Chemical Enhancement of Carbon Dioxide Transfer across the Air-Sea Interface

by

Klaus Keller

cand. Ing.
Technische Universität Berlin
March 1991

Submitted to the Department of Civil and Environmental Engineering in Partial

Fulfillment of the Requirements for the Degree of

Master of Science in Civil and Environmental Engineering
at the
Massachusetts Institute of Technology
September 1994

Signature of Author ____

Department of Civil and Environmental Engineering
August 4, 1994

Certified by ____

François M.M. Morel
Professor, Civil and Environmental Engineering
Thesis Supervisor

Accepted by ____

Joseph M. Sussman
Chairman, Department Graduate Committee

Chemical Enhancement of Carbon Dioxide Transfer across the Air-Sea Interface

by

Klaus Keller

Submitted to the Department of Civil and Environmental Engineering
on August 4, 1994 in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Civil and Environmental Engineering

Abstract

The question of chemically enhanced transport of carbon dioxide across the air-sea interface was reexamined using a general numerical model. The model shows that the enhancement factors are greater for a film penetration model than for a laminar film model. The results are in good agreement with independently determined experimental values. The model predicts higher enhancement factors than an earlier numerical model by *Quinn and Otto* (1971). This difference can be attributed to several factors: the consideration of changing pCO_2 and pH of the water bulk phase, the solution of the entire alkalinity expression, and the incorporation of the hydroxyl pathway for the hydration reaction. Calculations of the local effects of carbonic anhydrase in seawater show that the additional enhancement caused by the enzyme is less then 2 %.

Previous calculations with annually and globally averaged input parameters have lead to underestimations of the real effect of chemical enhancement because the relationship between the input parameters and the enhancement factor (EF) is highly nonlinear as well as asymmetrical and the input parameters are intercorrelated. Under absorbing conditions, the undersaturation with respect to CO_2 causes a higher pH. At higher pH the hydroxyl pathway and therefore chemical enhancement becomes more important. The problem is asymmetrical because the oceans can 'breathe in' more easily than they can 'breathe out'. The relatively small net influx of CO_2 is the result of much larger in- and out-fluxes. This amplifies the effect of chemical enhancement. As a result, the overall global effect of chemical enhancement can be bigger than the maximum local enhancement factor.

Assuming typically encountered wind speeds and air-sea CO_2 gradients for the Atlantic region, the actual effect of chemical enhancement could be as big as 15 % on a regional scale and up to 27 % on a basin scale. The higher enhancement for absorption conditions compared to desorption causes an average supersaturation of the oceans. The apparent disagreement between ^{14}C derived air-sea gas exchange rates and rates based on radon measurements can be explained by chemical reactions. Chemical enhancement cannot be neglected in modeling the fate of CO_2.

Thesis Supervisor: Dr. François M.M. Morel
Title: Professor of Civil and Environmental Engineering

Acknowledgments

After one great year in the Parsons Lab, I want to express my gratitude for the extraordinary hospitality of François Morel and his group. Their friendship helped me settle into a more than academically special environment.

Only a few of the people who helped me during the last year fit on one page. François Morel introduced me to this field, showed me new ideas, ways of thinking and helped me with his persistence to find a solution. He is much more than an advisor to me. Dianne Newman, John Reinfelder, Brian Fairchild and Mick Follows refined my humble English in this thesis. The discussions with them clarified the topic and gave me new ideas. Finally I want to thank my parents and Meltem for their encouragement, support and patience.

The author was supported by the Technische Universität Berlin and the Studienstiftung des Deutschen Volkes. Additional support from NSF and ONR is gratefully acknowledged.

Table of Contents:

List of Figures:

I. Introduction

Modeling the global cycle of carbon dioxide (CO_2) is becoming increasingly important due to the threat of global warming. The oceans play the biggest role in the geochemical cycle of CO_2, where the majority of anthropogenic CO_2 is finally absorbed over geological time scales (*Baes* (1982)). Yet to be resolved are two important questions: How fast can the oceans respond to the increasing atmospheric CO_2 concentrations? And how much are they absorbing in the end?

In recent years, scientists have approached this problem by incorporating more and more features (such as biological activity in the oceans or refined resolution of data retrieved by satellite measurements) into global carbon cycle models. But the problem has yet to be solved satisfactorily. Besides a missing sink of approximately 1.8 Gt of carbon per year in the global budget (*Winn* (1994)), the CO_2 exchange rate between the atmosphere and the oceans seems to be considerably higher when determined by carbon species than when calculated by the unreactive gas radon (*Broecker et al* (1986)).

Three approaches are commonly taken to determine the average oceanic CO_2 exchange rate (*Broecker and Peng* (1984)). The natural ^{14}C method predicts a rate of 21 moles*m^{-2}*$year^{-1}$. The artificial ^{14}C method deduces a CO_2 exchange rate of 23 plus or minus 5 moles*m^{-2}*$year^{-1}$. The radon method finally yields a globally averaged exchange rate of about 16 moles*m^{-2}*$year^{-1}$. Each of these predictive methods must wrestle with the inherent complexity of global air-sea gas exchange. Besides the question of chemical enhancement, other factors such as the influence of wind speed, different diffusivities, cooling of the surface due to evaporation and injection of bubbles at higher wind speeds still remain only partially quantified (*Broecker and Peng* (1984)).

Peng et al (1979) confirmed that the globally averaged mass transfer coefficient derived from carbon species is a factor of 1.2 to 1.4 bigger than values derived from the radon method. The possibility that this enhanced absorption of CO_2 could be due to chemical enhancement without the existence of a catalyst like carbonic anhydrase (CA) was denied by the authors based on the studies of *Bolin* (1960) and *Hoover and Berkshire* (1969). This thesis reexamines the question whether this effect could be explained by chemical enhancement and if so, under what conditions. First, previously used models are recalculated and extended; second, the needed concentration of carbonic anhydrase to explain the global enhancement is determined; and finally, the possible effects of insufficient data resolution and the resulting averaging over large space and time domains is estimated.

II. Conceptual and mathematical model

A. Physical mass transfer

The transfer of the anthropogenic CO_2 spike into the oceans is a transient problem. To answer the question of how fast CO_2 is absorbed by the oceans, mass transfer rates between the atmosphere and the water are relevant. The commonly used equation describing the flux F of CO_2 across the air-sea interface has the form:

$$F = EF * k_l * K_{\mathrm{Henry}} * \Delta pCO_2 \qquad (2.1)$$

with:

$$\Delta pCO_2 = pCO_{2,water} - pCO_{2,air} \tag{2.2}$$

The subscript $_l$ indicates that almost all resistance to mass transfer is in the liquid phase. Various transport models are distinguished by the methods used to evaluate the mass transfer coefficient k_l, which is proportional to the diffusivity constant to some power n.

$$k_l \propto D^n \tag{2.3}$$

In his laminar layer model, *Whitman* (1923) pictures the process as governed by a layer in which the rate of molecular diffusion limits transport (Figure 1). The adjacent phases are considered to be well mixed and the equation for the mass transfer coefficient becomes:

$$k_l = \frac{D}{z} \tag{2.4}$$

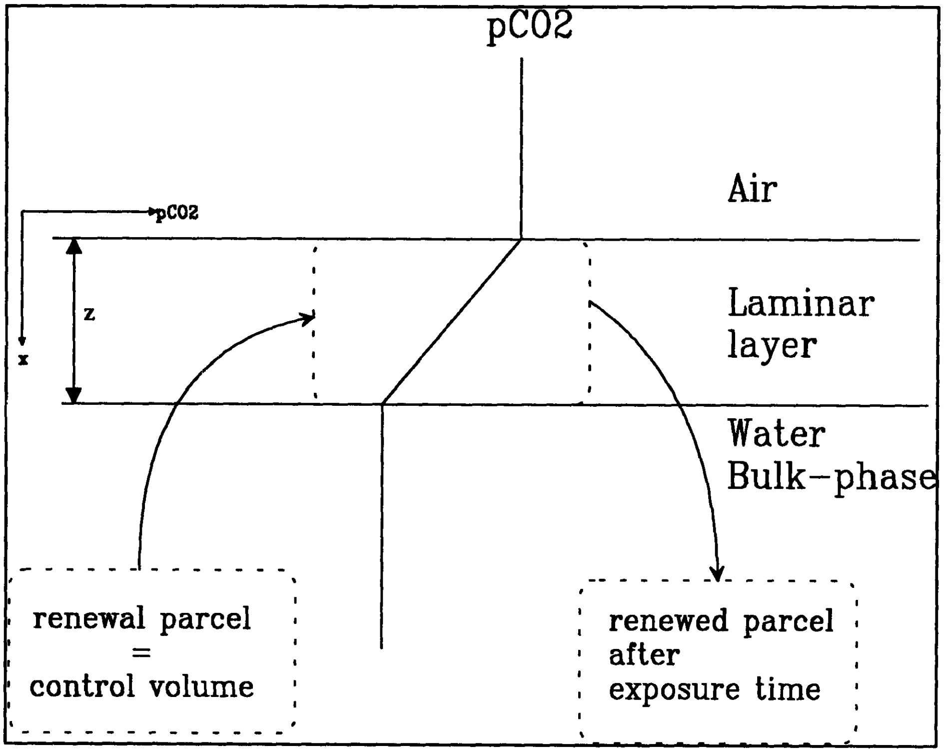

Figure 1: Conceptual picture of physical mass transfer

The mass transfer coefficient is then directly proportional to the molecular diffusivity of the gas. Although the real physical conditions at the air-sea interface are generally different from this picture, its conceptual power and mathematical simplicity cause the laminar film model to be widely popular.

Danckwerts (1951) changed the theoretical framework by assuming that an initially well mixed bulk phase is subject to absorption and is mixed by a turbulent eddy with a surface renewal rate of s. The mass transfer coefficient for the surface renewal model is then

$$k_l = \sqrt{Ds} \qquad (2.5)$$

with n equal to 0.5.

Roberts and Dändliker (1983) empirically determined the value of n to be around 0.6 for turbulent conditions; *Holmen and Liss* (1984) gave a value of approximately 0.57. The variation in n can be accounted for by the film penetration model of *Dobbins* (1955) which combines the film and surface renewal models. A finite laminar film is mixed after a time exposure of θ which results in:

$$k_l = \frac{D}{z}\left(1 + \frac{z^2}{3D\theta} - \frac{2z^2}{\pi D\theta}\sum_{m=1}^{\infty}\frac{1}{m^2}\exp\left(\frac{-\pi^2 m^2 D\theta}{z^2}\right)\right) \qquad (2.6)$$

where n lies in between 0.5 and 1 depending on how turbulent the conditions at the interface are. At high layer thicknesses and short exposure times, this model behaves like a surface renewal model, whereas in the opposite case, the model approaches the laminar film model.

B. Chemical reactions

The model of water chemistry is primarily based on water, carbon, and borate species. The equations describing the system are given in Appendix C.

All the acid-base reactions - except the hydration of CO_2 - can be treated as pseudo-equilibrium reactions at the time scales involved. Three pathways lead to the hydration or dehydration of CO_2: reaction with water, with the hydroxyl (OH^-) ion, and the reaction catalyzed by the enzyme carbonic anhydrase.

The kinetic rate expression for the carbonic anhydrase pathway is quite complex and treated in the literature in many different ways (*Lindskog et al* (1984); *Otto* (1971); *Lindskog* (1980)). I have assumed Michaelis-Menten kinetics with no equilibrium shift due to carbonic anhydrase addition, no limitation by protons in the dehydration direction, and equal half saturation constants for the hydration and dehydration direction. The hydration rate constant is a function of the ionization of the enzyme with an apparent $pK_{a,CA}$ value. Based on these assumptions, the dehydration rate constant is a direct function of the hydration rate constant.

C. Enhanced transport

Because reaction-diffusion systems have been described in the literature at length by many others (*Danckwerts* (1970); *Astarita* (1983); *Westerterp and Wijngaarden* (1992)) only a short outline is presented here.

If a molecule is able to react during its transport, the product can also be transported and an enhanced flux results. This effect is incorporated in the flux expression by the enhancement factor, defined as

$$EF = \frac{k_{l,\text{with reaction}}}{k_{l,\text{without reaction}}} \tag{2.7}$$

The relative magnitude of the diffusion time scale compared to the reaction time scale determines the degree of enhancement. This is expressed in the dimensionless ratio of these time scales, known as the Damkohler number, here defined for a first-order reaction (*Scharzenbach et al* (1993); *Zlokarnik* (1990)):

$$Da = \frac{D * k_{\text{obs}}}{k^2_{1,\text{without reaction}}} \tag{2.8}$$

At very high Damkohler numbers, the reaction is much faster than diffusion. Chemical equilibrium can be assumed everywhere, leading to equilibrium enhancement. At very low Damkohler numbers, diffusion is very fast compared to chemical reactions and chemical enhancement is negligible.

The mass balance of each species with concentration-independent diffusivities links reaction and diffusion processes to one another and yields the partial differential equation for the concentration c_i of each species:

$$\frac{\partial c_i}{\partial t} = D_i \frac{\partial^2 c_i}{\partial x^2} - k_{obs,i} * c_i \tag{2.9}$$

The indices i=1..3 denote the species carbon dioxide, bicarbonate, and carbonate, respectively. For the laminar film model, only the steady state solution is of interest, independent from the initial conditions. The film penetration model must be solved for a time history, starting from well-mixed initial conditions. The flux due to electrical potential differences can be neglected if the charged species have the same diffusivities. This is equivalent to meeting the electroneutrality condition at every point. These coupled nonlinear partial differential equations are subject to the following boundary and initial conditions.

At the air-sea interface the ionic species cannot partition into the air, which is expressed by the no-flux boundary conditions :

$$\frac{\partial [HCO_3^-]}{\partial x} = 0 \,\Big|_{x=0} \tag{2.10}$$

and

$$\frac{\partial [CO_3^{2-}]}{\partial x} = 0 \,\Big|_{x=0} \tag{2.11}$$

CO_2 at the interface is assumed to be always saturated with respect to the overlying atmosphere.

$$[CO_2] \,\Big|_{x=0} = K_{henry} * pCO_{2,\,\mathrm{air}} \tag{2.12}$$

At the lower boundary, (the bulk water phase) chemical equilibrium is assumed and all concentrations are fixed. The chemical composition of the bulk phase is expressed by the saturation ratio of total inorganic carbon (rsc_t) with

$$rsc_t = \frac{c_{\mathrm{t,saturated\ at\ pCO}_{2,\mathrm{air}}}}{c_{\mathrm{t,real}}} \tag{2.13}$$

Alk, B_t and c_t determine together with the appropriate constants the pH and speciation.

III. Analytical approximations and numerical solutions

To solve the system of partial differential equations numerically, I applied an explicit finite difference algorithm with time-splitting for diffusion and reaction. The pH of the solution was calculated by applying a bi-section root finder (*Press et al* (1990)) to the alkalinity expression. The space and time discretization was increased until the calculated EFs were constant.

The model is able to calculate the EF over the entire range of Damkohler numbers. As the Damkohler number approaches zero and infinity, the EFs approach their theoretical limits asymptotically.

In the laminar film model, steady state was reached when the variance of the total carbon fluxes through each layer was below 0.2 % and the change in EF was negligible. The EF with this mass transfer model was evaluated as the ratio of the average of the calculated total carbon flux at each layer to the theoretical diffusional CO_2 flux, defined by the boundary conditions. The routine converged from different initial conditions to a constant EF.

There are several analytical approximations available for the laminar film model. The analytical approximations given by *Smith* (1985) with

$$EF = r * z * \frac{\cosh(r * z)}{\sinh(r * z)} \qquad (3.1)$$

where

$$r = \sqrt{\frac{k_{co2} * \{H\} + k_{14} * K_W}{D_{CO_2} * \{H\}}} \tag{3.2}$$

is compared with the numerically obtained values and *Hoover and Berkshire*'s (1969) approximation in Figure 2.

Both analytical approximations assume that the pH in the laminar layer is constant and equal to that of the bulk phase and consider the hydration as well as the hydroxylation pathway. *Bolin*'s (1960) approximation neglects the hydroxylation pathway, and therefore fails to represent an important feature of the system. Because *Smith* (1985) treats the CO_2 reaction as a first-order irreversible reaction, his solution does not reflect the upper limit of the EF that would be achieved at equilibrium enhancement conditions (i.e., at high Damkohler numbers). His assumptions hold as long as the EF is not too high. Under oceanic conditions, the EF is always small compared to the EF under equilibrium enhancement. The maximum error introduced by using *Smith*'s (1985) approximation instead of the numerical solution is below 3 % at a laminar layer thickness of 200 µm and a pH of 8.38. The use of *Smith*'s (1985) approximation is therefore appropriate for estimating chemical enhancement of air-sea CO_2 transport if one assumes the validity of the laminar film model.

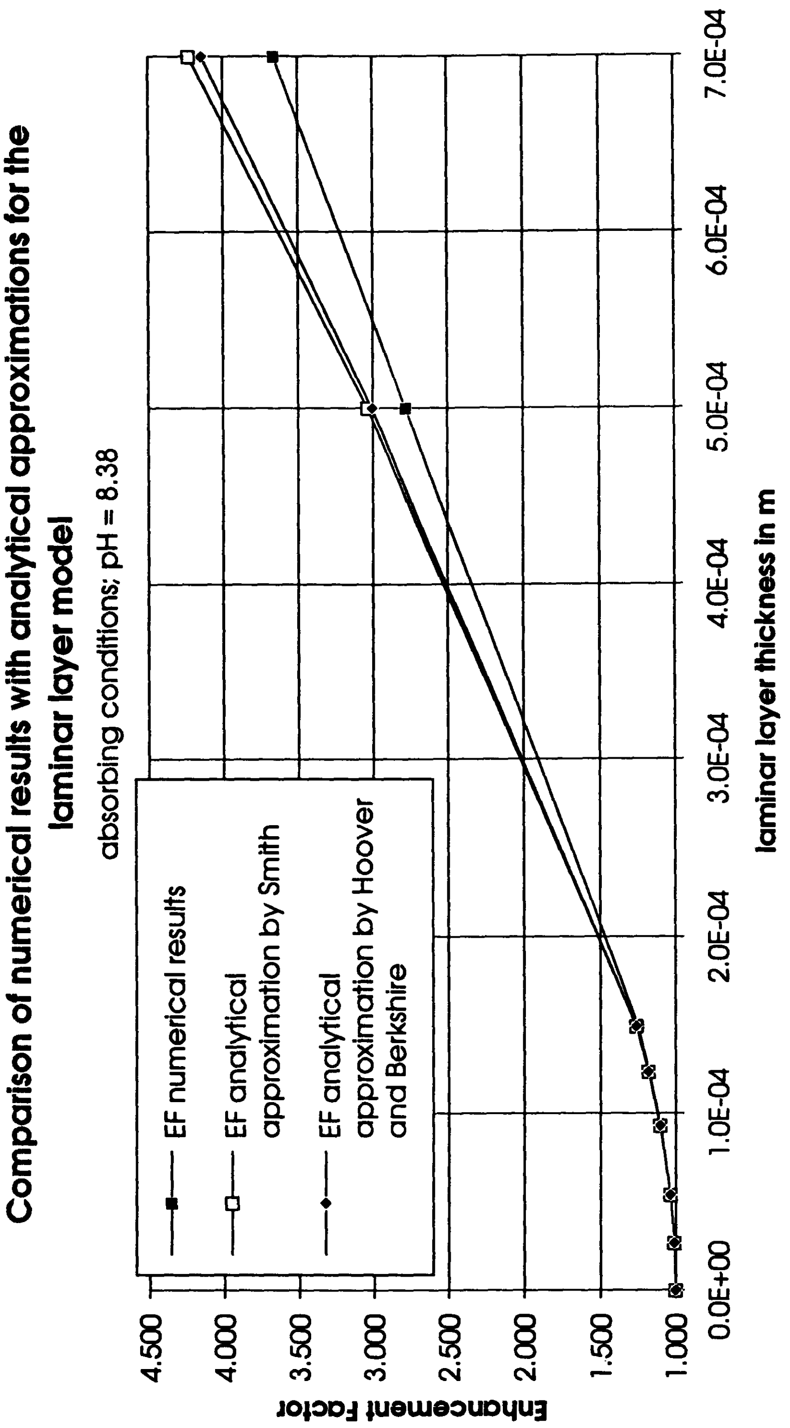

Figure 2: Comparison of numerical results with analytical approximations for the laminar film model

The computation of the EF with the film penetration model is numerically more complex. The evaluation of the fluxes and the EF in the analytical expressions for a film penetration model is done by evaluating the flux expression at the interface. When integrated with respect to time, this yields the amount of carbon absorbed per unit area and k_l is determined by

$$k_l = \frac{\int_{t=0}^{\theta} \left(D_{CO_2} \left. \frac{\partial [CO_2]}{\partial x} \right|_{x=0} \right) dt}{\theta \left([CO_2] \big|_{\text{bulk phase}} - [CO_2] \big|_{\text{interface}} \right)} \tag{3.3}$$

This equation, however, is very inaccurate if evaluated numerically. The reason is that at the interface the amount of reaction is at a maximum. To adequately represent this change numerically, very high time and space discretizations are required. A mass balance of c_t yields the following scheme which proves to be numerically more accurate.

$$k_l = \frac{\sum_{i=1}^{3} \int_{x=0}^{x=z} \left(c_{i,t=0} - c_{i,t=\theta} \right) dx - \int_{t=0}^{t=\theta} \left(\sum_{i=1}^{3} D_i \left. \frac{\partial c_i}{\partial x} \right|_{x=z} \right) dt}{\theta \left([CO_2] \big|_{\text{bulk phase}} - [CO_2] \big|_{\text{interface}} \right)} \tag{3.4}$$

The film penetration model was evaluated at a high degree of turbulence (i.e., at a high ratio of laminar layer thicknesses to exposure times). It represents the behavior of a surface renewal model. To calculate the EF for the film penetration model, the k_l values calculated with and without chemical reaction starting from the same initial conditions were compared.

IV. Results

A. Comparison with previous numerical results

Improvements over the model of *Quinn and Otto* (1971) are: the solution of the entire alkalinity expression, the consideration of the OH^- pathway, and the inclusion of variable bulk water pH. *Quinn and Otto* (1971) incorporated the OH^- pathway in their governing equations but neglected it in the actual calculations. This is only of minor importance if the pH of the bulk phase is held constant at 8, as it is in their study, but becomes an important factor when the bulk phase pH is variable. I have changed the $pCO_{2,water}$ rather than $pCO_{2,air}$. This is a more realistic approach, given that the atmospheric partial pressure of CO_2 can be assumed to be practically constant, whereas the seawater pCO_2 can vary (for example due to biological activity) over short time scales on the order of days (*Robertson et al* (1993)).

Actual measured pH values in the oceans range from 7.8 to 8.5 (*Simpson and Zirino* (1980)). At an average pH of 8.3, the CO_2 turnover in seawater due to the OH^- pathway is around 50 %. By neglecting this, one would underestimate the EF and fail to represent the nonlinearity of the EF with respect to ΔpCO_2. As we shall see, such nonlinearity, however, is a critical property of the system and must be taken into account when averaging.

Including the full alkalinity expression, the OH^- pathway, and the variable bulk phase pH significantly effects the predicted EF. While the model of *Quinn and Otto* (1971) predicts an equilibrium EF of 2.67 at ΔpCO_2=-227 μatm, a pH of 8.0 and T=20 °C, the new model calculates a pH of 8.6 and an equilibrium EF of 18. In the kinetic regime, the difference between EFs estimated by the two models increases with increasing Damkohler numbers. At a laminar layer thickness of 182 µm and the above specified boundary conditions this difference between the EFs is still 20 %.

B. Numerical results

The numerical results for the laminar film and the film penetration model at two different pH values are shown in Figure 3.

Danckwerts (1970) suggested that in most cases the film model would lead to almost the same prediction for the EF as the surface renewal model. Calculations with the two different models, however, predict different EFs. So why are the predicted EFs higher for the film penetration model than for the laminar film model ?

There are two reasons. First, *Danckwerts*' (1970) point is true as long as the diffusivities of all transported molecules are the same. In the case of CO_2 absorption, however, bicarbonate diffusivity is roughly only 50 % of that of CO_2. The equilibrium EF under these conditions is always higher for a surface renewal model than for a laminar film model, because, as discussed earlier, the ratio of diffusivities is weighted by a power of 0.5 as opposed to 1.

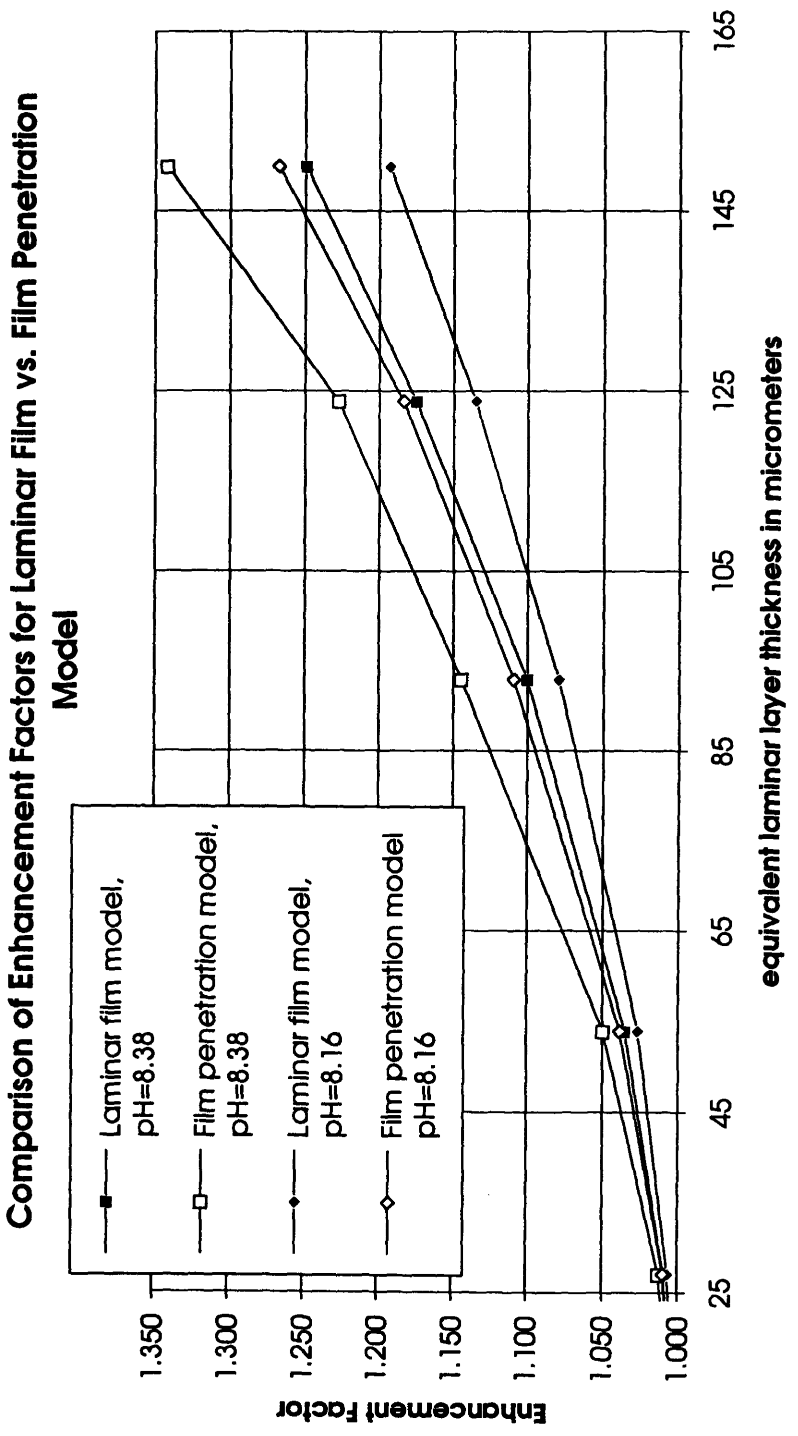

Figure 3: EFs calculated with laminar film and film penetration model as a function of the pH of the bulk phase

Second, even with equal diffusivities, the models predict slightly different EFs at intermediate Damkohler numbers. Chemical reactions are more important in surface renewal models than in laminar film models. The reason is that directly after a mixing event, the species concentrations at the interface are farthest from equilibrium and accordingly the reaction rate is at a maximum. In a laminar film model only the steady state solution is of interest and the "spike" in reaction rate that accompanies mixing is not considered. Predicted EFs at intermediate Damkohler numbers are therefore higher for a film penetration model.

This effect was illustrated by *Glasscock and Rochelle* (1989) in a numerical study with bimolecular reversible reactions and different diffusivities. They showed that the difference of the EFs has a maximum value at EFs around 1.5, which falls within the relevant range for CO_2 transfer across the air-sea interface. The additional reaction of bicarbonate to carbonate included here, but not considered by *Glasscock and Rochelle* (1989), further increases the reaction rate because the back-reaction of bicarbonate to CO_2 becomes less efficient with decreasing bicarbonate concentrations.

C. Comparison with measured data

Measurements of EFs for CO_2 transfer across the air-water interface have been reported by *Emerson* (1975), *Hoover and Berkshire* (1969), *Liss* (1983), *Broecker and Peng* (1973), *Goldman and Dennet* (1983), and *Berger and Libby* (1969). The last three reported measurements were for water at pH values typical for seawater. In all these papers important input parameters such as laminar layer thicknesses or pH values were either missing or variable over the course of the experiment. As a result, the measured EFs have a large degree of error. For example, *Emerson*'s (1975) measurement of one EF ranges from 4.1 to 6.5 with an error of plus or minus 23 percent. Additionally, the range of input parameters (such as kinetic rate constants) is so large that it is relatively easy for both numerical and analytical models to reproduce the measured EFs. *Smith*'s (1985) analytical approximation, as well as the presented numerical model, is able to reproduce the range of *Emerson*'s (1975) experimental values. For the same reason, it is not possible to discriminate against any of the existing mass transfer models on the basis of the cited experimental EF values.

D. Effects of carbonic anhydrase

Berger and Libby (1969) first hypothesized that the enzyme carbonic anhydrase could cause the globally observed enhancement of CO_2 transfer across the air-sea boundary. By adding 0.5 mg/l (which equals about $1.7*10^{-8}$ M) to aerated steel drums, they found a 30 fold increase in the apparent reactivity. Their experimental setup suggests rather high laminar layer thicknesses. *Quinn and Otto* (1971) calculated the carbonic anhydrase concentration needed to reduce the film thickness at which reactions become important to be 10^{-7} M. The numerical model predicts an EF of 1.6 at z=50 μm at this carbonic anhydrase concentration. This concentration is in excess of that necessary to cause the observed global enhancement. *Goldmann and Dennet* (1983) tried to reproduce the *Berger and Libby* (1969) results using a hydrodynamically more defined stirred cell with artificial and natural seawater. They measured EFs of 1.6 and 2 at 0.5 and 2 mg/l carbonic anhydrase respectively, but only at moderate degree of turbulence, to which they assigned a laminar layer thickness of 450 μm. Based on these measurements and the results of inhibitor studies, they concluded that carbonic anhydrase does not effect enhancement of CO_2 transfer in seawater.

These calculations and experiments show that carbonic anhydrase can cause considerable enhancement if the concentration is high enough. The maximum carbonic anhydrase concentration at the very surface of the oceans is constrained by the concentrations of zinc and perhaps cadmium there, because every active site of carbonic anhydrase requires a zinc atom. A substitution of zinc by cadmium in CA is possible. The upper limit of typically encountered total zinc and cadmium concentrations in oceanic surface waters is lower than $2*10^{-9}$ mol/kg (*Broecker and Peng* (1982)).

Surface layer partitioning could increase the carbonic anhydrase concentration in the microlayer. Measurements by *Duce et al* (1972) show an average microlayer enrichment factor for PCBs of 18.5. Assuming carbonic anhydrase partitions like PCBs and a conservative bulk surface water zink and cadmium concentration of 10^{-9} M, the carbonic anhydrase concentration in this microlayer could be as high as about $2*10^{-8}$ M. Using the conservative estimation for the carbonic anhydrase concentration and a globally averaged laminar layer thickness of 50 μm, the EF is less than 1.05 (Figure 4). A change to the film penetration model does not alter the situation significantly (Figure 5). Only at the maximum estimated carbonic anhydrase concentration and a high equivalent laminar layer thickness of $z = 65$ μm is the calculated EF of 1.24 close to the apparent global enhancement.

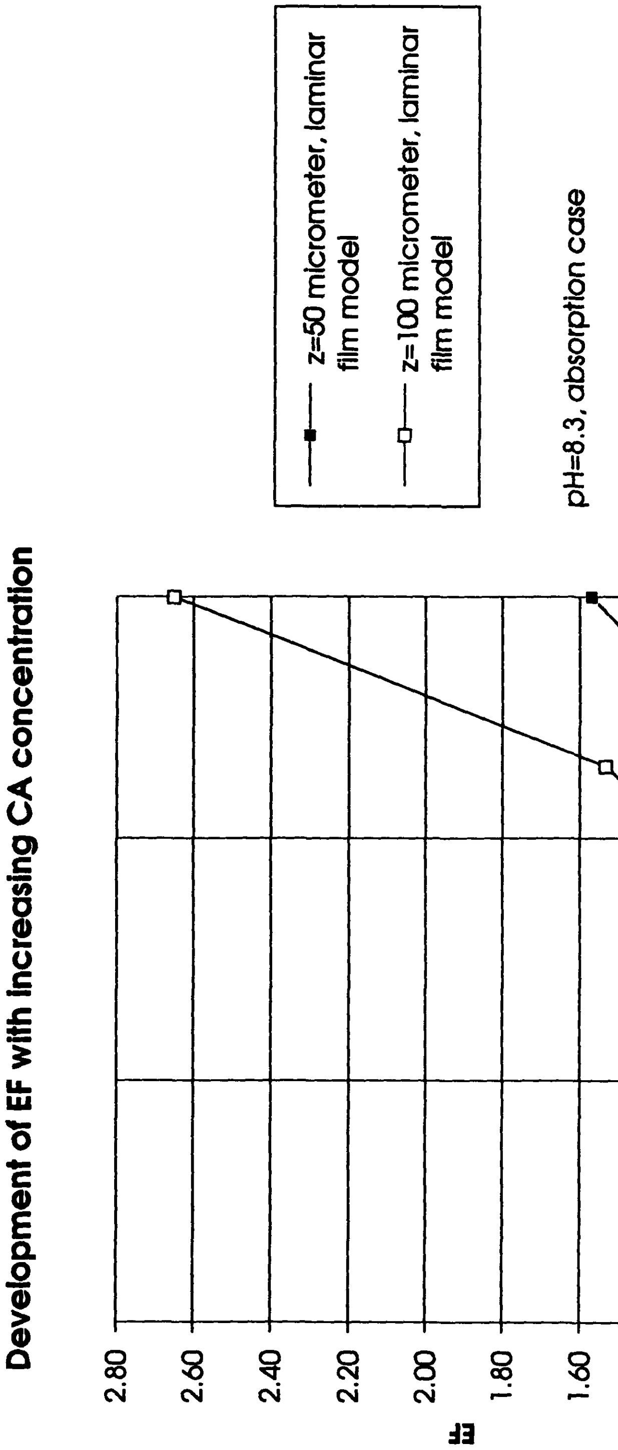

Figure 4: Effects of changing carbonic anhydrase concentration on the EF in a laminar film model at different laminar layer thicknesses

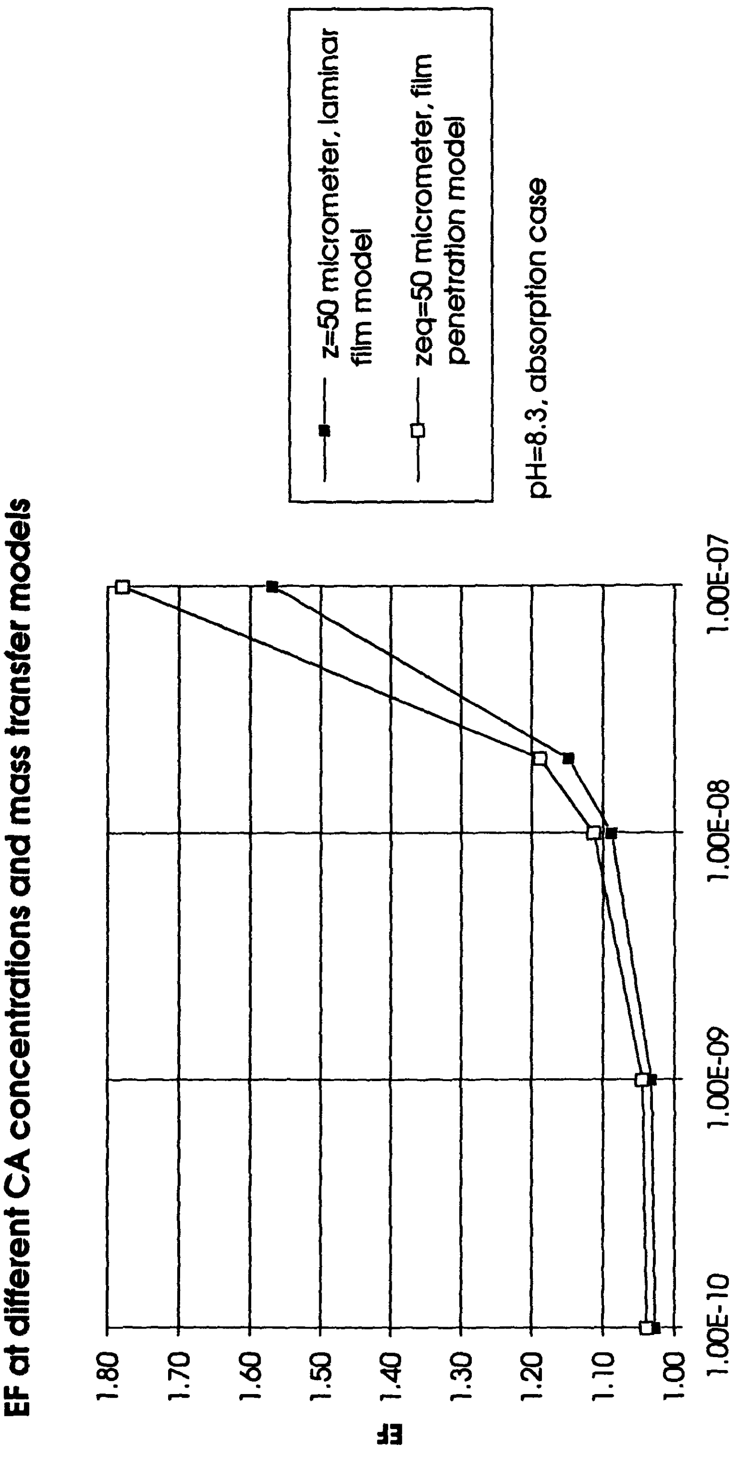

Figure 5: EFs as a function of carbonic anhydrase concentration evaluated with different mass transfer models

E. Effect of averaging on the effective EF

The globally averaged laminar layer thickness determined by the radon method is 50 plus or minus 30 μm (*Broecker and Peng* (1973)). Given this layer thickness, a pH of 8.3 and a temperature of 25 °C, the local EF is only 1.03. This calculation explains why chemical enhancement was thought to be negligible. The underlying assumption is that the global effect of chemical enhancement is the same as the effect that would result in a hypothetical situation where the constant local wind speed, temperature and pH would correspond to their globally averaged values. This hypothetical observed local EF with averaged input parameters, however, is not identical to the global effect expressed by the effective EF:

$$EF_{eff} = \frac{\int\limits_{space}\int\limits_{time} \Delta pCO_2 * k_l * K_{Henry} * EF}{\int\limits_{space}\int\limits_{time} \Delta pCO_2 * k_l * K_{Henry}} \qquad (4.1)$$

If the local EF is not changing as a function of the input parameters or if the conditions are constant, the effective EF simplifies to the local EF.

Averaging all input parameters neglects four major properties of the system: the correlation between the input parameters, asymmetric relationships between input parameters and the local EF, the distributions of the input parameters, and the effects of averaging input parameters in nonlinear functions. Because of these properties of the system, calculated effective EFs are a function of the temporal and spatial discretization of the input data. The evaluation of the averaging effect is complicated by the weighting factors (ΔpCO_2, k_l and K_{Henry}) in front of the term EF. These weighting factors as well as the EF itself are a function of the chemical and physical boundary conditions of the system.

I will show that situations exist in which the effective EF is much greater than the maximum local EF. To illustrate the effect of averaging, I first impose distributed values for one parameter at a time, considering wind speed and ΔpCO_2 values as the most important parameters. Finally, the case of changing wind speed and ΔpCO_2 simultaneously is examined. Measured values were used to calculate the amount of CO_2 absorbed without any averaging. When empirical parameter values were not available with sufficient resolution, their spatial and temporal distributions were modeled. Because of its numerical convenience, the analytical approximation for the laminar film model was used to calculate the EF. Relevant equations and assumptions are given in Appendix D.

Averaging wind speed values underpredicts the effective EF at typical oceanic conditions. The effect of averaging the wind speeds on the effective EF is a function of both the average wind speed and the distribution type. The EF at an effective laminar layer thickness of 59 µm, constant wind speed, and $\Delta pCO_2 = -100$ µatm is 1.04. To show the influence of distribution at one fixed wind speed, I changed from a constant wind to a uniform distribution of two values. The effective EF can increase up to 1.15 (Figure 6). Clearly this high maximum EF is an upper limit for the effect of wind averaging at these conditions, because at high wind speeds more and more CO_2 is absorbed at almost no enhancement and at low wind speeds the EF is limited by the assumed maximum laminar layer thickness of 700 µm. Hourly measured wind speed data

(*Trowbridge* (1994)) with the same chemical conditions and effective laminar layer thickness increased the effective EF only to 1.06.

The influence of distributed wind speeds changes with the average wind speed. Therefore I modeled the effect of wind averaging as a function of the effective laminar layer thickness (Figure 7). Again, the enhancement is, in the typical oceanic range of the laminar layer thickness, higher for a distributed wind. However, this effect cannot account for the global EF alone.

The additional effect of ΔpCO_2 averaging may explain this phenomenon. If the resolution of the ΔpCO_2 values is insufficient, the fact that huge in- and out-fluxes at different physical and chemical conditions yield a relatively small net flux is neglected. The invasion rate based on natural ^{14}C measurements is around 20 moles of carbon per m^2 and year. At an oceanic surface of $3.62*10^{14}$ m^2 and an atomic weight of 12 g per mol of carbon, this equals a steady state in- and out-flux of 87 Gt of carbon per year (*Broecker and Peng* (1993)). These huge in- and out-fluxes dominate the estimated net absorption rate of 2 Gt of carbon per year (*Winn* (1994)).

The number of passages (np), defined by

$$np = \left| \frac{\int_{space} \int_{time} \left| F_{enhanced} \right|}{\int_{space} \int_{time} F_{enhanced}} \right| \tag{4.2}$$

is one way to quantify this effect. It can be thought of as the average number of times a molecule has to pass through the interface until it stays at one side. The number of passages based on the above values for the invasion rate and net absorption is 87.

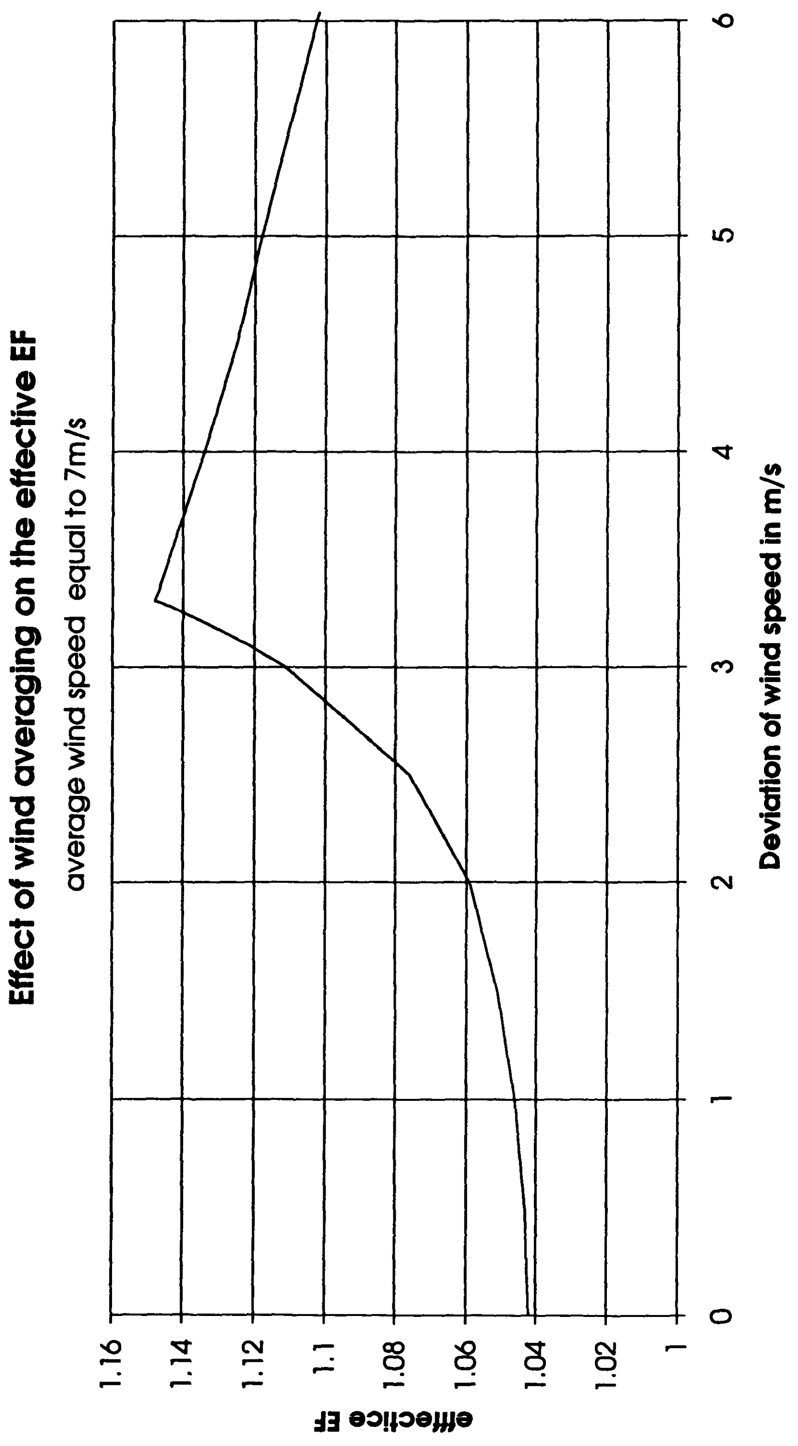

Figure 6: Effective EFs with a changing wind speed distribution

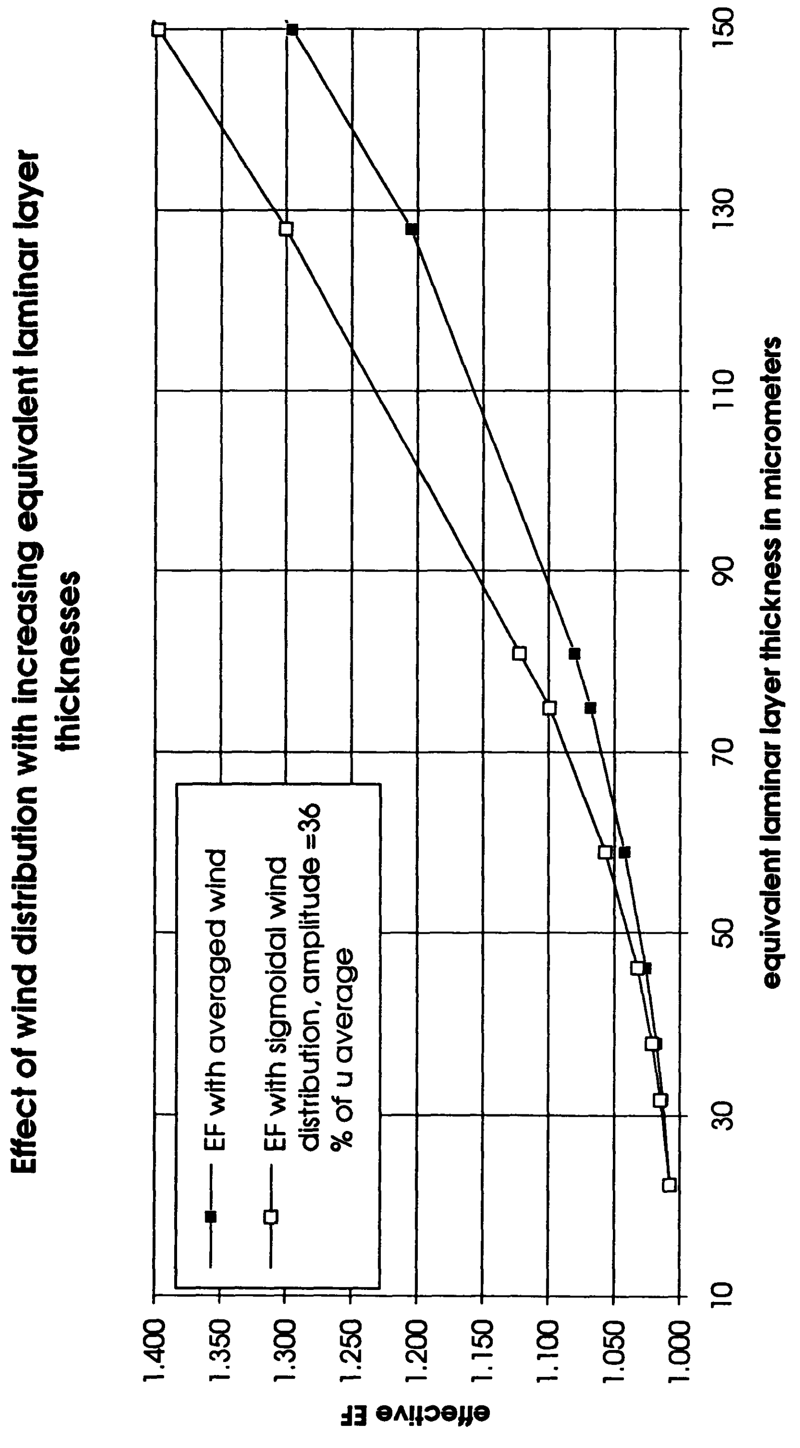

Figure 7: Effective EFs with constant wind distribution and varying effective laminar layer thickness

Seasonal cycling (*Codispoti et al* (1982)) and the geographic distribution of ΔpCO_2 values cause these huge in- and out-fluxes. Global maps of annually averaged ΔpCO_2 values range from minus 120 μatm to plus 140 μatm (*Keeling* (1968)). The global average is estimated to be around -8 plus or minus 8 μatm (*Broecker et al* (1979)). Besides these variations, diurnal cycles and small scale variations of ΔpCO_2 values have been observed. *Simpson and Zirino* (1980) reported spatial variations with a length scale down to 1 km and *Robertson et al* (1993) measured diurnal cycles with daily changes of 20 μatm.

The available resolution for ΔpCO_2 data used for global CO_2 flux estimations is very poor. Commonly used maps have a spatial resolution of about 5 degrees and are calculated with annual averages from real data sets or sometimes even with theoretical models (*Etcheto et al* (1991)). The maps of ΔpCO_2 that are used neglect the existing fine scale temporal and spatial variations. Calculation with one global average neglects the in- and out-fluxes entirely. Neither the effective EF nor the number of passages depends on the discretization used for the calculation. But if one calculates the effective EF on the basis of discrete values, the result is dependent on the resolution of the input parameters.

The gedanken experiment outlined in Table 1 illustrates the possible effects of neglecting chemical enhancement in this situation. Consider the situation of a two-box surface model, where for the sake of simplicity only the ΔpCO_2 and the pH values are variable.

Table 1: gedanken experiment ΔpCO_2 averaging

Assumptions: z = 65 µm; temperature = 25 ^{O}C; laminar film model; pH/ΔpCO_2 relationship from *Kempe and Pegler* (1991)	absorbing area	desorbing area	sums	averages
ΔpCO_2 in µatm	-55.87	53.75	-2.12	**-1.06**
pH	8.314	8.179		8.25
EF	1.047	1.040		**1.043**
% time*area	50 %	50 %	100 %	50 %
K_l relative ratio	1	1		1
flux enhanced	-29.24	27.95	**-1.29**	
absolute values flux	29.24	27.95	57.19	
flux unenhanced	-27.94	26.88	**-1.06**	

Effective EF	**1.22**
number of passages	**44.4**

The effective EF in this case is much higher than the maximum local EF. This effect cannot be observed if one neglects the OH^- pathway in the EF calculations, nor do highly averaged calculations adequately represent this effect. This amplifying effect is a major feature of the system. If we merely took the net effect of the in- and out-fluxes we would underestimate the cumulative effect. The higher the number of passages and the asymmetry of the EF, the more important are the errors introduced by averaging the input parameters. I calculated the same example with numerical values for the film penetration model because of its higher sensitivity to changes in pH. The effective EF increased to a value of 1.27.

The amplifying effect is caused by the asymmetric behavior of the EF with respect to ΔpCO_2. At absorbing conditions, the undersaturation with respect to CO_2 causes a higher pH. At higher pH the OH^- pathway becomes more important and the EF increases. The problem is asymmetrical because the oceans can 'breathe in' more easily than they can 'breathe out' (Figure 8). The effective EF as a function of $\overline{\Delta pCO_2}$ has three important features: a vertical asymptote at $\overline{\Delta pCO_2}$ equal to zero, negative EF values, and a positive intercept on the $\overline{\Delta pCO_2}$ axis (Figure 9).

The asymptotic behavior is based on the asymmetric properties of the EF. When the term

$$\int\limits_{space}\int\limits_{time} \Delta pCO_2 * k_l * K_{Henry} \qquad (4.3)$$

goes to zero, the effective EF goes to infinity. This is equivalent to a situation in which there would be no flux without enhancement and the nonzero flux is caused by chemical enhancement. Because the effective EF as a correction term is incorporated into the flux expression,

$$F \propto EF_{eff} * \overline{\Delta pCO_2} \qquad (4.4)$$

it has to increase as the weighting factor driving force decreases. To cause the nonzero flux at zero $\overline{\Delta pCO_2}$, the effective EF has to go to infinity.

At negative values of the effective EF, effective flux and averaged driving force have different signs. This means for example that there is a flux into the water, although the liquid is oversaturated on average. This can be explained by the no-flux condition for a system at steady-state, defined by

$$\int\limits_{space}\int\limits_{time} \Delta pCO_2 * k_l * K_{Henry} * EF = 0 \qquad (4.5)$$

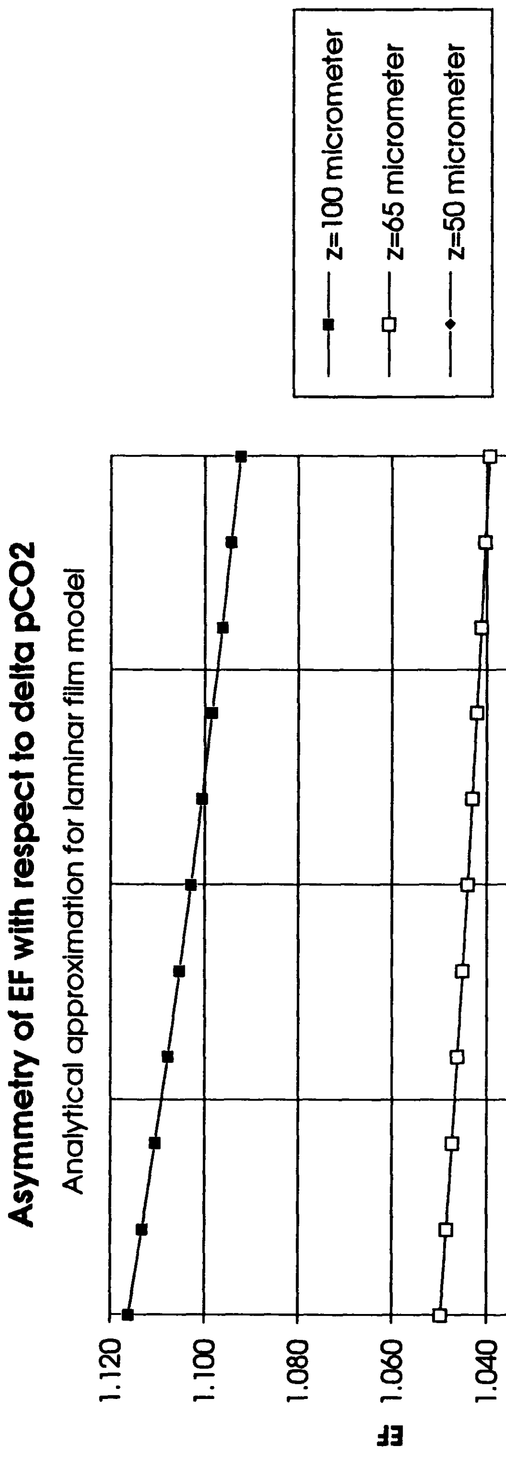

Figure 8: EFs as a function of ΔpCO_2 of the bulk phase at different laminar layer thicknesses

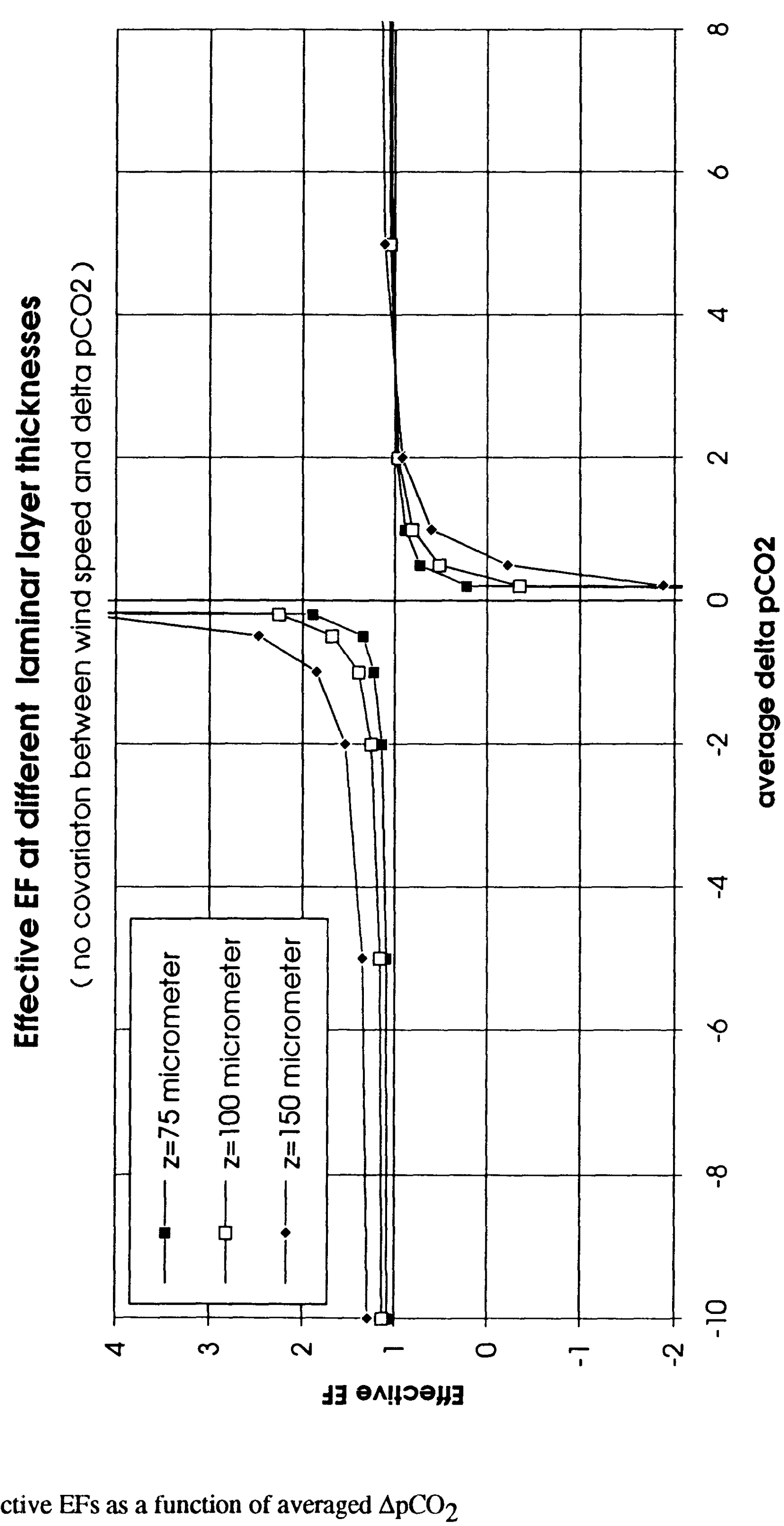

Figure 9: Effective EFs as a function of averaged ΔpCO_2

Because of the asymmetry of the EF, the oceans, which breathe in more easily than they breathe out, have to be slightly oversaturated with CO_2 on the average in order not to gain or lose CO_2. The equilibrium oversaturation is the intercept of the effective EF with the ΔpCO_2 axis. This oversaturation increases with increasing asymmetry of the EF and increasing number of passages.

Oversaturation with oxygen has been observed at high wind speeds by *Wallace and Wirick* (1992). This so called "wind pumping" is attributed to the asymmetrical effect of air entrainment. Breaking waves inject bubbles into deeper regions where, due to the increased hydrostatic pressure, the water becomes oversaturated. According to *Broecker et al* (1986), the oxygen oversaturation of the oceans is about 3 %. Much of this supersaturation is caused by net photosynthesis, acting as an oxygen source term. Supersaturation values for oxygen and CO_2 are different, because of the higher CO_2 solubility, the buffering effects of seawater, and chemical enhancement. *Memery and Merlivat* (1984) showed that the higher the solubility of a gas, the lower the effect of wind pumping. These two reasons suggest that CO_2 supersaturation caused by wind pumping should be negligible. Like the nonlinearity due to chemical enhancement, wind pumping is most important near equilibrium. The major difference between wind pumping and chemical enhancement pumping is that wind pumping is important at high wind speeds and on local scales, whereas chemical enhancement pumping is important at low wind speeds and on large scales.

In the case of more than one variable parameter, the correlation between them becomes important. The values of K_{Henry}, ΔpCO_2 and wind speed are all correlated to each other either by annual cycles, local feedback mechanisms, or geographic relationships.

The correlation between wind speed and ΔpCO_2 is considered here as an example. Because the parameters have annual cycles, the phase shift between them as well as their oscillation periods are important.

Consider a general absorbing condition. With a positive correlation between wind speed and ΔpCO_2 (i.e., high wind speeds at desorbing conditions and low wind speeds at absorbing conditions) the asymmetry of the EF is further increased and the effect of chemical enhancement is amplified. A negative correlation between these parameters decreases or even reverses the EF asymmetry and the effective EF decreases. The effective EF, however, does not increase in every case with a positive correlation. If the effect of increasing k_l with increasing wind and ΔpCO_2 is more important than the effect of chemical enhancement, the effective EF is diminished by the positive correlation. In a steady state situation with fixed net influx (as, for example, the natural ^{14}C case), the general result of chemical enhancement and a positive correlation would be to decrease the necessary averaged driving force.

A prediction of the correlation between wind speed and ΔpCO_2 from first principles is not possible. Several mechanisms link these two parameters, causing either a positive or negative correlation. A positive correlation between wind speed and ΔpCO_2 on a local scale and absorbing conditions is caused by a simple feedback mechanism. At a constant CO_2 sink term (i.e. CO_2 uptake due to biological activity), ΔpCO_2 becomes increasingly negative with lower wind speeds. On an annual basis, *Etcheto et al* (1991) reported a minimum wind speed in March and a maximum in October, roughly in phase with the annual ΔpCO_2 cycle which has a minimum in the spring to summer months (*Taylor et al* (1991); *Harvey* (1966); *Codispoti et al* (1982)). These annual cycles augment the local positive correlation. Two factors that support a negative correlation are: the general negative geographic correlation of higher wind speeds and more negative ΔpCO_2 values at high latitudes (*Etcheto et al* (1991)), and the solubility pump with desorption in the relatively calmer summer months. Although the general geographic correlation is important, it is unlikely that this effect is dominant. First, it is based on annually averaged values. Second, two-thirds of the world's oceans are between 40^o S and 40^o N latitude. Finally, absorbed bomb ^{14}C has maximum values in two symmetric bands located between 30^o and 40^o North and South (*Stuiver* (1980)) which indicates the importance of the tropical and subtropical latitudes. If the solubility pump were driving the ΔpCO_2 cycle, the actual measured annual cycles would have a maximum in the summer. As shown above, this is not the case in most of the reported time series.

The actual correlation coefficient on a global scale cannot be evaluated unless wind speed and ΔpCO_2 data are available in much finer resolution. Based on the reported annual ΔpCO_2 and wind speed cycles, the positive correlating mechanisms seem to be more important and the effect of chemical enhancement is therefore further amplified.

V. Summary and Conclusions

On the basis of theoretical and experimental arguments, previous authors have concluded that chemical enhancement of CO_2 transport across the air-sea interface is negligible despite contrary evidence from global ^{14}C and radon data. I have shown that by dropping some of the previously used simplifying assumptions, chemical reactions indeed can enhance the global flux of CO_2 significantly. Although the calculations reveal somewhat higher local EF, these predictions are primarily based on considering the effects of nonlinearity and poor data resolution in evaluating the global effects.

Averaging wind speed and ΔpCO_2 data drastically underpredicts the effective EF. Considering the multiplying effects of changing wind speed and ΔpCO_2 together with their possible positive correlation, the apparent global EF can be explained. Real conditions in the oceans are far more complex than those represented by the assumed distributions of the parameters. Yet these assumptions are arguably more realistic than past approaches which averaged input parameters.

Unless the uncertainties in the mass transfer models are resolved and the input data determined in finer temporal and spatial resolution, the calculation of a global EF is effectively an "inverse problem", rather than a prediction from first principles. This inverse problem, however, puts an additional constraint on the calibration of global carbon cycles.

Quantifying the ΔpCO_2 distribution in higher resolution is a critical next step for improving the accuracy of future models. Further research on how the spatial and temporal variability of the input parameters (chiefly wind speed and ΔpCO_2) can be described, measured and finally incorporated into global carbon cycle models is of the highest importance.

VI. References

Astarita, G., 1983, Gas treating with chemical solvents, Wiley-Interscience, New York

Baes, F.B., 1982, Effects of carbon chemistry and biology on atmospheric carbon dioxide, in: Carbon dioxide review 1982, Ed.: Clark, W.C., Oxford University Press, New York

Berger, R. and Libby, W.F., 1969, Equilibration of atmospheric carbon dioxide with sea water: Possible enzymatic control of the rate, Science, V. 164, pages 1395-1397

Bolin, B., 1960, On the exchange of carbon dioxide between the atmosphere and the sea, Tellus, V. 12, pages 274-281

Bolin, B., Björkström, A., Holmen, K. and Moore, B., 1983, The simultaneous use of tracers for ocean circulation studies, Tellus, V.35 B, pages 206-236

Broecker, W.S., Ledwell, J.R., Takahashi, T., Weiss, R., Merlivat, L., Memery, L., Peng, T.-H., Jahne, B. and Munnich, K.O., 1986, Isotopic versus micrometerologic ocean CO_2 fluxes: a serious conflict, Journal of Geophysical Research (JGR), V. 91, No. C9, pages 10517-10527

Broecker, W.S. and Peng, T.-H., 1973, Gas exchange rates between air and sea, Tellus, V. 26, No. 1-2, pages 21-35

Broecker, W.S. and Peng, T.-H., 1993, Carbon 13 constraint on fossil fuel CO_2 uptake, Global Biochemical Cycles (GBC), V. 7, No. 3, pages 619-626

Broecker, W.S. and Peng, T.-H., 1984, Gas exchange measurements in natural systems, in: Gas transfer at water surfaces, pages 479-493, Reidel Publishing

Broecker, W.S. and Peng, T.-H., 1982, Tracers in the sea, Lamount Doherty Geolocical Observatory, Columbia University, New York

Broecker, W.S., Peng, T.-H., Ostlund, G. and Stuiver, M., 1985, The distribution of bomb radiocarbon in the ocean, JGR, V. 90, No. C9, pages 6953-6970

Broecker, W.S., Takahashi, T., Simpson, H.J. and Peng, T.-H., 1979, Fate of fossil fuel carbon dioxide and the global carbon budget, Science, V. 206, No. 4417, pages 409-418

Codispoti, L.A., Friederich, G.E., Iverson, R.L. and Hood, D.W., 1982, Temporal changes in the inorganic carbon system of the south-eastern Bering Sea during spring 1980, Nature, V. 296, pages 242-245

Danckwerts, P.V., 1951, Significance of liquid-film coefficients in gas absorption, Industrial and engineering chemistry, V. 43, No. 6, pages 1460-1467

Danckwerts, P.V., 1970, Gas-liquid reactions, Mc. Graw Hill, New York

Dobbins, W.E., 1955, The nature of the oxygen transfer coefficient in aeration systems, in: Biological treatment of sewage and industrial wastes, V. 1, Ed. by: McCabe, Rheinold Publishing Corporation, New York

Duce, R.A., Quinn, J.G., Olney, C.E., Piotrowicz, S.R., Ray, B.J, and Wade, T.L., 1972, Enrichment of heavy metals and organic components in the surface microlayer at Narragansett Bay, Rhode Island, Science, V. 176, pages 161-163

Emerson, S., 1975, Chemical enhanced CO_2 gas exchange in an eutrophic lake - a general model, Limnology and Oceanography (L&O), V. 20, pages 743-753

Erickson, D.J., 1993, A stability dependent theory for air-sea gas exchange, JGR, No. C5, pages 8471-8488

Etcheto, J., Boutin, J., and Merlivat, L., 1991, Seasonal variations of the CO_2 exchange coefficient over the global ocean using satellite wind speed measurements, Tellus, V. 43 B, No. 2, pages 247-255

Glasscock, A.D. and Rochelle, G.T., 1989, Numerical simulation of theories for gas absorption with chemical reaction, AIChE Journal, V. 85, No. 8, pages 1271-1281

Goldman, J.C. and Dennett, M.R., 1983, Carbon dioxide exchange between air and seawater: no evidence for rate catalysis, Science, V. 220, pages 199-201

Harvey, H.W., 1966, The chemistry and fertility of sea waters, Cambridge University Press

Holmen, K. and Liss, P., 1984, Models for air water gas transfer, an experimental investigation, Tellus, V. 36 B, No. 2, pages 92-100

Hoover, T.E. and Berkshire, D.C., 1969, Effects of hydration on carbon dioxide exchange across the air-sea interface, JGR, V. 74, No. 2, pages 456-464

Johnson, K.S., 1982, Carbon dioxide hydration and dehydration kinetics in seawater, L&O, V. 27, No. 5, pages 849-855

Keeling, C.D., 1968, Carbon dioxide in surface ocean waters, 4. global distribution, JGR, V. 73, pages 4543-4554.

Kempe, S. and Pegler, K., 1991, Sinks and sources of CO_2 in coastal seas: the North Sea, Tellus, V. 43 B, pages 224-235

Kern, D.M., 1960, Hydration of carbon dioxide, Journal of Chemical Education ,V. 37, No. 1, pages 14-23

Lindskog, S., 1980, Rate limiting step in the catalytic action of carbonic anhydrase, in: Biophysics and physiology of carbon dioxide, Ed: Bauer, B. et al, Springer Verlag, Berlin

Lindskog, S., Engberg, P., Forsman, C., Ibrahim, S.A., Johnson, B.-H., Simonsson, I. and Tibell, L., 1984, Kinetics and mechanisms of carbonic anhydrase isoenzymes, in: Tashian and Emmett, Biology and chemistry of the carbonic anhydrase, The New York Academy of Sciences, New York

Liss, P.S., 1983, Gas transfer: Experimental and geochemical implications, in: Liss, P.S. and Slinn, W.G.N. (eds.), Air-sea exchange of gases and particles, pages 241-298

Meldon, J.H., Smith, K.A. and Colton, C.K., 1972, An analysis of electrical effects induced by carbon dioxide transport in alkaline solutions, in : Recent development in separation science, Ed. L. Norman, CRC Press, pages 1-10

Memery, L. and Merlivat, L., 1984, The contribution of bubbles to gas transfer across an air-water interface, in: Ocean whitecaps, Ed. Monahan, E.C and Niocaill, G.M, D. Reidel Publishing Company, pages 95 -100

Miller, R.F., Berkshire, D.C., Kelley, J.J., and Hood, D.W., 1971, Method for determination of reaction rates of carbon dioxide with water and hydroxyl ion in seawater, Environ. Sci. and Techol. (ES&T), V. 5, No. 2. pages 127-133

Morel, F.M.M. and Hering, J.G., 1993, Principles and applications of aquatic chemistry, Wiley-Interscience, New York

Nydal, R., 1968, Further investigation on the transfer of radiocarbon in nature, JGR, V. 73, No. 15, pages 3617-3635

Otto, N.C., 1971, The transport of carbon dioxide in bicarbonate solutions: studies of flux augmentation and the properties of carbonic anhydrase, Ph.D. thesis, Chemical Engineering, University of Illinois at Urbana-Champaign

Peng, T.-H., Broecker, W.S., Mathieu, G.G. and Li, Y.-H., 1979, Radon evasion rates in the Atlantic and Pacific ocean as determined during the GEOSEC program, JGR, V. 84, No. C5, pages 2471-2486

Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T., 1990, Numerical recipes, the art of scientific computing (FORTRAN version), Cambridge University Press

Quinn, J.A. and Otto, N.C., 1971, Carbon dioxide exchange at the air-sea interface: Flux augmentation by chemical reaction, JGR, V. 76, No. 6, pages 1539-1549

Roberts, P.V. and Dändliker, P.G., 1983, Mass transfer of volatile organic contaminants from aqueous solution to the atmosphere during surface aeration, ES&T, V. 17, No. 8, pages 484-489

Robertson, J.E., Watson, A.J., Langdon, C., Ling, R.D. and Wood, J.W., 1993, Diurnal variation in surface pCO_2 and O_2 at 60^oN, 20^oW in the North Atlantic, Deep Sea Research (DSR) II, V. 40, No. 1/2, pages 409-422

Schwarzenbach, R.P., Gschwend, P.M. and Imboden, D.M., 1993, Environmental organic chemistry, Wiley-Interscience

Simpson, J.J. and Zirino, A., 1980, Biological control of pH in the Peruvian coastal upwelling area, DSR, V. 27, pages 733-744

Smith, S.V., 1985, Physical, chemical and biological characteristics of CO_2 gas flux across the air-water interface, Plant Cell and Environment, V. 8, pages 387-398

Stuiver, M., 1980, ^{14}C distribution in the Atlantic ocean, JGR, V. 85, pages 2711-2718

Stumm, W. and Morgan, J.J., 1981, Aquatic Chemistry, Wiley-Interscience, New York

Taylor, A. H., Watson, A.J., Ainsworth, M., Robertson, J.E. and Turner, D.R., 1991, A modeling investigation of the role of phytoplankton in the balance of carbon at the surface of the north Atlantic, GBC, V. 5, No. 2. pages 151-171

Thomas, F., Perigaud, C., Merlivat, L. and Minister, J.-F., 1988, World scale monthly mapping of the CO_2 ocean atmosphere gas transfer coefficient, Phil. Trans. R. Soc. Lond. A., V. 325, pages 71-83

Trowbridge, P., 1994, Wind speed data obtained from Logan airport, Boston, personal communication

Wallace, D.W.R. and Wirick, C.D., 1992, Large air-sea fluxes associated with breaking waves, Nature, V. 356, page 694

Watson, A.J., Robinson, C., Robinson, J.E., Williams, P.J. le B. and Fasham, M.J.R., 1991, Spatial variability in the sink for atmospheric carbon dioxide in the North Atlantic, Nature, V. 350, pages 50-53

Westerterp, R.K. and Wijngaarden, R.J., 1992, Principles of chemical reaction engineering, in: Ullmann's encyclopedia of industrial chemistry, Ed: Elvers,B., Weinheim, V. B 4, pages 5-83

Wheast, R.C. and Lide, D.R. (eds.), 1990, Handbook of chemistry and physics, The Chemical Rubber Co.

Whitman, W.G., 1923, The two-film theory of gas absorption, Chem. Met. Eng., V.29, pages 146-148

Winn, C.D., Mackenzie, F.T., Carrillo, C.J., Sabine, C.L. and Karl, D.M., 1994, Air-sea exchange in the north Pacific subtropical gyre, implications for the global carbon budget, CBC, V. 8, No. 2, pages 157-163

Zlokarnik, M., 1990, Dimensional Analysis, in: Ullmann's encyclopedia of industrial chemistry, Ed: Elvers, B., Weinheim, V. B1, pages 3.1-3.27

VII. Appendices

Appendix A. Notation

Symbol	usage	units
[ ]	species concentration	$\text{moles}*l^{-1} = M$
{ }	species activity	M
$\overline{\textit{overbar}}$	averaged parameter	
θ	exposure time film penetration model	s
α_0	carbon dioxide dissociation fraction	dimensionless
$\alpha 1$	bicarbonate dissociation fraction	dimensionless
α_2	carbonate dissociation fraction	dimensionless
γ	adjusting parameter for oscillation period	dimensionless
μ	viscosity	$kg*m^{-1}*s^{-1}$
ρ	density	$kg*m^{-3}$
χ	phase shift	s
A	area	m^2
activity_{CA}	ratio of active enzyme at pertinent pH	dimensionless
Alk	total alkalinity	$eq*l^{-1}$
B_t	total borate concentration	M
c_i	concentration of species i	M
c_t	total inorganic carbon	M
Da	Damkohler number	dimensionless
D_i	diffusivity of species i	m^2*s^{-1}

ΔpCO_2	difference in partial pressure of CO_2	μatm
EF	enhancement factor	dimensionless
$[E_t]$	total concentration of enzyme CA	M
F_i	flux of species i (negative values imply absorption)	$moles*m^{-2}*s^{-1}$
$'K_1$	apparent first carbon dissociation constant	M
k_{14}	kinetic constant for CO_2 hydration, OH^- pathway	$Mol^{-1}s^{-1}$
$'K_2$	apparent second carbon dissociation constant	M
k_{41}	kinetic constant for CO_2 dehydration, OH^- pathway	s^{-1}
$'K_B$	apparent borate dissociation constant	M
$'K_{CA}$	apparent carbonic anhydrase dissociation constant	M
k_{co2}	kinetic constant for CO_2 hydration, H_2O pathway	s^{-1}
k_{dehydr}	dehydration rate constant, CA pathway	s^{-1}
k_{HCO3}	kinetic constant for CO_2 dehydration, H_2O pathway	$M^{-1}s^{-1}$
$'K_{Henry}$	apparent Henry's law constant	$M*atm^{-1}$
$k_{hydr\,max\,CA}$	hydration rate constant, CA pathway	s^{-1}
k_l	mass transfer coefficient, "piston velocity"	$m*s^{-1}$
K_{MM}	Michaelis-Menten half saturation constant	M
k_{obs}	pseudo-first-order rate constant	s^{-1}
K_w	thermodynamic water dissociation constant	M^2
$'K_w$	apparent water dissociation constant	M^2
np	number of passages	dimensionless
pH	negative $\log_{10}$ of {H}	
r	parameter for analytical approximation of EF	m^{-1}
rsc_t	saturation ratio of total inorganic carbon	dimensionless

s	surface renewal rate in Danckwerts' model	s^{-1}
Sc	Schmidt number	dimensionless
t	time	s
T	temperature	^{o}C
u_{10}	wind speed in 10 m height	$m*s^{-1}$
x	space coordinate	m
z	laminar layer thickness	m
z_{eq}	hypothetical laminar layer thickness with the same piston velocity as the film penetration model	m

Appendix B. Constants and interpolation methods

Constant	**Value**	**Source**
- log ('K_1)	6.0	*Stumm and Morgan* (1981)[1]
- log ('K_2)	9.1	*Stumm and Morgan* (1981)
- log ('K_b)	8.7	*Stumm and Morgan* (1981)
- log ('K_{CA})	7.5	*Lindskog* (1984)
- log ('K_{Henry})	1.53	*Stumm and Morgan* (1981)
- log ('K_w)	13.7	*Stumm and Morgan* (1981)
Alk	2.47E-3 eq/l	*Stumm and Morgan* (1981)
B_t	4.1E-4 M	*Stumm and Morgan* (1981)
D_{CO2}	1.94 E-9 m^2*s^{-1}	*Meldon* (1972)
D_{HCO3}	0.94 E-9 m^2*s^{-1}	*Meldon* (1972)[2]
D_{CO3}	0.94 E-9 m^2*s^{-1}	*Meldon* (1972)
$k_{14}*K_w$	1.7E-10	*Miller et al* (1971) and *Johnson* (1982)
k_{CO2}	0.03 s^{-1}	*Kern* (1960)
$k_{hydr,max,CA}$	7.08 E5 s^{-1}	*Lindskog* (1984)
K_{MM}	40E-3 M	*Lindskog* (1984)

The EF is very sensitive to the kinetic and equilibrium constants, which are only known within a certain degree of accuracy. The increase of the EF with increasing pH is caused by the greater reaction rate of the hydroxyl pathway which becomes rate de-

---

[1] All calculations in this thesis refer to 25° C and standard pressure, unless otherwise specified.

[2] We assumed equal diffusivities for the charged species to eliminate the potential term in the Nernst-Planck equation (*Quinn and Otto* (1971)).

termining in seawater at a pH of about 8.3. The assumption that the kinetic parameters in seawater are the same as in pure water (as done, for example, by *Bolin* (1960)) is very questionable. The hydroxyl pathway rate constant increases drastically with increasing salinity (*Miller et al* (1971)). Measured values for the hydroxyl pathway in seawater are reported as the product of the hydroxyl rate constant and the ionization product of water ($k_{14}*K_w$) (*Johnson* (1982) and *Miller et al* (1971)). The kinetic constants are reported as "apparent" constants, referring to the concentration scale for the carbon species. The activity scale, defined by the pH, is applied to the hydrogen and hydroxyl ions (*Johnson* (1982)).

Johnson (1982) reports a value for $k_{14}*K_w$ equal to 1.35E-10. The same parameter determined by *Miller et al* (1971) is about twice as large. A factor of 2 change in the rate-determining reaction rate has a large impact on the calculated EF.

The selected value of $k_{14}*K_w$ was chosen to represent the measured properties. First, the pH at which the hydroxyl and hydration pathway become equally important in the model is 8.3. *Miller et al* (1971) report this pH to be approximately 8.2, whereas *Johnson* (1982) calculates a value of 8.43. Also, the chosen value of $k_{14}*K_w$ lies in between the measured values of *Johnson* (1982) and *Miller et al* (1971). The hydroxyl pathway in the model is therefore not overestimated compared to the reported data.

Temperature interpolation methods

The dissociation constants and the Henry coefficient were interpolated using the relationships given by *Stumm and Morgan* (1981). The diffusivities were adjusted with the assumption that the diffusivity is proportional to $\mu^{-1.14}$ (*Schwarzenbach et al* (1983)) with viscosity data from *Wheast and Lide* (1990). Kinetic data was calculated, using the Arrenius equation to fit to data given in the references.

Appendix C. Equations describing the chemical system

$H_2O \Leftrightarrow H^+ + OH^-$ with $'K_w = \{H^+\}[OH^-]$ and $K_w = \{H^+\}\{OH^-\}$	C(1)
$CO_{2,\,gas} + H_2O \Leftrightarrow CO_{2,\,dissolved}$ with $pCO_{2,\,water} = \dfrac{[CO_{2,dissolved}]}{'K_{Henry}}$	C(2)
$CO_{2,aq} + H_2O \Leftrightarrow HCO_3^- + H^+$ with $'K_1 = \dfrac{[HCO_3^-]\{H^+\}}{[CO_2]}$	C(3)
$HCO_3^- + H_2O \Leftrightarrow CO_3^{2-} + H^+$ with $'K_2 = \dfrac{[CO_3^{2-}]\{H^+\}}{[HCO_3^-]}$	C(4)
$B(OH)_3 + H_2O \Leftrightarrow B(OH)_4^- + H^+$ with $'K_b = \dfrac{[B(OH)_3]\{H^+\}}{[B(OH)_4^-]}$	C(5)
$\alpha_0 = \left(1 + \dfrac{'K_1}{\{H^+\}} + \dfrac{'K_1 * 'K_2}{\{H^+\}^2}\right)^{-1}$	C(6)
$\alpha_1 = \left(1 + \dfrac{\{H^+\}}{'K_1} + \dfrac{'K_2}{\{H^+\}}\right)^{-1}$	C(7)
$\alpha_2 = \left(1 + \dfrac{\{H^+\}^{2}\,'K_1}{'K_1 * 'K_2} + \dfrac{\{H^+\}}{'K_2}\right)^{-1}$	C(8)
$\alpha_B = \dfrac{'K_B}{'K_B + \{H^+\}}$	C(9)
$c_t = [CO_2] + [HCO_3^-] + [CO_3^{2-}]$	C(10)
$Alk = -[H^+] + [OH^-] + [HCO_3^-] + 2[CO_3^{2-}] + [B(OH)_4^-]$	C(11)

$B_t = \left[B(OH)_3\right] + \left[B(OH)_4^-\right]$	C(12)
$\left.\frac{\partial\left[CO_2\right]}{\partial t}\right\vert_{OH} = -k_{14}\left[CO_2\right]\left\{OH^-\right\} + k_{41}\left[HCO_3^-\right]$	C(13)
$\left.\frac{\partial\left[CO_2\right]}{\partial t}\right\vert_{H_2O} = -k_{co_2}\left[CO_2\right] + k_{HCO_3}\left[HCO_3^-\right]\left\{H^+\right\}$	C(14)
$\left.\frac{\partial\left[CO_2\right]}{\partial t}\right\vert_{CA} = -\frac{\left[E_t\right]k_{hyd} * \left[CO_2\right]}{K_{mm} + \left[CO_2\right]} + \frac{\left[E_t\right]k_{dehydr} * \left[HCO_3^-\right]}{K_{mm} + \left[HCO_3^-\right]}$	C(15)
$\left.\frac{\partial\left[CO_2\right]}{\partial t}\right\vert_{total} = \left.\frac{\partial\left[CO_2\right]}{\partial t}\right\vert_{CA} + \left.\frac{\partial\left[CO_2\right]}{\partial t}\right\vert_{H_2O} + \left.\frac{\partial\left[CO_2\right]}{\partial t}\right\vert_{OH} = -k_{obs}\left[CO_2\right]$	C(16)
$activity_{CA,hydration} = \frac{1}{1 + \left(\frac{\{H\}}{K_{CA}}\right)}$	C(17)
$k_{hyd}(pH) = k_{hyd,\,max,\,ca} * activity_{ca,\,hydration}$	C(18)
at equilibrium : $\frac{\left[E_t\right]k_{hyd} * \left[CO_2\right]}{K_{MM} + \left[CO_2\right]} = \frac{\left[E_t\right]k_{dehydr} * \left[HCO_3^-\right]}{K_{MM} + \left[HCO_3^-\right]}$	C(19)

Appendix D. Equations and assumptions for the estimation of averaging effects

The main influencing factors used in deriving the global oceanic CO_2 flux are wind-speed, temperature and ΔpCO_2 values. The Liss and Merlivat relationship (*Thomas et al* (1988)) calculates the piston velocity k_l as a function of the wind speed :

$$\begin{array}{lll} \text{for } u_{10} \leq 3.6 & k_l = 0.17 * u_{10} * \left[\frac{Sc(T=20)}{Sc(T)}\right]^{2/3} & \\ \text{for } 3.6 < u_{10} \leq 13 & k_l = (2.85 * u_{10} - 9.65) * \left[\frac{Sc(T=20)}{Sc(T)}\right]^{1/2} & \text{(D1)} \\ \text{for } u_{10} > 13 & k_l = (5.9 * u_{10} - 49.3) * \left[\frac{Sc(T=20)}{Sc(T)}\right]^{1/2} & \end{array}$$

where k_l is in units of centimeter per hour. Sc is the Schmidt number defined by

$$Sc = \frac{\mu}{D * \rho} \qquad \text{(D2)}$$

The Schmidt number ratio for CO_2 was determined with the regression coefficients given by *Erickson* (1993). The calculated laminar layer thickness was corrected with an upper limit of 700 μm, to overcome the obvious shortcoming of the Liss and Merlivat relationship at low wind speed (z goes to infinity as u_{10} goes to zero).

The pH was calculated with a polynomial fit for the model results. The obtained pH values as well as the ΔpCO_2-EF relationships were similar to results obtained by the pH-$pCO_{2,water}$ regression equation given by *Kempe and Pegler* (1981). The EF finally was calculated using the analytical approximation for the laminar film model from *Smith* (1985), as defined in chapter III.

I assumed a sigmoidal deviation of the average wind speed to model the annual wind cycle, as given in:

$$\mathrm{u}_{10}(t) = \overline{u_{10}} + amplitude(u_{10}) * \sin(\gamma_{u10} * t + \chi) \qquad \text{(D3)}$$

An amplitude of 36 % of the average wind speed results in the same maximum variation of the k_l values at the average wind speed as given in values from *Thomas et al* (1988). The annual cycle of ΔpCO_2 was given by *Taylor et al* (1991) and *Harvey* (1966) and was modeled according to:

$$\Delta pCO_2(t) = \overline{\Delta pCO_2} + amplitude(\Delta pCO_2) * \sin(\gamma_{co2} * t) \qquad \text{(D4)}$$

The phase shift χ was used to adjust the correlation coefficient between wind speed and ΔpCO_2, whereas the factor γ represented the difference in the oscillation periods. This approach has a definite shortcoming in neglecting that the system is driven by the temperature changes and the biological pump. The ΔpCO_2 is the result of the forcing function and the feedback mechanisms, which are neglected. If one assumes a correlation coefficient of unity between the two parameters, the effective EF is a strong function of the variation of the wind speed. In order to incorporate the feedback mechanisms a phytoplancton bloom model with changing wind speed has to be implemented. The phytoplancton model used by *Taylor et al* (1991) uses a constant wind speed, so that his results cannot be used for a correlation analysis.

Appendix E. FORTRAN codes

1. Code laminar film model:

```
1      PROGRAM FLUX
C      ------------------------------------------------------------
C      |  DATE: 04/24/94                                            |
C      |  fluxit.f                                                  |
C      |  MODEL ENZYMATIC ENHANCEMENT FACTOR                        |
C      |  THIS PROGRAM CALCULATES THE ENHANCEMENT OF MASS TRANSFER |
C      |  OF CARBON DIOXIDE ACROSS THE AIR-SEA INTERFACE            |
C      |  AUTHOR: KLAUS KELLER                                      |
C      |  MIT                                                       |
c        ----------------------------------------------------------
C      ------------------------------------------------------------
C      Declaration of parameter
C      ------------------------------------------------------------
c
       REAL ALK,H,Z,DTDIFF,DTREACT
       REAL PCO2WATER, DELTAPCO2
       REAL K1,K2,KW,KB,TEMPC,TEMPABS
       REAL EQUILEF,EFCALCBC,EFCALCBCOLD,CHANGEEF,PCHANGEEF
       REAL AVFLUXCO2, AVFLUXTOTAL
       REAL CTDEFICIT, PATMCO2,HENRYCO2
       REAL MINPH,MAXPH
       REAL PHACC,EFACC,MASSBALACC
       REAL DIFFCO2, DIFFHCO3, DIFFCO3
       REAL KCO2,KOH,KCAF
       REAL SQRDAMMKOEHLER
       PARAMETER (MAXSIZETIME=2000)
       PARAMETER(MAXSIZESPACE=50)
       INTEGER I,K,NMAX,TSTEPMAX
       INTEGER COUNT,REALCOUNT,T
       CHARACTER*1 STEUER
       CHARACTER*8 OUTNAME
       DIMENSION CO2 (MAXSIZESPACE,MAXSIZETIME)
       DIMENSION H (MAXSIZESPACE,MAXSIZETIME)
       DIMENSION PH (MAXSIZESPACE,MAXSIZETIME)
       DIMENSION HCO3 (MAXSIZESPACE,MAXSIZETIME)
       DIMENSION CO3 (MAXSIZESPACE,MAXSIZETIME)
       DIMENSION CT (MAXSIZESPACE, MAXSIZETIME)
       DIMENSION CAPARTITION (MAXSIZESPACE)
       REAL CAFILM,CABULK
       INTEGER CALAYER
       DIMENSION SPECIES (MAXSIZESPACE)
       DIMENSION DIFFER (MAXSIZESPACE)
       DIMENSION DELTA (MAXSIZESPACE)
       Dimension CHANGECO2  (MAXSIZESPACE,MAXSIZETIME)
```

```
      DIMENSION CHANGEHCO3 (MAXSIZESPACE,MAXSIZETIME)
      DIMENSION CHANGECO3  (MAXSIZESPACE,MAXSIZETIME)
      DIMENSION CHANGEPH (MAXSIZESPACE,MAXSIZETIME)
      DIMENSION CHANGECT (MAXSIZESPACE,MAXSIZETIME)
      DIMENSION PCHANGECO2  (MAXSIZESPACE,MAXSIZETIME)
      DIMENSION PCHANGEHCO3 (MAXSIZESPACE,MAXSIZETIME)
      DIMENSION PCHANGECO3  (MAXSIZESPACE,MAXSIZETIME)
      DIMENSION PCHANGEPH (MAXSIZESPACE,MAXSIZETIME)
      DIMENSION PCHANGECT (MAXSIZESPACE,MAXSIZETIME)
      Dimension DCO2  (MAXSIZESPACE)
      DIMENSION DHCO3 (MAXSIZESPACE)
      DIMENSION DCO3  (MAXSIZESPACE)
      DIMENSION DPH (MAXSIZESPACE)
      DIMENSION DCT (MAXSIZESPACE)
      DIMENSION PDCO2  (MAXSIZESPACE)
      DIMENSION PDHCO3 (MAXSIZESPACE)
      DIMENSION PDCO3  (MAXSIZESPACE)
      DIMENSION PDPH (MAXSIZESPACE)
      DIMENSION PDCT (MAXSIZESPACE)
      DIMENSION CO2START (MAXSIZESPACE)
      DIMENSION CO2STOP  (MAXSIZESPACE)
      DIMENSION HCO3START(MAXSIZESPACE)
      DIMENSION HCO3STOP (MAXSIZESPACE)
      DIMENSION CO3START (MAXSIZESPACE)
      DIMENSION CO3STOP  (MAXSIZESPACE)
      DIMENSION PHSTART  (MAXSIZESPACE)
      DIMENSION PHSTOP   (MAXSIZESPACE)
      DIMENSION CTSTART  (MAXSIZESPACE)
      DIMENSION CTSTOP   (MAXSIZESPACE)
C     REFERS TO KINETIC ENHANCEMENT WITHOUT TIMEHISTORY
      REAL CO2T,CO2B,HCO3T,HCO3B,CO3T,CO3B
C     REFERS TO TOP AND BULK FOR EQUILIB. FACTOR ROUTINE
      REAL CO2IN, HCO3IN, CO3IN,PHIN
      REAL CO2EQ,HCO3EQ,CO3EQ,PHEQ,CTSAVE
      REAL CO2R, HCO3R, CO3R, PHR
      REAL CO2INTERFACE
C     K AND R REFER TO REACTION AND EQUILIBTATE OUTPUT RESPECTIVLY
      CHARACTER*20 ENHANCEMENT
      CHARACTER*1 DECICION
      REAL STABLECHANGE,MAXCHANGE
      REAL TIMEELAPSED,PROGRESS,PERCENT,maxtime,count2
      DIMENSION CO2TURNO (MAXSIZESPACE,MAXSIZETIME)
      INTEGER DIVIDETREACT,LASTCHANGESTEP
      LOGICAL REACTION,EQREACHED,SMOOTH,EXIT,PARTITION
      CHARACTER*12 PATH
      DIMENSION FLUXCO2(MAXSIZESPACE)
      DIMENSION FLUXHCO3(MAXSIZESPACE)
      DIMENSION FLUXCO3 (MAXSIZESPACE)
      DIMENSION FLUXTOTAL(MAXSIZESPACE)
      DIMENSION CO2END (MAXSIZESPACE)
      DIMENSION HCO3END (MAXSIZESPACE)
```

```
      DIMENSION CO3END (MAXSIZESPACE)
C     CONCENTRATIONEND REFERS TO LAST CONCENTARTION OF ITERATION
C     FOR INPUT IN FLUXES ROUTINE
C
c
c
c
20000 PRINT*,' '
C     LABEL FOR RETURN FOR NEW ITERATIONS AND IF ERRORS OCURR
      PRINT*,' **********************************************'
      PRINT*,' * ENHANCEMENT MODEL  VERSION #14 04/25/94     *'
      PRINT*,' *                                             *'
      PRINT*,' * AUTHOR: KLAUS KELLER                        *'
      PRINT*,' * MIT                                         *'
      PRINT*,' **********************************************'
      PRINT*,' '
      PRINT*, 'THE MAIN INPUT PARAMETER AND BOUNDARY CONDITIONS'
      print*,' ARE INITIALICED AS FOLLWOS:'
      print*,' *******************************************************'
      PRINT*,' '
C
c
c  -------------------------------------------------------------
C  ASSIGNMENT OF VALUES FOR FIXED PARAMETERS RESETTING CHANGED ONES
C  ------------------------------------------------------------
C
      MAXCHANGE=0
      MAXTIME=50
      EQREACHED=.FALSE.
      EXIT=.FALSE.
      SMOOTH=.TRUE.
      MAXCHANGE=0
      TIMEELAPSED=0
      COUNT=0
      COUNT2=0
      PROGRESS=0
      PERCENT=0
      STABLETIMESTEP=TSTEPMAX
      TSTEPMAX=0
      PARTITION=.FALSE.
      TEMPC=25
      PRINT*,'TEMPERATURE WATER : ', TEMPC, ' DEGREE CELSIUS'
      STABLECHANGE=1E-11
      PRINT*,'STABLECRITERIA : ',STABLECHANGE,' %'
c      CRITERION FOR STEADY STATE REACHED, IN %
      PHACC=0.000001
      PRINT*,'PHACC : ',PHACC,' pH units'
      EFACC=0.001
C     WANTED ACCURANCY EF IN %
      PRINT*,'ACCURANCY OF ENHANCEMENT FACTOR : ',EFACC ,' %'
      MASSBALACC=0.2
```

```
      PRINT*,'ALLOWED MASS BALLACE ERROR :',MASSBALACC,' %'
      PRINT*,'SMOOTING IC ',SMOOTH
      MAXPH=14
      MINPH=1
      PATMCO2=340E-6
      print*,'PARTIAL PRESSURE CO2 AIR = ',PATMCO2,' atm'
      ALK=2.47E-3
      PRINT*,'ALKALINITY :',ALK,' M'
      Z=0.0001
      PRINT*, 'LAYER THICKNESS  = ',Z,' m'
C     UNITS IN M
      DIFFCO2=1.94E-9
      print*,'DIFFUSIFITY CO2 ', DIFFCO2,' m2/s'
      DIFFHCO3=0.947E-9
      DIFFCO3=DIFFHCO3
      PRINT*,'DIFFUSIVITY CARBONATE AND BICARBONATE',DIFFCO3,' M2/S'
C     CALCULATE WITH EQUAL DIFFUSIVIIES FOR MAIN CHARGED SPECIES TO
C     ELIMINATE NERST TERM IN FLUX EQUATION
      NMAX=20
C     VALUE FOR MAXIMUM NUMBER OF LAYERS, normaly 20
      PRINT*,'NUMBER OF SUBLAYERS'
      PRINT*,'NUMBER OF SUBLAYERS =',NMAX
      DTDIFF=1E-3
      PRINT*,'TIMESTEP : ',DTDIFF ,'s'
C     VALUE FOR INKREMENT DIFFUSE, DEFAULT
      PRINT *, 'ALKALINITY =', ALK
      BT=4.1E-4
      print*,'BORATE CONCENTRATION = ',BT
      K1=10**(-6.0)
      K2=10**(-9.1)
      KW=2E-14
      henryco2=10**(-1.53)
      KB=10**(-8.71)
      print*,'HENRY COEFFICIENT CO2 = ',HENRYCO2,' M/atm'
C     25 DEGREE CELSIUS, 35 promille SALINITY
      PATH='CAoff'
      PRINT*,'CA PATHWAY = ',PATH
      KCO2=0.03
      KOH=8500
      KCAF=7.08E5
      PRINT*,'CO2-HYDRATION KINETIC CONSTANT : ',KCO2,' 1/s'
      sqrDAMMKOEHLER=SQRT(KCO2)*Z/(SQRT(DIFFCO2))
      PRINT*,'SQUAREROOT DAMMKOEHLER NUMBER : ',sqrDAMMKOEHLER
      CACONC=1E-8
      PRINT*,'CARBONIC ANHYDRASE CONCENTRATION ', CACONC,' MOL/l'
      PRINT*,'NO SURFACE LAYER PARTITIONING'
      PRINT*,' '
      PRINT*,'CALCULATION FOR SEAWATER'
      Print*,' '
c
c
```

```
C
C
C     CHANGES IN GENERAL CONDITIONS
C
      PRINT*,'DO YOU WANT TO CHANGE ANY OF THESE VALUES (Y/N)'
      READ*,STEUER
      IF(STEUER.EQ.'Y') THEN
 2001    PRINT*,'WHICH VALUE DO YOU WANT TO CHANGE'
         PRINT*,' '
         PRINT*,'    TEM(P)ERATURE '
         PRINT*,'    (L)AYER,'
         PRINT*,'    TIMESTEP (D)ELTA T'
         PRINT*,'    (B)ORATE CONCENTRATION,'
         PRINT*,'    (C)A-PATHWAY AND CONCENTRATION'
         PRINT*,'    (S)TABLECRITERION'
         PRINT*,'    (P)H ACCURANCY'
         PRINT*,'    (E)F ACCURANCY'
         PRINT*,'    (M)ASS BALANCE ACCURANCY'
         PRINT*,'    (T)HICHNESS OF LAYER '
         PRINT*,'    (S)MOOTHING I-CONDITIONS'
         PRINT*,'    (A)LKALINITY '
         PRINT*,'    (K)INETIC CONSTANT CO2HYDRATION'
         PRINT*,'    (F)RESHWATER VALUES'
         PRINT*,'     P(A)RTITIONING OF CA'
         PRINT*,' '
         READ*,DECICION
         IF(DECICION.EQ.'C')THEN
            PRINT*,'ENTER NEW PATHWAY (CAON OR CAOFF)'
            READ*,PATH
            print*,'CA PATHWAY : ',path
            PRINT*,'ENTER CA CONCENTRATION IN M'
            READ*,CACONC
            PRINT*,' THE NEW CA CONCENTRATION IS ', CACONC, 'M'
         ELSEIF(DECICION.EQ.'A')THEN
            PARTITION=.TRUE.
            PRINT*,'CALCULATING WITH SURFACE LAYER PARTITIONING'
            PRINT*,'PLEASE ENTER CATOP'
            READ*,CAfilm
            PRINT*,'PLEASE ENTER CABULK'
            READ*,CABULK
            print*,'enter # layers with enriched  CA.'
            READ*,CALAYER
         ELSEIF(DECICION.EQ.'L')THEN
            PRINT*,'ENTER NEW LAYER NUMBER'
            PRINT*,'MAXIMUM VALUE : ',MAXSIZESPACE
            READ*,NMAX
            IF(NMAX.GT.MAXSIZESPACE) THEN
               PRINT*,'SORRY, VALUE NOT ALLOWDED'
               GOTO 2001
            ENDIF
            PRINT*,'NEW LAYER NUMBER IS ',NMAX
```

```
         ELSEIF(DECICION.EQ.'E')THEN
            PRINT*,'ENTER NEW EF ACCURANCY IN %'
            READ*,EFACC
            PRINT*,'NEW EF ACCURANCY IS ',EFACC
         ELSEIF(DECICION.EQ.'F')THEN
            PRINT*,'CHANGING EQUILIBRIUM CONSTANTS TO FRESHWATER'
            K1=10**(-6.3)
            K2=10**(-10.3)
            KW=1E-14
            HENRYCO2=10**(-1.5)
            PRINT*,'SETTING ALK AND BORATE CONCENTRATION'
            ALK=1.5E-3
            BT=0
            PRINT*,'ALKALINITY ',ALK
            PRINT*,'BORATE ', BT
         ELSEIF(DECICION.EQ.'A')THEN
            PRINT*,'ENTER NEW ALKALINITY IN M'
            READ*,ALK
            PRINT*,'NEW ALKALINITY IS',ALK
         ELSEIF(DECICION.EQ.'P')THEN
            PRINT*,'ENTER NEW TEMPERATURE IN DEGREE CELSIUS'
            READ*,TEMPC
            PRINT*,'NEW TEMPERATURE IS ',TEMPC ,'CELSIUS'
            TEMPABS=TEMPC+273
            CALL NEWCONSTANTS (TEMPABS,K1,K2,HENRYCO2,KB ,
     &              DIFFCO2,DIFFHCO3, DIFFCO3,
     &              KCO2,KOH,KCAF)
C         CALCULATING NEW EQUILIBRIUM CONSTANTS
C
         ELSEIF(DECICION.EQ.'M')THEN
            PRINT*,'ENTER NEW ACCEPTABLE MASSBALACE MISTAKE IN %'
            READ*,MASSBALACC
            PRINT*,'NEW ACCEPTABLE MASS BALACE MISTAKE IS ',MASSBALACC
         ELSEIF(DECICION.EQ.'S')THEN
            PRINT*,'PLEASE ENTER SMOOTH (F or T )'
            READ*,SMOOTH
            IF(smooth.eqv..false.) THEN
               PRINT*,'SMOOTHIN DISABLED'
            ENDIF
         ELSEIF(DECICION.EQ.'D')THEN
            PRINT*,'ENTER NEW TIMESTEPSIZE IN SECONDS'
            READ*,DTDIFF
            print*,'DTDIFF NOW : ',DTDIFF
         ELSEIF(DECICION.EQ.'B') THEN
            PRINT*,'ENTER NEW BORATE CONCENTRATION'
            READ*,BT
            PRINT*,'NEW BORATE CONCENTRATION IS ',BT
         ELSEIF(DECICION.EQ.'T') THEN
            PRINT*,'ENTER NEW LAYER THICKNESS IN M'
            READ*,Z
            print*,'NEW LAYER THICKNESS IS : ',Z
```

```
          sqrDAMMKOEHLER=SQRT(KCO2)*Z/(SQRT(DIFFCO2))
          PRINT*,'NEW SQRT DAMMKOEHLER NUMBER : ',SQRDAMMKOEHLER
       ELSEIF(DECICION.EQ.'K') THEN
          PRINT*,'ENTER NEW KINETIC CONSTANT'
          READ*,KCO2
          print*,'THE NEW KINETIC CONSTANT IS : ',KCO2
          SQRDAMMKOEHLER=SQRT(KCO2)*Z/(SQRT(DIFFCO2))
          PRINT*,'NEW SQRT DAMMKOEHLER NUMBER : ',SQRDAMMKOEHLER
       ELSEIF(DECICION.EQ.'P') THEN
          PRINT*,'ENTER NEW ACCURRANCY'
          READ*,PHACC
          print*,'NEW PH ACCUARNCY IS : ',PHACC
       ELSE
          PRINT*,'YOU MADE A TYPO'
          GOTO 20000
       ENDIF
       PRINT*,' ANOTHER PARAMETERCAHANGE ? (Y/N)'
       READ*,STEUER
       IF (STEUER.EQ.'Y') THEN
          GOTO 2001
       ENDIF
     ENDIF
c
c
c     setting ca concentration ineach layer to allow partitioning
c
c
      IF (PARTITION) THEN
         DO 58392 K=1,CALAYER
           capartition(k)=cafilm
58392    continue
         do 58393 k=calayer+1,nmax
            capartition(k)=cabulk
58393    continue
      else
         do 58394 k=1,nmax
            capartition(k)=CAconc
58394    continue
      endif
c
      print*,'ca concentrations in layers'
      do 63956 k=1,nmax
         print*,'layer ',K, 'caconc ',capartition(k)
63956 continue
c      pause


c
      PRINT*,'PLEASE ENTER CT DEFICIT BULK PHASE'
      READ*,CTDEFICIT
      PRINT*,' '
```

```
      PRINT*,'CT-DEFICIT = ',CTDEFICIT
C     DEGREE OF TIC  UNDERSATUATION
c
c     -------------------------------------------------------
c     CALCULATION BOUNDARY AND INITIAL CONDITIONS
C     --------------------------------------------------------
c
c
      CO2(1,0)=PATMCO2*HENRYCO2
      CO2INTERFACE=CO2(1,0)
      PH(1,0)=SOLVEALKOPEN(MINPH,MAXPH,PHACC,ALK,BT,K1,K2,KW,KB,
     &        HENRYCO2)
      H(1,0)=10**(-1*ph(1,0))
      PRINT*,'H-CONCENTRATION TOP = ',H(1,0)
      CT(1,0)=CTEQUIL(PH(1,0),K1,K2,HENRYCO2)
      HCO3(1,0)=CO2(1,0)*K1/(10**(-1*PH(1,0)))
      CO3(1,0)=CO2(1,0)*K1*K2/((10**(-1*PH(1,0)))**2)
      PRINT*,'HCO3TOP = ',HCO3(1,0)
      CT(NMAX,0)=CT(1,0)*CTDEFICIT
      PRINT*,'CT AT THE BOTTOM  = ',CT(NMAX,0)
      maxph=14
      PH(NMAX,0)=SOLVEALKCLOSED(MINPH,MAXPH,PHACC,CT(NMAX,0),
     &                             ALK,BT,K1,K2,KW,KB)
      H(NMAX,0)=10**(-1*PH(NMAX,0))
C
C     EQUILIBRATE BULK PHASE
C
      CO2(NMAX,0)=CT(NMAX,0)*ALPHA0(H(NMAX,0),K1,K2)
      PCO2WATER=CO2(NMAX,0)/HENRYCO2
      PRINT*,'PCO2ATM ',PATMCO2 , 'ATM'
      PRINT*,'PCO2WATER ',PCO2WATER , 'ATM'
      DELTAPCO2=(pco2water-patmco2)*1e6
c     value in 10E6 atm equals ppm
      print*,'deltapco2 ',deltapco2
      print*,'pHbulk phase',ph(nmax,0)
      pause
      HCO3(NMAX,0)=CT(NMAX,0)*ALPHA1(H(NMAX,0),K1,K2)
      CO3(NMAX,0)=CT(NMAX,0)*ALPHA2(H(NMAX,0),K1,K2)
C
c     SETTING INITIAL CONDITIONS (T=0) AS FULLY MIXED
C     AND EQUAL TO THE BULK PHASE
c
      DO 8000 I=2,(NMAX-1)
         CO2(I,0)=CO2(NMAX,0)
         HCO3(I,0)=HCO3(NMAX,0)
         CO3(I,0)=CO3(NMAX,0)
 210     CT(I,0)=CT(NMAX,0)
         PH(I,0)=PH(NMAX,0)
 8000    CONTINUE
C
C     -------------------------------------------------------
```

```
C     END BOUNDARY AND INITIAL CONDITIONS
C     ------------------------------------------------------
C
C
C      --------------------------------------------------
C     OUTPUT TABLE INICIALIZED VALUES
C      ---------------------------------------------------
      PRINT*,' '
      PRINT*,'TABLE OF INITIALIZED VALUES, TIME STEP ZERO'
      PRINT*,'*******************************************'
      PRINT*,' '
      PRINT *,'LAYER   ','  CO2   ','         HCO3   ','   CO3   ',
     &'      CT   ','      pH'
 230  PRINT*,' '
      DO 27 I=1,NMAX
         PRINT(111),I, CO2(I,0),HCO3(I,0),CO3(I,0),CT(I,0),PH(I,0)
 27   CONTINUE
 111  FORMAT(I7,4E12.4,F8.2)
      PRINT*,' '
      PRINT*,'ALL CONCENTRATIONS IN THIS PROGRAM ARE IN MOL/LITER'
      PRINT*,' '
C     ---------------------------------------------------------
C     END OUTPUT INITIALIZED VALUES
C     --------------------------------------------------------
c
C    ---------------------------------------------------------
c     CALCULATING EQUILIBRIUM ENHANCEMENT FACTOR AS RATIO
C     OF THE GRADIENTS, MOREL BOOK
C     -------------------------------------------------------
      CO2T=CO2(1,0)
      CO2B=CO2(NMAX,0)
      HCO3T=HCO3(1,0)
 250  HCO3B=HCO3(NMAX,0)
      CO3T=CO3(1,0)
      CO3B=CO3(NMAX,0)
      EQUILEF= EQEF(CO2T,CO2B,HCO3T,HCO3B,CO3T,CO3B,CTDEFICIT,
     &         DIFFCO2, DIFFHCO3, DIFFCO3 )
      PRINT*,'THE CALCULATED EQUILIBRIUM ENHANCEMENT FACTOR IS'
      PRINT*,EQUILEF
C     -------------------
C     END CALCULATING EEF
C     -------------------
C
C
C
C     ---------------------------------------
C     INPUT OF TSTEPMAX AND KIND OF TRANSPORT
C     ---------------------------------------
C
c     CHECKING WHAT TYPE OF TRANSPORT USER WANTS TO MODEL
C
```

```
c
      PRINT*,' '
      PRINT*,'WHAT KIND OF TRANSPORT DO YOU WANT TO SIMULATE'
      PRINT*,' '
      PRINT*,'     (P)URE DIFFUSION,'
      PRINT*,'     (E)QUILIBRIUM ENHANCEMENT,'
      PRINT*,'     (K)INETIC ENHANCEMENT (WITHOUT TIME HISTORY)'
      PRINT*,'      KINETIC ENHANCEMENT WITH TIME (H)ISTORY'
      PRINT*,' '
      READ*,DECICION
      IF((DECICION.EQ.'K').OR.(DECICION.EQ.'H')) THEN
         DIVIDETREACT=1
         PRINT*,'CALCULATING KINETIC ENHANCEMENT'
         PRINT*,'WITH ',DIVIDETREACT,' ITERATIONS PER DIFFUSIONSSTEP'
         ENHANCEMENT='KINETIC'
         REACTION =.TRUE.
         print*,'CA PATHWAY : ',path
      ELSEIF(DECICION.EQ.'E')THEN
         PRINT*,'CALCULATING EQUILIBRIUM ENHANCEMENT'
         ENHANCEMENT='EQUILIBRIUM'
         REACTION = .TRUE.
      ELSEIF(DECICION.EQ.'P') THEN
         PRINT*,'CALCULATE PURE DIFFUSION'
         ENHANCEMENT='PURE DIFFUSION, NO'
         REACTION = .FALSE.
      ELSE
         PRINT*,'YOU MADE A TYPO'
         GOTO 20000
      ENDIF
c
      IF(decicion.NE.'K') THEN
         PRINT*,'PLEASE ENTER VALUE FOR NUMBER OF ITERATIONS'
         PRINT*,'MAXIMUM VALUE = ',MAXSIZETIME
         READ*,TSTEPMAX
         PRINT*,'THE NEW NUMBER OF ITERATIONS IS ',TSTEPMAX
      ELSE
         PRINT*,'MAXIMUM TIME TO ITERATE IN SECONDS'
         READ*,MAXTIME
      endif
C
C     ---------------------
C     END OF INPUT SEQUENCE
C     ---------------------
C
C
C
C
C
C      --------------------------------------------------------------
C      |   DRIVER FOR PURE PHYSICAL TRANSPORT                        |
c      |   PROGRAMED: 12/12/93                                       |
```

```
C       --------------------------------------------------------------
C
      IF (DECICION.EQ.'P') THEN
c
c     changing top concentration discontinuity, because meaningless
c     in pure diffusion case
c
         hco3(1,0)=hco3(nmax,0)
         co3(1,0)=co3(nmax,0)
         ph(1,0)=ph(nmax,0)

        PRINT*,'PERCENT OF ITERATIONSTEPS CALCULATED:'
        PRINT*,' '
C
C   DIFFUSING CO2

c    COPYING TWO-DIMENSIONAL ARRAY TO ONE-DIM. ARRAY
C
         DO 9000 I=1,NMAX
            SPECIES(I)=CO2(I,COUNT)
 9000    CONTINUE
c
c
c    BIG LOOP TIMESTEPS
c
           DO 700 T=1,TSTEPMAX
            CALL DIFFUSEEQ(SPECIES,DTDIFF,NMAX,Z,DIFFCO2,DIFFER,DELTA)
            COUNT=T
            count2=t
c
c   TELLING USER WHERE IN ITERATION
C
         PROGRESS=(COUNT2*10/TSTEPMAX)
         IF(INT(PROGRESS).EQ.PROGRESS) THEN
            PERCENT=PERCENT+10
            PRINT*,PERCENT
         ENDIF
c
c   COPYING SPECIES BACK IN PROPPER PLACE OF ARRAY
c   AND SETTING BACK OTHER CONCENTRATIONS


       DO 777 K=1,NMAX
          CO2(K,COUNT)=SPECIES(K)
          CHANGECO2(K,COUNT)=CO2(K,COUNT)-CO2(K,COUNT-1)
c         no change in other species because no reaction
c         and initial condition satisfies DEQ
          HCO3(K,COUNT)=hco3(K,count-1)
          CO3(K,COUNT)=co3(k,count-1)
          PH(K,COUNT)=ph(k,count-1)
          ct(k,count)=co2(k,count)+hco3(k,count)+co3(k,count)
```

```
C
C
C   CHECKING IF STEADY STATE IS REACHED
C   AND SETTING OTHER CHANGES TO ZERO

C         CHANGEPH(K,COUNT)=0
          PCHANGEPH(K,COUNT)=0
          CHANGECO2(K,COUNT)=ABS(CO2(K,COUNT)-CO2(K,COUNT-1))
          CHANGEHCO3(K,COUNT)=0
          PCHANGEHCO3(K,COUNT)=0
          CHANGECO3(K,COUNT)=0
          PCHANGECO3(K,COUNT)=0
          CHANGECT(K,COUNT)=CHANGECO2(K,COUNT)
C
C
          IF (CHANGECO2(K,COUNT).EQ.O) THEN
             PCHANGECO2(K,COUNT)=0
          ELSE
             PCHANGECO2(K,COUNT)=100*CHANGECO2(K,COUNT)/CO2(K,COUNT)
c             print*,'pchangeco2 = ',pchangeco2(k,count)
          ENDIF
C
C
c    CHECKING AGAINST STABLECHANGE
c
C
          IF(PCHANGECO2(K,COUNT).GT.STABLECHANGE) THEN
             LASTCHANGESTEP=COUNT
          ENDIF
C
 777      continue
c         CLOSING LOOP LAYERS CHECK
C
C
 700  CONTINUE
C   CLOSING LOOP ENTIRE DRIVER
C
c
c   TESTING IF STEADY STATE WAS REACHED

         IF(TSTEPMAX.GT.LASTCHANGESTEP) THEN
            EQREACHED=.TRUE.
         ENDIF
      ENDIF
C        CLOSING IF CONDITION FOR WHOLE DRIVER PURE DIFFUSION
C
c      ---------------------------------------------------------
C     |    END DRIVER PURE PHYSICAL TRANSPORT                  |
C     ---------------------------------------------------------
C     ------------------------------------------------------
C     |  DRIVER EQUILIBRIUM ENHANCEMENT                    |
```

```
C      |  CALCULATES THE CONCENTRATION PROFILE WITH EQUILENH  |
c      |  SUBROUTINES DIFFUSEEQ AND EQUILIBRATE               |
C      ----------------------------------------------------
C      CHECK IF EQUILIBRIUM ENHANCEMENT IS WANTED
         IF(DECICION.EQ.'E') THEN
             PRINT*,' '
             PRINT*,'ENTERING DRIVER EQUILIBRIUM ENHANCEMENT'
             PRINT*,' PERCENT OF MAX. TIMESTEP CALCULATED:'
C
c
C      DIFFUSING ONE SPECIES AFTER THE OTHER
C      SPECIESOUT REFERS TO OUTPUT DIFFUSE ROUTINE
C
C
       DO 6000 T=1,TSTEPMAX
C      BIG LOOP FOR ENTIRE DRIVER EQUILIBRIUMENHANCEMENT
          COUNT=T
          count2=t
c
c      TELLING USER WHERE IN ITERATION
C
          PROGRESS=(COUNT2*10/TSTEPMAX)
          IF(INT(PROGRESS).EQ.PROGRESS) THEN
             PERCENT=PERCENT+10
             PRINT*,PERCENT,' PERCENT CALCULATED'
          ENDIF
c
C         DIFFUSING CO2
C
C
             DO 11 I=1,NMAX
                SPECIES(I)=CO2(I,COUNT-1)
 11          CONTINUE
             CALL DIFFUSEEQ(SPECIES,DTDIFF,NMAX,Z,DIFFCO2,DIFFER,DELTA)
C            COPYING SPECIES IN PROPPER PLACE OF ARRAY
             DO 12 K=1,NMAX
              CO2(K,COUNT)=SPECIES(K)
 12          CONTINUE
c              PAUSE
C
C
C      DIFFUSING HCO3
C
C
 390         DO 13 I=1,NMAX
               SPECIES(I)=HCO3(I,COUNT-1)
 13          CONTINUE
             CALL DIFFUSEEQ(SPECIES,DTDIFF,NMAX,Z,DIFFHCO3,DIFFER,DELTA)
C            COPYING SPECIES IN PROPPER PLACE OF ARRAY
             DO 14 K=1,NMAX
              HCO3(K,COUNT)=SPECIES(K)
```

```
 14          CONTINUE
c             PAUSE
C
C
C     DIFFUSING CO3
C
C
             DO 15 I=1,NMAX
                SPECIES(I)=CO3(I,COUNT-1)
 15          CONTINUE
            CALL DIFFUSEEQ(SPECIES,DTDIFF,NMAX,Z,DIFFCO3,DIFFER,DELTA)
C            COPYING SPECIES IN PROPPER PLACE OF ARRAY
             DO 16 K=1,NMAX
               CO3(K,COUNT)=SPECIES(K)
 16          CONTINUE
c              PAUSE
C
C            DIFFUSION FINISHED, NOW REACTION OCCURS
c
c420          COPYING ARRAY DATA T0 INPUT DATA
c
c
          DO 18 K=1,NMAX
           CO2IN=CO2(K,COUNT)
           HCO3IN=HCO3(K,COUNT)
           CO3IN=CO3(K,COUNT)
           CTSAVE=CO2IN+HCO3IN+CO3IN
           minph=1
C
C     -------------------------------------------
C     EQUILIBRIUM ENHANCEMENT WITH ALPHA VALUES
C     AND CALCULATION OF CHANGES
C     --------------------------------------------

           CALL EQUILIBRATE(CO2IN,HCO3IN,CO3IN,CO2EQ,HCO3EQ,
     &                     CO3EQ,PHIN,PHEQ,MINPH,MAXPH,PHACC,
     &                     K1,K2,KW,BT,ALK,KB)
           CO2TURNO(K,COUNT)=CO2EQ-CO2IN
c
c          co2turnover is the amount of reaction and the
c          relevant figure to evaluate the importance of
c          carbonic anhydrase
c
c          COPYING DATA BACK IN PROPPER PLACE OF ARRAY
c
           CO2(K,COUNT)=CO2EQ
           HCO3(K,COUNT)=HCO3EQ
           CO3(K,COUNT)=CO3EQ
           PH(K,COUNT)=PHEQ
           CT(K,COUNT)=CTSAVE
```

```
C       -----------------------------------------------
C         CALCULATING ABSOLUT  AND PERCENTAGE CHANCES
C         AND TIME OF STABILITY
c       -------------------------------------------
          CHANGEPH(K,COUNT)=ABS(PH(K,COUNT)-PH(K,COUNT-1))
          CHANGECO2(K,COUNT)=ABS(CO2(K,COUNT)-CO2(K,COUNT-1))
          CHANGEHCO3(K,COUNT)=ABS(HCO3(K,COUNT)-HCO3(K,COUNT-1))
          CHANGECO3(K,COUNT)=ABS(CO3(K,COUNT)-CO3(K,COUNT-1))
          CHANGECT(K,COUNT)=ABS(CT(K,COUNT)-CT(K,COUNT-1))
          IF (CHANGEPH(K,COUNT).EQ.O) THEN
             PCHANGEPH(K,COUNT)=0
          ELSE
             PCHANGEPH(K,COUNT)=100*CHANGEPH(K,COUNT)/PH(K,COUNT)
          ENDIF
          IF (CHANGECO2(K,COUNT).EQ.O) THEN
             PCHANGECO2(K,COUNT)=0
          ELSE
             PCHANGECO2(K,COUNT)=100*CHANGECO2(K,COUNT)/CO2(K,COUNT)
          ENDIF
          IF (CHANGEHCO3(K,COUNT).EQ.O) THEN
             PCHANGEHCO3(K,COUNT)=0
          ELSE
             PCHANGEHCO3(K,COUNT)=100*CHANGEHCO3(K,COUNT)/HCO3(K,COUNT)
          ENDIF
          IF (CHANGECO3(K,COUNT).EQ.O) THEN
             PCHANGECO3(K,COUNT)=0
          ELSE
             PCHANGECO3(K,COUNT)=100*CHANGECO3(K,COUNT)/CT(K,COUNT)
          ENDIF
          IF (CHANGECT(K,COUNT).EQ.O) THEN
             PCHANGECT(K,COUNT)=0
          ELSE
             PCHANGECT(K,COUNT)=100*CHANGECT(K,COUNT)/CT(K,COUNT)
          ENDIF
C
C
C     --------------------------------------------------------------
c     FINDIG OUT THE MAXIMUM CHANGE AND CHECKING AGAINST STABLECHANGE
c     -------------------------------------------------------------
          maxchange=0
          IF (PCHANGECO2(K,COUNT).GT.MAXCHANGE) THEN
             MAXCHANGE=PCHANGECO3(K,COUNT)
          ENDIF
          IF (PCHANGEHCO3(K,COUNT).GT.MAXCHANGE) THEN
             MAXCHANGE=PCHANGEHCO3(K,COUNT)
          ENDIF
          IF (PCHANGECO3(K,COUNT).GT.MAXCHANGE) THEN
             MAXCHANGE=PCHANGECO3(K,COUNT)
          ENDIF
          IF (PCHANGEPH(K,COUNT).GT.MAXCHANGE) THEN
             MAXCHANGE=PCHANGEPH(K,COUNT)
```

```
          ENDIF
          IF (PCHANGECT(K,COUNT).GT.MAXCHANGE) THEN
             MAXCHANGE=PCHANGECT(K,COUNT)
          ENDIF
          IF(MAXCHANGE.GT.STABLECHANGE) THEN
             LASTCHANGESTEP=COUNT
          ENDIF
c
 18    CONTINUE
 6000    CONTINUE
C        CLOSING LOOP ENTIRE DRIVER
         ENDIF
C        CLOSING IF CONDITION FOR WHOLE DRIVER EQUILIBRIUM
C
C     TESTING IF STEADY STATE WAS REACHED
C
         IF(TSTEPMAX.GT.LASTCHANGESTEP) THEN
            EQREACHED=.TRUE.
         ENDIF
C     -------------------------------------------------------
C     |  END OF DRIVER EQUILIBRIUM ENHANCEMENT                |
C     -------------------------------------------------------
C
C
C
C

C     -------------------------------------------------------
C     |  DRIVER KINETIC ENHANCEMENT NO TIME HISTORY          |
C     |  CALCULATES THE CONCENTRATION PROFILE USING ROUTINES |
c     |  REACT AND DIFFUSEKI                                 |
C     -----------------------------------------------------
c
c
C     CHECK IF KINETIC ENHANCEMENT IS WANTED
        IF(DECICION.EQ.'K') THEN
            PRINT*,' '
            PRINT*,'ENTERING DRIVER KINETIC ENHANCEMENT'
c
c     copying array data to species data
c
            DO 1114 I=1,NMAX
               co2start(i)=CO2(I,0)
               hco3start(i)=hCO3(I,0)
               co3start(i)=CO3(I,0)
               phstart(i)=ph(I,0)
               ctstart(I)=ct(I,0)
 1114       CONTINUE
c
c
c     changing initial conditions to speed up
```

```
c
      IF(SMOOTH) THEN
         DO 11141 K=2,NMAX-1
             CO2START(K)=CO2START(1)-((CO2START(1)-co2start(nmax))
     &                /(nmax-1)*(K-1))
             CTSTART(K)=CO2START(K)+HCO3START(K)+CO3START(K)
11141   continue
          hco3start(1)=hco3start(nmax)
          co3start(1)=co3start(nmax)
          phstart(1)=phstart(nmax)
          ctstart(1)=CO2START(1)+HCO3START(1)+CO3START(1)
      ENDIF
c
c     printing out changed concentrations
c
      PRINT*,' '
      PRINT*,'CT-DEFICIT = ',CTDEFICIT
      PRINT*,' '
      PRINT*,'TABLE OF CHANGED INITIALIZED VALUES, TIME STEP ZERO'
      print*,'***************************************************'
      PRINT*,' '
      PRINT *,'LAYER   ','  CO2   ','        HCO3   ','  CO3   ',
     &'       CT   ','       pH'
      PRINT*,' '
      DO 11142 I=1,NMAX
         PRINT(111),I, CO2start(I),HCO3start(I),CO3start(I),
     &   CTstart(I),PHstart(I)
11142 CONTINUE
c
c     calculating fluxes of initialication
c
      print*,' '
      print*,' '
      print*,'TABLE OF CALCULATED EF AS A FUNCTION OF TIME'
      PRINT*,'*******************************************'
      PRINT*,' '
      PRINT*,'TIME in s   ','         EF     ',
     & '      STDEV TOTAL FLUX %','   LAST  CHANGE EF in % '
            call FLUXES(NMAX,Z,DIFFCO2,DIFFHCO3,DIFFCO3,CO2START,
     &           HCO3START,CO3START,
     &           FLUXCO2, FLUXHCO3, FLUXCO3, FLUXTOTAL,
     &           AVFLUXCO2, AVFLUXTOTAL,EFCALCBC,STDEVFTP)
            PRINT*,' '
            print*,' WITH INITIALICED PROFILES'
            PRINT*,TIMEELAPSED,EFCALCBC,STDEVFTP
            PRINT*,' '
c
c
c     ---------------------------------------------
C     BIG LOOP FOR ENTIRE DRIVER KINETIC ENHANCEMENT
c     ---------------------------------------------
```

```
78921     IF(COUNT2.EQ.1000) THEN
             TIMEELAPSED=COUNT*DTDIFF
             EFCALCBCOLD=EFCALCBC
             call FLUXES(NMAX,Z,DIFFCO2,DIFFHCO3,DIFFCO3,CO2START,
     &             HCO3START,CO3START,
     &             FLUXCO2, FLUXHCO3, FLUXCO3, FLUXTOTAL,
     &             AVFLUXCO2, AVFLUXTOTAL,EFCALCBC,STDEVFTP)
             CHANGEEF=ABS(EFCALCBCOLD-EFCALCBC)
             IF(CHANGEEF.EQ.0) THEN
                PCHANGEEF=0
             ELSE
                PCHANGEEF=100*CHANGEEF/EFCALCBC
             ENDIF
             IF(PCHANGEEF.LT.EFACC.AND.STDEVFTP.LT.MASSBALACC) THEN
                EXIT=.TRUE.
           ENDIF
             COUNT2=0
             PCO2WATER=CO2start(NMAx)/HENRYCO2
             DELTAPCO2=(pco2water-patmco2)*1e6
c             value in 10E6 atm equals ppm
             print*,'deltapco2 ',deltapco2
             print*,'pHbulk phase',phstart(nmax)
             PRINT*,TIMEELAPSED,EFCALCBC,STDEVFTP,PCHANGEEF
          ENDIF
c
C       DIFFUSING CO2
C
             DO 1219 L=1,NMAX
                SPECIES(L)=CO2Start(L)
 1219        CONTINUE
C
          CALL DIFFUSEKI(species,DTDIFF,NMAX,Z,DIFFCO2,DIFFER,DELTA)
             DO 1119 L=1,NMAX
                CO2STOP(L)=species(L)
 1119        CONTINUE
c
C            REPLENISHING TOP LAYER WITH CO2
             CO2STOP(1)=CO2INTERFACE
C      DIFFUSING HCO3
c
C
             DO 1220 L=1,NMAX
                SPECIES(L)=HCO3start(L)
 1220        CONTINUE
C
             CALL DIFFUSEKI(species,DTDIFF,NMAX,Z,DIFFHCO3,DIFFER,DELTA)
             DO 1120 L=1,NMAX
                hco3STOP(L)=species(L)
 1120        CONTINUE
c
c
```

```
C      DIFFUSING CO3
C
            DO 1221 L=1,NMAX
               SPECIES(L)=CO3START(L)
 1221       CONTINUE
c
           CALL DIFFUSEKI(SPECIES,DTDIFF,NMAX,Z,DIFFCO3,DIFFER,DELTA)
           DO 1121 L=1,NMAX
              CO3STOP(L)=species(L)
 1121      CONTINUE
c
C
C     BIG LOOP FOR REACTION IN EACH TIMESTEP
c
            maxchange=0
c           print*,'maxchange  = ',maxchange
c
       DO 1811 K=1,NMAX
c      REACTION IN EACH LAYER
C
C     ASSIGNING ARRAY DATA TO INPUT DATA
C
          CO2IN=CO2STOP(k)
          HCO3IN=HCO3STOP(K)
          CO3IN=CO3STOP(K)
          CACONC=CAPARTITION(K)
          CTSAVE=CO2STOP(K)+HCO3STOP(K)+CO3STOP(K)
          phin=phstart(k)
 1020     minph=1
           DTREACT=DTDIFF/DIVIDETREACT
C
c
       DO 7002 I=1,DIVIDETREACT
C         SUBLOOP FOR MULTIPLE REACTIONS IN ONE DIFFUSIONTIMESTEP
C
            CALL REACT(CO2IN,CO2R,HCO3IN,HCO3R
     &      ,CO3IN,CO3R,PHIN,PHR,DTREACT,PATH,
     &       KCO2,KOH, KCAF,K1,K2,KW,KB,caconc,BT,ALK)
c
C         TAKING RESULT OF REACT AS INPUT FOR NEXT
C         TIMESTEP & REPLENISHING CO2 @ INTERFACE
C
          IF(K.eq.1) THEN
             CO2IN=CO2INTERFACE
          ELSE
             CO2IN=CO2R
          ENDIF
          HCO3IN=HCO3R
          CO3IN=CO3R
          PHIN=PHR
c
```

```
C
 7002  CONTINUE
C            CLOSING LOOP FOR MULTIPLE REACTIONS IN ONE
C            DIFFUSION STEP
C
c          COPYING DATA BACK IN PROPER PLACE OF ARRAY
c
           CO2STOP(K)=CO2R
           HCO3STOP(K)=HCO3R
           CO3STOP(K)=CO3R
           PHSTOP(K)=PHR
           CTSTOP(K)=CTSAVE
c
C        -----------------------------------------------
C          CALCULATING ABSOLUT  AND PERCENTAGE CHANCES
C          AND TIME OF STABILITY
c        -------------------------------------------
C
           DPH(K)=ABS(PHSTOP(K)-PHSTART(K))
           DCO2(K)=ABS(CO2STOP(K)-CO2START(K))
           DHCO3(K)=ABS(HCO3STOP(K)-HCO3START(K))
           DCO3(K)=ABS(CO3STOP(K)-CO3START(K))
           DCT(K)=ABS(CTSTOP(K)-CTSTART(K))
c
           IF (DPH(K).EQ.O) THEN
              PDPH(K)=0
           ELSE
              PDPH(K)=100*DPH(K)/PHSTOP(K)
           ENDIF
c
           IF (DCO2(K).EQ.O) THEN
              PDCO2(K)=0
           ELSE
              PDCO2(K)=100*DCO2(K)/CO2STOP(K)
           ENDIF
c
           IF (DHCO3(K).EQ.O) THEN
              PDHCO3(K)=0
           ELSE
              PDHCO3(K)=100*DHCO3(K)/HCO3STOP(K)
           ENDIF
c
           IF (DCO3(K).EQ.O) THEN
              PDCO3(K)=0
           ELSE
              PDCO3(K)=100*DCO3(K)/CTSTOP(K)
           ENDIF
c
           IF (DCT(K).EQ.O) THEN
              PDCT(K)=0
```

```
          ELSE
             PDCT(K)=100*DCT(K)/CTSTOP(K)
          ENDIF
C
c     FINDIG OUT THE MAXIMUM CHANGE AND SORTING AGAINST MAXCHANGE
c
C
          IF (PDCO2(K).GT.MAXCHANGE) THEN
             MAXCHANGE=PDCO2(K)
          ENDIF
          IF (PDHCO3(K).GT.MAXCHANGE) THEN
             MAXCHANGE=PDHCO3(K)
          ENDIF
          IF (PDCO3(K).GT.MAXCHANGE) THEN
             MAXCHANGE=PDCO3(K)
          ENDIF
          IF (PDPH(K).GT.MAXCHANGE) THEN
             MAXCHANGE=PDPH(K)
          ENDIF
          IF (PDCT(K).GT.MAXCHANGE) THEN
             MAXCHANGE=PDCT(K)
          ENDIF
c
c
 1811  CONTINUE
c       closing loop reaction
c
c     checking maximum change of all layers in this timestep
c     against stablechange
c
       EQREACHED=.FALSE.
          IF(MAXCHANGE.GT.STABLECHANGE) THEN
             LASTCHANGESTEP=COUNT
          ELSE
             EQREACHED=.TRUE.
          Endif
       TIMEELAPSED=TIMEELAPSED + DTDIFF
       COUNT=COUNT+1
       count2=count2+1
c
C     TAKING RESULTS AS INPUT NEW ITERATION
C
            DO 1419 L=1,NMAX
               CO2Start(L)=CO2STOP(L)
               HCO3Start(L)=HCO3STOP(L)
               CO3Start(L)=CO3STOP(L)
               CTSTART(L)=CTSTOP(L)
               PHSTART(L)=PHSTOP(L)
 1419       CONTINUE
c
```

```
      IF (TIMEELAPSED.LT.MAXTIME.AND..NOT.EXIT) then
         GOTO 78921
C        QUASI WHILE CONDITION TIMEMAX
      endif
c
c
c     copying species data back to array data
c     AND CALCULATING TSTEPMAX
C
c
      IF (COUNT.GT.maxsizetime) THEN
          REALCOUNT=COUNT
          COUNT=maxsizetime
      ENDIF
C
       TSTEPMAX=COUNT
C
C
C
c       PRINT*,'ASSIGNING DATA INTO ARRAYS'
c       PRINT*,'COUNT = ',COUNT
c       pause

            DO 1115 I=1,NMAX
               CO2(I,COUNT)=CO2STOP(I)
               HCO3(I,COUNT)=HCO3STOP(I)
               CO3(I,COUNT)=CO3STOP(I)
               ph(I,COUNT)=PHSTOP(I)
               CT(I,COUNT)=CO2STOP(I)+HCO3STOP(I)+CO3STOP(I)
               CHANGEPH(I,COUNT)=DPH(I)
               CHANGECO2(I,COUNT)=DCO2(I)
               CHANGEHCO3(I,COUNT)=DHCO3(I)
               CHANGECO3(I,COUNT)=DCO3(I)
               CHANGECT(I,COUNT)=DCT(I)
               PCHANGEPH(I,COUNT)=PDPH(I)
               PCHANGECO2(I,COUNT)=PDCO2(I)
               PCHANGEHCO3(I,COUNT)=PDHCO3(I)
               PCHANGECO3(I,COUNT)=PDCO3(I)
               PCHANGECT(I,COUNT)=PDCT(I)
 1115       CONTINUE
c
      ENDIF
C        CLOSING IF CONDITION FOR WHOLE DRIVER KINETIC NO HISTORY
C
C
C     --------------------------------------------------------
C     |  END OF DRIVER KINETIC ENHANCEMENT NO TIME HISTORY    |
C     --------------------------------------------------------
c
C     assigning endconcentrations and calling fluxes routine
C
```

```
       DO 1911 N=1,NMAX
          CO2END(N)=CO2(N,COUNT)
          HCO3END(N)=HCO3(N,COUNT)
          CO3END(N)=CO3(N,COUNT)
1911  CONTINUE
      call FLUXES(NMAX,Z,DIFFCO2,DIFFHCO3,DIFFCO3,CO2END,HCO3END,CO3END,
     &            FLUXCO2, FLUXHCO3, FLUXCO3, FLUXTOTAL,
     &            AVFLUXCO2, AVFLUXTOTAL,EFCALCBC,STDEVFTP)
c
c
C     ---------------------------------------------------------
C     |                OUTPUT MODUL                           |
c     |                                                       |
C     ---------------------------------------------------------
C        OUTPUT TABLE CONCENTRATIONS
c
         PRINT*,'***********************'
         PRINT*,'TABLE OF CONCENTRATIONS'
         PRINT*,'***********************'
         PRINT*,' '
         IF(DECICION.NE.'K') THEN
            TIMEELAPSED=DTDIFF*TSTEPMAX
         ENDIF
         PRINT*,'WITH ',ENHANCEMENT,' ENHANCEMENT'
         PRINT*,' '
         IF(DECICION.NE.'K') THEN
           PRINT*,'AFTER',TSTEPMAX,' TIMESTEPS'
           Print*,' '
         ENDIF
         PRINT*,'AND ',TIMEELAPSED,' SECONDS'
         PRINT*,' '
         PRINT*,'THE END-CONCENTRATION PROFILES WERE'
         PRINT*,' '
c
c
c     ---------------------------------------------------
C     OUTPUT TABLE FOR PURE DIFFUSION, CO2-VALUES ONLY
C     --------------------------------------------------
c
      PRINT *,'LAYER   ','  CO2   ','      HCO3   ','     CO3  ',
     &'       CT   ','       pH'
      PRINT*,' '
      DO 88 I=1,NMAX
          PRINT(112),I,CO2(I,TSTEPMAX),HCO3(I,TSTEPMAX)
     &            ,CO3(I,TSTEPMAX),CT(I,TSTEPMAX),
     &            PH(I,TSTEPMAX)
 88    CONTINUE
 112   FORMAT(I7,4E12.4,F8.3)


         PRINT*,' '
```

```
         print*,' '
         PRINT*,'********************************'
         print*,'TABLE OF CALCULATED FLUX VALUES'
         print*,'*******************************'
         PRINT*,' '
         Print*,'ALL FLUXES IN MOL PER SQUARE METER AND SECOND '
         PRINT*,'POSITIVE FLUXES ARE DOWNWARDS'
         PRINT*,'FLUX'
         print*,'TO LAYER','      FLUX CO2 ','     FLUXHCO3   ',
     &   '    FLUXCO3 ', '       TOTAL FLUX'
          PRINT*,' '
            DO 89 I=2,NMAX
                  PRINT(115),I,FLUXCO2(I),FLUXHCO3(I)
     &           ,FLUXCO3(I),FLUXTOTAL(I)
 89          CONTINUE
 115         FORMAT(I9,4E15.4)
         PRINT*,' '
         PRINT*,'THE CALCULATED ENHANCEMENT FACTOR CALCULATED AS '
         PRINT*,'FLUXTOTAL OVER FLUXCO2 @ NO REACTION IS'
         PRINT*,' '
         PRINT*,EFCALCBC
         PRINT*,' '
         print*,'COMPARED TO THE EQUILIBRIUM ENHANCEMENT FACTOR OF'
         PRINT*,' '
         PRINT*,EQUILEF
         print*,' '
         PRINT*,'THE STANDART DEVIATION IN PERCENT OF THE '
         PRINT*,'TOTAL FLUX OVER THE LAYERS IS '
         print*,' '
         PRINT*,STDEVFTP
         print*,' '
C
C
C
C
C     ---------------------------------
C     OUTPUT END CONCENTRATION PROFILES
C     ---------------------------------
C
         PRINT*,'CREATE AN OUTPUT FILE OF CALCUL. PROFILES (Y/N)'
         READ*, STEUER
         IF (STEUER.EQ.'Y')THEN
            PRINT*, 'PLEASE ENTER OUTPUT FILENAME (8 CHARACTER)'
            READ*,OUTNAME
            OPEN(UNIT=16,FILE=OUTNAME, STATUS='NEW')
            IF (DECICION.EQ.'P') THEN
               DO 16500 I=1,NMAX
                  WRITE(16,*)CO2(I,COUNT)
16500             CONTINUE
            ELSE
            DO 17000 I=1,NMAX
```

```
                 WRITE(16,*)CO2(I,COUNT),HCO3(I,COUNT)
     &           ,CO3(I,COUNT),CT(I,COUNT),PH(I,COUNT)
C        PRINT*,CO2(I,COUNT),HCO3(I,COUNT),CO3(I,COUNT)
17000         CONTINUE
            CLOSE(UNIT=16)
            ENDIF
C           CLOSING IF FOR OUTPUT DECICION
          ENDIF
C           CLOSING IF OUTPUT PROFILES
C
c
c     ----------------
c     output of fluxes
c     ----------------
c
         PRINT*,'CREATE AN OUTPUT FILE OF CALCUL. FLUXES (Y/N)'
         READ*, STEUER
         IF (STEUER.EQ.'Y')THEN
            PRINT*, 'PLEASE ENTER OUTPUT FILENAME (8 CHARACTER)'
            READ*,OUTNAME
            OPEN(UNIT=16,FILE=OUTNAME, STATUS='NEW')
            DO 17001 I=2,NMAX
                 WRITE(16,*)FLUXCO2(I),FLUXHCO3(I)
     &           ,FLUXCO3(I),FLUXTOTAL(I)
17001         CONTINUE
            CLOSE(UNIT=16)
            ENDIF
C           CLOSING IF OUTPUT FLUXES
c
          PRINT*,'DO YOU WANT ANOTHER ITERATION (Y/N)'
          READ *,STEUER
          IF (STEUER.EQ.'Y') GOTO 20000
C    -----------------------------------------------
C   |     END OF OUTPUT                             |
C     ----------------------------------------------
C
       STOP
C
C     ******************END MAIN*******************************
C     *********************************************************
c
      END
C
C
C     ---------------------------------------------------------
C     |  FUNCTION ACTIVITY  DETERMINES REALTIVE ACTIFITY       |
C     |  OF CARBONIC ANHYDRASE AS FUNCTION OF PH               |
C     |  USING ACID BASE CHEMISTRY TO MODEL ENCYME ACTIFITY    |
c     |  INPUT PH, OUTPUT ACTIVITY IN FRACTION OF 1(=100%)     |
c     ----------------------------------------------------------
      FUNCTION ACTIVITY (PHIN)
```

```
      Real PHIN,KS
      REAL ACTIVITY
      KS=10**(-7.5)
      ACTIVITY=1/(1+(10**(-1*PHIN))/KS)
      RETURN
      END
C     ---------------------------------------------------------
C     END OF FUNCTION ACTIVITY
C     --------------------------------------------------------
C
C
C     ---------------------------------------------------------
C     |  SUBROUTINE ADJUSTPH                                  |
c     |  CALCULATES THE NEW PH VALUE AFTER REACT ROUTINE      |
c     |  BY SOLVING THE ALKALINITY EXPRESSION WITH            |
c     |  APPROXIMATION 'CTEQPOOL'                             |
C     |  SUBROUTINES CALLED: SOLVEALKREACT                    |
C     |  SUBROUTINES IMPLIED: ALPHACTPOOL                     |
C     |                       DELTAALKREACT                   |
c     ----------------------------------------------------------
      SUBROUTINE ADJUSTPH (HCO3IN,HCO3A,CO3IN,CO3A,PHIN,PHA,
     &                     K1,K2,KW,KB,BT,ALK)
C
      REAL  HCO3IN,CO3IN,PHIN
      REAL  HCO3A,CO3A,PHA,HA
      REAL K1,K2,KW,KB,BT,ALK
      REAL MINPH,MAXPH,PHACC
      MINPH=1
      MAXPH=14
      PHACC=0.000001
      CTEQPOOL=HCO3IN+CO3IN
      PHA=SOLVEALKREACT(MINPH,MAXPH,PHACC,CTEQPOOL,
     &                   ALK,BT,K2,KW,KB)
      HA=10**(-1*PHA)
      HCO3A=(1-ALPHACTPOOL(HA,K2))*CTEQPOOL
      CO3A=CTEQPOOL-HCO3A
      RETURN
      END
C     ---------------------------------------------------------
C                         END OF ADJUSTpH
C     ---------------------------------------------------------
C
C
C     -------------------------------------------------------
C     | FUNCTION ALPHAO TO CALCULATE INOIZATION FRACTION     |
C     -----------------------------------------------------
C
         FUNCTION ALPHA0 (H,K1,K2)
         REAL H,K1,K2
         ALPHA0 =1/(1+K1/H+K1*K2/H**2)
         RETURN
```

```
         END
C     -----------------------------------------------------
C     END FUNCTION ALPHA0
C     -------------------------------------------------------
C
C
C
C      -------------------------------------------------------
C       FUNCTION ALPHA1, SAME USAGE AS ABOVE
C     ------------------------------------------------------
c
         FUNCTION ALPHA1(H,K1,K2)
         REAL H,K1,K2
         ALPHA1=1/(H/K1+1+K2/H)
         RETURN
         END
C       -----------------------------------------------------
c     end of alpha1
c     ----------------------------------------------------
c
C
C     -----------------------------------------------------------
C     FUNCTION ALPHA2, THE SAME AS ABOVE
c     -------------------------------------------------------
c
C
         FUNCTION ALPHA2(H,K1,K2)
         REAL H,K1,K2
         ALPHA2=1/(H**2/(K1*K2)+H/K2+1)
         RETURN
         END
C       -----------------------------------------------------
C     end of alpha 2
c     ---------------------------------------------------
c
c
C     ------------------------------------------------------
C     FUNCTION ALPHABOR(H,KB)
C     IONICATION FRACTION BORATE ION
C     -----------------------------------------------------
       FUNCTION ALPHABOR(H,KB)
       REAL KB,H,ALPHA
       ALPHA=1/(1+(H/KB))
       ALPHABOR=ALPHA
       RETURN
       END
C     ------------------------------------------------------
C     END OF ALPHABOR
C     ------------------------------------------------------
C     ------------------------------------------------------
C     |  FUNCTION ALPHACTPOOL(H,K2)                         |
```

```
C     |  CALCULATES THE DISTIRIBUTION BETWEEN HCO3 AND CO3     |
C     |  (THE CTEQUILIBRIUM POO   AS A FUNCTION OF H           |
C     |  USAGE IN ADJUST PH                                    |
C     -------------------------------------------------------
       FUNCTION ALPHACTPOOL(H,K2)
       REAL K2,H,ALPHA
       ALPHA=1/(1+(H/K2))
       ALPHACTPOOL=ALPHA
       RETURN
       END
C     --------------------------------------------------------
C     END OF ALPHACTPOOL
C     --------------------------------------------------------
C      ---------------------------------------------------------
C     |   FUNCTION CTEQUIL                                     |
C     |   CALCULATES THE CT VALUE IN EQUILIBRIUM WITH THE      |
C     |   ATHMOSPHERE AND OPEN System with input ph            |
C     ----------------------------------------------------------
       FUNCTION CTEQUIL (X,K1,K2,HENRYCO2)
       REAL K1,K2,H,X
       REAL HENRYCO2, PATMCO2
       PATMCO2=340E-6
       H=10**(-1*X)
       CTEQUIL=(1/ALPHA0(H,K1,K2))*HENRYCO2* PATMCO2
       RETURN
       END
C     ---------------------------------------------------------
C     END OF CTEQUIL
C     ---------------------------------------------------------
C
C
C     -----------------------------------------------------------
C     |  SUBROUTINE  DIFFUSEEQ                                   |
C     |  DIFFUSES THE SPECIES BY ONE TIME STEP ACCORDING TO EMMERSON|
C     |  CALLED IN EQILIBRIUM ENHANCEMENT, ASSUMES INSTANTANIOUS    |
C     |  REPLENISHING OF TOP LAYER, SO THAT CHANGE IN LAYER #1 IS   |
C     |  NOT CALCULATED, TO SPEED CALCULATION UP                    |
C     ----------------------------------------------------------
       SUBROUTINE DIFFUSEEQ (SPECIES,DTDIFF,NMAX,Z,DIFF,DIFFER,DELTA)
       REAL R,Z,DTDIFF
       INTEGER NMAX,I
       DIMENSION SPECIES(NMAX)
       DIMENSION DIFFER(NMAX)
       DIMENSION DELTA(NMAX)
        R=DIFF/((Z/NMAX)**2)
      DO 2000 I=2,(NMAX-1)
        DIFFER(I)=SPECIES(I-1)+SPECIES(I+1)
     &-2*SPECIES(I)
       DELTA(I)=R*DIFFER(I)*DTDIFF
2000   CONTINUE
c     calculating new concentrations
```

```
       do 4000 I=2,(NMAX-1)
          SPECIES(I)=SPECIES(I)+DELTA(I)
4000   CONTINUE
       RETURN
       END
C     ------------------------------------------------------------
C     END OF DIFFUSEEQ
C     ------------------------------------------------------------
C
C     ------------------------------------------------------------
C     |  SUBROUTINE  DIFFUSEKI                                    |
C     |  DIFFUSES THE SPECIES BY ONE TIME STEP ACCORDING TO EMMERSON|
C     |  CALLEN IN DRIVER KINETIC ENHANCEMENT, CONSIDERS CHANGE IN |
C     |  TOP LAYER                                                |
C     ------------------------------------------------------------
       SUBROUTINE DIFFUSEKI (SPECIES,DTDIFF,NMAX,Z,DIFF,DIFFER,DELTA)
       REAL R,Z,DTDIFF
       INTEGER NMAX,I
       DIMENSION SPECIES(NMAX)
       DIMENSION DIFFER(NMAX)
       DIMENSION DELTA(NMAX)
        R=DIFF/((Z/NMAX)**2)
c     calculating changes top layer
C
       DIFFER(1)=SPECIES(2)-SPECIES(1)
       DELTA (1)=R*DIFFER(1)*DTDIFF
C
C     calculating changes other layers
C
      DO 2000 I=2,(NMAX-1)
        DIFFER(I)=SPECIES(I-1)+SPECIES(I+1)
     &-2*SPECIES(I)
       DELTA(I)=R*DIFFER(I)*DTDIFF
2000   CONTINUE
C
c     calculating new concentrations
C
       do 4000 I=1,(NMAX-1)
          SPECIES(I)=SPECIES(I)+DELTA(I)
4000   CONTINUE
       RETURN
       END
C     ------------------------------------------------------------
C     END OF DIFFUSEKI
C     ------------------------------------------------------------
c
C
C     -------------------------------------------------------
C     |  FUNCTION DELTAALKOPEN DETERMINES TOGETHER WITH       |
C     |  RTFLSP ROUTINE THE H CONCENTRATION OF TOP            |
C     |  BOUNDARY                                             |
```

```
C      -------------------------------------------------------
       FUNCTION DELTAALKOPEN (PH,ALK,BT,K1,K2,KW,KB,HENRYCO2)
       REAL K1,K2,KW,BT,H,ALK,ALPB,OH,KB
       REAL ph,ct,ALP1,ALP2,HENRYCO2
       H=10**(-1*ph)
       ALPB=ALPHABOR(H,KB)
       OH=KW/H
       CT=CTEQUIL(PH,K1,K2,HENRYCO2)
       alp1=alpha1(H,k1,k2)
       alp2=alpha2(h,k1,k2)
       ALKOPEN=ct*(alp1+2*alp2)+BT*ALPB-H+OH
       DELTAALKOPEN=ALKOPEN-alk
       RETURN
       END
C      ---------------------------------------------------------
C      END OF FUNCTION DELTAALKOPEN
C      ---------------------------------------------------------
C
C
C      --------------------------------------------------------
C      |  FUNCTION DELTAALKCLOSED                              |
C      |  CALCULATES H-CONCENTRATION IN EQUILIBRIUM TO FULFIL|
C      |  ALKALINITY RELATIONSHIP CLOSED SYSTEM !!!!!          |
C      |  INPUT PARAMETER PH VALUES                            |
C      --------------------------------------------------------
        FUNCTION DELTAALKCLOSED (PH,CT,ALK,BT,K1,K2,KW,KB)
C
        REAL CT,H,KW,BT,PH,K1,K2,OH,KB
        REAL DIFFER,ALK,ALKITER,ALP1,ALP2,ALPB
        H=10**(-1*PH)
        OH=KW/H
        ALP1=ALPHA1(H,K1,K2)
        ALP2=ALPHA2(H,K1,K2)
        ALPB=ALPHABOR(H,KB)
        ALKITER=CT*(ALP1+2*ALP2)+BT*ALPB-H+OH
        DIFFER=ALKITER-ALK
        DELTAALKCLOSED=DIFFER
        RETURN
        END
C      ----------------------------------------------------------
C      END OF FUNCTION DELTAALKCLOSED
C      --------------------------------------------------------
C
C      --------------------------------------------------------
C      |  FUNCTION DELTAALKREACT                               |
C      |  CALCULATES H-CONCENTRATION IN EQUILIBRIUM TO FULFIL|
C      |  ALKALINITY RELATIONSHIP CLOSED SYSTEM KINETIC        |
C      |  ENHANCEMENT. NEGLECTING H2CO3 AND TAKING HCO3 AND    |
C      |  CO3 AS THE CTEQPOOL. THIS APPROXIMATION IS VALID     |
C      |  AS LONG AS PH >7.                                    |
C      |  INPUT PARAMETER PH,CTEQPOOL                          |
```

```
C     ------------------------------------------------------
       FUNCTION DELTAALKREACT (PH,CTEQPOOL,ALK,BT,K2,KW,KB)
       REAL CTEQPOOL,H,KW,BT,PH,OH,K2,KB
       REAL DIFFER,ALK,ALKITER,ALPHAPOOL,ALPB
       H=10**(-1*PH)
       OH=KW/H
       ALPHAPOOL=ALPHACTPOOL(H,K2)
       ALPB=ALPHABOR(H,KB)
       ALKITER=CTEQPOOL*(1+ALPHAPOOL)+BT*ALPB-H+OH
       DIFFER=ALKITER-ALK
       DELTAALKREACT=DIFFER
       RETURN
       END
C     ---------------------------------------------------------
C     END OF FUNCTION DELTAALKCLOSED
C     -------------------------------------------------------

C     ---------------------------------------------------------
C     |  SUBROUTINE EQUILIBRATE  DETERMINES TOGETHER WITH      |
C     |  DIFFUSE DRIVER PH VALUE OF THE BOX AND                |
C     |  SPECIATION OF CT ASSUMING EQUILIBRIUM                 |
c     ----------------------------------------------------------
      SUBROUTINE EQUILIBRATE(CO2IN,HCO3IN,CO3IN,CO2EQ,HCO3EQ,
     &                    CO3EQ,PHIN,PHEQ,MINPH,MAXPH,PHACC,
     &                    K1,K2,KW,BT,ALK,KB)
c
      REAL CT,ALP0,ALP1,ALP2,K1,K2,H,KW,BT,ALK,KB
      REAL CO2IN,HCO3IN,CO3IN,PHIN
      REAL CO2EQ,HCO3EQ,CO3EQ,PHEQ
      REAL PH,PHACC,MINPH,MAXPH
      CT=CO2IN+HCO3IN+CO3IN
      PH=SOLVEALKCLOSED (MINPH,MAXPH,PHACC,CT,
     &                   ALK,BT,K1,K2,KW,KB)
      H=10**(-1*PH)
      ALP0=ALPHA0(H,K1,K2)
      ALP1=ALPHA1(H,K1,K2)
      ALP2=ALPHA2(H,K1,K2)
      CO2EQ=CT*ALP0
      HCO3EQ=CT*ALP1
      CO3EQ=CT*ALP2
       PHEQ=PH
      RETURN
      END
C     -------------------------------------------------------
C     END OF subroutine equilibrate
C    -----------------------------------------------------------
C
c     ------------------------------------------------
C     | FUNCTION EQUILIBRIUMEF                         |
C     | FINDS EQUILIRIUMENHANCEMENTFACTOR              |
c     | PRORAMED: 01/05/94                             |
```

```
C     | INPUT: CONCENTRATIONS                                 |
C      -------------------------------------------------------
      FUNCTION EQEF(CO2T,CO2B,HCO3T,HCO3B,CO3T,CO3B,CTDEFICIT,
     &         DIFFCO2,DIFFHCO3,DIFFCO3)
      REAL  CO2T,CO2B,HCO3T
      REAL  DIFFCO2, DIFFHCO3, DIFFCO3
      REAL  HCO3B,CO3T,CO3B,CTDEFICIT
      REAL  GRADIENTCO2,GRADIENTHCO3,GRADIENTCO3
      GRADIENTCO2=CO2T-CO2B
      GRADIENTHCO3=HCO3T-HCO3B
      GRADIENTCO3=CO3T-CO3B
      IF (CTDEFICIT.EQ.1) THEN
      PRINT*,'CT-DEFICIT=1 MEANS NO TRANSPORT AND NO ENHANCEMENT'
      PRINT*,'EQUILIBRIUMENHANCEMENTFACTOR IS NOT DEFINED'
      EQEF=1
      RETURN
      ENDIF
      EQEF=(GRADIENTCO2*DIFFCO2+GRADIENTHCO3*DIFFHCO3+
     &    GRADIENTCO3*DIFFCO3)/(GRADIENTCO2*DIFFCO2)
      RETURN
      END
C
C     --------------------------------------------------------
C     |  END EQUILIBRIUMENHANCEMENTFACTOR                      |
C     |                                                        |
C     -------------------------------------------------------
C
C     --------------------------------------------------------
C     |  SUBROUTINE FLUXES                                      |
c     |  CALCULATES THE FLUXES CAUSED BY THE SPECIES GRADIENTS |
c     ---------------------------------------------------------
      SUBROUTINE FLUXES(NMAX,Z,DIFFCO2,DIFFHCO3,DIFFCO3,
     &                  CO2END,HCO3END,CO3END,
     &                  FLUXCO2,FLUXHCO3,FLUXCO3,FLUXTOTAL,
     &                  AVFLUXCO2, AVFLUXTOT,EFCALCBC,STDEVFTP)
      REAL DIFFCO2, DIFFHCO3,DIFFCO3
      REAL PVCO2,PVHCO3,PVCO3
      REAL Z
      REAL AVFLUXCO2, AVFLUXTOT,FLUXCO2BC,EFCALCBC
      REAL SUMFLUXCO2,SUMFLUXTOTAL
      INTEGER NMAX,N
      REAL STDEVFTP,STDEVFT,SQSUMSTDEVFT,SQUARE
      DIMENSION CO2END(nmax)
      DIMENSION HCO3END(nmax)
      DIMENSION CO3END(nmax)
      DIMENSION FLUXCO2(nmax)
      DIMENSION FLUXHCO3(nmax)
      DIMENSION FLUXCO3(nmax)
      DIMENSION FLUXTOTAL(nmax)
      real nreal
C
```

```
      SUMFLUXCO2=0
      SUMFLUXTOTAL=0
      SQSUMSTDEVFT=0
      SQUARE=0
      nreal=nmax
      PVCO2=1000*DIFFCO2/(Z/(NMAX-1))
      PVHCO3=1000*DIFFHCO3/(Z/(NMAX-1))
      PVCO3=1000*DIFFCO3/(Z/(NMAX-1))
C     Factor 1000 for consistency in units from Mol/l to Mol/m**2
C
C  CALCULATING FLUXES
C
      DO 10 N=2,NMAX
         FLUXCO2(N)=PVCO2*(CO2END(N-1)-CO2END(N))
         FLUXHCO3(N)=PVHCO3*(HCO3END(N-1)-HCO3END(N))
         FLUXCO3(N)=PVCO3*(CO3END(N-1)-CO3END(N))
         FLUXTOTAL(N)=FLUXCO2(N)+FLUXHCO3(N)+FLUXCO3(N)
         SUMFLUXCO2=SUMFLUXCO2+FLUXCO2(N)
         SUMFLUXTOTAL=SUMFLUXTOTAL+FLUXTOTAL(N)
 10   CONTINUE
c
      AVFLUXCO2=SUMFLUXCO2/(Nreal-1)
      AVFLUXTOTAL=SUMFLUXTOTAL/(Nreal-1)
C
C   CALCULATING STANDART DEVIATION TOTAL FLUXES
C
      DO 20 N=2,NMAX
         SQUARE=(AVFLUXTOTAL-FLUXTOTAL(N))**2
         SQSUMSTDEVFT=SQSUMSTDEVFT+SQUARE
 20   CONTINUE
      STDEVFT=SQRT(SQSUMSTDEVFT/(nreal-2))

      IF(AVFLUXTOTAL.EQ.0) THEN
         STDEVFTP=0
      ELSE
         STDEVFTP=abs(STDEVFT*100/AVFLUXTOTAL)
      ENDIF
C
      IF ((CO2END(1)-CO2END(NMAX)).EQ.O) THEN
         FLUXCO2BC=0
         EFCALCBC=1
      ELSE
      fluxco2bc=(Diffco2*1000/z)*(co2end(1)-co2end(nmax))
      EFCALCBC=AVFLUXTOTAL/FLUXCO2BC
      ENDIF
      RETURN
      END
C
C     -------------------------------------------------------
C     END OF ROUTINE FLUXES
C     -------------------------------------------------------
```

```
C
C     ---------------------------------------------------------
C     | SUBROUTINE newconstants                                 |
c     | CALCULATES THE NEW CONSTANTS FOR EQUILIBRIUM AND        |
c     | THE CO2 HYDRATION  Step                                 |
C     | source: relationships Stumm morgan pages 205            |
C     | kinetic constants by arrenius equation and FM3 ans SM   |
C     | DIFFUSIVITY BY WATER VISKOSITY (PHIL P197 AND CRC       |
C     | EMPIRICAL VISKOSIOTY EQUATION                           |
C     | ASSUMING CHLORINITY OF 19%                              |
c     ----------------------------------------------------------
C
      Subroutine NEWCONSTANTS (TEMPABS,K1,K2,HENRYCO2,KB,
     &               DIFFCO2, DIFFHCO3, DIFFCO3,
     &               KCO2,KOH,KCAF)
c
      REAL TEMPABS,TEMPC,K1,K2,HENRYCO2,KB
      REAL PK1,PK2,PKHENRY,PKB,viskosity,DIFFTCORRF
      REAL KCO2,KOH,KCAF
      REAL LNKCO2, LNKOH,LNKCAF
C
      DIFFCO2=1.94E-9
      DIFFHCO3=0.947E-9
      DIFFCO3=DIFFHCO3
c
      PRINT*,' ENTERING NEWVALUES WITH INPUT'
      PRINT*,'K1 ', K1
      PRINT*,'K2 ', K2
      PRINT*,'KB', KB
      PRINT*,'HENRYCO2 ', HENRYCO2
      PRINT*,' DIFFCO2 ', DIFFCO2
      PRINT*,' DIFFHCO3 ', DIFFHCO3
      PRINT*,' DIFFCO3 ', DIFFCO3
      PRINT*,' KCO2 ', kco2
      PRINT*,' KOH ',KOH
      PRINT*,' KCAF ',KCAF
c
      TEMPC=TEMPABS-273
      PK1=3404.71/TEMPABS+0.032786*TEMPABS-15.22375
      PK2=2902.39/TEMPABS+0.02379*TEMPABS-7.7233
      PKHENRY=-1*(2385.73/TEMPABS-14.22100+1.570113E-2*TEMPABS)
      PKB=2291.9/TEMPABS+0.01756*TEMPABS-4.2402494
      LNKCO2=1/TEMPABS*(-9128.65)+27.35192
      LNKOH=1/TEMPABS*(-6872.34)+32.12553
      LNKCAF=1/TEMPABS*(-3860.06)+26.45475
C
      PRINT*,'PK1',PK1
      PRINT*,'PK2',PK2
      PRINT*,'PKB',PKB
      PRINT*,'PKHENRY ', PKHENRY
      PRINT*,'LNKCO2 ', LNKCO2
```

```
      PRINT*,'LNKOH',LNKOH
      PRINT*,'LNKCAF',LNKCAF
C
      viskosity=10**(1301/(834.6230+8.1855*TEMPC
     &    +0.00585*(TEMPC-20)**2) -1.30233)
      print*,' viskosity ', viskosity
      DIFFTCORRF=VISKOSITY**(-1.14)*0.87605
      DIFFCO2=DIFFCO2*DIFFtCORRF
      DIFFHCO3=DIFFHCO3*DIFFtCORRF
      DIFFCO3=DIFFHCO3
      K1=10**(-1*PK1)
      K2=10**(-1*PK2)
      KB=10**(-1*PKB)
      HENRYCO2=10**(-1*PKHENRY)
      KCO2=EXP(LNKCO2)
      KOH=EXP(LNKOH)
      KCAF=EXP(LNKCAF)
C
      PRINT*,'THE NEW CONSTANTS ARE: '
      PRINT*,'K1 ', K1
      PRINT*,'K2 ', K2
      PRINT*,'KB', KB
      PRINT*,'HENRYCO2 ', HENRYCO2
      PRINT*,' DIFFCO2 ', DIFFCO2
      PRINT*,' DIFFHCO3 ', DIFFHCO3
      PRINT*,' DIFFCO3 ', DIFFCO3
      PRINT*,' KCO2 ', KCO2
      PRINT*,' KOH ',KOH
      PRINT*,' KCAF ', KCAF
      RETURN
      END
c
C     ---------------------------------------------------------
C     END OF SUBROUTINE NEWCONSTANTS
C     ---------------------------------------------------------
C
C     ---------------------------------------------------------
C     |  SUBROUTINE REACT                                      |
c     |  CALCULATES THE TURNOVER AS WELL AS THE EFFECTS OF     |
c     |  THE CO2 HYDRATION DEHYDRATION AND HYDDROXOLATION      |
c     |  DEHYDROXOLATIONSTEP                                   |
C     |  CAN RUN ON IT'S ONE TO TEST KINETIC PARAMETER BUT THEN|
C     |  GEDANKENEXPERIMENT PURE WATER                         |
C     |  SUBBROUTINE ADJUST PH MUST THEN BE OFF                |
C     |  DATA AND EXAMPLE : STUMM MORGAN                       |
c     |  KINETIC DATA ADJUSTED TO RESULT IN SEAWATER EQUIL.    |
C     |  DIFFERENCIATING BETWEEN NORMAL, OH AND CA PATHWAY     |
c     ----------------------------------------------------------
C
      SUBROUTINE REACT(CO2IN,CO2R,HCO3IN,HCO3R
     &       ,CO3IN,CO3R,PHIN,PHR,DTREACT,PATH,
```

```
     &        KCO2,KOH, KCAF,K1,K2,KW,KB,CACONC,BT,ALK)
c
      REAL KW,K1,K2,KB,BT,ALK
      REAL CO2IN,HCO3IN,CO3IN,PHIN,HIN,OHIN
      REAL CO2R,HCO3R,CO3R,HR,PHR
      REAL HCO3A,CO3A,PHA
      REAL DTREACT
      REAL KCO2,KHCO3,KOH,K14,K41
      REAL DCO2DTN,DCO2DTOH,DCO2DTCA
      REAL DELTACO2
      REAL ACV,KCAF,KCAFEFF,KCABEFF
      REAL CACONC
      REAL SUMDCO2DT
      REAL KMM
      CHARACTER*12 PATH
c
      KHCO3=KCO2/K1
      K14=KOH
      K41=K14*KW/K1
C     KINETIC CONSTANTS FOR OH REACTION PATH
      HIN=10**(-1*PHIN)

c
C      CALCULATING VALUES AND NORMAL PATH
C
      OHIN=KW/HIN
      DCO2DTN=-1* KCO2*CO2IN+KHCO3*HCO3IN*HIN
      DCO2DTOH= -1*K14*CO2IN*OHIN + K41 * HCO3IN
C
C     SWITCH ON OFF CA
C
      IF (PATH.EQ.'CAON') THEN
C
C     CALCULATING ACTIVITY AND EFFECTIVE TURNOVER
C     OLD EQUATION WITH ACTIVITY AND ASSUMPTION OF
C     SATURATION OF ENZYME

      KMM=40E-3
C     Michaelis-Menten HALF SATURATION CONSTANT IN MOL
C     SOURCE CA BOOK P227
         ACV=ACTIVITY (PHIN)
         KCAFEFF=KCAF*ACV*CACONC/KMM
         KCABEFF=KCAFEFF*HIN/K1
         DCO2DTCA=-1*KCAFEFF*CO2IN + KCABEFF*HCO3IN
      ENDIF
c
      SUMDCO2DT=DCO2DTN+DCO2DTOH+DCO2DTCA
      DELTACO2=SUMDCO2DT*DTREACT
      CO2R=CO2IN +  DELTACO2
      HCO3R=HCO3IN - DELTACO2
      CO3R = CO3IN
```

```
C         NO EFFECT OF CO2-HYDRATION ON  CO3 IN FIRST TIMESTEP
      HR=HIN - DELTACO2
      PHR=LOG10(HR)*(-1)

C     CALLING ADJUSTPH TO READJUST THE PH WITH ALKALINITY
C     EXPRESION
C
      HCO3A=0
      CO3A=0
      CALL ADJUSTPH(HCO3R,HCO3A,CO3R,CO3A,PHR,PHA,
     &                     K1,K2,KW,KB,BT,ALK)
C     RENAMING THE CHANGED PARAMETERS
C
      HCO3R=HCO3A
      CO3R =CO3A
      PHR = PHA
      RETURN
      END
c     -------------------------------------------------------
c     END OF SUBROUTINE REACT
c     -------------------------------------------------------
C
C
C     ----------------------------------------------------
C     | FUNCTION SOLVEALKCLOSED                           |
C     | BISEKTION METHOD ROOTFINDER                       |
c     | SOLVER DELTAALKCLOSED TO ZERO                     |
C     | WITH OUTPUVALUE FOR PH                            |
c     ----------------------------------------------------
      FUNCTION solvealkclosed(x1,x2,xacc,CT,
     &                        ALK,BT,K1,K2,KW,KB)
c
      INTEGER MAXIT
      REAL rtflsp,x1,x2,xacc,CT
      REAL ALK,BT,K1,K2,KW,KB
      PARAMETER (MAXIT=100)
      INTEGER j
      REAL del,dx,f,fh,fl,swap,xh,xl
      fl=DELTAALKclosed(x1,CT,ALK,BT,K1,K2,KW,KB)
      fh=DELTAALKclosed(x2,CT,ALK,BT,K1,K2,KW,KB)
      if(fl*fh.gt.0.) then
         PRINT*,'root must be bracketed in solvealkc'
         print*,'enterd solveaklclosed values ='
         PRINT*,'PHMIN ',x1
         PRINT*,'PHMAX',x2
         PRINT*,'PHACC',xacc
         PRINT*,'CT ',CT
         PRINT*,'ALK',ALK
         PRINT*,'BT',BT
         PRINT*,'K1 ',K1
         PRINT*,'K2 ',K2
```

```
           print*,'Bracketvalues are ',fl,fh
           pause
        endif
        if(fl.lt.0.)then
          xl=x1
          xh=x2
        else
          xl=x2
          xh=x1
          swap=fl
          fl=fh
          fh=swap
 760    endif
        dx=xh-xl
        do 11 j=1,MAXIT
          rtflsp=xl+dx*0.5
          f=DELTAALKCLOSED(rtflsp,CT,ALK,BT,K1,K2,KW,KB)
          if(f.lt.0.) then
            del=xl-rtflsp
            xl=rtflsp
            fl=f
          else
            del=xh-rtflsp
            xh=rtflsp
            fh=f
          endif
          dx=xh-xl
          if(abs(del).lt.xacc.or.f.eq.0.)then
              SOLVEALKCLOSED=rtflsp
c             PAUSE
             return
          endif
11      continue
        pause 'sovealkclosed exceed maximum iterations'
        END

C       ------------------------------------------------------
C       END SOLVEALKCLOSED
C       ------------------------------------------------------
c
c
c
C
C       ----------------------------------------------------
C       | FUNCTION SOLVEALKOPEN                             |
C       | BISEKTION METHOD ROOTFINDER                       |
c       | SOLVER DELTAALKOPEN TO ZERO                       |
C       | WITH OUTPUVALUE FOR H-CONCENTRATION               |
C       | INPUT AND OUTPUTVALUES PH                         |
C       ----------------------------------------------------
        FUNCTION solvealkopen(x1,x2,xacc,ALK,BT,K1,K2,KW,KB,
```

```
     &                      HENRYCO2)
      INTEGER MAXIT
      REAL rtflsp,x1,x2,xacc
      REAL ALK,BT,K1,K2,KW,KB,HENRYCO2
      PARAMETER (MAXIT=100)
      INTEGER j
      REAL del,dx,f,fh,fl,swap,xh,xl
      fl=DELTAALKOPEN (x1,ALK,BT,K1,K2,KW,KB,HENRYCO2)
      fh=DELTAALKOPEN (x2,ALK,BT,K1,K2,KW,KB,HENRYCO2)
      if(fl*fh.gt.0.) pause 'root must be bracketed in rtflsp'
      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
        rtflsp=xl+dx*0.5
        f=DELTAALKOPEN(rtflsp,ALK,BT,K1,K2,KW,KB,HENRYCO2)
        if(f.lt.0.) then
          del=xl-rtflsp
         xl=rtflsp
 520     fl=f
        else
          del=xh-rtflsp
          xh=rtflsp
          fh=f
        endif
        dx=xh-xl
        if(abs(del).lt.xacc.or.f.eq.0.)then
           solvealkopen=rtflsp
           return
        endif
11    continue
      pause 'solvealkopen exceed maximum iterations'
      END

C     -------------------------------------------------------
C     |    END SOLVEALKOPEN                                 |
C     -------------------------------------------------------
C

C
C     -----------------------------------------------------
C     | FUNCTION SOLVEALKREACT                            |
C     | BISEKTION METHOD ROOTFINDER                       |
```

```
c     | SOLVES DELTAALKREACT  TO ZERO                         |
C     | WITH OUTPUVALUE FOR PH                                |
C     | INPUT AND OUTPUTVALUES PH                             |
c     -------------------------------------------------------
      FUNCTION SOLVEALKREACT(x1,x2,xacc,CTEQPOOL,ALK,BT,K2,KW,KB)
      INTEGER MAXIT
      REAL rtflsp,x1,x2,xacc,CTEQPOOL
      REAL ALK,BT,K2,KW,KB
      PARAMETER (MAXIT=100)
      INTEGER j
      REAL del,dx,f,fh,fl,swap,xh,xl
      fl=DELTAALKREACT(x1,CTEQPOOL,ALK,BT,K2,KW,KB)
      fh=DELTAALKREACT(x2,CTEQPOOL,ALK,BT,K2,KW,KB)
      if(fl*fh.gt.0.) then
         print*,'values are:'
         print*,'phmin = x1 ',x1
         print*,'phmax = x2 ',x2
         print*,'cteqpool ',cteqpool
         print*,'xacc  = phacc ',xacc
         print*,'layer number ',l
         print*,'count ',count
         pause 'root must be bracketed in SOLVEALKREACT'
      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
        rtflsp=xl+dx*0.5
        f=DELTAALKREACT(rtflsp,CTEQPOOL,ALK,BT,K2,KW,KB)
        if(f.lt.0.) then
          del=xl-rtflsp
          xl=rtflsp
          fl=f
        else
          del=xh-rtflsp
          xh=rtflsp
          fh=f
        endif
        dx=xh-xl
        if(abs(del).lt.xacc.or.f.eq.0.)then
            SOLVEALKREACT=rtflsp
c           PAUSE
          return
```

```
          endif
11      continue
        pause 'sovealkclosed exceed maximum iterations'
        END

C       -------------------------------------------------------
C       END SOLVEALKREACT
C       -------------------------------------------------------
```

2.: Code film penetration model

Here only the driver and the nessesary subroutines and functions are provided, everything else, exept the parameter definition is the same as in the implementation of the lamianr film model.

1. Driver:

```
C       ---------------------------------------
C       INPUT OF TSTEPMAX AND KIND OF TRANSPORT
C       ---------------------------------------
C
c       CHECKING WHAT TYPE OF TRANSPORT USER WANTS TO MODEL
C
C
c
           PRINT*,'PLEASE ENTER MEDIUM SURFACE RENEWAL TIME IN S'
           READ *,Smean
           dpenetrate=3.6*Sqrt(Diffco2*smean)
           print*,'penetration depth equals',dpenetrate
           PRINT*,'PLEASE ENTER LAYER THICKNESS IN M'
           READ*,Z
           maxiterlfm=1
C
C       ---------------------
C       END OF INPUT SEQUENCE
C       ---------------------
C       -------------------------------------------------------
C       | DRIVER FILM PENETRATION MODEL                       |
C       | CALCULATES THE CONCENTRATION PROFILE USING ROUTINES |
c       | REACT AND DIFFUSEK, INTEGRATES THE FLUXES, USING    |
C       | SIMP ROUTINE AND CALCULATES THE EF OVER TETA        |
C       ----------------------------------------------------
          IF(model.eq.'FILMPENETRATION') THEN
              PRINT*,' '
              PRINT*,'ENTERING DRIVER FILM PENETRATION'
           deltac=-1*(co2(1,0)-co2(nmax,0))
```

```
         PRINT*,'DELTAC ',DELTAC
         DZ=Z/(NMAX-1)
         dividereact=1
         K=0
         count=0
         timeelapsed=0
c
c     start big loop transport cycle
c
79921    K=k+1
c
c     copying array data to species data
c
            DO 1114 I=1,NMAX
               co2start(i)=CO2b
               hco3start(i)=hCO3b
               co3start(i)=CO3b
               phstart(i)=phb
1114        CONTINUE
            co2start(1)=co2interface
c
c
         s(k)= smean
         print*,'s(k)',s(k)
         PRINT*,'DTDIFF',DTDIFF
         tstepmax=nint(s(k)/dtdiff)
         print*,'tstepmax ',tstepmax
         print*,'maxsizetimeFPM ',maxsizetimeFPM
         if(tstepmax.gt.maxsizetimeFPM) then
            tstepmax=maxsizetimeFPM
            print*,'change maxsizetime, to low'
            pause
         endif
         if (tstepmax.eq.0)then
            print*,'change time regime'
            pause
         endif
c         print*,' '
c
c     ***********************************************************
C     PURE PHYSICAL TRANSPORT
C
         print*,'enter diffusion part'
c
c    big loop for timestep
c
      do 702 t=1,tstepmax
         count=t
         timeelapsed=timeelapsed+dtdiff
           DO 9002 I=1,NMAX
              SPECIES(I)=CO2start(I)
```

```
9002       CONTINUE
           CALL DIFFUSEKI(SPECIES,DTDIFF,NMAX,Z,
     &                        DIFFCO2,DIFFER,DELTA)
            DO 7772 I=1,NMAX
               IF(I.EQ.1) THEN
                  CO2stop(I)=CO2INTERFACE
               ELSE
                  CO2STOP(i)=SPECIES(i)
               ENDIF
7772        continue
c
c     calculating flux loss through bottom layer
       call FLUXARRAY (CO2STOP,fluxinterm,MAXSIZESPACE,tstepmax,
     &                Z,DIFFco2,nmax)
      FLUXINTERMCO2(k) = fluxinterm
c
c     calculating flux through top  layer
c
       call FLUXARRAYtop (CO2STOP,fluxinterm,MAXSIZESPACE,tstepmax,
     &                Z,DIFFco2,nmax)
      FLUXco2toppur(k) = fluxinterm

C     TAKING RESULTS AS INPUT NEW ITERATION
C
            DO 1419 L=1,NMAX
               CO2Start(L)=CO2STOP(L)
 1419       CONTINUE
702   CONTINUE
c
c      integrate for kl-value no enhancement

      intfluxco2pur(K)=simpflux(FLUXINTERMCO2,dtdiff,
     &                           maxsizetimefpm,count)
      intfluxco2topp(K)=simpflux(fluxco2toppur,
     &                 dtdiff,maxsizetimefpm,count)
      accuco2(k)= simpconc(co2stop,dz,maxsizespace,nmax)
     &               -(1000*co2b*z)
c     amount accumulated in cv, integrates over the
c     difference of the concentration profile
      print*,' accuco2(k) ',accuco2(k)
      accuco2save=accuco2(k)
      klpur(K)=-1*(accuco2(k)+intfluxco2pur(k))
     &         /(1000*deltac*timeelapsed)
c      units in m/s piston velocity
c     factor off 1000 for M in moles/m3
      print*,' klpur(k) ',klpur(k)
      klanalytik(K)=(2*deltac*sqrt(diffco2*timeelapsed/3.14159))
     &         /(deltac*dtdiff*tstepmax)
c     analytical sollution for infinite deep layer
      print*,'klanalytic(k) ',klanalytik(k)
      zequiv=DIFFCO2/klpur(k)
```

```
      print*,'equicalent layer thickness is',zequiv ,'m'

c
c     **********************************************************
c     transport with chemical reaction
C     **********************************************************
c
c     reinitialicing concentrations because changed by diffusepur

         do 24462 l=1,nmax
            co2start(l)=co2b
            hco3start(l)=hco3b
            co3start(l)=co3b
            phstart(l)=ph(nmax,0)
24462    continue
         co2start (1)=co2interface
         phstart (1)=ph(1,0)
c
      DO 6007 T=1,TSTEPMAX
C     BIG LOOP FOR ENTIRE DRIVER KINETIC ENHANCEMENT
C

C     DIFFUSING CO2
C
          count=t
           co2start(1)=co2interface
            DO 12111 I=1,NMAX
               SPECIES(I)=CO2start(I)
12111       CONTINUE
            CALL DIFFUSEKI(SPECIES,DTDIFF,NMAX,Z
      &           ,DIFFCO2,DIFFER,DELTA)
            CO2start(1)=CO2INTERFACE
            DO 12211 i=2,NMAX
               CO2stop(i)=SPECIES(i)
12211     CONTINUE
c
C     DIFFUSING HCO3
C
           DO 12311 I=1,NMAX
              SPECIES(I)=HCO3start(I)
12311      CONTINUE
            CALL DIFFUSEKI(SPECIES,DTDIFF,NMAX,Z,
      &            DIFFHCO3,DIFFER,DELTA)
C           COPYING SPECIES IN PROPPER PLACE OF ARRAY
              DO 12411 L=1,NMAX
              HCO3stop(L)=SPECIES(L)
12411     CONTINUE
C
C     DIFFUSING CO3
C
            DO 12511 I=1,NMAX
```

```
               SPECIES(I)=CO3start(I)
12511       CONTINUE
           CALL DIFFUSEKI(SPECIES,DTDIFF,NMAX,Z,DIFFCO3,DIFFER,DELTA)
C           COPYING SPECIES IN PROPPER PLACE OF ARRAY
            DO 16121 L=1,NMAX
              CO3stop(L)=SPECIES(L)
16121       CONTINUE
c
c
c     ****************************************************************
c     reaction part
c
         DO 12811 M=1,NMAX
c
C        BIG LOOP FOR REACTION IN EACH LAYER AND TIMESTEP
C
C         ASSIGNING ARRAY DATA TO INPUT DATA
C
          CO2IN=CO2stop(M)
          HCO3IN=HCO3stop(M)
          CO3IN=CO3stop(M)
          CTSAVE=CO2IN+HCO3IN+CO3IN
          PHin=phstart(M)
          minph=1
          DTREACT=DTDIFF/DIVIDETREACT
          do 70004 i=1,dividetreact
C
             call REACT(CO2IN,CO2R,HCO3IN,HCO3R
     &            ,CO3IN,CO3R,PHIN,PHR,DTREACT,PATH,
     &            KCO2,KOH, KCAF,K1,K2,KW,KB,CACONC,BT,ALK)
c
             IF(M.eq.1) THEN
               CO2IN=CO2INTERFACE
             ELSE
              CO2IN=CO2R
             ENDIF
             HCO3IN=HCO3R
             CO3IN=CO3R
             PHIN=PHR
c     ***************************************************
C
70004     continue
c
c
          CO2stop(M)=CO2R
          HCO3stop(M)=HCO3R
          CO3stop(M)=CO3R
          PHstop(M)=PHR
c
12811   CONTINUE
c       closing loop reaction
```

```
c
c
        co2stop(1)=co2interface
      call FLUXARRAY (co2stop,fluxinterm,MAXSIZESPACE,tstepmax,
     &              Z,DIFFco2,nmax)
      FLUXINTERMCO2(k) = fluxinterm
c       PRINT*,'FLUXINTERM',FLUXINTERM
      call FLUXARRAY (hco3stop,fluxinterm,
     &     MAXSIZESPACE,tstepmax, Z,DIFFco2,nmax)
      FLUXINTERMHCO3(k) = fluxinterm
c      print*,'fluxintermHo2(k) ', FLUXINTERMHCO3(K)
      call FLUXARRAY (co3stop,fluxinterm,
     &         MAXSIZESPACE,tstepmax, Z,DIFFco2,nmax)
      FLUXINTERMCO3(k) = fluxinterm
c      print*,'fluxintermco3(k) ', FLUXINTERMCO3(K)
c
c     calculating flux through top  layer
       call FLUXARRAYtop (CO2STOP,fluxinterm,MAXSIZESPACE,tstepmax,
     &                Z,DIFFco2,nmax)
c      print*,'fluxinterm ',fluxinterm
      FLUXco2topenhanced(k) = fluxinterm
c      print*,'fluxco2topenhanced(k) ', FLUXCO2topenhanced(K)
c

c     taking result iteration as new input
c
         do 12864 M=1,nmax
            co2start(M)=co2stop(M)
            hco3start(M)=hco3stop(M)
            co3start(M)=co3stop(M)
            phstart(M)=phstop(M)
12864     continue
c
c
 6007 CONTINUE
c     caculating fluxes through bottom boundary,
c     accumulation in the layer and
c     kl-value with enhancement
c
c
      intfluxco2tope(K)=simpflux(fluxco2topenhanced,
     &                dtdiff,maxsizetimefpm,count)
      print*,'intfluxco2tope ',intfluxco2tope(k)
c
c
      intfluxco2enh(K)=simpflux(fluxintermco2,dtdiff
     &                        ,maxsizetimefpm,count)
      print*,'intfluxco2enh ',intfluxco2enh(k)
      intfluxhco3(K)=simpflux(fluxintermhco3,dtdiff,maxsizetime,count)
      print*,'intfluxhco3 ',intfluxhco3 (k)
```

```
      intfluxco3(K)=simpflux(fluxintermco3,dtdiff,maxsizetime,count)
      print*,'intfluxco3 ',intfluxco3(k)
      intfluxtotal(K)=intfluxco2enh(K)+intfluxhco3(K)+intfluxco3(K)
c     total amount of ct transported out of controll volume
      print*,'intfluxtotal (k)',intfluxtotal(k)
c
c
c     calculating amount accumulated in cv
c
      accuco2(k)=simpconc(co2stop,dz,maxsizespace,nmax)
     &            -(co2b*z*1000)
      accuhco3(k)=simpconc(hco3stop,dz,maxsizespace,nmax)
     &            -(1000*hco3b*z)
      accuco3(k)= simpconc(co3stop,dz,maxsizespace,nmax)
     &            -(1000*co3b*z)
c
      klenhanced(K)=-1*(accuco2(k)+accuhco3(k)+accuco3(k)
     &         +intfluxtotal(k)) /(1000*deltac*timeelapsed)
c
c     calculating enhancement factor
c
      EFFPM=KLENHANCED(K)/KLPUR(K)

      zequiv=DIFFCO2/klpur(k)
      PRINT*,'ZEQUIV ',ZEQUIV
      PRINT*,' '
      print*,'EFFECTIVE EQUILVALENT LAYER THICKNESS (m)',ZEQUIV
      PRINT*,'THE OVERALL EFFECTIVE ENHANCEMENT FACTOR IS'
      PRINT*,EFFPM
c
c
c
         endif
C        CLOSING IF CONDITION FOR WHOLE DRIVER FILM PENETRATION
C     -------------------------------------------------------
C     | END OF DRIVER FILM PENETRATION MODEL                 |
C     -------------------------------------------------------
```

2. Subroutines and functions:

```
C     --------------------------------------------------------
C     | SUBROUTINE FLUXARRAY                                  |
c     | CALCULATES THE FLUXES through the bottom boundary     |
c     ---------------------------------------------------------
      SUBROUTINE FLUXARRAY (dummy,fluxinterm,MAXSIZESPACE,
     &               tstepmax,Z,DIFF,nmax)
C
      REAL DIFF,pv,z
```

```
      INTEGER tstepmax,MAXSIZESPACE
      DIMENSION dummy (maxsizespace)
      REAL FLUXINTERM
c
      PV=1000*DIFF/(Z/(NMAX-1))
C     Factor 1000 for consistency in units from Mol/1 to Mol/m**2
c     flux counts positive for flux out of cv
       FLUXINTERM=-1*PV*(DUMMY(nmax)-DUMMY(nmax-1))
       RETURN
       END
C     -------------------------------------------------------
C     END OF ROUTINE FLUxarray
C     ---------------------------------------------------
C
C     ----------------------------------------------------------
C     |  SUBROUTINE FLUXARRAYTOP                               |
c     |  CALCULATES THE FLUXES through the TOP    boundary     |
c     -----------------------------------------------------------
      SUBROUTINE FLUXARRAYTOP (dummy,fluxinterm,MAXSIZESPACE,
     &               tstepmax,Z,DIFF,nmax)
C
      REAL DIFF,pv,z
      INTEGER tstepmax,MAXSIZESPACE
      DIMENSION dummy (maxsizespace)
      REAL FLUXINTERM
      PV=1000*DIFF/(Z/(NMAX-1))
C     Factor 1000 for consistency in units from Mol/1 to Mol/m**2
c     flux counts positive for flux out of cv
       FLUXINTERM=-1*PV*(DUMMY(2)-DUMMY(1))
       RETURN
       END
C
C     -------------------------------------------------------
C     END OF ROUTINE FLUxarray
C     ---------------------------------------------------
C
c     ------------------------------------------------------
c     | FUNCTION SIMPCONC  integrates the concentrations    |
c     | over the layer thickness at time tstepmax, given by |
c     | equaly spaced values                                |
c     | using simpsons rule                                 |
c     | source equation: numerical recipes                  |
c     ------------------------------------------------------
      function  simpconc (y,dx,maxsizespace,nmax)
c
      integer maxsizespace,nmax
      dimension y (maxsizespace)
      real dx,sum1,sum2
      integer 1
c
      sum1=0
```

```
      sum2=0
      sum1=0.5*dx*(y(1)+y(nmax))
      do 10 l=2,nmax-1
         sum2=sum2+y(l)
 10   continue
      sum2=sum2*dx
      simpconc=1000*(sum1+sum2)
c     factor 1000 to go from moles/liter to moles/m2 absorbed
c     =output units
      return
      end
c
c     ----------------------------------------------------
c     end of simpconc routine
c     ----------------------------------------------------
c     -----------------------------------------------------
c     | FUNCTION SIMPFLUX  integrates the fluxes through the|
c     | bottom boundary, the values are given by            |
c     | equaly spaced values of the fluxes using simpsons   |
c     | rule                                                |
c     | source equation: numerical recipes                  |
c     -----------------------------------------------------
      function  simpflux (y,dx,maxsizetimefpm,n)
c
      integer maxsizetimefpm
      dimension y (maxsizetimefpm)
      real dx,sum1,sum2
      integer n,l
c
      sum1=0
      sum2=0
      sum1=0.5*dx*(y(0)+y(n))
      do 10 l=1,n-1
         sum2=sum2+y(l)
 10   continue
      sum2=sum2*dx
      simpflux=sum1+sum2
      return
      end
c
c     ----------------------------------------------------
c     end of simpflux routine
c     ----------------------------------------------------
```