
□
     1                           ,CAT_DIA)
 44   CONTINUE

      IF (NOD1C.GE.1) THEN
      DO 46 KK=1,NOD1C
              MTC_D1C(KK) = FUNC_MTC(RE_CAT,SCH_D1C(KK),DIFU_D1C(KK)
     1                           ,CAT_DIA)
         IF (CBD1CSTA(KK).GT.CBD1C(KK)) THEN
              MTC_D1CSTA(KK) = (1.+LOG(DIFU_D1CSTA(KK)*CBD1CSTA(KK)/
     1            DIFU_D1C(KK)/CBD1C(KK)))*
     1        FUNC_MTC(RE_CAT,SCH_D1CSTA(KK),DIFU_D1CSTA(KK),CAT_DIA)
         ELSE
              MTC_D1CSTA(KK) = FUNC_MTC(RE_CAT,SCH_D1CSTA(KK),
     1            DIFU_D1CSTA(KK),CAT_DIA)
         END IF
 46   CONTINUE
      END IF

      IF (NOD2C.GE.1) THEN
         DO 47 NN=1,NOD2C
            MTC_D2C(NN) = FUNC_MTC(RE_CAT,SCH_D2C(NN),DIFU_D2C(NN)
     1                           ,CAT_DIA)
 47      CONTINUE
         IF (CBD2CSTA.GT.CBD2C(1)) THEN
            MTC_D2CSTA = (1.+LOG(DIFU_D2CSTA*CBD2CSTA/
     1              DIFU_D2C(1)/CBD2C(1)))*
     1              FUNC_MTC(RE_CAT,SCH_D2CSTA,DIFU_D2CSTA,CAT_DIA)
         ELSE
            MTC_D2CSTA = FUNC_MTC(RE_CAT,SCH_D2CSTA,DIFU_D2CSTA,
     1              CAT_DIA)
         ENDIF
      END IF

      IF (NOD3C.GE.1) THEN
         DO 647 KT=1,NOD3C
            MTC_D3C(KT) = FUNC_MTC(RE_CAT,SCH_D3C(KT),DIFU_D3C(KT)
     1                           ,CAT_DIA)
647      CONTINUE
         IF (CBD3CSTA.GT.CBD3C(1)) THEN
```

```
            MTC_D3CSTA = (1.+LOG(DIFU_D3CSTA*CBD3CSTA/
     1              DIFU_D3C(1)/CBD3C(1)))*
     1              FUNC_MTC(RE_CAT,SCH_D3CSTA,DIFU_D3CSTA,CAT_DIA)
         ELSE
            MTC_D3CSTA = FUNC_MTC(RE_CAT,SCH_D3CSTA,DIFU_D3CSTA,
     1              CAT_DIA)
         ENDIF
      END IF

      DO 48 JJ=1,NOANI
           MTC_ANI(JJ) = FUNC_MTC(RE_ANI,SCH_ANI(JJ),DIFU_ANI(JJ)
     1                          ,ANI_DIA)
 48   CONTINUE

      IF (NOD1A.GE.1) THEN
       DO 49 MM=1,NOD1A
          MTC_D1A(MM) = FUNC_MTC(RE_ANI,SCH_D1A(MM),DIFU_D1A(MM)
     1                          ,ANI_DIA)
          IF (CBD1ASTA(MM).GT.CBD1A(MM)) THEN
              MTC_D1ASTA(MM) = (1.+LOG(DIFU_D1ASTA(MM)*CBD1ASTA(MM)/
     1            DIFU_D1A(MM)/CBD1A(MM)))*
     1        FUNC_MTC(RE_ANI,SCH_D1ASTA(MM),DIFU_D1ASTA(MM),ANI_DIA)
          ELSE
              MTC_D1ASTA(MM) = FUNC_MTC(RE_ANI,SCH_D1ASTA(MM),
     1            DIFU_D1ASTA(MM),ANI_DIA)
          END IF
 49    CONTINUE
      ENDIF

      IF (NOD2A.GE.1) THEN
         DO 50 LL=1,NOD2A
            MTC_D2A(LL) = FUNC_MTC(RE_ANI,SCH_D2A(LL),DIFU_D2A(LL)
     1                          ,ANI_DIA)
 50      CONTINUE
         IF (CBD2ASTA.GT.CBD2A(1)) THEN
            MTC_D2ASTA = (1.+LOG(DIFU_D2ASTA*CBD2ASTA/
     1              DIFU_D2A(1)/CBD2A(1)))*
     1              FUNC_MTC(RE_ANI,SCH_D2ASTA,DIFU_D2ASTA,ANI_DIA)
         ELSE
            MTC_D2ASTA = FUNC_MTC(RE_ANI,SCH_D2ASTA,DIFU_D2ASTA,
     1              ANI_DIA)
         ENDIF
      END IF

      IF (NOD3A.GE.1) THEN
         DO 650 KP=1,NOD3A
            MTC_D3A(KP) = FUNC_MTC(RE_ANI,SCH_D3A(KP),DIFU_D3A(KP)
     1                          ,ANI_DIA)
650      CONTINUE
         IF (CBD3ASTA.GT.CBD3A(1)) THEN
            MTC_D3ASTA = (1.+LOG(DIFU_D3ASTA*CBD3ASTA/
     1              DIFU_D3A(1)/CBD3A(1)))*
     1              FUNC_MTC(RE_ANI,SCH_D3ASTA,DIFU_D3ASTA,ANI_DIA)
         ELSE
            MTC_D3ASTA = FUNC_MTC(RE_ANI,SCH_D3ASTA,DIFU_D3ASTA,
     1              ANI_DIA)
         ENDIF
      END IF

      IF (NOSLCA.GE.1) THEN
         DO 750 KS=1,NOSLCA
            MTC_slca(KS) = FUNC_MTC(RE_ANI,SCH_slca(KS),DIFU_slca(KS)
     1                          ,ANI_DIA)
750      CONTINUE
         IF (CBslcaSH.GT.CBslca(1)) THEN
            MTC_slcaSH = (1.+LOG(DIFU_slcaSH*CBslcaSH/
     1           DIFU_slca(1)/CBslca(1)))*
     1        FUNC_MTC(RE_ANI,SCH_slcaSH,DIFU_slcaSH,ANI_DIA)
         ELSE
            MTC_slcaSH = FUNC_MTC(RE_ANI,SCH_slcaSH,DIFU_slcaSH,
     1              ANI_DIA)    ! for molecular silica Si(OH)4
```

```
         ENDIF
            MTC_slcaSC = FUNC_MTC(RE_ANI,SCH_slcaSC,DIFU_slcaSC,
     1               ANI_DIA)     ! for colloidal silica

            MTC_slcaPT = FUNC_MTC(RE_ANI,SCH_slcaPT,DIFU_slcaPT,
     1               ANI_DIA)     ! for particular silica
      END IF
*     ______________________________________________________________________
*     Call subroutine "COMBPROP" to combine physlcal properties like
*     selectivities, valences, diffusivities and mass transfer
*     coefficients for the interfacial and flux calculations.
*     ______________________________________________________________________

      IF (NOCAT.GE.1) THEN
         CALL COMBPROP(SEL_CAT,SELCOMC,VACAT,VACOMC,DIFU_CAT,
     1        DIFUCOMC,MTC_CAT,MTC_COMC,EWCAT,EWC,NOCAT,0)
        ENDIF

        IF (NOD1C.GE.1) THEN
           CALL COMBPROP(SELD1C,SELCOMC,VAD1C,VACOMC,DIFU_D1C,
     1        DIFUCOMC,MTC_D1C,MTC_COMC,EWD1C,EWC,NOD1C,NOCAT)
        ENDIF

        IF (NOD2C.GE.1) THEN
           CALL COMBPROP(SELD2C,SELCOMC,VAD2C,VACOMC,DIFU_D2C,
     1     DIFUCOMC,MTC_D2C,MTC_COMC,EWD2C,EWC,NOD2C,NOCAT+NOD1C)
        ENDIF

        IF (NOD3C.GE.1) THEN
           CALL COMBPROP(SELD3C,SELCOMC,VAD3C,VACOMC,DIFU_D3C,DIFUCOMC,
     1     MTC_D3C,MTC_COMC,EWD3C,EWC,NOD3C,NOCAT+NOD1C+NOD2C)
        ENDIF

      IF (NOANI.GE.1) THEN
         CALL COMBPROP(SEL_ANI,SELCOMA,VAANI,VACOMA,DIFU_ANI,
     1       DIFUCOMA,MTC_ANI,MTC_COMA,EWANI,EWA,NOANI,0)
        ENDIF

      IF (NOD1A.GE.1) THEN
         CALL COMBPROP(SELD1A,SELCOMA,VAD1A,VACOMA,DIFU_D1A,
     1     DIFUCOMA,MTC_D1A,MTC_COMA,EWD1A,EWA,NOD1A,NOANI)
        ENDIF

      IF (NOD2A.GE.1) THEN
         CALL COMBPROP(SELD2A,SELCOMA,VAD2A,VACOMA,DIFU_D2A,
     1     DIFUCOMA,MTC_D2A,MTC_COMA,EWD2A,EWA,NOD2A,NOANI+NOD1A)
        ENDIF

      IF (NOD3A.GE.1) THEN
         CALL COMBPROP(SELD3A,SELCOMA,VAD3A,VACOMA,DIFU_D3A,DIFUCOMA,
     1     MTC_D3A,MTC_COMA,EWD3A,EWA,NOD3A,NOANI+NOD1A+NOD2A)
        ENDIF

        IF (NOSLCA.GE.1) THEN
           CALL COMBPROP(SELslca,SELCOMA,VAslca,VACOMA,DIFU_slca,
     1                DIFUCOMA,MTC_slca,MTC_COMA,EWslca,EWA,NOSLCA,
     1                NOANI+NOD1A+NOD2A+NOD3A)
        ENDIF

      SELCOMC(NC+1) = SELH
      VACOMC(NC+1) = VAH
      DIFUCOMC(NC+1) = DIFUH

      SELCOMA(NA+1) = SELOH
      VACOMA(NA+1) = VAOH
      DIFUCOMA(NA+1) = DIFUOH

C-------implementing the effects of cationic resin fouling ------
      IF (FFLAG.eq.'Y'.or.FFLAG.eq.'y') THEN
      DO 55 I=1,NC
         MTC_COMC(I)=(1.-p)*MTC_COMC(I)
```

```
 55   CONTINUE
      ENDIF
C ------- Implementing the effects of anionic resin fouling
      IF (FFLAG.eq.'Y'.or.FFLAG.eq.'y') THEN
      DO 59 I = 1, NA
         MTC_COMA(I)=(1.-q)*MTC_COMA(I)
 59   CONTINUE
      ENDIF

       CALL SORT_IONS(MTC_COMC,MTC_COMA)
*    __________________________________________________________________
*    Choosing the reference ion for calculations. The anion with
*    largest mass transfer coefficient is chosen as reference
*    __________________________________________________________________
      IF (NA.EQ.1) THEN              ! if only one anion, itself is reference
         MTC_REF = MTC_COMA(NA)
         GOTO 65
      END IF

*---- Pick hydroxide as the reference, if no other anion is present.

      IF(NA.EQ.0)THEN
         MTC_OH = FUNC_MTC(RE_ANI,SCH_OH,DIFUOH,ANI_DIA)
         MTC_REF = MTC_OH
         GOTO 65
      ENDIF
*------------- find the larges MTC for anions
      NN = 1
      DO 64 JJ = 2, NA
         IF (MTC_COMA(NN).LT.MTC_COMA(JJ)) THEN
             NN = JJ
         END IF
 64   CONTINUE
      MTC_REF = MTC_COMA(NN)
*    __________________________________________________________________
*    Calculating the dimensionless column height(CHTD) and the total
*    number of distance steps (NOSLICE) down the column.
*    __________________________________________________________________

 65   IF (VS.LE.0.0 .OR. ANI_DIA .LE. 0.0) THEN
          WRITE(*,*)"ERROR: VS <= 0.0 and/or ANI_DIA <= 0.0"
          GOTO 838
      ENDIF

      CHTD = MTC_REF*(1.-VOID_FRAC)*RES_HT/(VS*ANI_DIA)
      NOSLICE = CHTD/XI-3+3*XI/XI1
*    __________________________________________________________________
*    Calculating dimensionless program time limit based on inlet
*    conditions (at Z = 0)
*    __________________________________________________________________

      IF (VOL_FLOW .LE.0.0 .OR. CF.LE.0.0) THEN
         WRITE(*,*) "ERROR: VOL_FLOW <= 0.0 and/or CF <= 0.0"
         GOTO 838
      ENDIF

      TMAXC = CAT_CAP*3.142*(BED_DIA/2.)**2.*RES_HT*FCR
     1          /(VOL_FLOW*CF*60.)
      TMAXA = ANI_CAP*3.142*(BED_DIA/2.)**2.*RES_HT*FAR
     1          /(VOL_FLOW*CF*60.)

       IF(TMAXC.GE.TMAXA) THEN
          TMAX = TMAXC
      ELSE
          TMAX = TMAXA
      ENDIF

      IF (ANI_DIA .LE.0.0 .OR. ANI_CAP .LE.0.0) THEN
          WRITE(*,*)"ERROR: ANI_DIA <= 0.0 OR ANI_CAP <= 0.0"
          GOTO 838
      END IF
```

```
      TAUMAX = MTC_REF*CF*(TMAX*60.)/(ANI_DIA*ANI_CAP)
      DMAX=TMAX/1440.
*    _____________________________________________________
*    Defining the desulphonation term (Fisher's data)
*    _____________________________________________________

      IF (SFLAG.EQ.'Y'.OR.SFLAG.EQ.'y') THEN
         S1 = (7.5E+6*EXP(-10278.6/(TMPC+273.15))*CHTD
     1        *3.1415927*(BED_DIA**2.)*CAT_CAP)*(VS*ANI_DIA)*FCR
     1        /(NOSLICE*3600.*4.0*VOL_FLOW*MTC_REF*(1.-VOID_FRAC))

        DD = S1/CF
      ELSEIF (SFLAG.EQ.'N'.OR.SFLAG.EQ.'n') THEN
        DD = 0.0
      END IF
*    __________________________________________________________________
*     Calculating the constants outside the loops to avoid repititive
*     calculations inside the loops. These constants are used inside
*     the loops during rate calculations.
*    __________________________________________________________________
      IF (MTC_REF .LE. 0.0) THEN
          WRITE(*,*) "ERROR: MTC_REF <= 0.0"
          GOTO 838
      END IF

      CONP = -6.*ANI_DIA/(MTC_REF*CAT_DIA*CF)
      CONS = -6./(MTC_REF*CF)
c      CONY = ANI_CAP/CAT_CAP

838    RETURN
       END

***************************************************************
*    Subroutine 'cubspline' is the main program which calls
*    'ZSPL3' and 'SPL3' to interpolate values
***************************************************************
      SUBROUTINE cubspline(CHTD,XI1,XI,N,NMT,Y,U,V,Z,W,B,S,T)
       IMPLICIT REAL*8 (A-H,O-Z)
       REAL*8 T(N),Y(N),U(N),V(N),Z(N),W(N),B(N),S(NMT)
         NF=3*10+1   ! No of slice for the first few small steps

        DO 2 I=1,N
            if (I.LE.NF) then
                T(I)=(I-1)*XI1
            else
               T(I)=(I-1)*XI
            endif
 2      CONTINUE

         Call ZSPL3(N,T,Y,W,B,U,V,Z)

         XINEW=CHTD/(NMT-3+30+1)    ! because of the change of Noslice,
         XI1NEW=XINEW/10.       ! the stepsize looks changed too

        DO 4 I=1,NMT
             if (NMT.LE.NF) then
                 X=(I-1)*XI1NEW
             else
                X=(I-1)*XINEW
             endif
             S(I)=SPL3(N,T,Y,Z,X)
 4      CONTINUE
      RETURN
      END

□
**********************************************************************
□
*     Subroutine 'ZSPL3' is used to determine a natural cubic spline
□
```

```
* by taking N table values (ti,yi) in arrays T and Y and computes
* the Zi's, storing them in array Z
**********************************************************************
      SUBROUTINE ZSPL3(N,T,Y,H,B,U,V,Z)
        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 T(N),Y(N),H(N),B(N),U(N),V(N),Z(N)
        DO 2 I=1,N-1
           H(I)=T(I+1)-T(I)
           B(I)=(Y(I+1)-Y(I))/H(I)
 2    CONTINUE
      U(2)=2.0*(H(1)+H(2))
        V(2)=6.0*(B(2)-B(1))
        DO 4 I=3,N-1
           U(I)=2.0*(H(I)+H(I-1))-H(I-1)**2./U(I-1)
           V(I)=6.0*(B(I)-B(I-1))-H(I-1)*V(I-1)/U(I-1)
 4    CONTINUE
      Z(N)=0.0
        DO 6 I=N-1, 2, -1
           Z(I)=(V(I)-H(I)*Z(I+1))/U(I)
 6    CONTINUE
      Z(1)=0.0
      RETURN
      END

□
*********************************************************************
□
*     FUNCTION 'SPL3' is used to evaluate the natural cubic spline
* function S(x), for a given x value.  SPL3 first determines the
* interval [ti, ti+1] containning x and then evaluates S(x)
*********************************************************************
      FUNCTION SPL3(N,T,Y,Z,X)
        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 T(N),Y(N),Z(N)
        DO 5 I=N-1,2,-1
           DIFF=X-T(I)
           IF (DIFF .GE. 0.0) GOTO 10
 5    CONTINUE
      I=1
        DIFF=X-T(1)
 10   H=T(I+1)-T(I)
      B=(Y(I+1)-Y(I))/H-H*(Z(I+1)+2.0*Z(I))/6.0
        P=0.5*Z(I)+DIFF*(Z(I+1)-Z(I))/(6.0*H)
        P=B+DIFF*P
        SPL3=Y(I)+DIFF*P
        RETURN
        END

**********************************************************************
*    Function 'FUNC_MTC' to calculate the mass transfer coefficients
**********************************************************************
        REAL*8 FUNCTION FUNC_MTC(R,S,DIFU,DIA)
        IMPLICIT REAL*8 (A-H,O-Z)

        COMMON /Q/BED_DIA,RES_HT,VOID_FRAC,FCR,FAR,DEN,VISCO,
     1              CAT_DIA,ANI_DIA,SFLAG,FFLAG
        CHARACTER*1 SFLAG,FFLAG
*    ______________________________________________________________
*    Dwivedi & Upadhyay's correlation for calculating mass
*    transfer coefficients (Dwivedi & Upadhyay, 1977,Ind.Eng.Chem.
*    Pro. Des. Dev. 16(2), 157-165)
*    ______________________________________________________________

      FUNC_MTC = DIFU/DIA *(S**(1./3.))*R*
     1  (0.765/(VOID_FRAC*R)**0.82 + 0.365/(VOID_FRAC*R)**0.386)

      RETURN
      END
```

# VITA

Yunquan Liu

Candidate for the Degree of

Doctor of Philosophy

Dissertation: GENERALIZATION OF A MULTICOMPONENT MIXED-BED ION EXCHANGE MODEL WITH MORE COMPLEX WATER CHEMISTRY

Major Field: Chemical Engineering

Biographical:

Personal Data: Born in Ganzhou, Jiangxi, People's Republic of China (PRC), August 9, 1963, the son of Yuandong Liu and Shilan Zhong.

Education: Graduated from Ganzhou No. 4 High School, Ganzhou, Jiangxi, PRC, in July 1981; received Bachelor of Science degree in Chemical Engineering from Dalian University of Technology, Dalian, China in July 1985; received Master of Science in Chemical Engineering from Dalian University of Technology, Dalian, China in July 1988; completed requirements for the Doctor of Philosophy Degree in Chemical Engineering at Oklahoma State University in December, 1998.

Professional Experience: Employed as a research engineer at the Research Institute of Shanghai Petrochemical Corporation, 1988-1989; employed as an assistant professor at Shanghai College of Petrochemical Technology, 1989-1994; employed as a research associate, School of Chemical Engineering, Oklahoma State University, August 1994 to May 1998.

Professional Organizations: AICHE, Lambda Phi Epsilon.