This is to certify that the

dissertation entitled

Engineering Applications of Microbial Chemotaxis

presented by

Mark Thomas Widman

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Chemical Engineering

R Mark Worden
Major professor

Date 8/18/97

ENGINEERING APPLICATIONS OF MICROBIAL CHEMOTAXIS

By

Mark Thomas Widman

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Chemical Engineering

1997

ENG

Chemo

chemicals call

chemotaxis to

to develop or r

tools and meth

the laser diffra

Several mather

experimental o

predictions we

modeling simu

chemotactic re

saturation calcu

The sec

which chemota

fully take adva

applications th

in situ bioremedi

ABSTRACT

ENGINEERING APPLICATIONS OF MICROBIAL CHEMOTAXIS

By

Mark Thomas Widman

Chemotaxis is the ability of an organism to move in response to gradients of chemicals called chemoattractants. Many species of bacteria are known to exhibit chemotaxis to a wide variety of chemoattractants. The first objective of this research was to develop or refine tools and methods to study the chemotactic response of bacteria. The tools and methods include the diffusion gradient chamber (DGC) and its associated units, the laser diffraction capillary assay (LDCA), and microelectrodes and microbiosensors. Several mathematical models of the DGC system were written that allowed predictions of experimental outcomes to be made without running the actual experiments. The modeling predictions were verified by comparison to experimental data. Methods to analyze the modeling simulations were developed to further enhance the understanding of the chemotactic response. These analysis methods included bacterial flux and receptor saturation calculations.

The second objective of the research was to identify applications or systems in which chemotaxis was beneficial, and then to develop ways to engineer those systems to fully take advantage of the benefits offered through the chemotactic response. The three applications that were selected were microbial competition, selection of mutants, and *in situ* bioremediation. Chemotaxis was predicted by the mathematical model to impart a

competitive ad

preliminary w

response was u

coli. *In situ*

chemotactic r

population of

being used to r

chemotaxis to

response of *P*

KC and *Esche*

competitive advantage to a bacterial population in certain non-mixed environments, and preliminary work was done to experimentally validate the predictions. The chemotactic response was used as a selection agent in the isolation of desirable mutants of *Escherichia coli*. *In situ* mutagenesis was performed on feedback inhibited *E. coli*, and the chemotactic response was used to separate feedback resistant mutants from the population of inhibited bacteria. *Pseudomonas stutzeri* strain KC, a bacterium currently being used to remediate a carbon tetrachloride contaminated aquifer, was found to exhibit chemotaxis to chemicals present in the aquifer. Experimental protocols to study the response of *Pseudomonas* KC around objects, and competition between *Pseudomonas* KC and *Escherichia coli* were developed.

The ef
work reported
performed ma
work on the
responsible fo
I woul
with me on a
whose researc
Marshall Brec
Dr. M
I have reache
DGC, and mo
microorganis
ADI algorithr
and Faith Pet
I owe
thank my wif
has worked e
her devotion

ACKNOWLEDGMENTS

The efforts of many people have been instrumental in helping me to complete the work reported in this thesis. I am grateful for the help of a number of undergraduates who performed many of the experiments. Laura Booms and Petty Setiawan both did good work on the *Pseudomonas* KC swarm plate experiments. Sebastian Schmidt was responsible for making the Laser Diffraction Capillary Assay a reality.

I would like to thank my fellow graduate students, Tyler Ames, who collaborated with me on a project that impacted both of our research programs, and Mark Mikola, whose research on mutant selection led to another use for chemotaxis. Both of them and Marshall Bredwell provided me with many insights into my research.

Dr. Mark Worden has mentored me through my four years, and with his guidance I have reached the goals that I set for myself. Dave Emerson taught me how to use the DGC, and more importantly, gave me a good introduction to handling and thinking about microorganisms, from a non-engineering viewpoint. Chichia Chiu supplied me with the ADI algorithm used to solve some of the math models. Candy McMaster, Julie Caywood and Faith Peterson in the graduate office have also been invaluable resources and friends.

I owe a lot to my Mom and Dad, for everything. Most importantly, I would like to thank my wife, Shelly. She has put up with me all through my graduate school years, and has worked extremely hard so that I could stay in school. I hope to finally be able to repay her devotion to me by proving to her that all this was really worth it.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

$C_$ nutrie

C_Q chem

C_S chem

D_H diffus

D_Q diffu

D_S diffu

$F_{=\Delta}$ avera

G gluco

H nutri

$J_$ cell f

K_{DQ} cell r

K_{DS} cell r

K_I inhib

k_Q mass

k_S mass

J_{uS} flux

J_{uu} flux

J_{uu} flux

N_T total

P inhib

LIST OF NOMENCLATURE

C_0	nutrient saturation constant
C_Q	chemoattractant Q saturation constant
C_S	chemoattractant S saturation constant
D_H	diffusion coefficient of H
D_Q	diffusion coefficient of Q
D_S	diffusion coefficient of S
F_{max}	average maximum flux
G	glucose concentration
H	nutrient density
J_u	cell flux
K_{DQ}	cell receptor saturation constant for chemoattractant Q
K_{DS}	cell receptor saturation constant for chemoattractant S
K_I	inhibition constant
k_Q	mass transfer coefficient for Q
k_S	mass transfer coefficient for S
J_{uS}	flux of cells due to chemotaxis to S
J_{uu}	flux of cells due to chemotaxis to Q
J_{uu}	flux of cells due to random motility
N_T	total number of receptors for a chemoattractant
P	inhibitor P density

Q	chem
R	micro
S	chem
t	time
u	cell d
V_{arena}	volur
v	maxi
v'	maxi
v_Q	speci
v_S	speci
V_u	chem
X	perce
Y_H	yield
σ	cell
υ	singl
χ_0	chem
χ_2	chem
μ	rand
μ'	rand
ϕ_S	chem
Ψ	dyna

chemoattractant Q density

microbiosensor reading

chemoattractant S density

time

cell density

ena volume of DGC arena

maximum growth rate on H

maximum specific growth rate modified for inhibition

specific chemoattractant consumption coefficient for Q

specific chemoattractant consumption coefficient for S

chemotactic velocity

percent oxygen saturation

yield coefficient for cells growing on H

cell tumbling frequency

single cell swimming speed

chemotactic sensitivity coefficient

chemotactic sensitivity coefficient modified for inhibition

random motility coefficient

random motility coefficient modified for inhibition

chemotactic response factor for chemoattractant S

dynamic competition factor

1. INTRO

Non-

energy, temp

Microorganis

a competitiv

response to a

been shown

1990; Kato e

There

develop or r

tools that w

mathematica

and microel

The

possible for

those systen

three applic

in situ biore

Sect

significance

and modeli

1. INTRODUCTION

Non-uniform distributions of chemical compounds, pH, dissolved oxygen, light energy, temperature and other factors are common in many microbial ecosystems. Microorganisms able to position themselves optimally with respect to gradients may have a competitive advantage over other organisms. The ability of an organism to move in response to a gradient of a chemical species is known as chemotaxis. Many bacteria have been shown to exhibit chemotaxis (Adler, 1972; Berg and Tedesco, 1975; Harwood *et al.*, 1990; Kato *et al.*, 1990; Macnab, 1987; Yamamoto and Imae, 1993).

There were two main objectives for this research. The first objective was to develop or refine tools and methods to study the chemotactic response of bacteria. The tools that were used include the diffusion gradient chamber (DGC), several versions of mathematical models of the DGC system, the laser diffraction capillary assay (LDCA), and microelectrodes and microbiosensors.

The second objective was to identify applications or systems in which it might be possible for chemotaxis to be of a beneficial nature, and then to develop ways to engineer those systems so as to realize the full benefits offered by the chemotactic response. The three applications that were studied were microbial competition, selection of mutants, and *in situ* bioremediation.

Section 2 explores the two main objectives in more detail, and describes their significance to the scientific community. Section 3 gives background on the experimental and modeling methods employed by others to study chemotaxis. The results of this

research proje

methods that

5.7 describe t

and the work

summary of th

Include

wall in the ma

They are incl

chemotactic r

outlines the al

equations. App

which is prese

program. The

the quantities p

the model is de

earch project are presented in the next four sections. Section 4 examines the tools and thods that were developed to study chemotaxis over the course of the project. Sections 7 describe the applications that were identified for improvement through chemotaxis, d the work that was done to engineer the systems to make use of chemotaxis. A mmary of the work, and the conclusions that were obtained are given in Section 8.

Included in Appendix A are results of some interesting experiments that did not fit ll in the main text of the thesis, or have not been fully analyzed for their importance. ey are included to show the range of growth patterns that can arise due to the emotactic response, including rings and geometric patterns. Appendix B briefly tlines the alternating-direction implicit (ADI) algorithm for solving partial differential uations. Appendix C gives instructions for the FORTRAN ADI model, one version of ich is presented in Appendix D. Appendix E gives the input files for the FORTRAN gram. The Matlab files used to produce many of the graphs and to calculate some of quantities presented in the thesis are given in Appendix F. A dimensionless version of model is developed in Appendix G.

2. OBJEC

2.1 Develo

The fi

modeling met

The diffusion

used in this re

Emerson *et al*

system. Cons

different oper

responding t

chemoattracta

inhibitor grad

quantitative in

without actua

To pr

and microorg

these paramet

the laser diffr

of the random

1996), which

OBJECTIVES AND SIGNIFICANCE

Development of tools

The first objective of this research was to improve existing experimental and deling methods for studying chemotaxis, or to develop new methods, as necessary. e diffusion gradient chamber (Emerson *et al.*, 1994) was the main experimental tool ed in this research. The DGC was well established and tested for chemotaxis studies by erson *et al.*, but some further enhancements to the method were made to optimize the stem. Considerable research was devoted to developing mathematical models of fferent operating modes of the DGC system. These modes included one cell population ponding to one chemoattractant; two cell populations responding to multiple emoattractants; and two cell populations responding to two chemoattractants and an ibitor gradient. These models were experimentally validated and then used for antitative interpretation of experimental results, and for exploring chemotactic behavior hout actually running time-consuming experiments.

To produce useful simulations that accurately modeled real chemical gradients l microorganisms, parameter values had to be obtained. Several methods of measuring se parameters were developed as part of the research effort. These methods included laser diffraction capillary assay (Schmidt *et al.*, 1997) which allowed for measurement he random motility coefficient, and microelectrodes and microbiosensors (Peteu *et al.*, 6), which enabled high-resolution measurements of gradients of oxygen and glucose.

2.2 Applic

The s

chemotactic r

was the first s

of bacteria c

competition i

hypothesized

position them

the DGC prov

methods to va

Anoth

mutant strains

to draw muta

concentration

method of se

obtained for th

A thir

The bacterium

producing har

currently bein

Applications for chemotaxis

The second objective of the research was to identify applications in which the notactic response could be exploited to benefit the application. Microbial competition the first such application selected. Many studies have examined how two populations acteria compete in well-mixed environments, but much less is known about petition in spatially structured systems, where chemical gradients can occur. It is thesized that chemotaxis can impart a competitive advantage by allowing cells to tion themselves in conditions optimal for their survival. The mathematical model of DGC provided a means to explore this hypothesis. Preliminary work on experimental nods to validate the model's predictions was also initiated.

Another application benefitted by the chemotactic response was the selection of ant strains of bacteria that exhibited a desired trait. Specifically, chemotaxis was used raw mutant strains of *Escherichia coli* into a region of the DGC having a higher entration of an inhibitor than their non-mutagenized progenitors could tolerate. This od of selection by chemotaxis resulted in an improved strain of *E. coli* being ned for the industrial fermentation of a valuable product.

A third application for which chemotaxis was studied was *in situ* bioremediation. bacterium *Pseudomonas* KC, which is able to degrade carbon tetrachloride without ucing harmful by-products, was chosen as the model organism. *Pseudomonas* KC is ntly being used in Schoolcraft, MI, to bioremediate a contaminated aquifer. The

chemotactic b

medium on m

otactic behavior of *Pseudomonas* KC was studied, and the effects of a porous

um on motility and transport were observed.

3. CHEM

3.1 The ch

The motility

Salmonella t

1987). When

the flagella r

When the cel

tumble and ra

of this series

Figure 1.

In an environ

the presence

EMOTAXIS BACKGROUND

he chemotactic mechanism

otility and chemotactic response of the enteric bacteria *Escherichia coli* and *ella typhimurium* have been studied in detail (Berg and Brown, 1972; Macnab, When these peritrichously flagellated bacteria spin their flagella counterclockwise, gella move in a synchronized bundle, causing the cell to swim forward, or run. the cell reverses the spin of the flagella, the bundle separates, causing the cell to and randomly reorient itself before running in a new direction. The overall effect series of runs followed by tumbles is referred to as a random walk, as shown in 1.

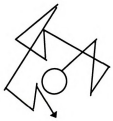


Figure 1. Random walk

vironment with no chemoattractants, the run length is independent of direction. In ence of a chemoattractant gradient, the cells monitor the time rate of change of

occupancy

concentratio

extended run

frequency ap

result is a bia

in Figure 2.

3.2 Experi

Exper

assay (Adler,

1972; Nossal

(SFDC) (Forc

upancy of receptors for the chemoattractant during a run. If an increasing centration gradient is encountered, the tumbling frequency decreases, causing an nded run length. If a decreasing concentration gradient is encountered, the tumbling uency apparently remains at the basal level (Berg, 1988; Macnab, 1987). The net lt is a biased random walk toward the higher chemoattractant concentration, as shown igure 2.

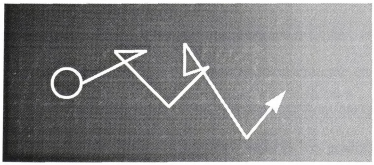


Figure 2. Runs and tumbles in the presence of a gradient.

Experimental systems for measuring chemotaxis

Experimental systems used to study microbial chemotaxis include the capillary (Adler, 1972; Nikita *et al.*, 1992; Staffeld *et al.*, 1987), the motility plate (Adler, ; Nossal, 1972; Wolfe and Berg, 1989), and the Stopped Flow Diffusion Chamber C) (Ford, 1992; Ford *et al.*, 1990). In the capillary assay, the open end of a capillary

filled with bu

for a fixed ar

entered the

different co

mathematical

Lauffenburge

random motil

A mo

having a unif

of the plate i

gradient is fo

The motility

cellular meta

chemotactic

motility plate

The S

are introduce

of interest. W

approximated

an initially st

by light diffra

:d with buffer solution containing a chemoattractant is inserted into a cell suspension a fixed amount of time. The capillary is then removed, and the number of cells that :red the capillary are counted. By comparing this number for capillaries having erent concentrations of chemoattractant, chemotaxis can be quantified. A hematical model of the capillary assay has been developed (Rivero-Hudec and ffenburger, 1986) that allows calculation of two important modeling parameters, the lom motility coefficient and the chemotactic sensitivity coefficient.

A motility plate consists of a petri dish containing a semi-solid agar medium ng a uniform distribution of a consumable chemoattractant (Adler, 1972). The center ne plate is inoculated with cells. As they grow and consume the chemoattractant, a ient is formed that induces migration of the cells outward from the inoculation zone. motility plate is a simple system to use, but, because the gradient is formed by lar metabolism, the gradient is difficult to control and quantify. Also, the notactic response to non-metabolizable chemoattractants cannot be studied in a lity plate.

The SFDC is a rectangular chamber into which two impinging streams of medium ntroduced. One stream contains the chemoattractant, and the other contains the cells terest. When the flows are suddenly stopped, a step-change in chemoattractant is oximated where the two streams intersect. Diffusion of the chemoattractant creates itially steep gradient that decays over time. The chemotactic migration is measured ght diffraction. A mathematical model of the system has been developed to calculate

the chemoatt

cells.

None

well characte

have develop

microbial che

previously pu

DGC consist

chemoattractant gradients and to analyze the resulting chemotactic migration of the s.

None of the experimental systems described above are convenient for establishing l characterized, steady-state gradients or multiple gradients in multiple directions. We e developed the Diffusion Gradient Chamber (DGC), shown in Figure 3, for studying robial chemotaxis under such conditions. Details of the DGC system have been

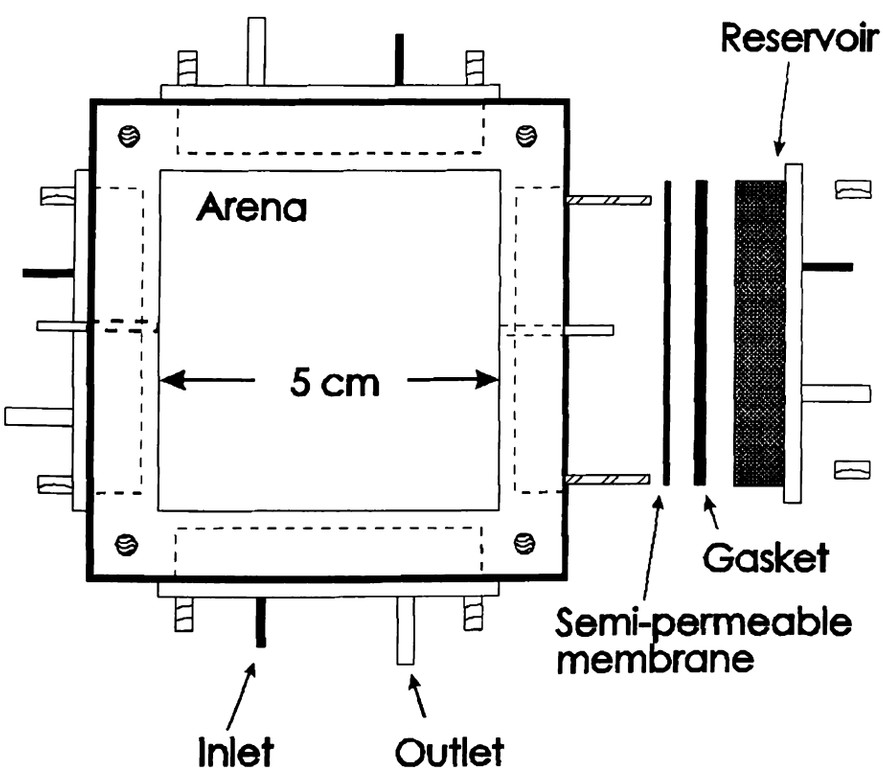


Figure 3. Diagram of the diffusion gradient chamber.

ously published (Emerson *et al.*, 1994), so only a brief description is given here. The consists of a square arena (5 cm x 5 cm x 1 cm) bounded by a reservoir on each

side. Each re

by an imperr

arena and res

gel through

chemoattract

concentration

concentration

across the ge

then the chan

monitored fr

The

Multiple gra

chamber. Gr

inoculation,

simultaneous

The DGC ha

microsensor

inoculation

inoculation i

maintain an

gaseous reag

side. Each reservoir is separated from the arena either by a semi-permeable membrane, or by an impermeable silicone sheet, depending on whether substrate diffusion between the arena and reservoir is desired. The arena is filled with medium containing dilute agarose gel through which the cells can readily swim. Different concentrations of the chemoattractant(s) are maintained in the reservoirs. Diffusion from the higher concentration reservoir (the source) through the membrane and across the gel to the lower concentration reservoir (the sink) results in the development of a continuous gradient across the gel. The gradient is allowed to establish for a specified amount of time, and then the chamber is inoculated. Growth and movement of the microbial populations are monitored from above by light diffraction.

The DGC has several advantages over other methods of studying chemotaxis. Multiple gradients in different directions may be established simultaneously in the chamber. Gradients may be allowed to approach a linear steady-state profile before inoculation, or transient gradients can be used. These gradients can be initiated simultaneously, or in a staggered fashion to simulate pulsed influxes of chemoattractants. The DGC has been designed with a removable lid so that samples may be withdrawn or microsensor readings taken without sacrificing the experiment. Many different inoculation protocols may be used, including uniform inoculation across the gel, inoculation in a line, or at a point. Gas ports in the chamber provide the possibility to maintain an inert gas headspace above the arena for anaerobic experiments, or to supply gaseous reagents to the microbes.

4. DEVEL

4.1 Impro

4.1.1 Image

In pre

the DGC we

was laborio

course of th

were develop

The f

7CN CCD-c

The Pulnix c

(Quanta, Mc

written by Se

Chemotactic

Studienarbeit

pictures at pr

4.0 for Windo

Two

cameras (Co

other to a Po

DEVELOPMENT OF TOOLS

Improvements to the DGC method

1 Image acquisition system

In previous work done in the DGC system (Emerson *et al.*, 1994), photographs of DGC were taken manually with a 35 mm camera. This method gave good results, but laborious and did not lend itself well to computerized image analysis. Over the rse of the work presented in this thesis, several automated image analysis systems e developed.

The first computer-based image analysis system was built using a PULNiX TM- CCD-camera (PULNiX America Inc., Sunnyvale, CA) mounted above the DGC. Pulnix camera was attached to a PC through a WinVisionPro video capture board nta, Mountain View, CA). A time-lapse capture program called AutoCap was en by Sebastian Schmidt ("Development of Novel Methods to Measure Random and notactic Microbial Motility at the Community Level", Sebastian Schmidt, ienarbeit, Michigan State University, 1995) and used to automatically capture res at pre-set intervals. Captured images were converted into RAW format in Matlab or Windows, and then analyzed.

Two other image analysis systems were installed based on Color QuickCam ras (Connectix, San Mateo, CA). One Quickcam was connected to a PC, and the to a Power Macintosh 7200/90. The software included with the Quickcams allowed

for timed-im

PULNiX sys

a freeware ir

4.1.2 Quan

To q

curve relatir

was grown

glycerol for

were taken

added to a c

acquired. Th

of the cultur

bar. Anothe

medium, op

added to th

inoculated.

The

the 1:100,0(

average nur

colonies. Th

ned-image capture. Images from these units, and, later in the research, from the iX system, were analyzed using NIH Image (http://rsb.info.nih.gov/NIH-IMAGE/), ware image analysis program.

Quantification of cells by grayscale analysis

To quantify the number of cells in a DGC through image analysis, a calibration relating cell number to image grayscale was developed. *Escherichia coli* HCB 33 rown in M63 medium (described in more detail in Section 4.2.3) with 5 mM ol for 30 hours at 30°C. Replicate measures of the optical density of the culture taken at a wavelength of 630 nm. Forty milliliters of the cell culture were then to a clean DGC containing no agar, and three consecutive images of the DGC were ed. The DGC was then emptied and cleaned for the next reading. A 0.1 ml aliquot culture was placed in triplicate on LB agar plates and spread uniformly with a glass nother volume of the cell culture was then diluted by a factor of 1:1.5 with M63 m, optical density measurements were taken, 40 ml of the diluted solution was to the DGC, three more images were acquired, and three more LB plates were ated. This process was repeated seven times at the dilutions shown in Table 1.

The plates from each dilution were incubated at 30°C for 18 hours. At this time, 00,000 dilution plates had the most suitable number of colonies to count. The e number of colonies in each plate was 448, with a standard deviation of 36 s. The original culture therefore contained approximately 4.48×10^8 cells/ml.

Table 1. Dilution factor and OD for grayscale calibration

Dilution (ml cells/ml total)	Optical Density (λ=630 nm)	
no dilution	0.762	0.789
1:1.5	0.559	0.577
1:2	0.453	0.424
1:3	0.327	0.299
1:4	0.322	0.270
1:5	0.191	0.192
1:10	0.082	0.066
1:15	0.054	0.041
1:20	0.025	0.028
1:100	--	--
1:1000	--	--
1:10,000	--	--
1:100,000	--	--

Matlab 4.0 for Windows was used to analyze the images. The image was ed to RAW format with Image Alchemy, and then read into Matlab as an image

file. The ave

images for ea

dilution. Fina

to normalize

averaged at

illustrated in

The

DGC. This

with Tyler

this dissert

relationship

. The average grayscale value of each picture was calculated in Matlab. The three ges for each dilution were then averaged together to give one grand average value per tion. Finally, each grand average was divided by the maximum of the grand averages normalize the values into a range from 0 to 1. The optical density readings were also raged at each dilution. The relationship between optical density and grayscale is strated in Figure 4.

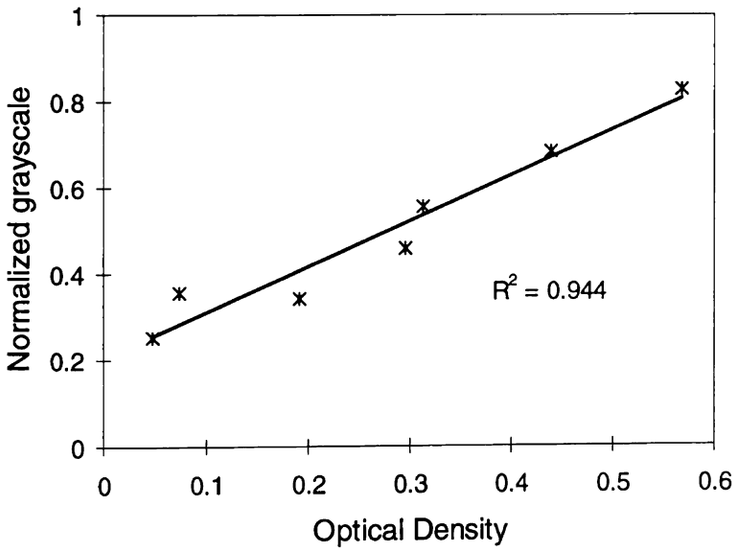


Figure 4. Normalized grayscale vs. optical density for *E. coli*.

The normalized grayscale was next plotted against the number of cells in the C. This yielded a linear calibration, as shown in Figure 5. Other work in collaboration n Tyler Ames, a Ph.D. candidate in Dr. Worden's research group, but not included in dissertation, has also shown that soybean plant cell concentration has a linear tionship with grayscale in the DGC (Ames, 1997).

4.1.3 Meth

An e

the membra

following se

arena by a s

was fixed in

to the grour

harden. The

take place in

a horizontal

A su

3000 mL of

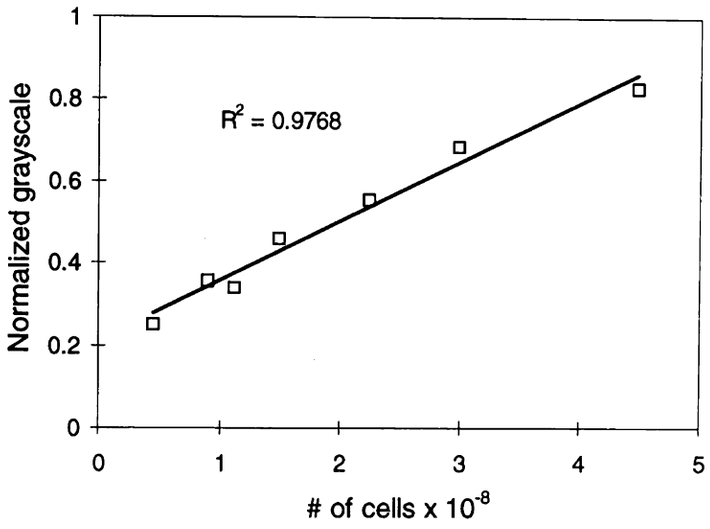


Figure 5. Calibration curve for *E. coli* in DGC.

3 Method to measure membrane mass transfer coefficient

An experiment was designed to measure the mass transfer coefficient, k_S, across membrane. This parameter is important in the modeling work introduced in the owing sections. A DGC system was prepared having one reservoir separated from the a by a semi-permeable membrane, and the other three sealed. After the membrane fixed in the chamber, the chamber was stood on one side with the membrane parallel ne ground. 400 μL of 0.3% agar were spread onto the membrane and allowed to en. The agar layer was applied to duplicate any fouling of the membrane that could place in a DGC experiment. After the agar layer solidified, the DGC was returned to rizontal position, and filled with 45 ml of RO water.

A sucrose solution was pumped through the reservoir from a flask containing) mL of 30 g/L sucrose. The solution was recycled back into the 3000 mL flask. A stir

bar was plac

model for thi

where S is th

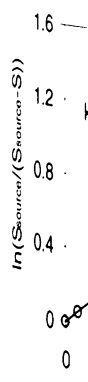
A is the are

concentratio

S=0 and line

Sugar conc

measuremer



)ar was placed in the DGC, and the DGC sat on a magnetic stir plate. The mathematical model for this system is

$$V_{arena}\frac{dS}{dt} = k_S A(S_{source} - S) \qquad (1)$$

Where S is the concentration of sugar in the DGC arena, V_{arena} is the volume of the arena, A is the area of the membrane (3.06 cm^2) available for mass transfer, and S_{source} is the concentration in the source flask. Equation (1) can be solved with the initial condition =0 and linearized as

$$\ln\left(\frac{S_{source}}{S_{source} - S}\right) = \frac{k_S A}{V_{arena}} t \qquad (2)$$

ugar concentrations were measured over time by HPLC analysis. The results of neasurements for both sucrose and glucose are shown in Figure 6.

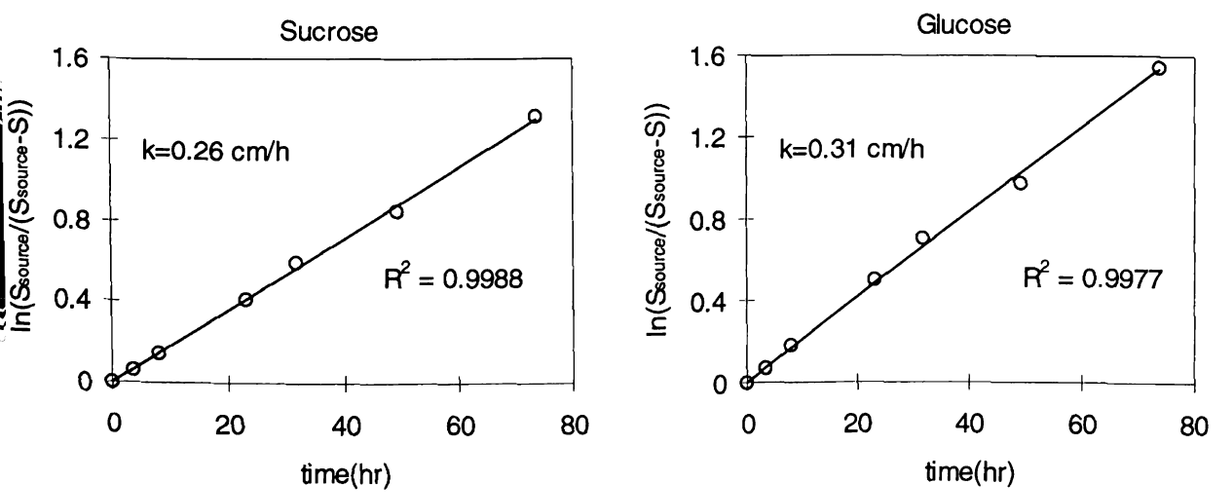


Figure 6. Determination of the membrane mass transfer coefficient.

4.1.4 Faster

In sc

conventiona

arena. Wher

process beg

mathematica

state diffusio

where S is t

Equation (3

where y is

boundary co

for the sink

4 Faster approach to steady-state gradients

In some instances, steady-state gradients were desirable in the DGC. In the ventional operating mode, gel without the gradient chemical was poured into the a. When the gradient chemical was introduced into one reservoir, the diffusion cess began, and eventually a linear steady-state gradient would be established. A hematical model of the system was developed to describe this process. The unsteady-e diffusion equation, for the case with no cells present, is given by

$$\frac{\partial S}{\partial t} = D_S \nabla^2 S \qquad (3)$$

re S is the gradient chemical and D_S is the diffusion coefficient. In two-dimensions, ation (3) can be rewritten as

$$\frac{\partial S}{\partial t} = D_S\left(\frac{\partial^2 S}{\partial y^2} + \frac{\partial^2 S}{\partial z^2}\right) \qquad (4)$$

re y is the direction parallel to the gradient and z is the vertical direction. The ndary conditions at the membranes are given by

$$\left.\frac{\partial S}{\partial y}\right|_{y=0} = \frac{k_S}{D_S}\left(S|_{y=0} - S_{Sink}\right) \qquad (5)$$

he sink reservoir (position y=0 cm) and

$$\left.\frac{\partial S}{\partial y}\right|_{y=5} = \frac{-k_S}{D_S}\left(S|_{y=5} - S_{Source}\right) \qquad (6)$$

for the sourc

assumed to

boundary co

The

equation wi

conventiona

steady-state

state solutic

solution at

within 95%

necessary to

A ne

achieve the

work refere

schematical

concentratic

poured and

no gradient

50% of the

· the source reservoir (position y=5 cm). The source and sink flask concentrations were sumed to be constant throughout the experiment. For all other boundaries, the no-flux undary condition was applied

$$D_S \nabla S = 0 \tag{7}$$

The Matlab Partial Differential Equation Toolbox was used to solve the diffusion uation with a finite element method. The predicted time to reach steady-state for the nventional method was 33 days, using sucrose as the model chemical. To define ady-state, the elliptical equation was solved by the tool box to give the actual steady-te solution. Then the parabolic equation was solved, and compared to the elliptical ution at several time points. The first time point at which the parabolic solution was hin 95% of the elliptic solution at every node in the DGC was defined as the time essary to reach steady-state.

A new method of pouring gel slabs was developed to reduce the time necessary to ieve the steady-state gradient. This method was developed as part of the plant cell rk referenced in Section 4.1.2. The new method of pouring the slabs is shown ematically in Figure 7. The DGC was tilted at an angle and Layer 1, containing a centration of the gradient chemical equal to that used in the source reservoir, was ired and allowed to solidify. The DGC was returned to level, and Layer 2, containing gradient chemical, was poured and allowed to solidify. Finally, Layer 3, containing % of the gradient chemical concentration used in Layer 1, was poured.

The Matlab

distance from

The time nece

be 40 hours, v

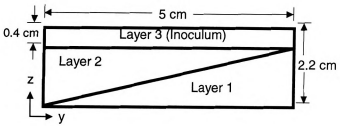


Figure 7. Three-slab pouring method.

Matlab simulation of this system, depicting the concentration as a function of ance from the sink reservoir, is shown in Figure 8 for three time-points.

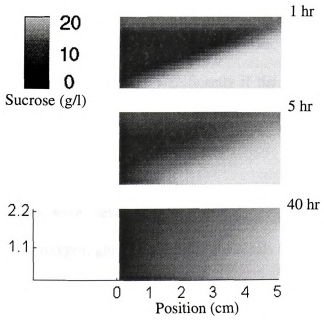


Figure 8. Simulation of approach to steady-state.

ime necessary to reach steady-state for the three-slab pouring method is predicted to hours, which is approximately 20 times less than the conventional pouring method.

4.1.5 Recy

Anc

flasks. In t

through the

method wo

the flasks a

flask was

concentratio

1.3%/week)

result is sp

flask conce

mode.

4.1.6 Micr

Mic

Worden to

al., 1996a;

microelectr

microelectr

consumes g

microelectr

5 Recycle of source and sink flasks

Another in the operating method was the use of recycle in the source and sink ks. In the original DGC method, fluid is pumped out of the source or sink flask, ough the corresponding reservoir on the DGC, and then to a waste container. This thod worked well, but the time an experiment could run was limited by the volume of flasks and the pump speed. The idea of recycling medium from the reservoir to the sk was tested. Sucrose was the gradient chemical, at an initial source flask ncentration of 20 g/L. In six different experiments, only 0.26±0.13 g/L-week (or %/week) were lost from the flasks due to transfer across the membrane. Although this ult is specific to the plant cell system and to sucrose, it suggests that in general, the sk concentrations will not change significantly if the system is operated in recycle de.

.6 Microsensors and microbiosensors

Microsensors were developed by David Emerson, Serban Peteu, and Mark rden to measure oxygen, glucose and other chemicals (Peteu *et al.*, 1996; Emerson *et* 1996a; Emerson *et al.*, 1996b). Briefly, the microsensors are Clark-type oxygen oelectrodes. The addition of an enzyme, such as glucose oxidase, to the tip of the oelectrode produces a microbiosensor. The enzyme catalyzes a reaction that umes glucose and oxygen. The rate of disappearance of oxygen is measured by the electrode, and allows for the glucose concentration to be measured.

A po

for the gluc

linked to th

be perform

function of

fit the data t

where R is

percent oxy

least-square

parameters

calibration p

A portion of this research project was devoted to developing a calibration method the glucose microbiosensors. Because the reaction at the tip of the microbiosensor is ked to the oxygen concentration, calibrations at varying oxygen concentrations had to performed. A group of microbiosensor readings, in picoamps, was obtained as a nction of percent oxygen saturation and glucose concentration. The simplest model to the data to is a linear plane, whose equation is given by

$$R = aG + bX + d \tag{8}$$

here R is the microbiosensor reading, G is the glucose concentration, and X is the ercent oxygen saturation. The Solver tool in Microsoft Excel 7.0 was used to find the ast-squares best fit between the experimental points and the model, by varying the rameters a, b, and d simultaneously. Four isoclines of the resulting three-dimensional libration plane are shown in Figure 9.

An

microbiose

simultaneo

medium wa

for more de

bacteria ha

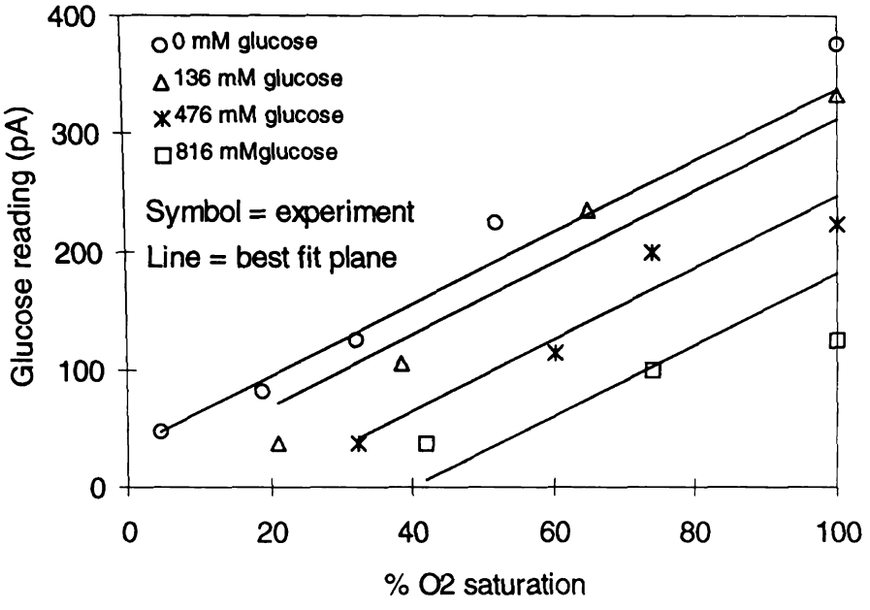


Figure 9. Three dimensional glucose microsensor calibration plane.

An example of an experiment in which the oxygen microsensor, the glucose icrobiosensor, and the image analysis system for cell grayscale were used multaneously is shown in Figure 11. A swarm plate containing 1 mM glucose in M63 edium was inoculated with 10 μL actively growing *E. coli* HCB 33 (see Section 4.2.3 r more details on medium and strain). After approximately 24 hours of growth, a ring of cteria had formed, as shown in Figure 10.

Microsensc

An image v

11 shows

chemotacti

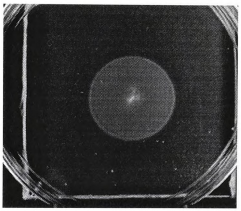


Figure 10. *E. coli* forming ring in response to glucose.

Microsensor and microbiosensor readings were taken across the ring at a depth of 1 mm. An image was recorded of the swarm plate with the Pulnix image analysis system. Figure 11 shows the combined results of the sensor readings and the image analysis. The chemotactic wave (see Section 4.2) is visible where the gradients have the steepest slopes.

Figure 1

4.2 Deve

The

coupled co

or carbon s

where J_u i

nutrient (H

review, se

migration

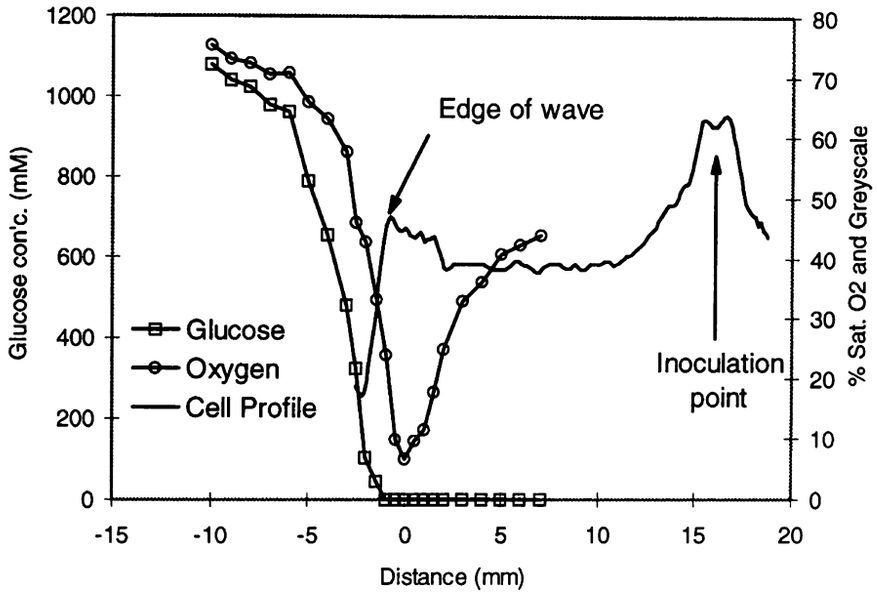


Figure 11. Use of microsensors, microbiosensors, and image analysis, DGC modeling

4.2 Development of model

The mathematical model of the Diffusion Gradient Chamber system consists of coupled conservation equations for the microbes, the chemoattractant(s), and the nutrient or carbon source. The cell balance equation may be written as

$$\frac{\partial u}{\partial t} = -\nabla \cdot J_u + f(H)u \tag{9}$$

where J_u is the cell flux, u is the cell density, and f(H) is a function for cell growth on a nutrient (H). Many forms of the flux equations for chemotaxis have been suggested (for a review, see Ford, 1992). A commonly used constitutive equation to describe cell migration at the population level is that derived by Keller and Segel (1971):

where μ is

model pres

chemotacti

where V_{cs}

chemotacti

The

the RTBL r

but has be

model for

assumes th

to the spat

(Frymier, *e*

$$J_u = -\mu \nabla u + V_u u \tag{10}$$

where μ is the random motility coefficient and V_u is the chemotactic velocity. For the model presented here, the Keller-Segel equation has been modified by adding a second chemotactic flux, to give

$$J_u = -\mu \nabla u + V_{uS} u + V_{uQ} u \tag{11}$$

where V_{uS} is the chemotactic velocity in response to a chemoattractant S, and V_{uQ} is the chemotactic velocity to a chemoattractant Q. Combining Equations (9) and (11) gives

$$\frac{\partial u}{\partial t} = \mu \nabla^2 u - \nabla \cdot (V_{uS} u) - \nabla \cdot (V_{uQ} u) + f(H) u \tag{12}$$

The chemotactic velocity equation proposed by Rivero, *et al.* (1989) (known as the RTBL model) is strictly applicable only to one-dimensional movement of the bacteria, but has been shown to give good agreement with a more rigorous three-dimensional model for a range of parameter values (Frymier, *et al.*, 1992). The RTBL model also assumes that the temporal gradient of chemoattractant has a small contribution compared to the spatial gradient, and evidence to support this assumption has been provided (Frymier, *et al.*, 1994). The RTBL chemotaxis term is given by

$$V_{uS} = \upsilon \tanh\left\{ \sigma \upsilon \frac{N_{TS} K_{DS}}{(K_{DS} + S)^2} \nabla S \right\} \tag{13}$$

where υ is

total numbe

for the rec

velocity can

where χ_{os},

terms of in

A similar e

where χ_{oq}

dissociatio

chemotacti

from indiv

that both c

assay.

here υ is the swimming speed of a single cell, σ is the tumbling frequency, N_{TS} is the ɔtal number of receptors for the chemoattractant S, and K_{DS} is the dissociation constant ɔr the receptor-S complex. For shallow gradients, the modified RTBL chemotactic ɛlocity can be approximated by

$$V_{uS} = \chi_{0S} \frac{K_{DS}}{(K_{DS} + S)^2} \nabla S \tag{14}$$

here χ_{0S}, the chemotactic sensitivity coefficient to the attractant S, can be expressed in rms of individual cell parameters by

$$\chi_{0S} = \sigma \upsilon^2 N_{TS} \tag{15}$$

similar equation can be written for the chemoattractant Q as

$$V_{uQ} = \chi_{0Q} \frac{K_{DQ}}{(K_{DQ} + Q)^2} \nabla Q \tag{16}$$

here χ_{OQ} is the chemotactic sensitivity coefficient to the attractant Q and K_{DQ} is the ssociation constant for the receptor-Q complex. Rivero *et al.* (1989) showed that the ıemotactic sensitivity coefficient and the random motility coefficient can be calculated ɔm individual cell parameters, and Rivero-Hudec and Lauffenburger (1986) showed at both can be measured experimentally through population assays such as the capillary say.

The

and (16) w

yield

$$\frac{\partial u}{\partial t}=\mu\nabla^2$$

where v is

half-saturat

chemotacti

better thar

reproduce

achieved g

coli to simu

The

Monod-typ

chemoattra

uptake. Th

nutrient (

chemoattra

The

The cell balance used for the DGC system is derived by combining Equations (14) d (16) with the random motility flux term and a term for growth (on substrate H) to eld

$$\frac{\partial u}{\partial t}=\mu\nabla^2 u-\chi_{0S}\nabla\cdot\left[\left(\frac{K_{DS}}{(K_{DS}+S)^2}\right)u\nabla S\right]-\chi_{0Q}\nabla\cdot\left[\left(\frac{K_{DQ}}{(K_{DQ}+Q)^2}\right)u\nabla Q\right]+\frac{vH}{C_0+H}u \qquad (17)$$

here v is the maximum specific growth rate of the cells growing on H, and C_0 is the lf-saturation constant for growth on H. The assumption of simple additivity of the emotactic responses to multiple attractant gradients has been shown to be as good or tter than more complex interaction models, but lacks the ability to completely produce experimental observations (Strauss, *et al.*, 1995). Boon and Herpigny (1986) hieved good agreement with experimental results when modeling the response of *E. li* to simultaneous gradients of glucose and oxygen using this assumption.

The cell growth term and the chemoattractant consumption terms are modeled as nod-type (Bailey and Ollis, 1986) saturation processes. Growth of cells due to moattractant uptake is assumed to be negligible compared to growth due to nutrient ake. This assumption is reasonable for many experimental systems because the rient (usually glycerol) is present at a much higher concentration than the moattractant.

The two chemoattractant balances are given by

$$\frac{\partial S}{\partial t}=D_S\nabla^2 S-\frac{v_S S}{C_S+S}u \qquad (18)$$

and

where D_S a

chemoattra

consumptio

The

where D_H i

on H.

In t

diffusion c

previously

model (Em

4.2.1 Bou

A t

illustrate th

2.5 cm, and

on all boun

and

$$\frac{\partial Q}{\partial t} = D_Q \nabla^2 Q - \frac{v_Q Q}{C_Q + Q} u \tag{19}$$

where D_S and D_Q are the chemoattractant diffusion coefficients; v_S and v_Q are the specific chemoattractant consumption coefficients; and C_S and C_Q are the saturation constants for consumption of S and Q, respectively.

The nutrient balance is given by

$$\frac{\partial H}{\partial t} = D_H \nabla^2 H - \frac{vH}{C_0 + H} \frac{u}{Y_H} \tag{20}$$

where D_H is the nutrient diffusion coefficient, and Y_H is the yield coefficient for growth on H.

In both of the chemoattractant balances and in the nutrient balance, the respective diffusion coefficients are assumed to be constant and the medium isotropic. We have previously confirmed that substrate diffusion into the DGC is accurately described by the model (Emerson *et al.*, 1994).

4.2.1 Boundary conditions

A two-dimensional schematic representation of the DGC is shown in Figure 12 to illustrate the geometric parameters used in the mathematical model. The dimension R is 2.5 cm, and r is 2 cm. For the cell balance, a zero total flux boundary condition is applied on all boundaries (Ω) of the chamber. This boundary condition, shown below, states that

the overall

to zero on

$\mu\nabla$

Ne

boundaries

diffuse ac

transfer. T

where S_{res}

reservoirs,

permeable

Ou

walls of th

If the rese

condition

e overall flux, given by the sum of the random motility and chemotaxis fluxes, is equal zero on the boundaries.

$$\left\{\mu\nabla^2 u - \chi_{0S}\nabla\cdot\left[\left(\frac{K_{DS}}{(K_{DS}+S)^2}\right)u\nabla S\right] - \chi_{0Q}\nabla\cdot\left[\left(\frac{K_{DQ}}{(K_{DQ}+Q)^2}\right)u\nabla Q\right]\right\}\Bigg|_{\Omega} = 0 \qquad (21)$$

Neumann boundary conditions are used for the chemoattractant S on the undaries where the reservoir is open to the arena. The chemoattractant is assumed to ffuse across the semi-permeable membrane, which imposes a resistance to mass nsfer. This boundary condition is written as

$$\left.\frac{\partial S}{\partial y}\right|_{y=0} = \frac{k_S}{D_S}\left(S|_{y=0} - S_{res,S}\right) \text{ and } \left.\frac{\partial S}{\partial y}\right|_{y=5} = \frac{-k_S}{D_S}\left(S|_{y=5} - S_{res,N}\right) \qquad (22)$$

ere $S_{res,S}$ and $S_{res,N}$ are the concentrations of the chemoattractant in the south and north ervoirs, respectively, and k_S is the mass transfer coefficient for S across the semi-rmeable membrane.

Outside the reservoir openings, there is no flux of chemoattractant across the lls of the chamber:

$$\left.\frac{\partial S}{\partial y}\right|_{y=0} = 0 \quad \text{and} \quad \left.\frac{\partial S}{\partial y}\right|_{y=5} = 0 \qquad (23)$$

he reservoir is sealed with a non-permeable membrane, then the no flux boundary dition would apply across the entire side. The boundary conditions for Q and H are

similar ma

thesis, the

initially pr

concentrat

4.2.2 Init

In

inoculated

shape of th

where u_0 i

and w we

experimen

that x'=0 a

Th

initial cor

concentrat

y, where C

4.2.3 Exp

Ex

streptomy

imilar mathematically to those of S. For all simulations presented in this section of the hesis, the concentration of Q in each open reservoir was equal to the concentration of Q nitially present in the arena, and the concentration of H in each reservoir was equal to the oncentration of H initially in the arena.

.2.2 Initial conditions

In the experiments used to validate the model, the center of the chamber was noculated with a micropipette. An exponential function was used to approximate the hape of the injected cell peak:

$$u(x',y') = \frac{u_0}{\exp(w\sqrt{x'^2 + y'^2})} \tag{24}$$

here u_0 is the initial concentration of cells and w is a peak width factor. The values of u_0 nd w were calculated to yield the same number of cells in the peak as were added xperimentally (~$4x10^7$ cells, Emerson *et al.*, 1994). The variables x' and y' are defined so hat x'=0 and y'=0 occur at the center of the arena.

The initial condition for the chemoattractant S is S(x,y)=0 for all x and y. The itial condition for H is that $H(x,y)=H_0$ for all x and y, where H_0 is the initial oncentration of H in the arena. The initial condition on Q is that $Q(x,y)=Q_0$ for all x and , where Q_0 is the initial concentration of Q in the arena.

.2.3 Experiments used to validate model

Experiments in the DGC were carried out as described in Emerson *et al.* (1994). A reptomycin resistant strain of *E. coli* HCB 33 (=RP437) that is wild type for motility

and chemotaxis w

This medium was

methionine, and th

The glycerol ser

chemoattractant f

and either 0.1 m

M63 medium on

initially containe

in LB broth.

In these e

north the sink. T

μm pore-size pol

chemotaxis was used. A mineral salts medium (M63) was used for all experiments. ; medium was supplemented with 2 mM glycerol, the amino acids histidine, leucine, hionine, and threonine required for this auxotrophic strain, and 125 μg/l streptomycin. glycerol served as the carbon and energy source for growth, but it is not a moattractant for *E. coli.* The source reservoir contained supplemented M63 medium either 0.1 mM or 1 mM aspartate, while the sink reservoir contained supplemented 3 medium only. The medium in the arena was stabilized with 0.15% agarose and ially contained no aspartate. The *E. coli* were grown overnight with aeration at 30°C LB broth.

In these experiments, the south reservoir (see Figure 12) was the source, and the th the sink. These source and sink reservoirs were separated from the arena by 0.05 pore-size polycarbonate filter membranes (Poretics Corp., Livermore, CA).

Figure

The east

silastic sheeting

autoclaved at 12

in the flasks as l

sterile filters. Th

through a sterile

Both the

reservoirs at 2.5

operated for a

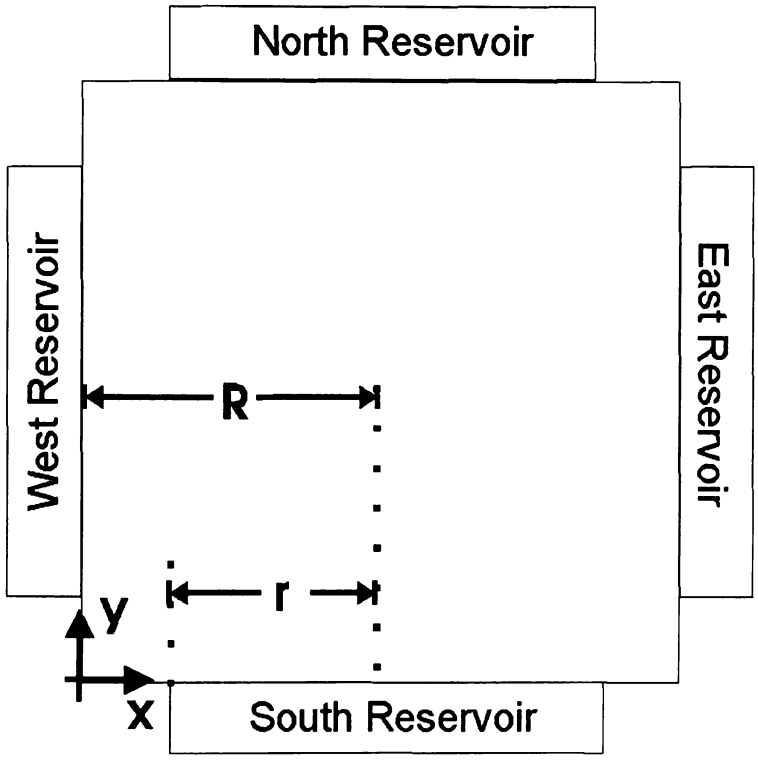


Figure 12. Schematic diagram of DGC, showing modeling coordinates.

The east and west reservoirs were sealed off from the arena with non-permeable silastic sheeting (Dow Corning, Inc., Midland, MI). The source and sink liquids were autoclaved at 121°C to sterilize, and allowed to cool. To prevent a vacuum from forming in the flasks as liquid was pumped out, the flasks were vented to the atmosphere through sterile filters. The gas port to the DGC (see Figure 3) was also vented to the atmosphere through a sterile filter, to allow the headspace to be replenished with fresh air.

Both the source and sink solutions were pumped through their respective reservoirs at 2.5 ml/hr. To allow the aspartate gradient to partially form, the system was operated for a given time before inoculation. The amount of time the gradient was

allowed to form b

the arena of the D

the cells evenly th

4.2.4 Computer

The mathe

algorithm (Carna

difference equati

equation. The fir

only in the y-dire

ADI method is a

discretization err

More details on t

FORTRAN 77 a

Q, and H were sc

was added, in or

from the FORTR

MATLAB progr

4.2.5 Modeling

In order

determined inde

laser diffraction

allowed to form before inoculation was varied between experiments. The center point in the arena of the DGC was inoculated with 10 µl of *E. coli* using a micropipette to disperse the cells evenly throughout the depth of the agarose.

4.2.4 Computer simulations

The mathematical model was solved with an Alternating Direction Implicit (ADI) algorithm (Carnahan *et al.*, 1969; Chapra and Canale, 1988). The ADI method uses two difference equations to solve each two-dimensional unsteady-state partial differential equation. The first difference equation is implicit only in the x-direction and the second only in the y-direction. The equations are solved in succession at time steps of $\Delta t/2$. The ADI method is an unconditionally stable method with which convergence occurs with a discretization error of the order $[(\Delta t)^2+(\Delta x)^2]$. For the model presented here, $\Delta x=\Delta y$. More details on the ADI method can be found in Appendix B. The program is written in FORTRAN 77 and executed in the UNIX operating system. The balance equations for S, Q, and H were solved by the model for a specified amount of time before the cell balance was added, in order to simulate the gradient initiation time before inoculation. Output from the FORTRAN program was imported to MATLAB version 4.0 for Windows. The MATLAB program was used for image analysis and for graphical output.

4.2.5 Modeling parameters

In order to validate the mathematical model, all of the parameter values were determined independently. The random motility coefficient, µ, was measured using a laser diffraction capillary assay (LDCA) developed in our laboratory (Schmidt *et al.*,

1997). The LDC

movement in var

and modeling. th

indicate that the

range of 0.15 to

the experimenta

experiments. The

assumed to be ir

rate, v_B, was de

bacterial populat

to calculate the g

The diffu

correlation (Will

and glycerol, re

30°C by a correl

Green, 1984); A

cells growing on

Values

dissociation con

from the iterat

1997). The LDCA consists of a capillary tube through which a laser is directed. Cell movement in varying agar concentrations can be observed, and through image analysis and modeling, the random motility coefficient can be determined. LDCA experiments indicate that the random motility decreases linearly with agar concentration over the range of 0.15 to 0.30% agar. The random motility coefficient used in the simulations is the experimentally measured value for the agar concentration used in the DGC experiments. The LDCA is described in more detail in Section 4.5. For this model, μ is assumed to be independent of the chemoattractant gradient. The maximum cell growth rate, v_H, was determined from the experimentally measured doubling time, t_d, of the bacterial population using the relationship

$$v_H = \frac{\ln(2)}{t_d} \tag{25}$$

to calculate the growth rate.

The diffusion coefficient for aspartate (D_S) was calculated by the Wilke-Chang correlation (Wilke and Chang, 1955) The diffusion coefficients for D_Q and D_H (oxygen and glycerol, respectively) at 25°C were obtained from the literature, and adjusted to 30°C by a correlation for the temperature dependence of diffusion coefficients (Perry and Green, 1984). An approximate value for Y_H was calculated by an electron balance for cells growing on glycerol.

Values for the chemotactic sensitivity coefficient for aspartate, χ_{os}, and the dissociation constant for the receptor-attractant complex for aspartate, K_{DS}, were taken from the literature. Values for the chemotactic sensitivity to various chemoattractants

have been publi

published by Mac

The value

range of those c

specific chemoat

the maximum ce

estimated by adj

DGC (Emerson,

(glycerol), S (asp

is given by $k=D_e$

of the membrane

and Chang, 195

predicted outcor

coefficient. Valu

e been published (Ford, 1992). Values for the dissociation constant have been
lished by Macnab (1987).

The values for the saturation constants C_H, C_S, and C_Q were chosen to be in the
ge of those commonly reported for *E. coli* (Bailey and Ollis, 1986). The maximum
cific chemoattractant consumption coefficient (v_S) was taken to be close in value to
maximum cell growth rate on H. The mass transfer coefficients k_H, k_S, and k_Q were
mated by adjusting the k value for glucose, determined in previous work with the
C (Emerson, *et al.*, 1994) by the ratio of the molecular weight of glucose to H
cerol), S (aspartate), or Q (oxygen). The mass transfer coefficient across a membrane
iven by $k=D_{eff}/l$, where D_{eff} is the effective diffusion coefficient, and l is the thickness
he membrane (Cussler, 1984). D_{eff} varies with the molar volume, V, to the -0.5 (Wilke
Chang, 1955) to -0.6 power (Sitaraman *et al.*, 1963). In results not shown, the
dicted outcomes were relatively insensitive to small variations in the mass transfer
fficient. Values for parameters used in the modeling simulations are given in Table 2.

T

4.2.6 Results o

The cell

those observed

was introduced

chamber at a co

center of the ch

The exp

Figure 13. The

Table 2. Parameter values for validation of model

Parameter	Value
μ	0.0010 $cm^2 h^{-1}$
χ_{0Q}	0.080 $cm^2 h^{-1}$
χ_{0S}	0.085 $cm^2 h^{-1}$
C_0	4.08×10^{-6} $g_H cm^{-3}$
C_Q	6.70×10^{-8} $g_Q cm^{-3}$
C_S	5.50×10^{-5} $g_S cm^{-3}$
D_H	0.01 $cm^2 h^{-1}$
D_Q	0.09 $cm^2 h^{-1}$
D_S	0.033 $cm^2 h^{-1}$
K_{DQ}	3.30×10^{-5} $g_Q cm^{-3}$
K_{DS}	2.00×10^{-6} $g_S cm^{-3}$
v	0.35 h^{-1}
v_Q	0.02 $g_Q g_u^{-1} h^{-1}$
v_S	0.60 $g_S g_u^{-1} h^{-1}$
Y_H	0.50 g_u/g_H

.6 Results of validation simulations

The cell growth and migration patterns predicted by the model were compared to se observed experimentally under two sets of conditions. In Run 1, 1.0 mM aspartate s introduced into the south reservoir. Glycerol was initially present throughout the ımber at a concentration of 5 mM. After the gradient was established for 6 hours, the ıter of the chamber was inoculated with *E. coli*.

The experimental photos and modeling results at four time points are shown in ;ure 13. The results of the computer simulation are shown in the left-hand column and

photographs of t

modeling simula

observed experim

wave of cells. cha

developed and m

increased with ti

approached the

source slowed n

source. The thi

chemoattractant

In Run

(0.1mM aspartat

in the left-hand

experiment, the

Run 1, chemota

wave were evi

shapes of the

growth pattern

approached the

tographs of the experiment in the right-hand column. In both the photos and the deling simulations, lighter areas correspond to higher cell density. Several trends erved experimentally were reproduced by the model. The first trend was the band or e of cells, characterized by a locally high cell concentration near the pattern edge, that eloped and migrated toward the chemoattractant source. The height of the wave peak eased with time. The second trend was that the velocity of the wave decreased as it roached the chemoattractant source. Because the edge of the pattern closest to the rce slowed more than the rest, the pattern tended to broaden as it approached the rce. The third trend was the less prominent wave that migrated away from the moattractant source reservoir.

In Run 2, the chemoattractant concentration in the source (south) reservoir mM aspartate) was one tenth that in Run 1. Figure 14 shows the computer simulation he left-hand column and experimental photographs in the right-hand column. In this eriment, the chamber was inoculated 6.5 hours after the gradient was initiated. As in n 1, chemotactic migration toward the aspartate source and formation of a chemotactic ve were evident in both the experiment and the model predictions. However, the pes of the patterns were significantly different in the two runs. In particular, the wth pattern for Run 2 was more elongated and exhibited less flattening as it roached the source.

Figure 1

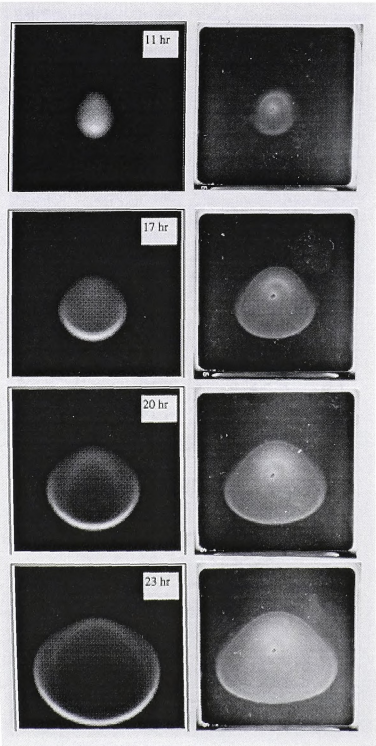


Figure 13. Simulation compared to experiment for 1.0 mM aspartate source

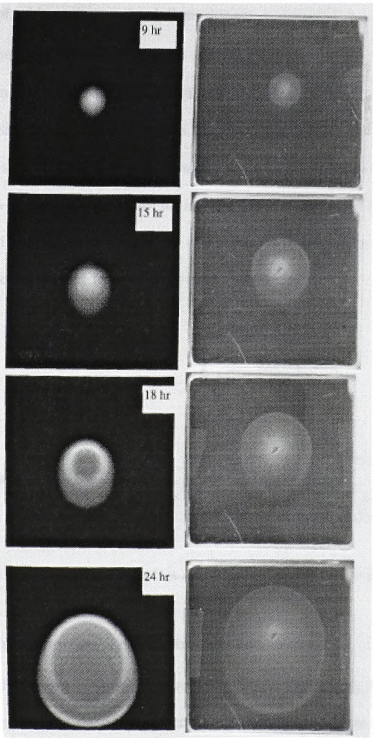


Figure 14. Simulation compared to experiment for 0.1 mM aspartate source.

4.2.7 Effects of

The simu

equally spaced n

of the interval si

and Run 1 was s

two central axes

on the predicted

did show a mark

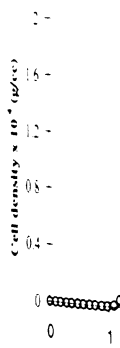


Figure 15

4.2.8 Concent

The ce

concentration

profiles are sho

All four graph

Effects of grid-spacing on simulations

The simulations shown in Figure 13 and Figure 14 used a grid consisting of 81 ly spaced nodes in each direction, for a total of 81^2 (6561) points. To test the effects : interval size on the predicted profiles, the grid was reduced to 51^2 (2601) points, :un 1 was simulated again. Figure 15, which gives the cell concentrations along the entral axes of the DGC, shows that the increase from 51^2 to 81^2 had almost no effect e predicted profiles. Other simulations (not shown) at grid-spacings of 25^2 and 15^2 iow a marked difference from the results at 51^2 intervals.

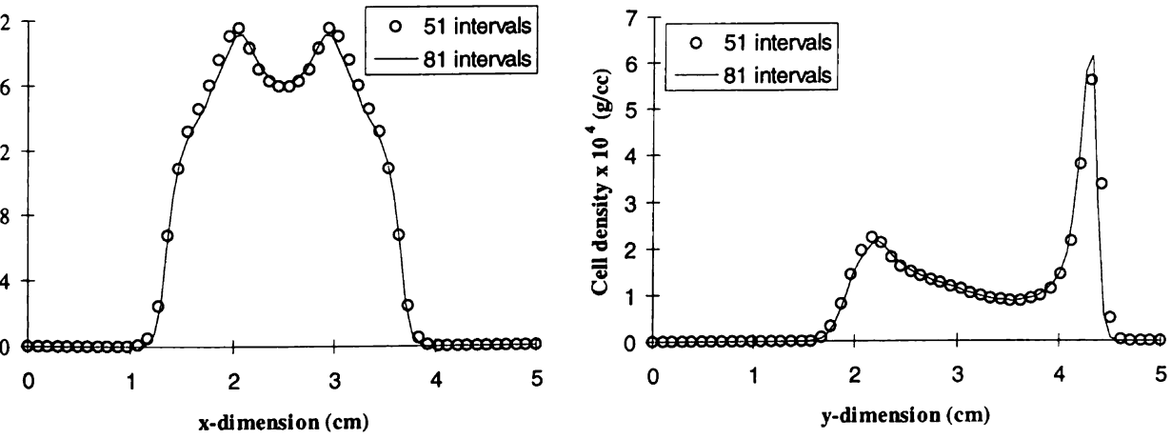


Figure 15. Cell concentration profiles across the two central axes of the DGC.

Concentration profiles of chemoattractants and nutrient

The cell patterns developed in response to the underlying, time-dependent ntration profiles of the chemoattractants and nutrient. Examples of these latter es are shown, along with the corresponding cell profile, in Figure 16 and Figure 17. our graphs have the concentration, g/cm^3, on the vertical axis, and the spatial

4.2.7 Effects of

The simu

equally spaced n

of the interval s

and Run 1 was s

two central axes

on the predicted

did show a mark

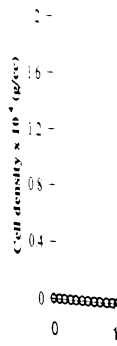


Figure 1

4.2.8 Concen

The c

concentration

profiles are sh

All four grap

4.2.7 Effects of grid-spacing on simulations

The simulations shown in Figure 13 and Figure 14 used a grid consisting of 81 equally spaced nodes in each direction, for a total of 81^2 (6561) points. To test the effects of the interval size on the predicted profiles, the grid was reduced to 51^2 (2601) points, and Run 1 was simulated again. Figure 15, which gives the cell concentrations along the two central axes of the DGC, shows that the increase from 51^2 to 81^2 had almost no effect on the predicted profiles. Other simulations (not shown) at grid-spacings of 25^2 and 15^2 did show a marked difference from the results at 51^2 intervals.

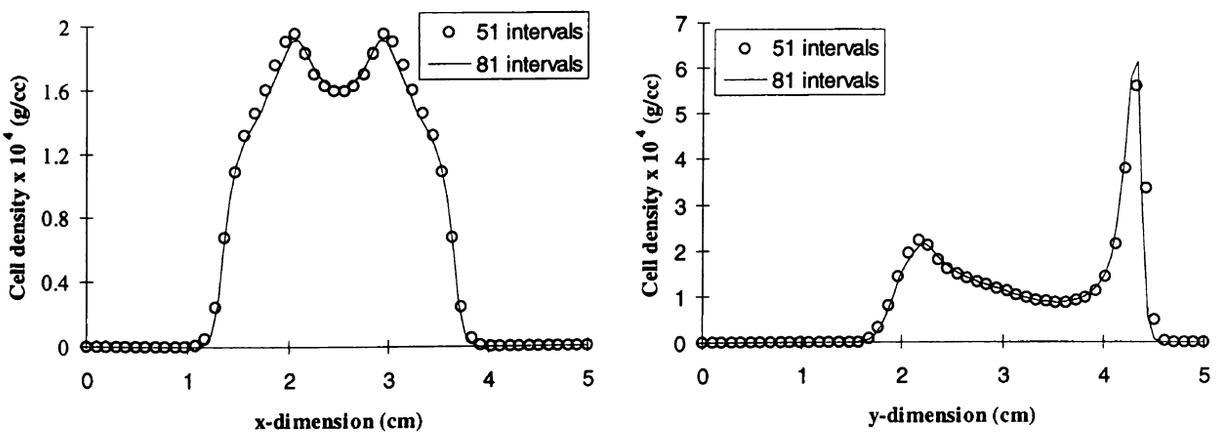


Figure 15. Cell concentration profiles across the two central axes of the DGC.

4.2.8 Concentration profiles of chemoattractants and nutrient

The cell patterns developed in response to the underlying, time-dependent concentration profiles of the chemoattractants and nutrient. Examples of these latter profiles are shown, along with the corresponding cell profile, in Figure 16 and Figure 17. All four graphs have the concentration, g/cm^3, on the vertical axis, and the spatial

position, in cm.

graphing progra

sition, in cm, on the horizontal axes. Note that the jagged edges are artifacts of the

aphing program.

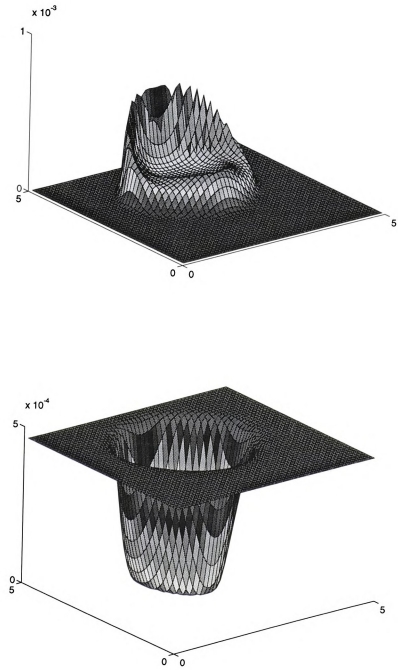


Figure 16. Top: Typical cell profile. Bottom: nutrient profile.

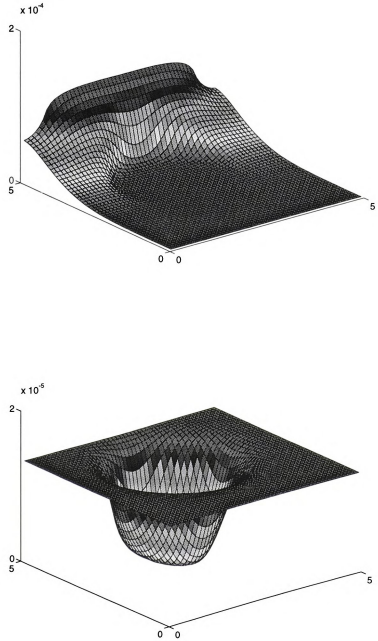


Figure 17 Top: Typical S profile. Bottom: Q profile.

Experimental v

Figure 18 (Wid

oxygen profiles

away from the a

Oxygen Concentration (μM)

To val

measured in a

the same as

compared to

results of the

imental verification of these profiles was more challenging. However, as shown in e 18 (Widman *et al.*, 1997), microelectrodes have been used to confirm that sharp n profiles exist across both the cell front moving toward the aspartate source and from the aspartate source.

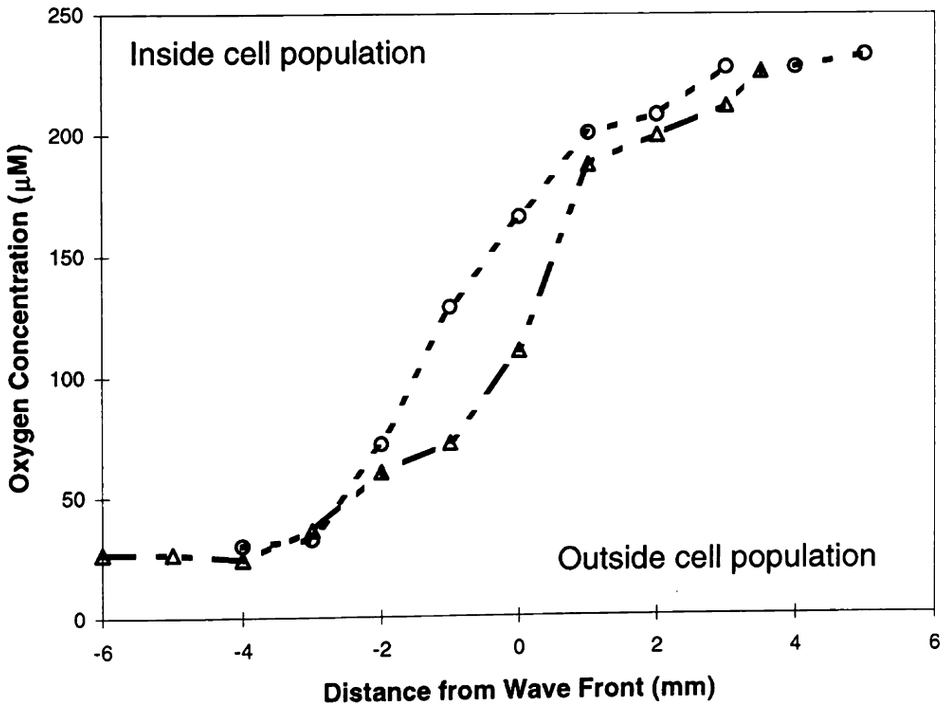


Figure 18. Oxygen gradients in the DGC arena

To validate the model's predictions of chemical gradients, glucose gradients were ured in a miniature DGC using microbiosensors. The miniature DGC is fashioned ame as a regular DGC, but has an arena length and width of only 3.0 cm, as ared to 5.0 cm for the regular DGC. Figure 19 (Widman *et al.*, 1997) shows the s of these measurements for a single time point, along four different lines in the

miniature DGC.

openings not co

model accounts

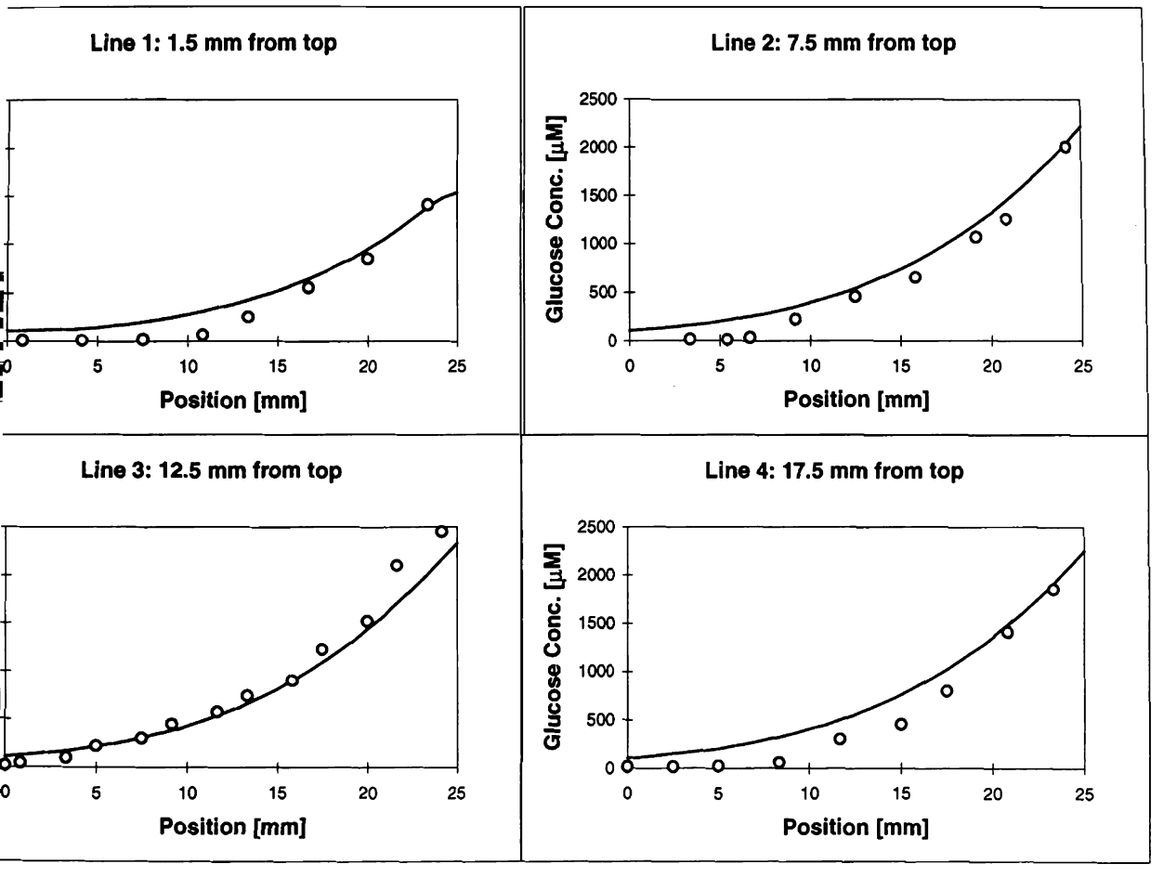
e DGC. The variations in the profiles for the different lines arise from reservoir

s not completely spanning the width of the DGC, as shown in Figure 12. The

ccounts for this geometry.

Glucose Conc. [µM]
2500
2000
1500
1000
500
0

Glucose Conc. [µM]
2500
2000
1500
1000
500
0



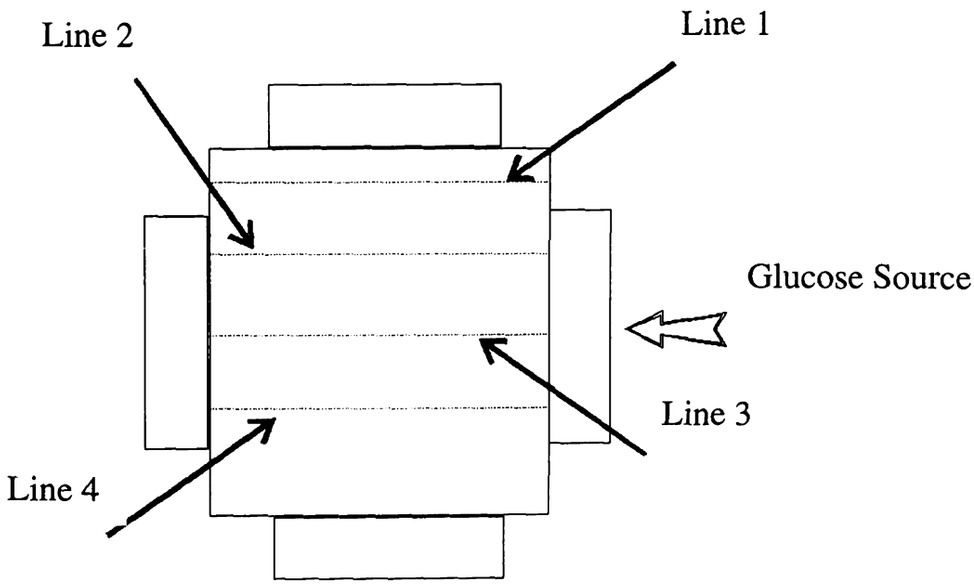


Figure 19. Glucose gradients at several positions in the DGC

4.2.9 Assumpt

The moc

and boundary c

between the ex

assumptions us

In the F

depend upon b

neglect these c

random motilit

to chemotaxis.

relatively inser

Anothe

bacteria are r

Equation (13))

assay (Ford *et*

concentrations

chemoattractar

prevents the p

cells. In our r

the simulation

speed of 7.92

gradient simp

mptions and simplifications in the mathematical model

model was able to reproduce several experimental trends for different initial ary conditions using the same set of parameters. The reasonable agreement ne experimental and modeling results provides support for the simplifying s used. These assumptions are discussed below in more detail.

he RTBL model of chemotaxis, the random motility coefficient is shown to on both temporal and spatial chemoattractant gradients. We have chosen to se dependencies and assume constant μ. The magnitude of the flux due to otility is predicted to be several orders of magnitude smaller than the fluxes due axis. In results not shown, the predicted growth patterns have been found to be nsensitive to μ. Therefore an exact calculation of μ is not deemed necessary.

other assumption is that the gradients of chemoattractant encountered by the re relatively shallow (i.e. Equation (14) is a reasonable approximation to 13)). This assumption has been found to be adequate for modeling the capillary d *et al.*, 1990; Rivero and Lauffenburger, 1986). However, the locally high ions of cells in the chemotactic waves are associated with significant actant gradients. The use of the hyperbolic tangent term in Equation (13) ne predicted chemotactic velocity from exceeding the swimming speed of the ur modeling simulations, the maximum predicted velocity at any time during tion is only 0.154 cm/h (0.43 μm/s), much less than a typical cell swimming .92 cm/h (22 μm/s) (Frymier, *et al.*, 1994). To further validate the shallow-implification, velocity profiles from the hyperbolic tangent model and the

shallow gradien

gradients of S p

(Ford and Lauff

swimming spee

the number of r

used in the sim

sensitivity of 6.

in Figure 20 fo

introduced by

encountered in

ow gradient model were calculated by Equations (13) and (14), respectively, for the ients of S predicted by the model. Typical values for the individual cell parameters d and Lauffenburger, 1990) for *E. coli* were chosen for use in Equation (13). The cell ming speed, υ, was taken to be 30 μm/s, and the tumbling frequency multiplied by umber of receptors, σN_T, was 75 s. The value for K_D, 2.0×10^{-6}, was the same as that in the simulations. Equation (15) was used to calculate a value for the chemotactic tivity of 6.74×10^{-4} cm^2/s from the single cell parameters. The comparisons are shown gure 20 for the first and last time points in Runs 1 and 2. There is virtually no error duced by substituting Equation (14) for Equation (13) for any of the gradients untered in either run.

Chemotactic velocity to S [cm/s]
Chemotactic velocity to S [cm/s]

Figure

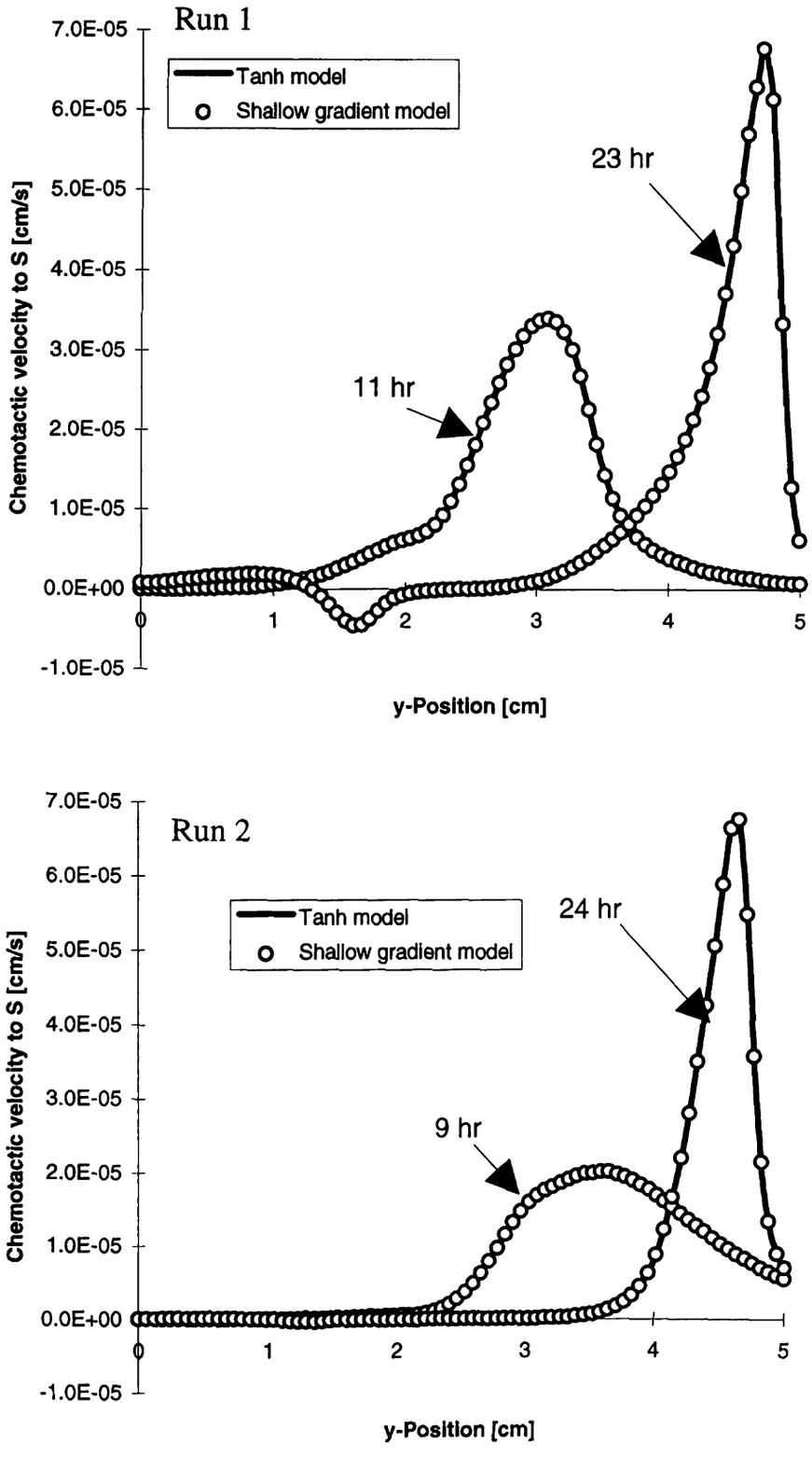


Figure 20. Comparison of model with and without hyperbolic tangent term.

A third

constant throu

uniformly disp

gradients to f

significant ve

microelectrode

partially respo

4.3 Analysi

4.3.1 Flux ca

The m

from chemota

of fluxes due t

where J_{uu} is th

to S-gradients

magnitude an

20 hr and are

arrow corresp

corresponds t

the flux is cal

A third assumption is that the concentration of all components in the model is onstant throughout the depth (the z-direction) of the gel. Oxygen, which is initially iformly dispersed in the gel, is quickly consumed by the cells, causing vertical adients to form. In experiments (Emerson, unpublished data), we have measured gnificant vertical oxygen gradients in the zone of cell growth with oxygen icroelectrodes. The variation in oxygen concentration with depth is thought to be rtially responsible for the deviations between the experimental and predicted profiles.

.3 Analysis of modeling results

3.1 Flux calculations

The model was used to calculate the direction and magnitude of cell fluxes arising om chemotaxis and random motility in the DGC. Equation (11) was rewritten in terms f fluxes due to single driving forces as

$$J_u = J_{uu} + J_{uS} + J_{uQ} \tag{26}$$

here J_{uu} is the flux due to random motility, J_{uS} is the flux due to chemotaxis in response S-gradients, and J_{uQ} is the flux due to chemotaxis in response to Q-gradients. The agnitude and direction of each flux term were calculated individually for Run 1 at time) hr and are presented at each node point in Figure 21-Figure 24. The length of each row corresponds to the magnitude of the flux, and the direction of the arrow rresponds to the direction of the flux. The tail of the arrow is located at the point where e flux is calculated. (Note that in Figure 21-Figure 24 only the top 20% of the fluxes are

shown to reduc

the lighter (wh

illustrated. Th

chemoattractan

arrows onto th

profile. and th

Figure 23 sho

obtained by ad

type of cell flu

T

Fl

vn to reduce the number of arrows and to improve clarity.) In each of these figures, ighter (whiter) shading corresponds to higher concentration of the component being trated. The top graph in Figure 21 overlays the J_{uS} flux arrows onto the noattractant S gradient profile, and the lower graph in Figure 21 overlays the J_{uS} flux ws onto the cell profile. The top graph in Figure 22 shows J_{uQ} overlayed onto the Q ile, and the bottom graph in Figure 22 shows J_{uQ} overlayed onto the cell profile. re 23 shows J_{uu}, the flux due to random motility. Figure 24 shows the total flux, ined by adding the three flux components together. The maximum magnitude of each of cell flux is given in Table 3.

Table 3. Maximum predicted cell fluxes for Run 1 at 20 hours.

Flux component	**Maximum magnitude of flux ($\times 10^5$ $g_{cells}/cm^2 \cdot h$)**
J_{uS}	8.05
J_{uQ}	3.83
J_{uu}	0.0000131
J_u	11.4

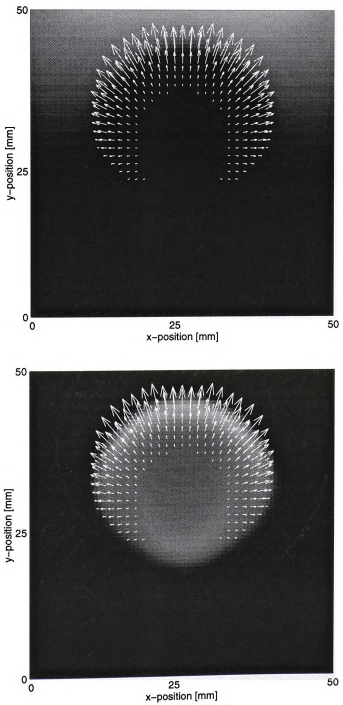


Figure 21. Fluxes due to response to S chemoattractant.

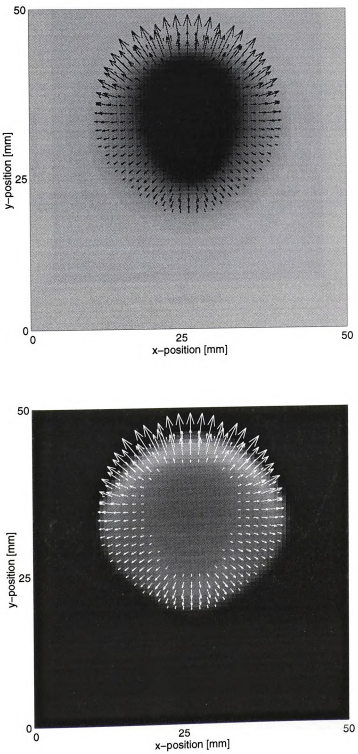


Figure 22. Fluxes due to response to Q chemoattractant.

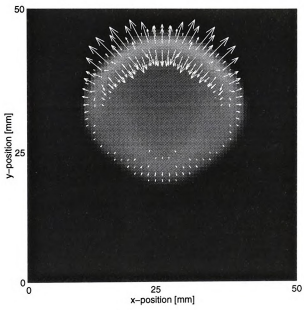


Figure 23. Fluxes due to random motility.

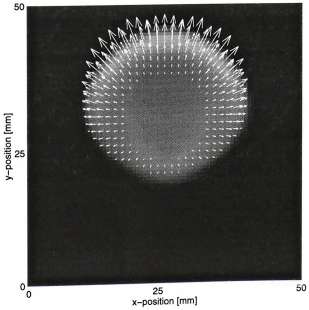


Figure 24. Total mass fluxes.

4.3.2 Recepto

The de

be calculated

ϕ_S, as the aver

S divided by t

receptors wer

$$\phi_s = -$$

The value of

response is c

zero, the che

factor has be

13 and Figu

Calculation c

of the maxim

3.2 Receptor saturation

The degree to which receptor saturation suppresses the chemotactic response can e calculated from the simulation results. We define a global chemotactic response factor, s, as the average cell flux (along a line of constant x) in response to the chemoattractant divided by the average cell flux (along the same constant x line) that would result if the ceptors were completely free of any S. This definition is expressed mathematically as

$$\phi_S = \frac{\int_0^{2R}\left[\chi_{0S}\frac{K_{DS}}{(K_{DS}+S)^2}\frac{\partial S}{\partial y}u\right]dy \Big/ \int_0^{2R} dy}{\int_0^{2R}\left[\chi_{0S}\frac{1}{K_{DS}}\frac{\partial S}{\partial y}u\right]dy \Big/ \int_0^{2R} dy} = \frac{\int_0^{2R}\left[\left(1+\frac{S}{K_{DS}}\right)^{-2}\frac{\partial S}{\partial y}u\right]dy}{\int_0^{2R}\left[\frac{\partial S}{\partial y}u\right]dy} \qquad (27)$$

he value of ϕ_S will vary between 0 and 1. If ϕ_S is close to one, then the chemotactic sponse is close to the maximum it can attain for the given gradient. As ϕ_S approaches ero, the chemotactic response to the gradient of S diminishes. The chemotactic response ctor has been calculated for Run 1 and Run 2 at each of the time points shown in Figure 3 and Figure 14. Figure 25 shows ϕ_S plotted against the x-position for both runs. alculation of ϕ_S was limited to areas where the cell concentration was greater than 0.1% f the maximum cell concentration occurring at that time point.

0.2

ϕ_S 0.

0

0

0

0

0

ϕ_S 0

0

0

0

0

Figure 2

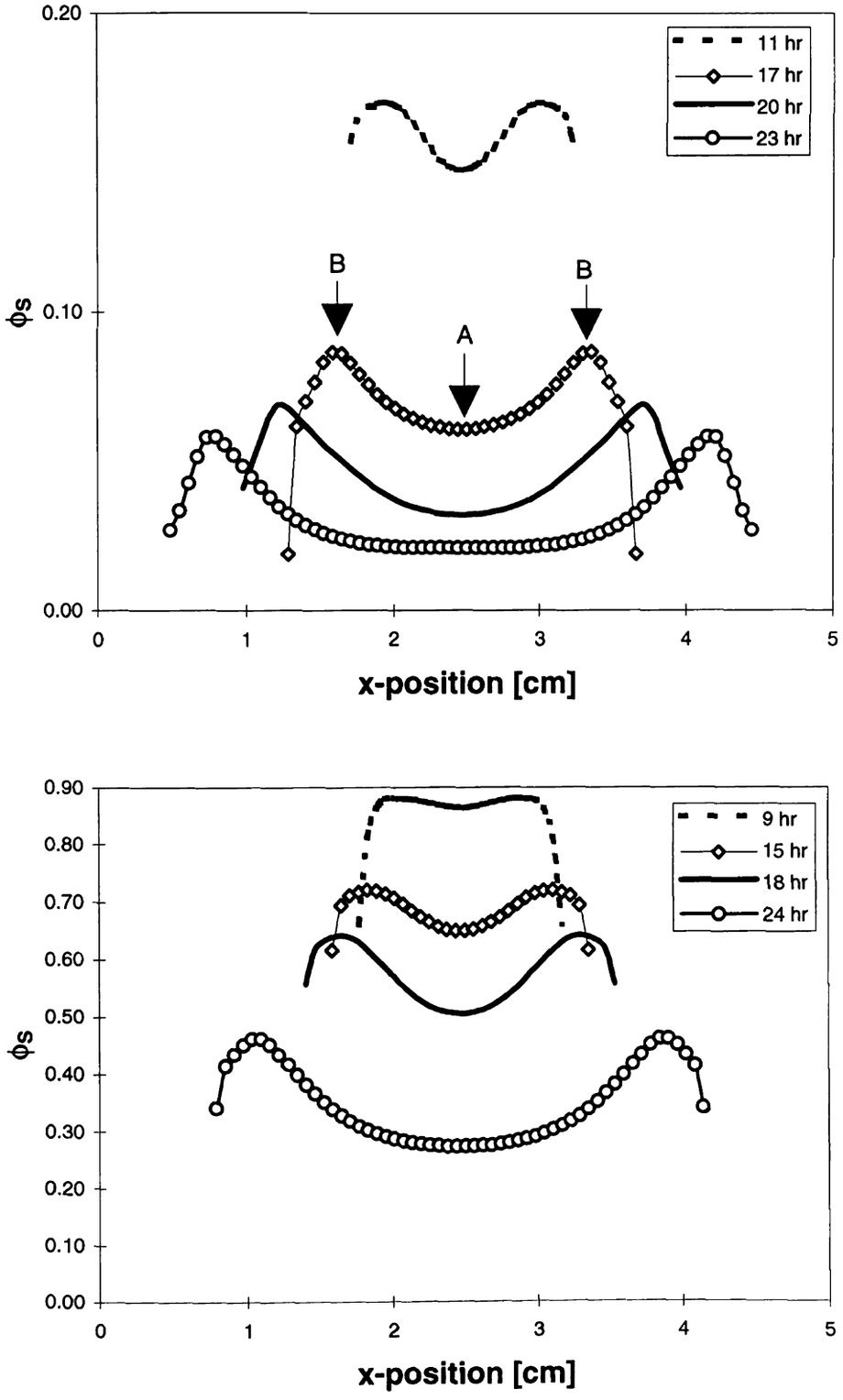


Figure 25 Top: ϕ_S for Run 1. Bottom: ϕ_S for Run 2. Note different vertical scale.

4.3.3 Pattern

Two h

first hypothes

reducing the c

takes longer f

near the sourc

The ch

first hypothes

σ_S was deter

point A at tim

of the pattern

values vary b

lesser degree

the maximun

Run 1.

The 2

while the 24

occur, the y-

in the center.

suggest that

4.3.3 Pattern dynamics

Two hypotheses could explain the flattening of the pattern seen in Figure 13. The first hypothesis is that the cell receptors become saturated for the chemoattractant S, reducing the cell's chemotactic response to gradients in S. The second hypothesis is that it takes longer for the cells to consume the higher concentrations of chemoattractant found near the source.

The chemotactic response factor values (Figure 25) provide strong support for the first hypothesis. In Run 1, ϕ_S values lie between 0.02 and 0.18 at all time points for which ϕ_S was determined. The values are the lowest in the middle of the pattern (indicated by point A at time 17 hr), where the flattening is the most evident, and highest near the edges of the pattern (indicated by the two points labeled B at time 17 hr). In Run 2, the ϕ_S values vary between 0.25 and 0.9, indicating that chemotaxis is suppressed to a much lesser degree by receptor saturation than in Run 1. This result would be expected since the maximum chemoattractant concentration in Run 2 is approximately one tenth that in Run 1.

The 23 hr ϕ_S curve for for Run 1 is flat from approximately 1.7 cm to 3.2 cm, while the 24 hr ϕ_S curve for Run 2 has a much rounder appearance. For broadening to occur, the y-component of the wave's velocity at the sides of the pattern must exceed that in the center. The shapes of the ϕ_S curves are consistent with such a velocity gradient, and suggest that receptor saturation is at least partially responsible for the pattern broadening.

4.3.4 Chemo

Mover

source reserv

was not repr

that this beha

second chemo

second chem

coli (Adler,

significant o

away from th

moving away

gradient.

4.4 Variat

The

experiments

compounds,

however, an

complex mo

First

competition

modificatio

4.3.4 Chemotactic wave in response to oxygen

Movement of a portion of the chemotactic wave away from the chemoattractant source reservoir (i.e. toward the north) was an unexpected experimental result. This effect was not reproduced in the simulations with a single chemoattractant. We hypothesized that this behavior could arise in response to gradients formed by the consumption of a second chemoattractant that is uniformly distributed throughout the chamber initially. The second chemoattractant has been identified as oxygen, a known chemoattractant for *E. coli* (Adler, 1972). The microelectrode measurements shown in Figure 18 indicate that significant oxygen gradients coincide with the cell wave fronts moving both toward and away from the applied chemoattractant gradient. Model predictions indicate that the wave moving away from the aspartate source tracks an oxygen gradient rather than an aspartate gradient.

4.4 Variations on original model

The original model, presented in Section 4.2, gave good agreement with experiments for one bacterial population responding to gradients of two chemical compounds, and growing on a third. The resulting simulations were not perfect matches, however, and experimental data, such as the oxygen gradients, suggested that a more complex model was needed to match the experimental results more closely.

First, a second bacterial balance was added to the model. This balance allowed competition or other multi-population phenomena to be studied. More details on this modification to the model are given in Sections 5 and 6.

In addi

second-genera

and the nutrie

speaking, this

Q were oxyge

oxygen. If

improvement

and Ollis, 198

takes into ac

period of l

chemoattract

growth. The

motility, for

An i

cells to gro

compound w

coli, or aceta

The

$$\frac{\partial u_i}{\partial t} = \mu_i \nabla^2$$

where v_{ei} i

population.

In addition to the second cell balance, several improvements were built into this second-generation model. These included making consumption of the chemoattractant S and the nutrient H dependent on the concentration of chemoattractant, Q. Biologically speaking, this can be interpreted as modeling Q as the electron acceptor in the system. If Q were oxygen, then the cell's metabolism is aerobic, with electrons being donated to oxygen. If Q were nitrate, anaerobic denitrification could be modeled. Another improvement was the addition of death terms and endogenous metabolism terms (Bailey and Ollis, 1986). The death terms model the death of cells, while endogenous metabolism takes into account the loss of cell mass due to internal consumption, such as during a period of low nutrient availability. Cell maintenance terms were added to the chemoattractant and nutrient balances to allow for consumption for uses other than growth. The cell maintenance terms could include nutrient consumed for energy for motility, for example.

An important modification to the second-generation model was the ability of the cells to grow on the chemoattractant S. This gives the model the ability to model a compound which is both a chemoattractant and a growth nutrient, such as glucose for *E. coli*, or acetate for *Pseudomonas* KC (see Section 7).

The second-generation model for cell population i (i=a or b) is given by

$$\frac{\partial u_i}{\partial t} = \mu_i \nabla^2 \; u_i - \nabla \bullet (V_c u)_{is} - \nabla \bullet (V_c u)_{iq} + \frac{Q}{C_{iq} + Q}\left[\frac{v_{ih} H}{C_{ih} + H} + \frac{v_{is} S}{C_{is} + S} - v_{ei}\right] u_i - d_{thi} u_i \qquad (28)$$

where v_{ei} is the endogenous metabolism term and d_{thi} is the death term for the i^{th} population. The balance for the consumable compound j (j=S or H) is now given by

where v_{ijm} is

term is given

where Y_{ij} is t

term. for the

$$\frac{\partial Q}{\partial t} =$$

where Y_{iq} is

endogenous

Two

shown in F

metabolism

to obtain acc

results that r

$$\frac{\partial j}{\partial t} = D_j \nabla^2 \quad j - \sum_{i=a}^{b} \left(v_{ijm} \frac{j}{C_{ij} + j} \frac{Q}{C_{iq} + Q} u_i \right) \tag{29}$$

where v_{ijm} is the total consumption coefficient for the i^{th} population growing on j. This term is given by

$$v_{ijm} = \frac{v_{ij}}{Y_{ij}} + cm_{ij} \tag{30}$$

where Y_{ij} is the yield coefficient for cells growing on j and cm_{ij} is the cell maintenance term, for the consumption of j. The balance for the compound Q is given by

$$\frac{\partial Q}{\partial t} = D_q \nabla^2 Q - \sum_{i=a}^{b} \left(\left(\frac{v_{ihm}}{Y_{ihq}} \frac{H}{C_{ih} + H} + \frac{v_{ism}}{Y_{isq}} \frac{S}{C_{is} + S} + \frac{v_{ei}}{Y_{iq}} \right) \frac{Q}{C_{iq} + Q} u_i \right) \tag{31}$$

where Y_{iq} is the yield coefficient for the consumption of Q by the i^{th} population used for endogenous metabolism.

Two time-points from a sample simulation using the second-generation model are shown in Figure 26. Values for unknown parameters, such as the endogeneous metabolism terms and the consumption terms were estimated. More work will be needed to obtain accurate values for these parameters, but the model has been shown to produce results that resemble the experimental data.

4.5 Laser

The l

this research

student, Seb

will be pres

1997.

Moti

coefficient

moving thro

Therefore, a

allows the r

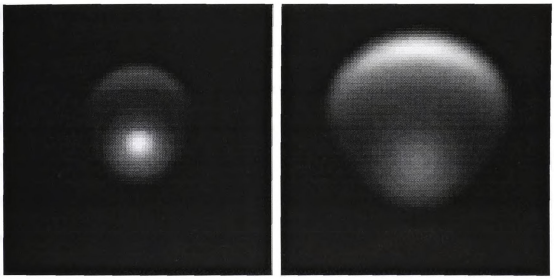


Figure 26. Sample simulations from second-generation model.

4.5 Laser diffraction capillary assay

The laser diffraction capillary assay was conceived and developed as a portion of this research project, but the majority of the work was done by an Aachen exchange student, Sebastian Schmidt. For this reason, only a brief summary of the LDCA method will be presented within the body of this work. For further details, see Schmidt *et al.*, 1997.

Motility parameters necessary for the modeling work, such as the random motility coefficient and the chemotactic sensitivity coefficient, were not known for bacteria moving through a semi-solid medium, such as the dilute agar gel in the arena of the DGC. Therefore, a method to independently measure these parameters was needed. The LDCA allows the random motility coefficient to be measured at various agar gel concentrations,

and prelimin

coefficients t

The L

mounted on a

was achieved

suspensions i

of 30 mm, th

a He/Ne lase

experiment.

and preliminary experiments indicate that it will also allow chemotactic sensitivity coefficients to be measured as a function of agar concentration.

The LDCA consists of a 1.6 mm inside diameter glass capillary, which can be mounted on a microscope stage. A step change in cell concentration inside the capillary was achieved by successively inserting the tip of the capillary into each of the two agar suspensions in the culture tubes (Figure 27). The capillaries were filled over a total length of 30 mm, then mounted on the microscope stage. The end of the capillary opposite from a He/Ne laser was sealed with silicone grease to avoid unwanted convection during the experiment.

The laser wa

was mounte

15 minutes,

maintained

The

Lauffenburg

dimensional

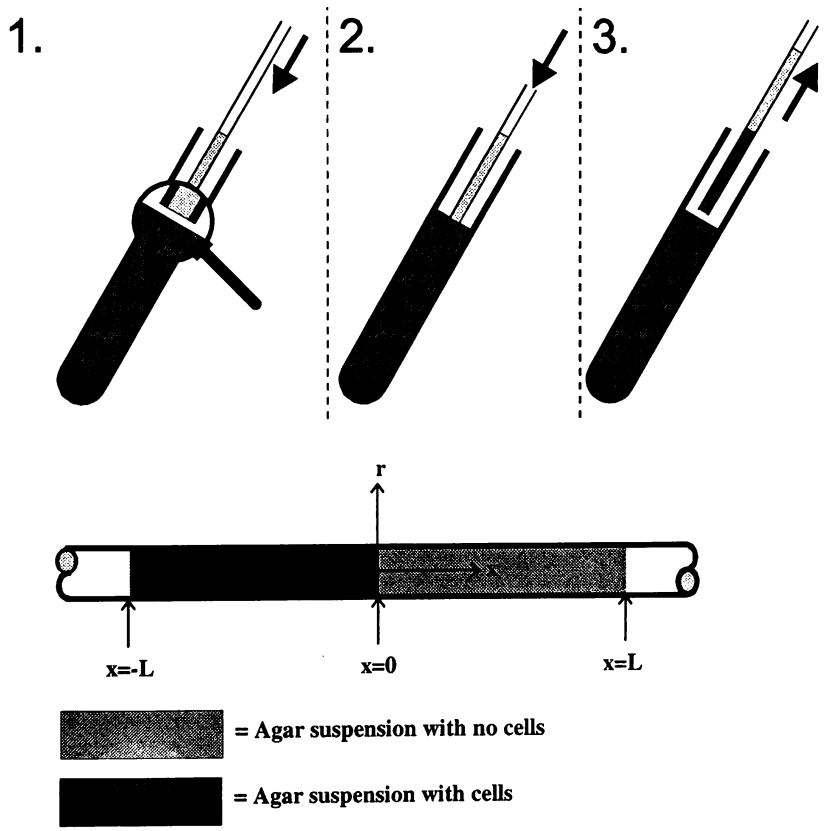


Figure 27. Filling the capillary in the LDCA

The laser was shone axially through the capillary. A CCD camera, attached to a computer, was mounted on the optical lens of the microscope. Images were captured at 1.5 minutes, 15 minutes, and then at 30 minute intervals. A typical run lasted 3 hours. The LDCA was maintained at room temperature for all runs.

The mathematical model for the LDCA is similar to that used by Ford and Lauffenburger (1992) for the SFDC, with the chemotaxis terms omitted. The one-dimensional cell balance is given by

The analytica

using the bou

The concent

This slope c

The model i

motility coe

a function c

points. The

confidence

$$\frac{\partial B}{\partial t} = \mu \frac{\partial^2 B}{\partial x^2} \tag{32}$$

The analytical solution to this model is

$$B = \frac{1}{2} erfc\left[\frac{x}{\sqrt{4\mu t}}\right] \tag{33}$$

using the boundary conditions

$$B(x,0) = 1 \text{ for } -L < x < 0 \tag{34}$$

$$B(x,0) = 0 \text{ for } 0 < x < L \tag{35}$$

$$\frac{\partial B}{\partial x}(-L,t) = \frac{\partial B}{\partial x}(L,t) = 0 \tag{36}$$

The concentration gradient at x=0 is given by differentiating Equation (33) to yield

$$\frac{\partial B}{\partial x} = \frac{-0.5}{\sqrt{\pi \mu t}} \tag{37}$$

This slope can be experimentally measured at each time-point from the captured images. The model is fit to the slopes by adjusting the random motility coefficient, μ. The random motility coefficient for the bacteria *Pseudomonas stutzeri* KC, measured by the LDCA as a function of agar concentration, is shown in Figure 28. The open circles show the data points. The solid line is the best fit of the model, and the dashed lines represent the 95% confidence interval for the fit. The random motility coefficient for a zero agar

concentration

order of mag

cm^2/s). The i

displayed a

concentratio

concentration, estimated by extrapolation, is 2.0×10^{-6} cm^2/s. This value is of the same order of magnitude as that reported by Segel *et al.* (1977) for *P. fluorenscens* (6×10^{-6} cm^2/s). The important finding of the LDCA work was that the random motility coefficient displayed a linear relationship with agar concentration, at least in the region of agar concentration tested.

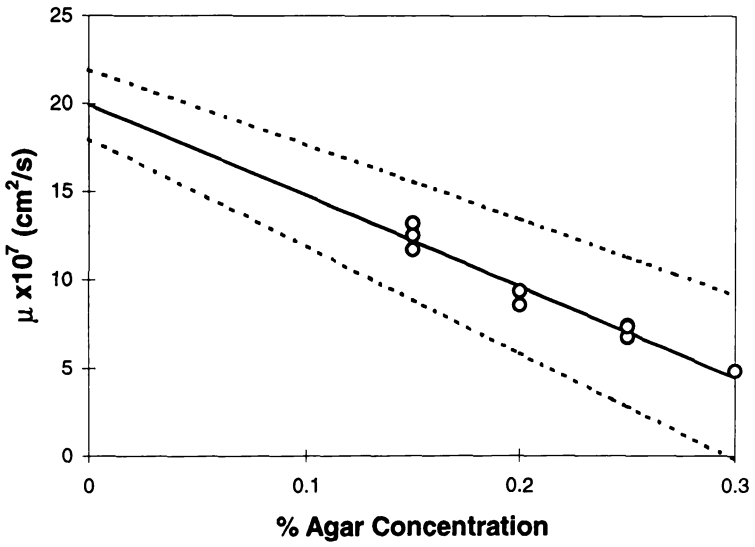


Figure 28. The random motility coefficient of *Pseudomas* KC.

5. MICRO

5.1 Compe

Chem

to gradients

(1976) stated

Since most e

sole criterion

The e

utilizing mat

microbial po

confined dor

motility cou

al. (1982) w

of chemotax

growth rates

bacterial po

(Lauffenbur

populations

sufficiently

had a highe

geometry

5. MICROBIAL COMPETITION

5.1 Competition background

Chemotactic microorganisms able to position themselves optimally with respect to gradients may have a competitive advantage over other organisms. Chet and Mitchell (1976) stated that chemotaxis "presumably gives...microorganisms a selective advantage. Since most ecosystems are not fully mixed, enzyme kinetics cannot be postulated as the sole criterion for the competitive advantage of one microorganism over another."

The effects of chemotaxis on competition have been studied on a theoretical basis, utilizing mathematical models to predict possible outcomes of competition between two microbial populations. Lauffenburger *et al.* (1981) showed that in a one-dimensional, confined domain, with a substrate diffusing in from one side, increasing levels of random motility could be detrimental to the survival of a bacterial population. Lauffenburger *et al.* (1982) went on to show that if the substrate was also a chemoattractant, a certain level of chemotaxis could impart an advantage to a population, canceling the effects of lower growth rates or higher random motilities. Both of these studies involved only a single bacterial population. Lauffenburger and Calcagno (1983), building on the previous work (Lauffenburger *et al.*, 1981; Lauffenburger *et al.*, 1982) modeled two randomly motile populations simultaneously. They found that if a slower growing population also had sufficiently lower random motility, it could out-compete a faster growing population that had a higher random motility. Kelly *et al.* (1988), using a model system with the same geometry as the Lauffenburger papers, modeled two chemotactic populations

simultaneousl

above which

immotile pop

The r

steady-state s

first populati

would surviv

authors poin

order of a ye

occur over t

motility and

environmen

Rela

from an exp

strain of *Pr*

culture, but

(1985) stud

strain of th

chemotacti

similar gro

Oth

organisms

simultaneously. Their model predicted that there was a minimum level of chemotaxis above which a population would have a competitive advantage over a non-chemotactic, immotile population.

The model of Kelly *et al.*, which included a death term and therefore allowed for steady-state solutions, predicted that at steady-state, three conditions could arise: (1) the first population would survive and the second would disappear, (2) the second population would survive, while the first disappeared, or (3) both populations would coexist. The authors pointed out that the time to attain steady-state predicted by their model is on the order of a year, and in real systems, it is likely that variations in chemical gradients would occur over that time period. They concluded that the ability to "consider the effects of motility and chemotaxis on population growth and competition in rapidly changing environments is clear".

Relatively few studies have addressed the effects of chemotaxis on competition from an experimental viewpoint. Pilgram and Williams (1976) found that a chemotactic strain of *Proteus mirabilis* outgrew a non-chemotactic but still motile strain in stationary culture, but that the two strains grew equally well in mixed culture. Kennedy and Lawless (1985) studied a motile, chemotactic strain of *Pseudomonas fluorescens* and a immotile strain of the same species, and found that in unmixed aerobic and anaerobic soils, the chemotactic strain survived significantly better than the non-chemotactic strain. Both had similar growth characteristics in mixed culture.

Other work has focused on the interaction of two or more populations of organisms growing in the presence of spatially varying environments. Caldwell and

Hirsch (1972
gradients in t
layer, and gr
study did no
could have
the local con

Tilm
structured ha
dispersal rat
movement c
studied how
competition

In th
spatially str
called the d
the format
Microorgan
movement
mathematic
balance, an
Widman *et*
(1) the sub

Hirsch (1972) studied the growth of microorganisms in two-dimensional concentration gradients in their steady-state diffusion plate. The organisms were immobilized in an agar layer, and growth was quantified as a function of position in the gradients. Although this study did not address competition directly, the researchers did find that the organisms could have vastly different growth characteristics, depending upon their adaptability to the local concentration conditions.

Tilman (1994), taking a broad ecological perspective of competition in spatially structured habitats, noted that coexistence of species could occur if a species with a high dispersal rate could move into a zone not inhabited by a superior competitor with less movement capability. Holmes *et al.* (1994), again from an ecologically based viewpoint, studied how partial differential equations could be used to study, among other things, competition, dispersal, and dispersal-mediated coexistence in spatially structured systems.

In this study, a system of partial differential equations was used to describe a spatially structured system in which two bacterial populations were growing. The system, called the diffusion gradient chamber (DGC), has been established as a tool that allows the formation of two-dimensional chemical gradients (Emerson *et al.*, 1994). Microorganisms inoculated into the chamber can be observed, and their growth and movement properties recorded. Widman *et al.*, (1997) (see Section 4.2), developed a mathematical model of the DGC that included two chemoattractant balances, a nutrient balance, and a cell balance. In addition to modeling a different physical geometry, the Widman *et al.* model offered three features not found in the Kelly *et al.*, (1988) model: (1) the substrate does not necessarily have to be a chemoattractant, and vice versa; (2) the

model equati

transient mo

analyzed wit

extended to

competitions

5.2 Comp

The

system, intr

balance. Th

$$\frac{\partial u_i}{\partial t} = \mu_i \nabla^2$$

where, for i

is the nutri

coefficient

the recepto

K_{DiQ} is the

growth rat

Th

the second

model equations are solved in two dimensions, as opposed to one dimension; and (3) the transient model can be solved over relatively short time periods, and the data can be analyzed without requiring a steady-state solution. In this study, the model of the DGC is extended to include a second cell balance, which allows predictions of the outcomes of competitions between two microbial populations.

5.2 Competition mathematical model

The system of coupled partial differential equations that describes the DGC system, introduced in Widman *et al.* (1997), was extended to include a second cell balance. The cell balances for Populations A and B are given by

$$\frac{\partial u_i}{\partial t} = \mu_i \nabla^2 u_i - \chi_{0iS} \nabla \cdot \left[\left(\frac{K_{DiS}}{\left(K_{DiS} + S\right)^2} \right) u_i \nabla S \right] - \chi_{0iQ} \nabla \cdot \left[\left(\frac{K_{DiQ}}{\left(K_{DiQ} + Q\right)^2} \right) u_i \nabla Q \right] + \frac{\nu_{iH} H}{C_{iH} + H} u_i \tag{38}$$

where, for i=A or B, u_i is the cell concentration, S and Q are the two chemoattractants, H is the nutrient, μ_i is the random motility coefficient, χ_{0iS} is the chemotactic sensitivity coefficient to the attractant S for the i^{th} population, K_{DiS} is the dissociation constant for the receptor-S complex, χ_{0iQ} is the chemotactic sensitivity coefficient to the attractant Q, K_{DiQ} is the dissociation constant for the receptor-S complex, ν_{iH} is the maximum specific growth rate, and C_{iH} is the half-saturation constant for growth on H.

The chemoattractant and nutrient balances were changed to reflect the addition of the second cell population. The chemoattractant balances are given by

where, for j

chemoattract

chemoattract

chemoattract

As d

more of the

chemoattrac

the reservoi

which is a c

The

where D_H i

growth of th

The

justified in

balance ar

nutrient are

cell popula

$$\frac{\partial j}{\partial t} = D_j \nabla^2 j - \sum_{i=A}^{B} \frac{v_{ij} j}{C_{ij} + j} u_i \tag{39}$$

where, for j=S or Q, D_j is the chemoattractant diffusion coefficient, v_{ij} is the specific chemoattractant consumption coefficient for the i^{th} population consuming the j^{th} chemoattractant, and C_{ij} is the saturation constants for i^{th} population consuming the j^{th} chemoattractant.

As described in Section 4.2.3, the chemoattractant S is introduced from one or more of the DGC reservoirs, and forms diffusion gradient across the gel in the arena. The chemoattractant Q is initially present at a constant concentration throughout the gel and the reservoirs. In the experiments used to validate the model, Q was most likely oxygen, which is a chemoattractant for many bacteria.

The nutrient balance becomes

$$\frac{\partial H}{\partial t} = D_H \nabla^2 H - \sum_{i=A}^{B} \frac{v_{iH} H}{C_{iH} + H} \frac{u_i}{Y_{iH}} \tag{40}$$

where D_H is the nutrient diffusion coefficient, and Y_{iH} is the yield coefficient for the growth of the i^{th} population on H.

The assumptions made in the original, single population model are given and justified in Section 4.2.9. The new assumptions introduced by including the second cell balance are that the effects of the two populations on the chemoattractants and the nutrient are additive (Lauffenburger and Calcagno, 1983; Kelly *et al.*, 1988); that neither cell population preferentially consumes one chemoattractant or nutrient over the other;

and that the

predation or

5.3 Model

Many

parameters v

correlations.

model was v

In th

and that there are no direct interaction effects between the two populations, such as predation or parasitism.

5.3 Modeling parameters

Many of the parameters used in the model are given in Section 4.2.5. The parameters were measured independently, obtained from the literature, or calculated from correlations. All the parameters had reasonable values for a population of *E. coli*, and the model was verified by comparing its simulations to experiments with *E. coli*.

In this study, the base set of parameters for each population is given in Table 4.

Fro

their effec

where the

parameter

a dimensi

Table 4. Base case parameters for competition simulations.

Parameter (i = a or b)	**Value**
μ_i	0.0010 $cm^2 h^{-1}$
χ_{0AQ}	0.085 $cm^2 h^{-1}$
χ_{0BQ}	0
χ_{0AS}	0.020 $cm^2 h^{-1}$
χ_{0BS}	0
C_{iH}	$4.08x10^{-6}$ $g_H cm^{-3}$
C_{iQ}	$6.70x10^{-8}$ $g_Q cm^{-3}$
C_{iS}	$5.50x10^{-5}$ $g_S cm^{-3}$
D_H	0.01 $cm^2 h^{-1}$
D_Q	0.033 $cm^2 h^{-1}$
D_S	0.033 $cm^2 h^{-1}$
K_{DiQ}	$3.30x10^{-5}$ $g_Q cm^{-3}$
K_{DiS}	$2.00x10^{-6}$ $g_S cm^{-3}$
v_{AH}	0.35 h^{-1}
v_{BH}	0.5 h^{-1}
v_{AQ}	0.02 $g_Q g_u^{-1} h^{-1}$
v_{BQ}	0
v_{AS}	0.60 $g_S g_u^{-1} h^{-1}$
v_{BS}	0
Y_{iH}	0.50 g_u/g_H

From this base set, individual parameters for Population A were varied to test their effects on the outcomes of competition simulations. In many mathematical models, where the effects of parameters are unknown, a dimensionless model enables many parameter effects to be studied at one time. In the case of this particular model, however, a dimensionless model did not decrease the number of parameters, and was therefore not

useful for pa

the attractan

the specific

chosen as t

inoculation

simulations,

while that f

rate for Pop

(Hansen an

including c

that it is on

In

inoculation

The base c

chamber, a

5.4 Anal

5.4.1 The

Ty

populatio

populatio

result is

useful for parametric studies (see Appendix G). The chemotactic sensitivity coefficient to the attractant S (χ_{0AS}), the dissociation constant for the receptor-S complex (K_{DAS}), and the specific chemoattractant consumption coefficient for the attractant S (ν_{AS}) were chosen as the variables to be manipulated. The initial amount of H present and the inoculation pattern of the two populations were also varied in some simulations. For all simulations, the maximum specific growth rate of Population B (μ_B) was set to 0.5 h^{-1}, while that for Population A was left at the base case value of 0.35 h^{-1}. The larger growth rate for Population B would give it a competitive advantage in a well-mixed environment (Hansen and Hubbell, 1980). Population B does not interact with either chemoattractant, including chemotaxis or consumption. In other words, the base case for Population B is that it is only able to move by random motility, and it consumes only H.

In all simulations, the gradient was allowed to initialize for 6 hours before inoculation. After inoculation, the duration of the simulated experiment was 30 hours. The base case initial condition for the cells is that they are inoculated in the center of the chamber, at an equal concentration, as described in Widman *et al.* (1997).

5.4 Analysis of competition simulations

5.4.1 The dynamic competition factor

Typically, the result of a competition experiment between two microbial populations in well-mixed culture is quantified by the death or disappearance of one population and the continued survival of the other. This type of steady-state competition result is not readily amenable to the transient DGC model. Therefore, a dynamic

competition

masses of th

where V is

the model.

dimensiona

depth of the

Wh

value of ψ

a value of

competition factor (ψ), was defined to give a time-variant measure of the ratio of the masses of the populations; ψ is defined by

$$\psi(t) = \frac{\iiint_V u_A(x, y, z, t)dV}{\iiint_V u_B(x, y, z, t)dV} \tag{41}$$

where V is the total volume of the gel, and $u_a(x,y,z,t)$ and $u_b(x,y,z,t)$ are calculated from the model. The coordinate axes are defined in Figure 29. It is assumed that the two-dimensional concentration profiles predicted by the model are constant throughout the depth of the gel (the z-direction).

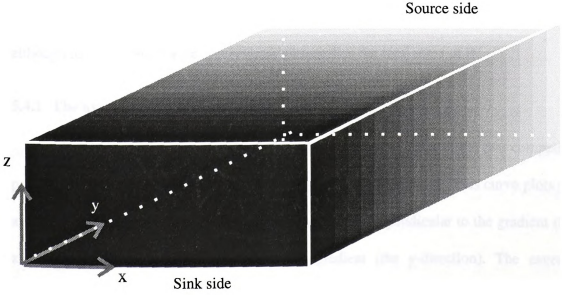


Figure 29. Coordinate axes used for competition analysis.

When $\psi = 1$, both populations have the same total mass present in the chamber. A value of $\psi < 1$, indicates that Population B has a higher total mass than Population A, and a value of $\psi > 1$ indicates that Population B has a lower total mass than Population A.

Therefore, f

instantaneou

time, and vic

Anot

derived fron

masses, and

If $\psi' > 0$, th

B. If $\psi <$ 1

although its

5.4.2 The

An

population

average co

x-direction

concentrat

$\bar{u}_i$

Therefore, for the purposes of this study, Population A will be said to have an instantaneous competitive advantage over Population B at a given time if $\psi > 1$ at that time, and vice versa.

Another measure of the competitive relationship of the two populations can be derived from ψ. The local competition rate, ψ', is the rate of change of the ratio of the masses, and can be found by

$$\psi' = \frac{d\psi}{dt} \tag{42}$$

If $\psi' > 0$, then the mass of Population A is increasing relative to the mass of Population B. If $\psi < 1$, but $\psi' > 1$, Population A may still have the overall competitive advantage, although its total mass has not yet become greater than the total mass of B.

5.4.2 The average concentration curve

Another useful approach to studying the spatial arrangement of the two competing populations was the development of the average concentration curve. This curve plots the average concentration of the population along each line perpendicular to the gradient (the x-direction) vs. the position parallel to the gradient (the y-direction). The average concentration is calculated (for Population A, for example) by

$$\bar{u}_A(y) = \frac{\int_{x=0}^{x=L} u_A(x,y)\big|_y \Delta y \Delta z dx}{\Delta x \Delta y \Delta z} \approx \frac{\sum_{i=1}^{n} u_A(i,y)\big|_y \Delta y \Delta z \Delta x_i}{n \Delta x \Delta y \Delta z} = \frac{\sum_{i=1}^{n} u_A(i,y)\big|_y}{n} \tag{43}$$

where L is th

and Δy are

the gel. Figu

point on the

image.

15

Figure

It

dynamic c

for any

concentra

the simul

where L is the width of the chamber, n is the number of grid points in the x-direction, Δx and Δy are determined by the grid spacing and are always equal, and Δz is the depth of the gel. Figure 30 shows a typical competition simulation, and one line used to calculate a point on the total mass curve. The chemoattractant S is diffusing in from the top of the image.

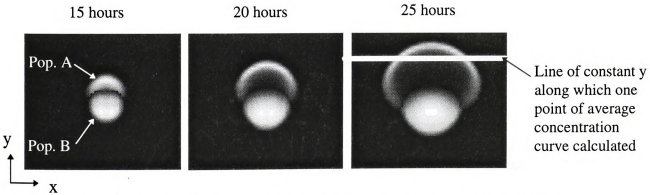


Figure 30. Typical simulation, illustrating average concentration curve calculation.

It should be noted that the average concentration curve can be related to the dynamic competition factor as

$$\psi(t) = \frac{\int \bar{u}_A(y,t)dy}{\int \bar{u}_B(y,t)dy} \tag{44}$$

for any time point. By studying the dynamic competition factor and the average concentration curve, some measure of the spatially developing competitive outcomes of the simulations can be found.

5.4.3 Satur

The

utilized to e

quantity use

for a given S

the equality

where F_{max}

of constant

5.5 Com

Ma

was also

curve, and

The Matl

approxima

5.5.1 Eff

In

in Table

5.4.3 Saturation of chemoreceptors

The global chemotactic response factor, ϕ_S, defined in Section 4.3.2, can be utilized to explain some of the responses that occur in the following simulations. Another quantity used to analyze some of the simulation results was the maximum flux attainable for a given S gradient. This term is the integrand of the denominator of the second term of the equality in Equation (27), and can be defined as

$$F_{\max}(x) = \frac{\int \frac{\chi_{0AS}}{K_{DAS}} \frac{\partial S}{\partial y} dy \Big|_x}{\int_0^L dy} \tag{45}$$

where $F_{max}(x)$ is the average maximum flux attainable for a given S gradient, along a line of constant x.

5.5 Competition simulations

Matlab v. 4.2c1 was used to visualize the output of the FORTRAN model. Matlab was also used to solve for the dynamic competition factor, the average concentration curve, and the global chemotactic response factor (see Appendix F for the Matlab code). The Matlab function "trapz.m", which utilizes the trapezoidal method, was used to approximate the solution to the integrals.

5.5.1 Effect of modeling parameters on competition results

In the following sets of simulations, the parameters for Population B are as shown in Table 4. The indicated parameter for Population A is varied in each set of simulations.

Range findi

showed a g

range to pr

concentrati

hours after

prior to cel

number is l

al., 1997),

simulation

5.5.2 Var

The

0.01-0.07

points in

correspon

gray. Pop

chemoattr

the inocul

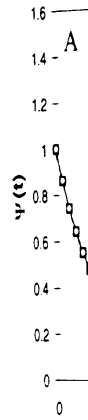
dynamic

trend.

Range finding runs (results not shown) were performed to find a parameter range that showed a good variation in results. The parameter of interest was then varied within this range to produce the desired simulation sets. Both populations were inoculated in equal concentrations in the center of the chamber. The simulated experimental time was 30 hours after inoculation of the cells. The chemoattractant gradient was initiated for 6 hours prior to cell inoculation. The grid-spacing used for all simulations was 41^2 points. This number is less than the 51^2 points used in the study that validated the model (Widman *et al.*, 1997), but was found to be sufficiently accurate and allowed for the fastest possible simulation times.

5.5.2 Variation of χ_{0aS}

The chemotactic sensitivity factor for Population A, χ_{0AS}, was varied between 0.01-0.07 cm^2/h to assess its affect on the outcome of simulated competitions. Three time points in this simulation for χ_{0AS}=0.07 are shown in Figure 30. The lighter areas correspond to higher cell densities. Both Populations A and B are shown in shades of gray. Population A is the one which moves out in a chemotactic wave toward the chemoattractant source, while Population B remains in the center of the chamber, around the inoculation point. Figure 31A shows the dynamic competition factor. In many of the dynamic competition graphs shown, the initial 10 hour period shows a similar decreasing trend.



Thi

center of t

Figure 31E

graph and

the positic

chamber t

reached a

both Popu

concentra

so they w

hour time

wave. Th

the chemo

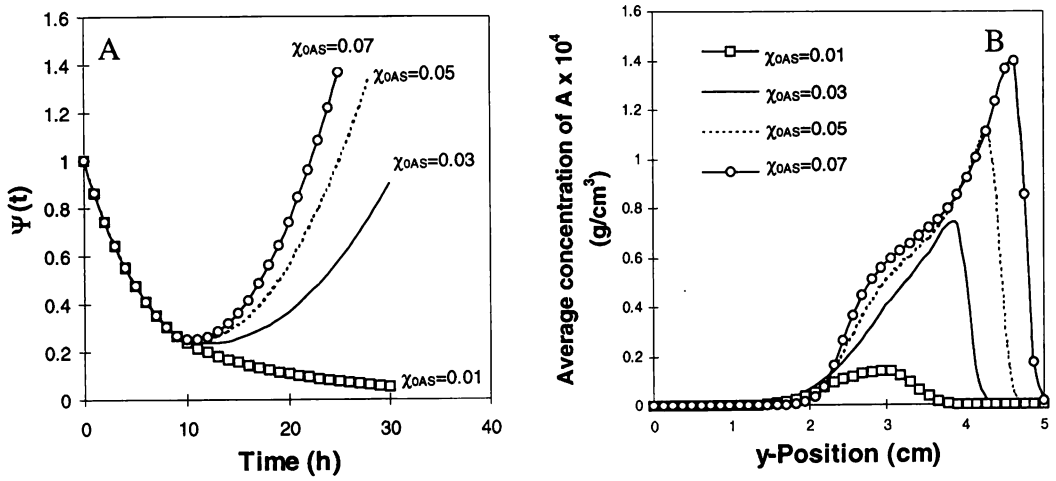


Figure 31 Variation of χ_{0AS}.

This trend occurs because significant concentrations of S do not diffuse into the center of the chamber until approximately 10 hours after inoculation (results not shown). Figure 31B shows the average concentration curves for four values of χ_{0AS}, at 25 h. In this graph and all subsequent average concentration curve figures, the attractant source is at the position y=5 cm. For higher levels of χ_{0AS}, Population A encountered the wall of the chamber before 30 hours. In the graphs of Figure 31, only times before the wall was reached are shown in the graphs. Figure 32A shows the average concentration curve for both Populations A and B, for the case where χ_{0AS}=0.03 cm^2/h, at 25 hours. The average concentration curves for Population B in most of the other simulations were very similar, so they will not be shown in future graphs. Figure 32B shows the density maps of the 25 hour time-point of Populations A and B for χ_{0AS}=0.03 cm^2/h, illustrating the chemotactic wave. The wave is the high concentration band of cells of Population A moving toward the chemoattractant source.

Note that

the highes

This does

that it is a

Th

the cell p

Figure 31

growth a

Populatio

(χ_{OAS}≥0.(

highest l

Populati

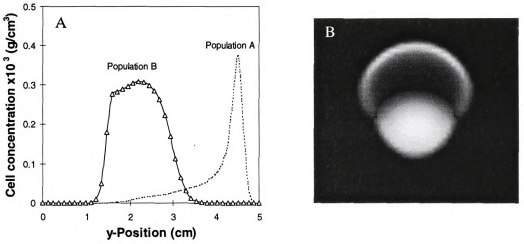


Figure 32. Other results of variation of χ_{0AS}

Note that in this figure, and all figures of the density map type, the cell population with the highest concentration at any point in the chamber is the one that appears in the graph. This does not necessarily mean that the other population does not exist at this point, only that it is at a lower concentration.

The chemotactic sensitivity coefficient is a measure of the magnitude with which the cell population migrates chemotactically in response to a chemoattractant gradient. In Figure 31A it is apparent that below a certain χ_0 value (in this case, χ_{0AS}=0.01 cm^2/h), the growth advantage of Population B is so large that, in the time-frame of the experiment, Population A is overgrown by B, as indicated by $\psi' < 0$ at all times. At higher levels ($\chi_{0AS} \geq 0.03$ cm^2/h), the sign of ψ' becomes positive after about 10 hours, and at the highest levels of χ_{0AS}, ψ reaches a value greater than one, indicating that the total mass of Population A has become greater than the total mass of Population B. For this set of

conditions,

total mass o

The

the chemot

average co

side of the

Population

peak of Po

two popula

of higher

with B.

Th

sensitivity

This indic

carrying c

increasing

cells mov

cells may

and there

carrying

advantage

and the n

conditions, chemotaxis can be said to impart a competitive advantage, as measured by the total mass of microorganisms present, to Population A.

The average concentration curves of Figure 31B give another view of the effect of the chemotactic sensitivity. As χ_{0AS} increases, two things occur. First, the peak of the average concentration curve shifts toward the attractant source, which is located at the side of the chamber corresponding to y=5 cm. The average concentration curve for Population B remains close to the center of the chamber, as illustrated in Figure 32A. The peak of Population A moving further from the center indicates greater separation of the two populations. By moving away from Population B, Population A is able to enter areas of higher nutrient (H) concentration, that can be consumed without having to compete with B.

The second trend that can be observed in Figure 31B is that as the chemotactic sensitivity increases, the maximum height of the average concentration curve increases. This indicates that not only is the total mass of Population A increasing, but also that the carrying capacity of the chemotactic wave (Widman *et al.*, 1997) of Population A is increasing. The chemotactic wave is better illustrated in Figure 32B. The bright band of cells moving toward the chemoattractant source is the chemotactic wave. This wave of cells may allow for faster consumption of the chemoattractant, creating sharper gradients, and therefore allowing for faster chemotaxis toward the source. By increasing the carrying capacity of the wave, the cells may be gaining an even larger competitive advantage by being able to consume larger concentrations of both the chemoattractant S and the nutrient H as the wave travels through the region.

5.5.3 Varia

The

coefficient,

dynamic co

cm) profile

33C shows

33D show

5.5.3 Variation of v_{as}

The second parameter varied was the specific chemoattractant consumption coefficient, v_{AS}. It was varied between 0 and 1.0 $g_S g_{uA}^{-1} h^{-1}$. Figure 33A shows the dynamic competition factor for five values of v_{AS}. Figure 33B shows a centerline (x=2.5 cm) profile of the chemoattractant S gradient at 30 hours for each value of v_{AS}. Figure 33C shows a density map profile for both populations with v_{AS}=1.0 $g_S g_{uA}^{-1} h^{-1}$ and Figure 33D shows the profiles for v_{AS}=0 $g_S g_{uA}^{-1} h^{-1}$, both at 30 hours.

I

on the d

Figure

howeve

other w

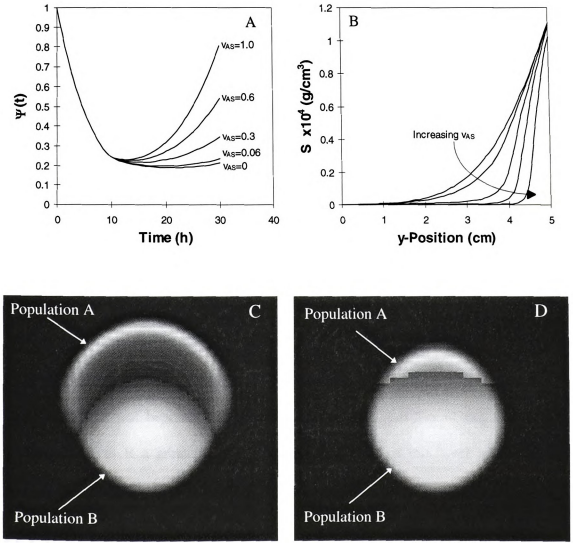


Figure 33. Variation of v_{AS}.

Increasing the specific chemoattractant consumption coefficient, v_{AS}, had an effect on the dynamic competition factor curves similar to that of increasing χ_{0AS}, as shown in Figure 33A. The reasons for the increased competitive advantage are not the same, however. Figure 33B shows the S concentration profiles for the position x=2.5 cm, or, in other words, down the center of the DGC. As v_{AS} is increased, a given mass of cells can

consume

encountere

response,

v_{AS}=0, me

positive n

were grea

This can

of S, the

When v_{A}

subseque

33C, whe

and the

consume

(χ_{0AS}=0.

5.5.4 V

complex

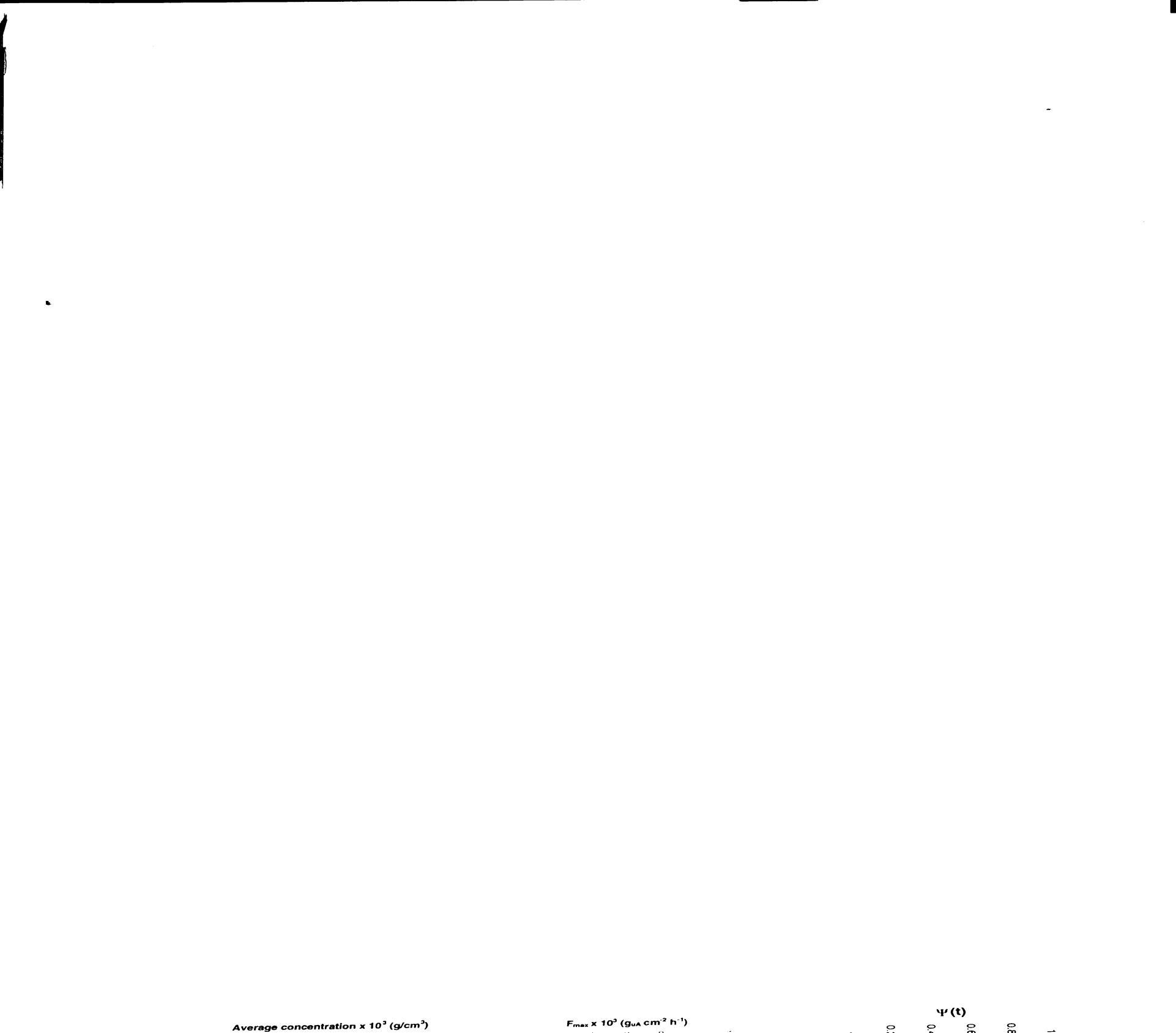
34B sho

shown i

consume a larger quantity of S, and therefore increase the gradient of S that is encountered by the wave of A. This increase in slope enables a faster chemotactic response, and therefore a more rapid separation of Populations A and B. Even with $v_{AS}=0$, meaning no consumption of the chemoattractant S, the value of ψ' does become positive near the end of the simulation (Figure 33A). If the dimensions of the chamber were greater, it is likely that ψ' would become positive, even with no consumption of S. This can again be explained by the carrying capacity of the wave. With no consumption of S, the gradient is relatively shallow, and the carrying capacity of the wave is low. When v_{AS} is high, the gradients are sharper, which increases chemotaxis, and subsequently increases the carrying capacity of the wave. This is illustrated in Figure 33C, where v_{AS} is high and the wave is very apparent, and in Figure 33D, where $v_{AS}=0$, and the wave is just beginning to appear at 30 hours. The ability of Population A to consume S gives it a greater competitive advantage, but at this level of chemotaxis ($\chi_{0AS}=0.02$), the ability to consume S is not necessary for A to compete successfully.

5.5.4 Variation of K_{DAS}

The third parameter varied was the dissociation constant for the receptor-S complex, K_{DAS}. The parameter range was 1×10^{-7} - 8×10^{-6} g/cm^3. Figure 34A and Figure 34B show the $\psi(t)$ values for five values of K_{DAS}. Note that the curve for $K_{DAS}=1\times10^{-6}$ is shown in both figures, for comparison. The arrows show the direction of increasing K_{DAS}.


Average concentration x 10³ (g/cm³)
Fmax x 10³ (guA cm⁻² h⁻¹)
Ψ (t)

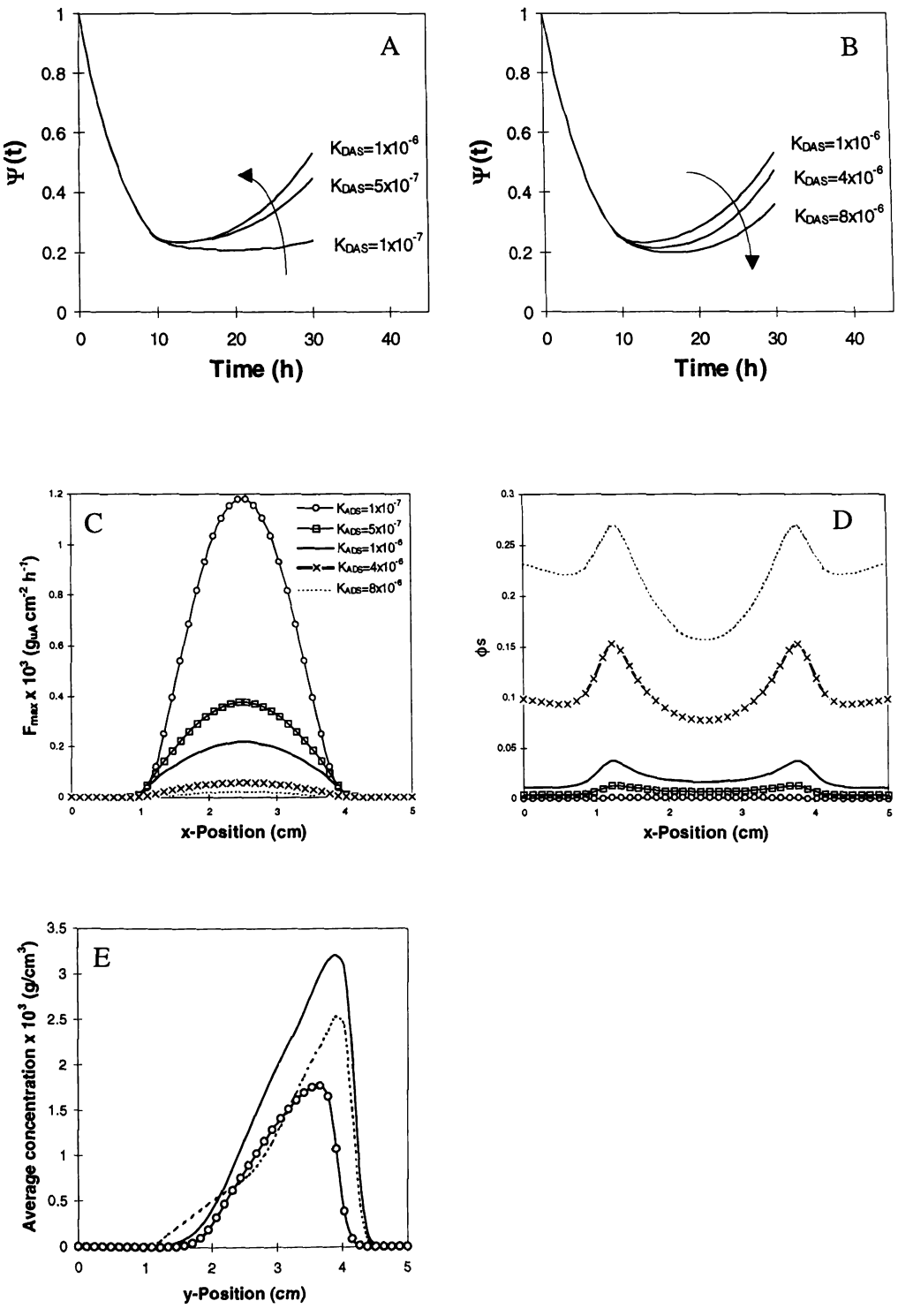


Figure 34. Variation of K_{DAS}. Time=30 h for all figures.

In

competitiv

1×10^{-6}, inc

reasons fo

average m

legend in

the global

maximum

Conseque

when S >

points in

decrease

gradient.

value fo

optimum

gradient

the ave

value o

values

interme

In Figure 34A, for $K_{DAS} < 1\times10^{-6}$, increasing K_{DAS} made Population A more competitive, as indicated by larger ψ values at the later time points. However, above 1×10^{-6}, increasing K_{DAS} caused ψ to decrease at later times, as shown in Figure 34B. The reasons for these trends are shown in Figure 34C and Figure 34D. Figure 34C shows the average maximum attainable fluxes for the corresponding S-gradient, at 30 hours. The legend in Figure 34C applies to Figure 34D and Figure 34E, as well. Figure 34D shows the global chemotactic response factor for the five parameter values, also at 30 hours. The maximum mass flux is inversely proportional to K_{DAS}, as shown in Equation (45). Consequently, at the lowest values of K_{DAS}, F_{max} reaches its highest levels. However, when $S >> K_{DAS}$, (e.g. for the lowest values of K_{DAS}), the value of ϕ_S approaches 0 at all points in the chamber. As K_{DAS} increases, the level of saturation of the receptors decreases, and therefore the population responds with a higher chemotactic flux to the S-gradient. The trade-off between the opposing trends of F_{max} and ϕ_S results in an optimum value for K_{DAS} of about 1×10^{-6} g/cm^3, where flux is high and saturation is low. This optimum K_{DAS} value is most likely dependent on the concentration and slope of the gradient. Evidence for an optimum K_{DAS} value is also given in Figure 34E, which shows the average concentration profiles for Population A for a low, intermediate, and high value of K_{DAS} at 30 hours. Although the peaks of the curves for the intermediate and high values appear at about the same y-position, the carrying capacity of the wave for the intermediate value is clearly greatest.

5.5.5 Var

In

varied. Th

in Table 5

coefficien

values, wi

T

1.32×10

concentr

Figure 3

source

concent

10 and

5.5.5 Variation of chemoattractant concentration

In this group of simulations, the initial conditions for the chemoattractant S were varied. The base case set of initial conditions for all of the chemical compounds is given in Table 5. For all simulations, the east and west reservoirs were sealed (the mass transfer coefficient was set to 0). The parameters for Population A were set to the base case values, with the exception of χ_{0AS}, which had a value of 0.05 cm^2/h.

Table 5. Base case initial conditions for competition simulations.

Compound and location	Value (g/cm^3)
S in arena, south reservoir	0
S in north reservoir	1.32×10^{-4}
Q in arena, N and S reservoirs	1.32×10^{-5}
H in arena, N and S reservoirs	4.6×10^{-4}
Population A, Population B	5×10^{-6}

The concentration of S in the north reservoir was varied from 1.32×10^{-6} to 1.32×10^{-2} g/cm^3. Figure 35A and Figure 35B show the dynamic competition factor for six concentrations of S in the source reservoir. Like the results shown in Figure 34A and B, Figure 35A and B also show a reversal in the ψ trend above a certain chemoattractant source concentration. For values of $S \leq 1.32\times10^{-6}$, ψ' is always negative. As the concentration is increased, the sign of ψ' changes after a certain amount of time (between 10 and 20 hours) in each simulation. Above a certain concentration, however, the ψ trend

begins to

the trade-

Fig

concentra

using sim

time-poin

35C and

is that as

F

concentra

opposing

simulati

conditio

for three

indicate

growth

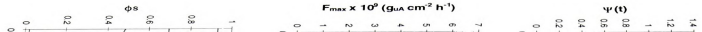
wall of

carrying

begins to decrease with increasing concentration. This effect can again be explained by the trade-off between the maximum flux and saturation of receptors.

Figure 35C and Figure 35D show the maximum obtainable flux for the six concentrations. Note that the y-axis has a different scale in the two figures. To avoid using simulations where the cells had encountered the wall of the chamber, the 28 hour time-point was used for all graphs that show a trend at a single time. The graphs in Figure 35C and Figure 35D have been split because the scales are so different. The overall trend is that as the concentration of S increases, so does the maximum attainable flux.

Figure 35E shows the ϕ_S curves for the four lowest concentrations. At the higher concentrations, the ϕ_S values were extremely close to 0 at every position. The same opposing trends that were observed in the response to K_{DAS} are apparent in this set of simulations. The level of S that minimizes saturation and maximizes flux imparts the best conditions for survival to Population A. In Figure 35F, the average concentration curves for three concentrations are shown. At the lowest concentration, a wave does not form, as indicated by the lack of a local maximum in cell concentration near the source side of the growth region. At the intermediate concentration, a strong wave has traveled nearly to the wall of the chamber. At the highest concentration, a wave is apparent, but has a low carrying capacity and has not moved as far from the inoculation point.


ϕs
0
0.2
0.4
0.6
0.8
1
F_{max} x 10^5 (g_{UA} cm^{-2} h^{-1})
0
1
2
3
4
5
6
7
Ψ (t)
0
0.2
0.4
0.6
0.8
1
1.2
1.4

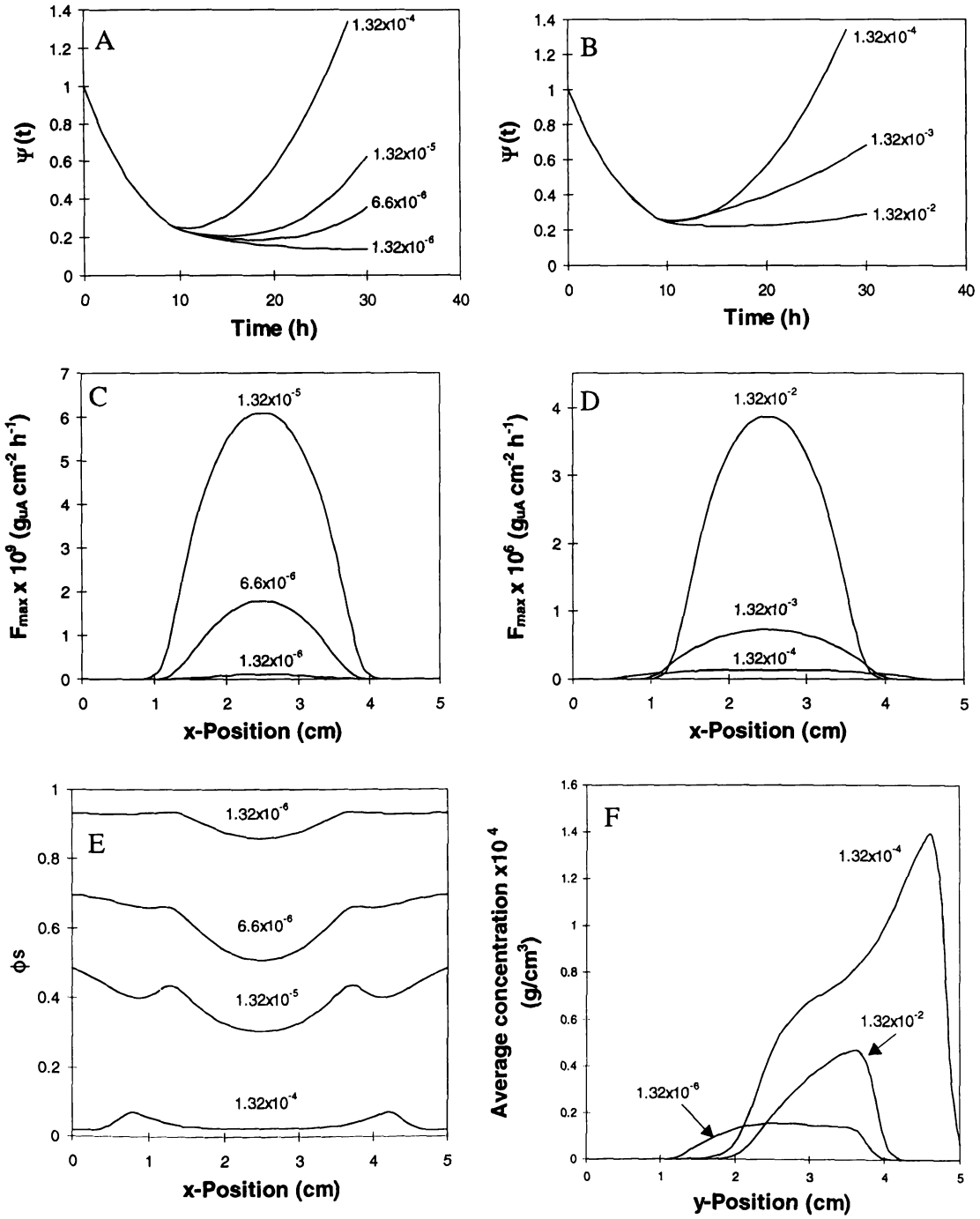


Figure 35: Variation of S source concentration. Time=28 h for all figures.

5.5.6 Va

A

microbial

S-source

Alternati

chemoatt

(top) in

A was

chemo

respon

enter a

compe

5.5.6 Variation of inoculation conditions

A large variety of inoculation patterns can be envisioned that could influence the microbial competition. For example, one population could be inoculated further from the S-source than the other, possibly disadvantaging the population further from the S-source. Alternatively, two different inoculation points equidistant from the source of the chemoattractant S could be used.

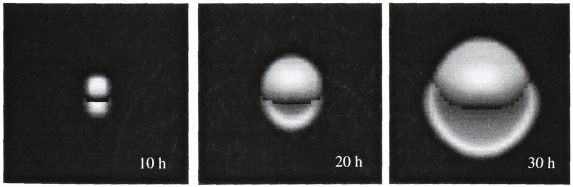


Figure 36. Pop. A inoculated farther from S source than pop. B.

In Figure 36, a the first approach was tested. The source of S is from the north (top) in each image. The cell properties are the same as used in Section 5.5.5. Population A was inoculated further from the S-source, but directly in line with Population B. The chemotactic wave in Population A still was able to form, although it may be more in response to the chemoattractant Q than to S. As the S gradient develops over time, it will enter areas of the DGC where Population A is present, and Population A may gain the competitive advantage.

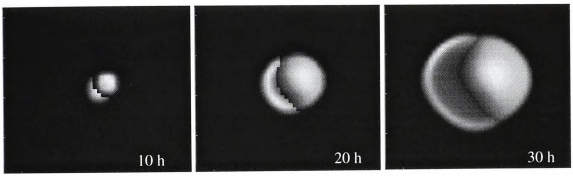


Figure 37. Pop. A inoculated to the side and away from S source.

Figure 37 shows the results for a combination of the two inoculation classes discussed. Population B has been moved closer to the S-source, while Population A was moved to one side, away from Population B. Again the chemotactic wave is apparent, and Population A is able to move into a larger area of the chamber than Population B.

Figure 38, which shows the dynamic competition factor curves for the simulations in Figure 36 and Figure 37, indicates that in both figures, the S-gradient is giving Population A some advantage, because it is only after S reaches Population A (approximately 10 hours), that the dynamic competition factor begins to increase ($\psi'>0$), indicating that Population A is gaining mass at a faster rate than B.

It is apparent from these few test situations that the inoculation pattern of the two populations can play an important role in determining which population will have the competitive advantage. Another situation that could be tested is to have one population well-established in the DGC, and introducing the other at a much lower concentration, to test if it can successfully "invade" and be coexistent with the initial population. The

results of

the time

5.5.7

compo

being

popula

nutrie

hypot

given

results of all of these variation of inoculation point simulations are highly dependent upon the time point being studied.

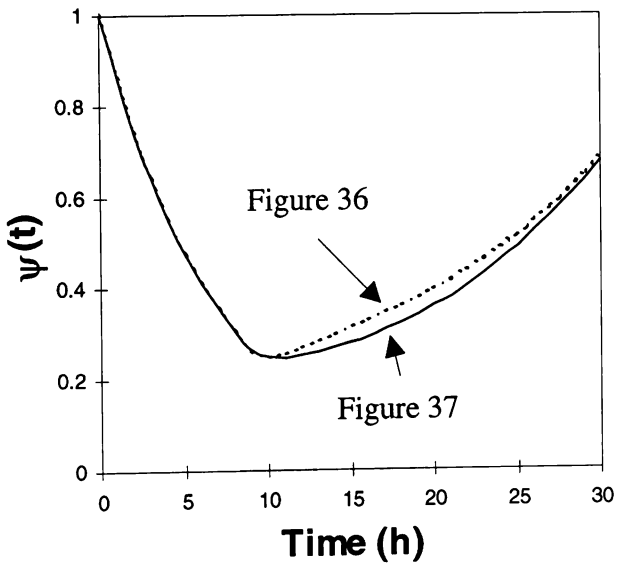


Figure 38. Dynamic competition factors for two inoculation patterns.

5.5.7 Variation of nutrient initial conditions

Another hypothesis that was explored with the model was that if the nutrient compound was diffusing in from one reservoir with the chemoattractant, as opposed to being uniformly distributed through the gel, then the chemotactic response of a population might give it the competitive advantage by moving that population toward the nutrient, while the other was left behind in a low nutrient environment. To test this hypothesis, a simulation was performed where the two populations had the parameters given in Table 4, except χ_{0AS}=0.05 cm^2/h, v_{AH}=0.5 h^{-1}, and v_{BH}=0.7 h^{-1}. The nutrient was

present i

was pres

same as

simulati

39C sho

present in the north reservoir (x = 5 cm) at a concentration of 8.6×10^{-4} g/cm^3, and no H was present in the gel initially. The chemoattractant and cell initial conditions were the same as given in Table 5. Figure 39A shows the dynamic competition factor for this simulation, and Figure 39B shows the average concentration curves at time = 30 h. Figure 39C shows the centerline profiles (y=2.5 cm) of the H and S compounds.

never

Popula

39B, t

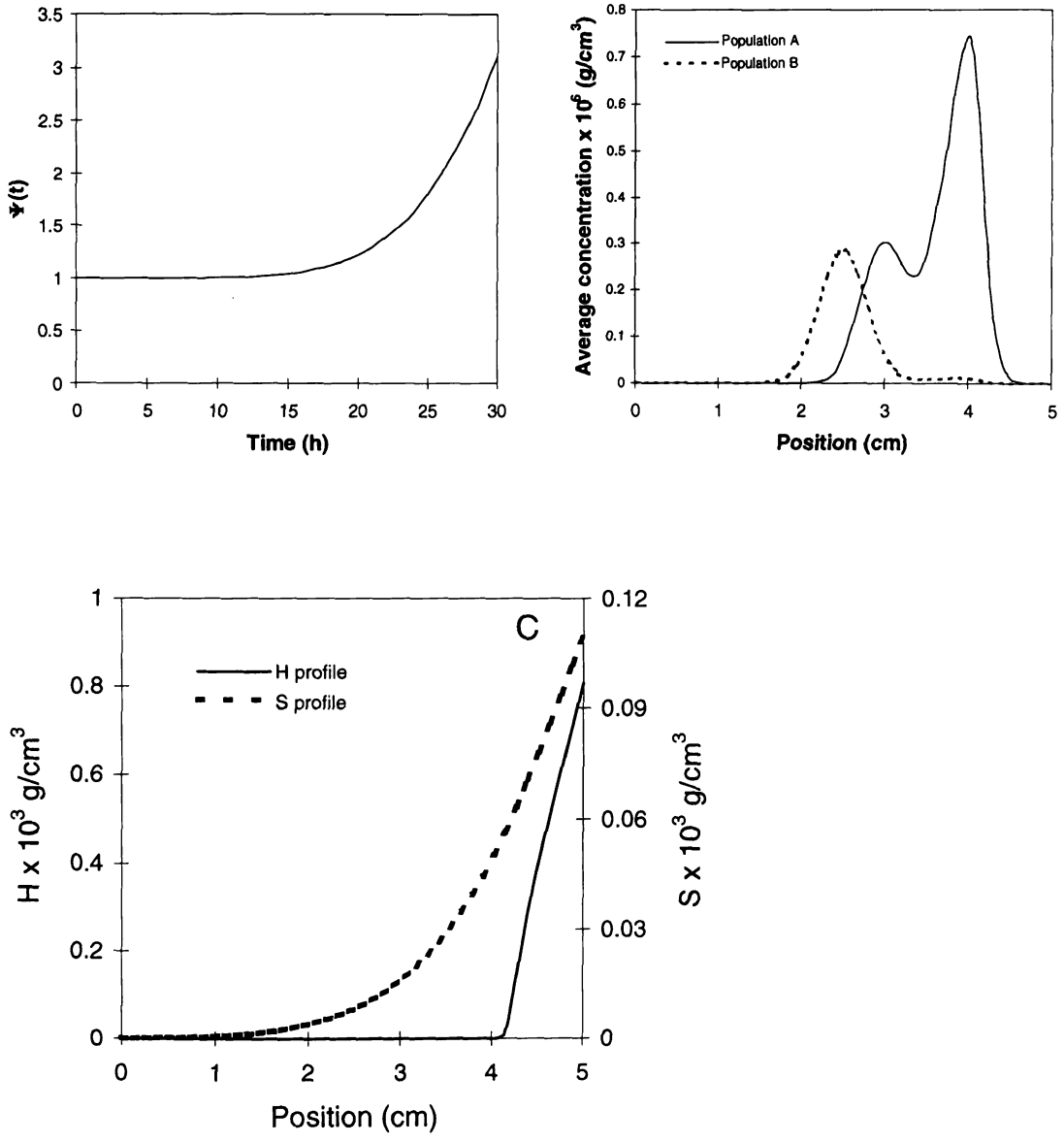


Figure 39. Competition for nutrient diffusing in at x = 5 cm.

Figure 39A illustrates that even though Population B has a growth advantage, it never obtains a higher mass than Population A. This is explained by the fact that Population B never experiences a high concentration of the nutrient. As shown in Figure 39B, the chemotactic response of Population A has allowed it to move toward the source

of nutrie

complete

mass of

likely is

consume

pronoun

was a fa

exist, ch

exampl

discuss

5.6 F

method

should

single

chemo

and th

studie

howe

of nutrient and chemoattractant. Figure 39C illustrates that the nutrient has been completely consumed by Population A, and is not available to Population B. The small mass of Population B that appears close to the source of S and H in Figure 39B most likely is the product of a small amount of growth that occurred before Population A fully consumed the H that had diffused to the center of the chamber.

In this simulation, the chemotactic response of Population A gave it an even more pronounced advantage in that ψ never had a value less than 1, even though Population B was a faster grower. In a system where gradients of both chemoattractants and nutrients exist, chemotaxis may be extremely important in determining the competitive winner. An example of a real system where these conditions might occur is bioremediation, as discussed in Section 7.

5.6 Future work on competition

The next step needed in the competition studies is to develop experimental methods to corroborate the modeling predictions. The focus of the experimental work should be on finding mutant bacteria that would differ from the main strain only in a single transport property, such as random motility, or response to a single chemoattractant.

Many transport mutants of *E. coli* have been previously isolated and cataloged, and the genes involved in the specific mutations are well defined. Emerson *et al.* (1994) studied the growth of such *E. coli* mutants in the DGC. At the time of this writing, however, *Pseudomonas* KC (see Section 7) was the focus of this portion of the research

project. V

efforts to

working

mutants

desired p

(

possess

mutant a

after the

marker

competi

work) t

Escheri

more c

mediur

asparta

growin

optical

Figure

project. Well-characterized mutants of *Pseudomonas* KC are not currently available, but efforts to develop them are underway in Dr. Craig Criddle's laboratory. Petty Setiawan, working on a project that was part of this thesis work, screened 120 *Pseudomonas* KC mutants for their motility and chemotaxis parameters. None were found that have the desired properties that would make them useful in the competition experiments.

One way to perform these experiments would be to select transport mutants which possess a selection marker, such as antibiotic resistance or fluorescence. A mixture of the mutant and non-mutant strains could then be inoculated in a swarm plate or the DGC, and after the growth and motility patterns had formed, samples could be taken. The selection marker would allow each strain to be counted individually, and the outcome of the competition could be quantified.

A preliminary competition experiment has been performed (Petty Setiawan's work) to begin to study the effects of chemotaxis on competitions. In this experiment, *Escherichia coli* was competed against *Pseudomona stutzeri* strain KC (see Section 7 for more details on Pseudomonas KC). Swarm plates were poured that contained M9 medium at pH 7.5. The plates contained 2 mM glycerol as the nutrient source and 0.1 mM aspartate as the chemoattractant. Two plates were inoculated with 20 µl of either actively growing *Pseudomonas* KC or *E. coli* culture that had been adjusted to have the same optical densities. The plates were incubated at 27 °C for 48 hours. These plates, shown in Figure 40, allowed the response of the individual populations to be observed.

Fi

density

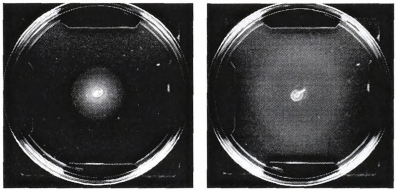
for 48

sample

diluted

were i

KC co



E. coli only Pseudomonas KC only

Mixture of two populations

Figure 40. Competition between *Pseudomonas* KC and *E. coli* in swarm plates.

In a third swarm plate, a 20 µl aliquot of an equal mixture, based on optical density, of the two populations was inoculated in a swarm plate and incubated at 27 °C for 48 hours. An image of this swarm plate is also given in Figure 40. After 48 hours, samples were taken at 5 locations in the competition swarm plate. The samples were diluted by a factor of 1:10,000 and plated on nutrient agar plates. The nutrient agar plates were incubated for two days at 27 °C. Individual colonies of *E. coli* and *Pseudomonas* KC could be differentiated by their unique morphologies, and counted. The results are

shown i

plate.

results

The *Ps*

the cer

into ar

gradie

(glyce

was m

shown in Figure 41. The origin of the x-axis corresponds to the center point of the swarm plate.

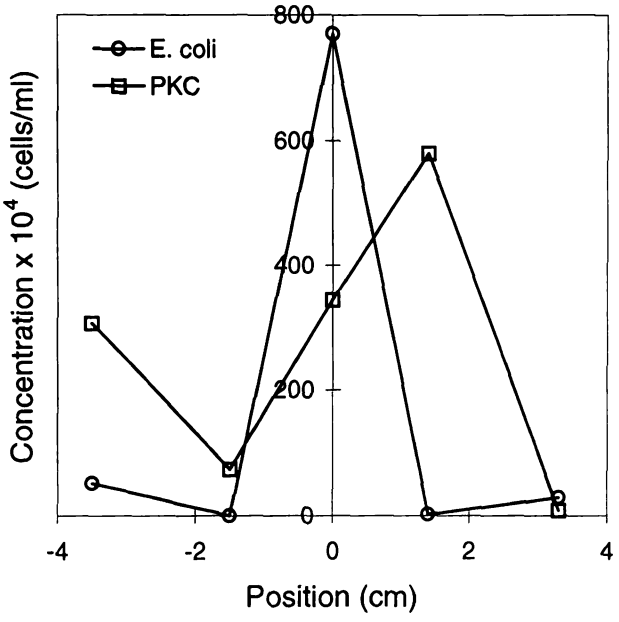


Figure 41. Colony counts in competition experiment.

The *E. coli* remained primarily in the center of the swarm plate. Although the results are not symmetrical, the *Pseudomonas* KC have moved outward from the center. The *Pseudomonas* KC may be at a competitive advantage in this case if the resources in the center of the plate are fully consumed. The *Pseudomonas* KC would be able to move into areas at the edge of the growth front where resources are still plentiful.

More work needs to be done with experiments such as this one. For instance, gradient measurements would be useful in determining if resources such as the nutrient (glycerol) were still available in the center of the plate. Also, in this experiment, no effort was made to control the relative growth rates of the two populations.

6. SEL

6.1 I

T

a M. S.

were pe

Mutage

Chambe

Biochen

DAHP

metabo

feedbac

tested

had in

analog

away f

lid was

for 10

that w

The cl

glucos

6. SELECTION OF MUTANTS

6.1 Experimental work

This portion of the research was completed as a collaboration with Mark Mikola, a M. S. candidate in Dr. Worden's research group. The experimental portions of the work were performed entirely by Mr. Mikola, and are detailed in Mikola (1996) and in "In-situ Mutagenesis and Chemotactic Selection of Microorganisms in a Diffusion Gradient Chamber" by M. R. Mikola, M. T. Widman, and R. M. Worden, submitted to *Applied Biochemistry and Biotechnology.*

To summarize, it was desired to obtain a strain of *E. coli* whose isozyme of DAHP synthase, designated AroF, was not inhibited by L-tyrosine, a product in the metabolic pathway. To achieve this goal, a population of *E. coli* that experienced feedback resistance from tyrosine was inoculated into a DGC. The *E. coli* had previously tested positive for chemotaxis toward glucose in a separate DGC experiment. The DGC had in its source flask 125 μM *m*-fluorotyrosine (m-FT), a non-metabolizable tyrosine analogue, and 5 mM glucose. The bacteria only grew in the portion of the DGC farthest away from the source of m-FT (see Figure 42A). After the initial growth period, the DGC lid was removed, and the DGC was exposed to ultraviolet radiation at a distance of 60 cm for 10 seconds. The lid was replaced and the incubation continued. Blooms of mutants that were less inhibited by the m-FT appeared after the mutation process (Figure 42B). The chemotactic response to the glucose gradient drew the mutants toward the source of glucose and m-FT, as seen in Figure 42C.

Fig

6.2

modifie

cell ba

Also, a

by both

where

consu

satura

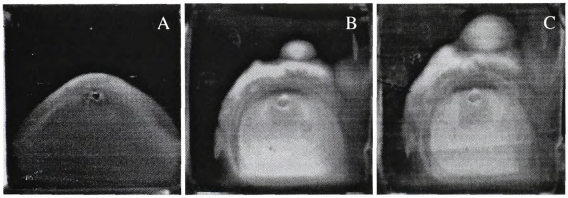


Figure 42. Mutant selection by chemotactic response (source at top of pictures).

6.2 Modeling selection of mutants

The mathematical model of the DGC system presented in Section 4.2 was modified and used to simulate strain selection in the DGC. For this application, a second cell balance was added to account for both the original (A) and the mutant (B) strains. Also, a balance was added to account for diffusion and cellular uptake of the inhibitor (P) by both strains:

$$\frac{\partial P}{\partial t} = D_P \nabla^2 P - \frac{v_{aP} P}{C_{aP} + P} u_a - \frac{v_{bP} P}{C_{bP} + P} u_b \qquad (46)$$

where D_P is the diffusion coefficient of the inhibitor; v_{aP} and v_{bP} are the specific consumption coefficients for Populations A and B consuming P; and C_{aP} and C_{bP} are the saturation constants for consumption of P by Populations A and B, respectively.

T

chemoat

growth r

terms ch

Ollis, 19

where

respons

sensitiv

popula

by

where

uninh

motil

The chemotactic sensitivity coefficients for both populations responding to the chemoattractants S and Q, the random motility coefficients, and the maximum specific growth rates were modified to incorporate the inhibition effects. The form of the modified terms chosen is analogous to noncompetitive inhibition in enzyme kinetics (Bailey and Ollis, 1986). The modified chemotactic sensitivity is given by

$$\chi'_{0ij} = \frac{\chi_{0ij}}{1 + \frac{P}{K_{Iij\chi}}} \tag{47}$$

where χ'_{0ij} is the inhibited chemotactic sensitivity for the i^{th} population (i=a or b) responding to the j^{th} chemoattractant (j=S or Q), χ_{0ij} is the uninhibited chemotactic sensitivity, and $K_{Iij\chi}$ is the inhibition constant for the chemotactic sensitivity of the i^{th} population to the j^{th} chemoattractant. The modified random motility coefficient is given by

$$\mu'_i = \frac{\mu_i}{1 + \frac{P}{K_{Ii\mu}}} \tag{48}$$

where μ'_i is the inhibited random motility coefficient for the i^{th} population, μ_i is the uninhibited random motility coefficient, and $K_{Ii\mu}$ is the inhibition constant for the random motility of the i^{th} population. The inhibited maximum specific growth rate is given by

where v

H, v_{iH} i

specific

cell bal

where

includ

where

the yi

(Widr

Equat

trends

funct

calcu

$$v'_{iH} = \frac{v_{iH}}{1 + \frac{P}{K_{Iiv}}} \tag{49}$$

where v'_{iH} is the inhibited maximum specific growth rate for the i^{th} population growing on H, v_{iH} is the maximum specific growth rate, and K_{Iiv} is the inhibition constant for the specific growth rate of the i^{th} population. In terms of the variables defined above, the two cell balance equations are given by

$$\frac{\partial u_i}{\partial t} = \mu'_i \nabla^2 u_i - \sum_{j=S}^{Q} \chi'_{0ij} \nabla \cdot \left[\left(\frac{K_{Dii}}{\left(K_{Dij} + j\right)^2} \right) u_i \nabla j \right] + \frac{v'_{iH} H}{C_{iH} + H} u_i \tag{50}$$

where u_i is the i^{th} cell population (either a or b). The nutrient balance was also modified to include the inhibited maximum specific growth rates:

$$\frac{\partial H}{\partial t} = D_H \nabla^2 H - \frac{v'_{aH} H}{C_{aH} + H} \frac{u_a}{Y_{aH}} - \frac{v'_{bH} H}{C_{bH} + H} \frac{u_b}{Y_{bH}} \tag{51}$$

where D_H is the diffusion coefficient for H, C_{iH} is the half-saturation constant, and Y_{iH} is the yield coefficient. Values for the modeling constants were taken from the literature (Widman *et al.*, 1997) insofar as possible. Values for the inhibition constants included in Equations (47)-(49) were chosen that gave reasonable agreement with the experimental trends.

Figure 43 shows a time sequence of contour plots depicting cell concentration as a function of position. The contour lines show constant concentration isoclines, as calculated by a built-in Matlab program. The chemoattractant (glucose) gradient is

indicate

There is

the top

observe

indicated by gray shading, with lighter gray indicating a higher glucose concentration. There is no m-FT in this simulation. The source reservoir is on the side corresponding to the top of each figure. Bias of the cell's migration in this direction is comparable to that observed experimentally.

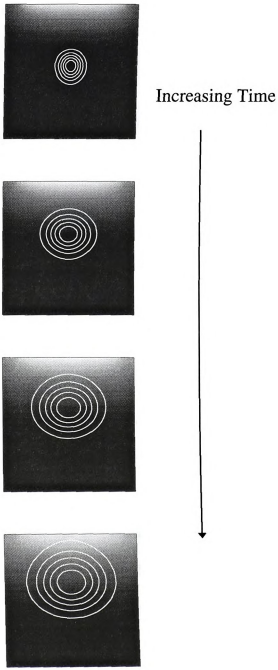


Figure 43. *E. coli* responding to chemoattractant gradient.

shown

evoluti

only th

indicat

43, is

inhibit

DGC.

param

Popul

bound

The

chem

secon

the g

reser

and

inocu

Popu

The ability of the mathematical model to reproduce the experimental trends shown in Figure 42 was evaluated. Figure 44 shows the predicted growth-pattern evolution in which gradients of both a chemoattractant and an inhibitor are applied, and only the population sensitive to the inhibitor is present. In this figure, the gray shading indicates the inhibitor gradient. The effect of chemotaxis, which was significant in Figure 43, is overwhelmed by the effect of the inhibitor. As the concentration gradient of the inhibitor becomes established, cell growth occurs predominantly in the sink end of the DGC. This trend was observed experimentally in Figure 42. The base case set of parameters for all of the simulations shown is given in Table 6. In these simulations, Population A was the inhibited population, while Population B was the mutant. The boundary and initial conditions are similar to those presented in Sections 4.2.1 and 4.2.2. The inhibitor concentration (P) in the source reservoir was 0.0132 g/cm^3. The chemoattractant concentration (S) in the source reservoir was 0.000132 g/cm^3. The second chemoattractant concentration (Q) in the source and sink reservoirs and initially in the gel was 0.0000132 g/cm^3. The nutrient concentration (H) in the source and sink reservoirs and initially in the gel was 0.000046 g/cm^3. The value of u_{a0} was 3×10^{-6} g/cm^3, and ub0 was 3×10^{-7} g/cm^3. The gradients were allowed to initiate for 12 h before inoculation of Population A. Population B appeared in the simulation 15 h after Population A. The simulation continued for 25 h after the appearance of Population B.

The

shown in F

evolution in

only the po

indicates th

43, is overr

inhibitor be

DGC. This

parameters

Population

boundary a

The inhib

chemoattra

second ch

the gel w

reservoirs

and ub0

inoculatio

Populatio

The ability of the mathematical model to reproduce the experimental trends shown in Figure 42 was evaluated. Figure 44 shows the predicted growth-pattern evolution in which gradients of both a chemoattractant and an inhibitor are applied, and only the population sensitive to the inhibitor is present. In this figure, the gray shading indicates the inhibitor gradient. The effect of chemotaxis, which was significant in Figure 43, is overwhelmed by the effect of the inhibitor. As the concentration gradient of the inhibitor becomes established, cell growth occurs predominantly in the sink end of the DGC. This trend was observed experimentally in Figure 42. The base case set of parameters for all of the simulations shown is given in Table 6. In these simulations, Population A was the inhibited population, while Population B was the mutant. The boundary and initial conditions are similar to those presented in Sections 4.2.1 and 4.2.2. The inhibitor concentration (P) in the source reservoir was 0.0132 g/cm^3. The chemoattractant concentration (S) in the source reservoir was 0.000132 g/cm^3. The second chemoattractant concentration (Q) in the source and sink reservoirs and initially in the gel was 0.0000132 g/cm^3. The nutrient concentration (H) in the source and sink reservoirs and initially in the gel was 0.000046 g/cm^3. The value of u_{a0} was 3×10^{-6} g/cm^3, and ub0 was 3×10^{-7} g/cm^3. The gradients were allowed to initiate for 12 h before inoculation of Population A. Population B appeared in the simulation 15 h after Population A. The simulation continued for 25 h after the appearance of Population B.

Table 6. Parameter values for inhibition model

Parameter (i=a or b)	**Value**
μ_i	0.010 $cm^2 h^{-1}$
χ_{0iQ}	0.08 $cm^2 h^{-1}$
χ_{0iS}	0.08 $cm^2 h^{-1}$
C_{iH}	$4.08x10^{-6}$ $g_H cm^{-3}$
Ci_Q	$6.70x10^{-8}$ $g_Q cm^{-3}$
C_S	$5.50x10^{-5}$ $g_S cm^{-3}$
D_H	0.01 $cm^2 h^{-1}$
D_Q	0.033 $cm^2 h^{-1}$
D_S	0.033 $cm^2 h^{-1}$
D_P	0.033 $cm^2 h^{-1}$
K_{DiQ}	$3.30x10^{-5}$ $g_Q cm^{-3}$
K_{DiS}	$2.00x10^{-6}$ $g_S cm^{-3}$
v_{iH}	0.35 h^{-1}
v_{iQ}	0.02 $g_Q g_u^{-1} h^{-1}$
v_{iS}	0.60 $g_S g_u^{-1} h^{-1}$
Y_{iH}	0.50 g_u/g_H
$K_{Iaj\chi}$	$1.0x10^{-6}$ g cm^{-3}
$K_{Ia\mu}$	$1.0x10^{-6}$ g cm^{-3}
K_{Iav}	$1.0x10^{-6}$ g cm^{-3}

Figu

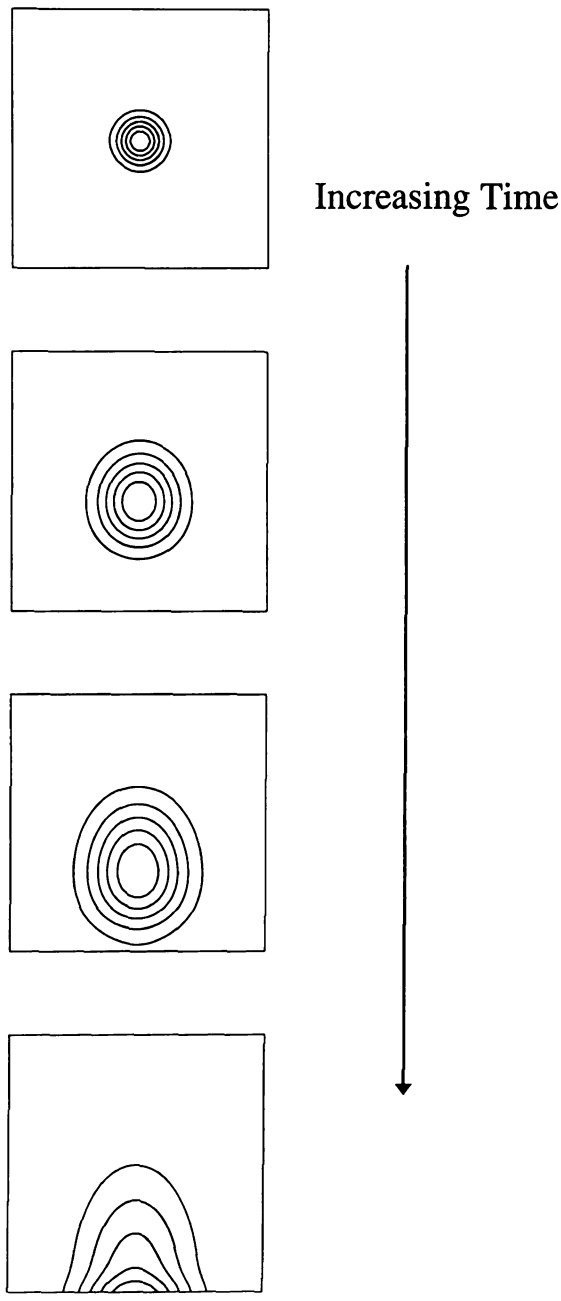


Figure 44. *E. coli* responding to inhibitor gradient and chemoattractant gradient.

Figu

B) that is ir

the inhibito

significant

a lower co

effect of th

moves pr

predomina

experimen

Th

favorable

that lose t

chemoattra

method sl

inhibtion f

the trends

be elucida

Figure 45 is a simulation of a mutation event giving rise to a mutant (Population B) that is insensitive to the inhibitor. As in Figure 44, Population A, which is sensitive to the inhibitor, grows preferentially near the sink. Population B takes longer to appear in significant concentration, because this population was initiated after Population A, and at a lower concentration than Population A, to simulate the UV mutagenesis. The beneficial effect of the chemoattractant in separating the two populations is evident, as Population B moves preferentially toward the source reservoir, while Population A remains predominantly near the sink reservoir. These trends are similar to those observed experimentally in Figure 42.

The model allowed the effectiveness of the method to be explored under less favorable conditions than were experienced in the experimental work, such as mutants that lose their chemotactic response, or that appear in areas of the DGC far from the chemoattractant source. The model predicts that even in these worse case scenarios, the method should still allow the desired mutants to be selected. The noncompetitive inhibtion form used in the modeling (Equation (47), for example) adequately reproduced the trends of the experiment, but other forms may be more physically realistic, and may be elucidated through further experiments and modeling.

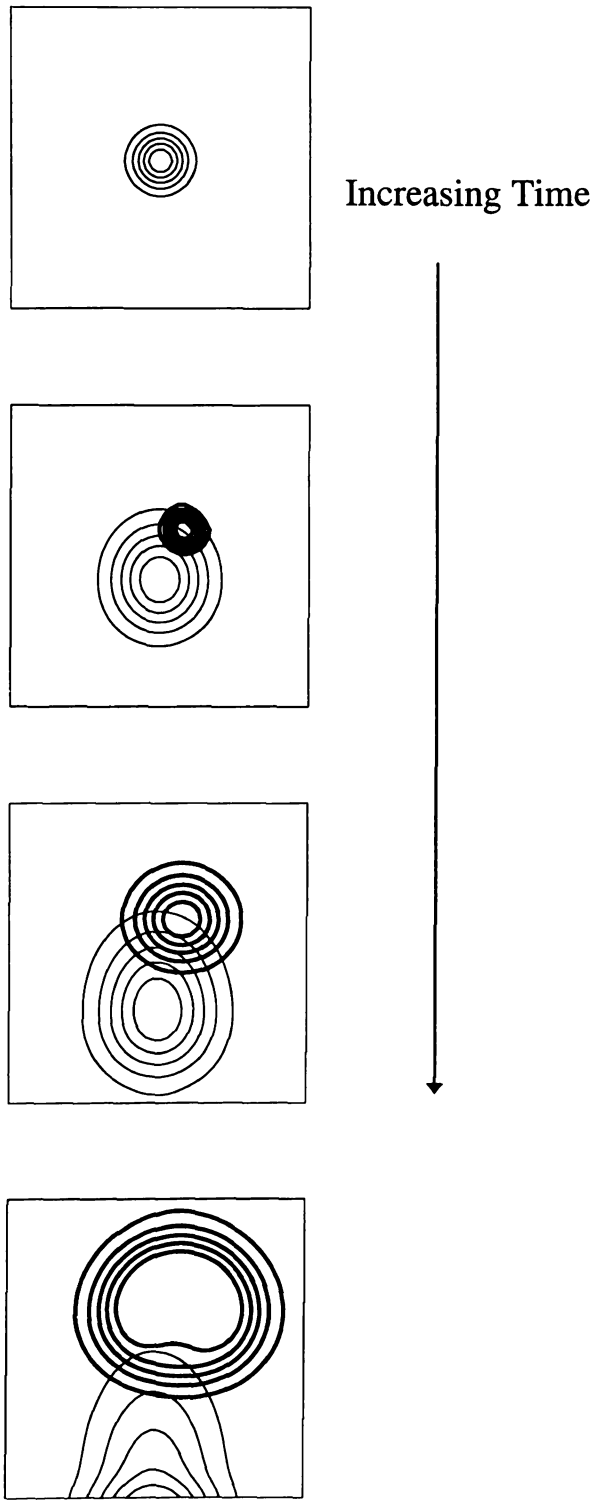


Figure 45. Simulation of experiment shown in Figure 42.

The r

a simulation

of Populatio

two populat

predominan

to Q. To es

repeated w

chemotaxis

potent the

The model can also be used to test other situations. For example, Figure 46 shows a simulation of the case where the mutation occurred in the middle of the growth pattern of Population A, and the mutant (Population B) was not chemotactic to S. In this case, the two populations were still able to be separated, but in this case the separation is based predominantly on the influence of the inhibitor gradient and spreading due to chemotaxis to Q. To estimate the benefit obtained by chemotaxis toward S, this simulation was then repeated with chemotaxis to S reinstated. The results, shown in Figure 47, indicate that chemotaxis toward S does further enhance the rate of separation. Presumably, the more potent the chemoattractant (i.e., the higher the χ'_{0ij} value) the greater the enhancement.

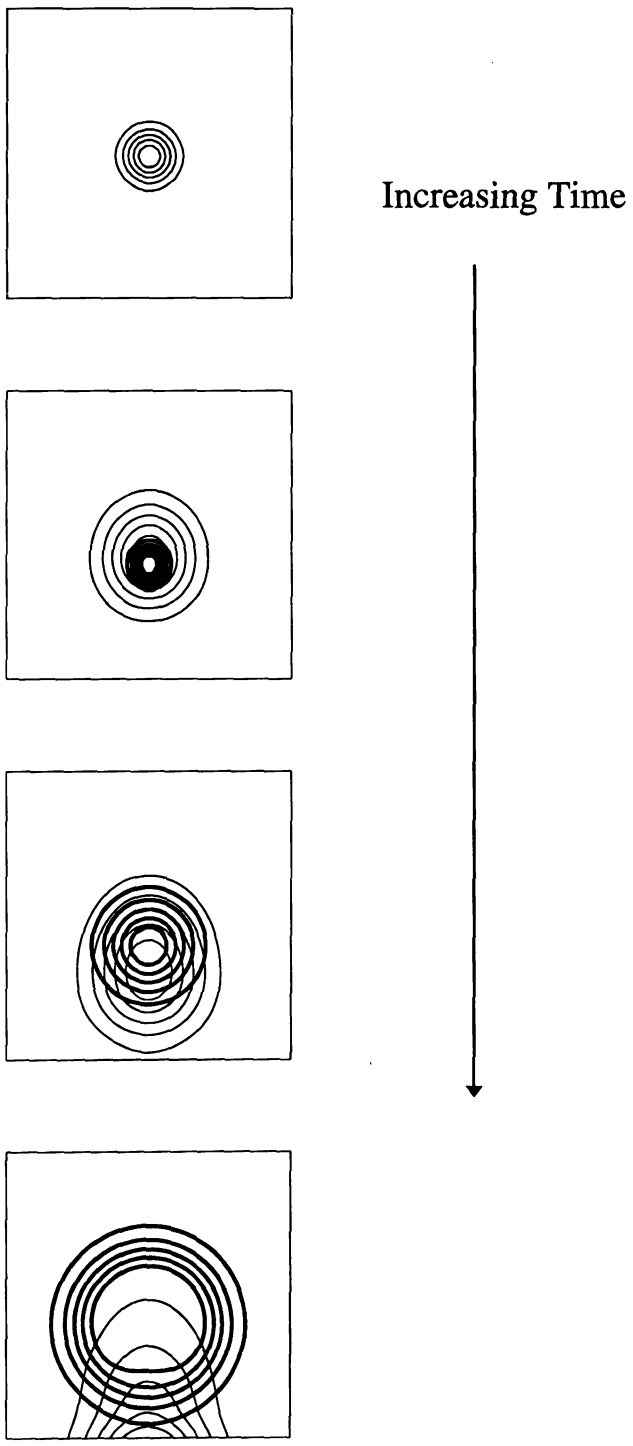


Figure 46. Mutant appears far back in population.

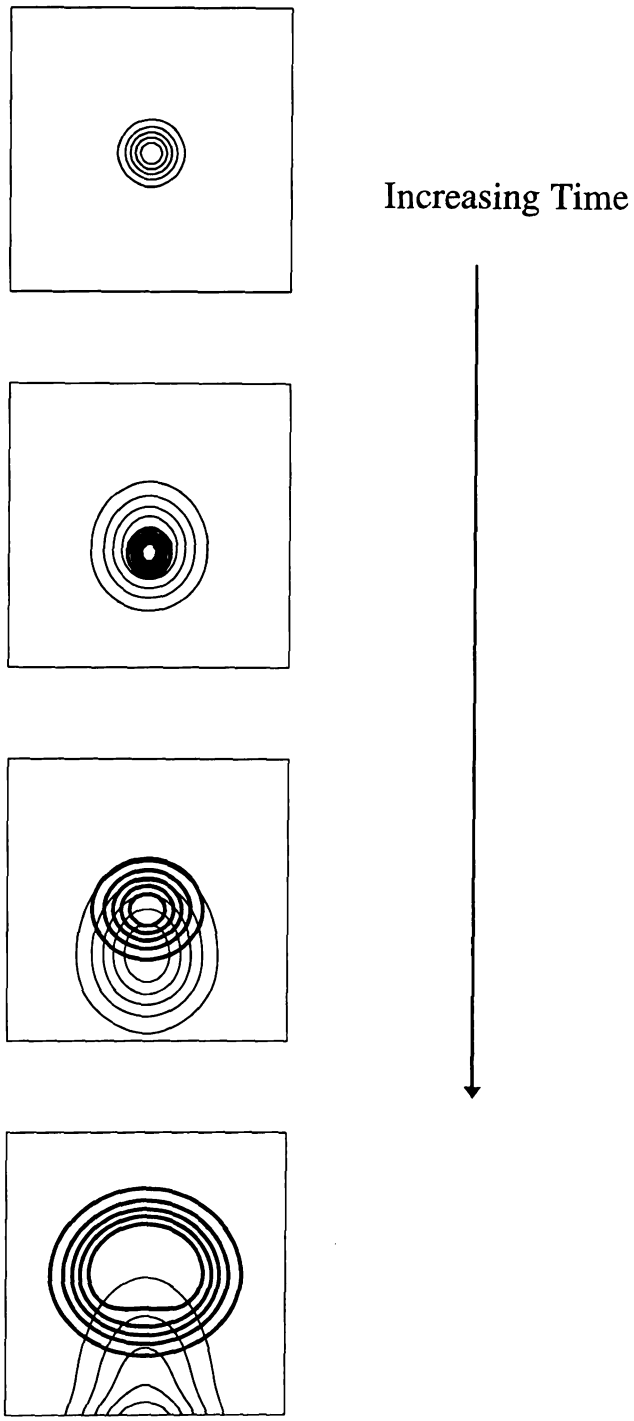


Figure 47. Mutant now chemotactic to S.

7. BIORE

The

environmen

et al., 1993

references,

or consider

might sugg

that severa

through so

Rec

porous me

walk mod

the ratio

porous me

column th

both the r

replaced

ecologica

roots by

and chem

7. BIOREMEDIATION

The transport of bacteria through porous media, specifically soil or aquifer environments, has been studied in both experimental and modeling systems (Abu-Ashour, *et al.*, 1993; Sarkar, *et al.*, 1994a; Sarkar *et al.*, 1994b, for example). In a number of these references, the chemotactic movement of the bacteria has been ignored (Tan, *et al.*, 1994) or considered too complex to include, even though results have been recorded which might suggest chemotaxis is important. For example, Abu-Ashour *et al.* (1993) reported that several field and laboratory experiments found that the average velocity of bacteria through soil was faster than the velocity of tracers and of the groundwater.

Recently, a few studies have addressed the role motility plays in transport through porous media. Duffy *et al.* (1995) attempted to adapt the homogeneous media random walk model to a porous media. They utilized a tortuosity factor that was proportional to the ratio of the random motility in bulk liquid to the effective random motility in the porous media. Their model gave good qualitative agreement with an experimental sand column through which *Pseudomonas putida* swam. Barton and Ford (1997) showed that both the random motility coefficient and the chemotactic sensitivity coefficient could be replaced with effective values that incorporate the effect of the porous media. An ecological system in which chemotaxis through soil is important is the colonization of roots by *Rhizobium meliloti.* Soby and Bergman (1983) demonstrated that active motility and chemotaxis were necessary for efficient spreading of the bacteria through soil.

The e

an *in situ*

phenanthren

the areas of

They blame

sands, reter

contaminar

presented a

model inc

experimen

Ps

tetrachlori

chloroforr

other mic

microbe

experime

1996). T

bioaugm

Schoolcr

as an ele

acceptor

contami

The effective transport of bacteria through the soil is an important requirement for an *in situ* bioremediation project. Devare and Alexander (1995) found that a phenanthrene-metabolizing *Pseudomonas* sp. could adequately remove phenanthrene in the areas of a soil column in which the bacteria was inoculated, but not in other areas. They blamed this result on bacterial retention by the clay soil, and noted that in aquifer sands, retention was much less. Their conclusion was that transport of bacteria to the contaminant was very important to the success of bioremediation. Bosma *et al.* (1988) presented a model for bacteria consuming xenobiotic chemicals in a soil column. Their model incorporated chemotactic movement, and its results compared well with experimental data.

Pseudomonas stutzeri KC is a denitrifying bacterium able to degrade carbon tetrachloride (CT) into carbon dioxide and nonvolatile products without the production of chloroform (Criddle *et al.*, 1990; Dybas *et al.*, 1995; Mayotte *et al.*, 1996). In contrast, other microbes typically convert CT to chloroform. The outstanding potential of this microbe for bioremediation of CT spills has been demonstrated both in shaker-flask experiments (Tatara *et al.*, 1993), and a model aquifer system (Witt 1994; Mayotte *et al.*, 1996). The Michigan Department of Natural Resources is sponsoring a major bioaugmentation field experiment with this organism in a CT-contaminated aquifer near Schoolcraft, Michigan. In this experiment, acetate is periodically injected into the ground as an electron donor for the reaction, and the naturally occurring nitrate is the electron acceptor. The *Pseudomonas* KC is introduced into the aquifer soil, and the flow of contaminated water is directed through a "biofence" of the bacteria.

Che

in the labo

packed with

equal to tha

indicated th

front was r

15 cm/day.

7.1 Expe

7.1.1 Swa

Stu

KC to var

were prepa

adjusted t

solutions

into petri

Pseudomo

experimen

Scientific

Fi

levels of

nitrate co

Chemotactic movement of *Pseudomonas* KC has been observed in sand columns in the laboratory (personal communication with Mike Witt). The sand column was packed with aquifer sand, and supplemented with acetate. The nitrate concentration was equal to that observed in the Schoolcraft aquifer. Preliminary experimental measurements indicated that the chemotactic movement was in response to nitrate gradients. The cell front was moving at a velocity approximately 8 cm/day faster than the water velocity of 15 cm/day.

7.1 Experimental results

7.1.1 Swarm plates

Studies were initiated to characterize the chemotactic response of *Pseudomonas* KC to various possible chemoattractants. Solutions of Medium D (Tatara, *et al.*, 1993) were prepared with varying levels of acetate and/or nitrate. The pH of the medium was adjusted to 8.2. Swarm plates (Petty Setiawan's work) were made using the Medium D solutions supplemented with 0.25% high strength agar gel. The gel mixture was poured into petri dishes and allowed to solidify. A 20 µl aliquot of actively growing *Pseudomonas* KC culture was inoculated in the center of the plate. For the anaerobic experiments, the plates were incubated in a GasPak 150^{TM} Anaerobic System (VWR Scientific). For aerobic experiments, the plates were incubated on the lab bench.

Figure 48 shows the chemotactic ring patterns formed in response to varying levels of nitrate, at a constant acetate concentration of 1 g/l. In the image on the left, the nitrate concentration was 1 g/l, and on the right, the nitrate concentration was 2.5 g/l. All

swarm plate

to 1 g nitra

Pseudomon

acetate to n

image of F

occurred. C

inner ring c

consumpti

may be du

F

experim

the left

swarm plate images were taken 48 hours after inoculation. The mass ratio of 1 g acetate to 1 g nitrate is the balanced ratio for consumption of the two chemical species by *Pseudomonas* KC (personal communication with Dr. Craig Criddle). When the ratio of acetate to nitrate was equal to one, only a single ring appeared, as shown in the left-hand image of Figure 48. In the right-hand image, when nitrate is in excess, a double ring occurred. One hypothesis is that the outer ring is due to consumption of acetate, and the inner ring due to the use of the remaining nitrate, or an intermediate such as nitrite, for the consumption of another carbon source, such as endogenous metabolism. The outer ring may be due to acetate, nitrate, or a combination of both.

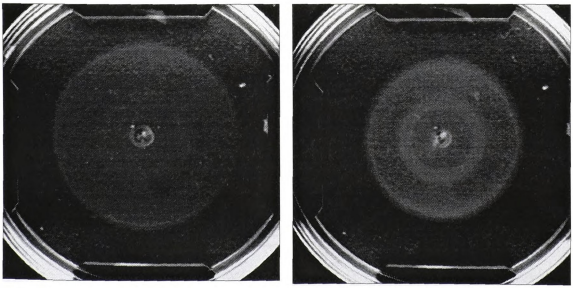


Figure 48. Chemotactic rings of *Pseudomonas* KC at varying nitrate levels.

In the next experiments, the acetate concentration was varied. Figure 49 shows experiments at a constant nitrate level of 50 mg/l, and acetate varying from 100 mg/l on the left to 1000 mg/l on the right. In the low acetate image, there is no noticeable

movement f

minimal lev

considerable

Figure 48. I

hand image

image 1000

the many

appeared w

bubbles m

denitrifica

movement from the inoculation condition. Both growth and motility seem to be at a minimal level. In the higher acetate image, however, the high density ring indicates considerable growth, but the diameter of the ring indicates that motility is slower than in Figure 48. In the next set of experiments, the nitrate level was set to 500 mg/l. In the left-hand image in Figure 50, the acetate concentration was 100 mg/l, and in the right-hand image 1000 mg/l. At 100 mg/l acetate, only a single ring forms. In the 1000 mg/l image, the many bright white dots are bubbles that appeared in the gel. The bubbles only appeared when the acetate to nitrate ratio was greater than one. We hypothesize that the bubbles may contain nitrogen that forms when excess acetate is present to complete the denitrification reaction shown in Equation (52)

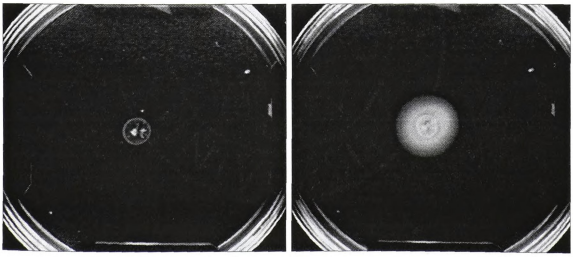


Figure 49. Nitrate concentration = 50 mg/l, varying acetate.

Pseudomo

acetate is

acceptor t

nitrogen,

I

medium.

found th

ranging

formed.

at the hi

Pseudo

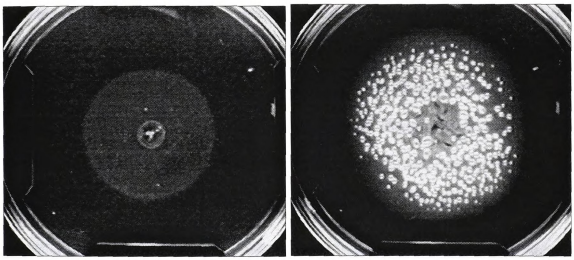


Figure 50. Nitrate concentration = 500 mg/l, varying acetate.

Pseudomonas KC is known to preferentially use nitrate before consuming nitrite. When acetate is in excess, however, the *Pseudomonas* may utilize the nitrite as an electron acceptor to further consume the remaining acetate. This would result in the formation of nitrogen, and possibly the bubbles.

$$NO_3^- \rightarrow NO_2^- \rightarrow N_2 \qquad (52)$$

In aerobic experiments with the *Pseudomonas* KC, no nitrate was added to the medium. Instead, oxygen was used as the electron acceptor. In results not shown, it was found that no ring formed at an acetate concentration of 100 mg/l. For concentrations ranging between 1000 mg/l and 1750 mg/l, a ring of approximately 4.7 cm in diameter formed. At a concentration of 2500 mg/l, no ring was observed. These results suggest that at the higher concentrations of acetate, the chemotactic receptors may be saturated.

Additional experiments were performed to test the aerobic response of *Pseudomonas* KC to attractants other than acetate. Medium M9 (Maniatis *et al.*, 1982),

adjusted to

medium wa

more cells

experiment

supplemen

carbon sou

Fig

7.1.2 DC

E

KC to ap

1.24 g/l

experime

adjusted to pH 7.5, was used for the medium. In the left-hand image in Figure 52, the M9 medium was supplemented with 1 mM glucose. A distinct chemotactic ring formed, but more cells seemed to remain in the center of the plate than in the acetate/nitrate experiments shown above. In the right-hand image of Figure 52, the medium was supplemented with 0.1 mM aspartate as the chemoattractant, and 2 mM glycerol as the carbon source. In this instance, a fairly wide, diffuse ring was observed.

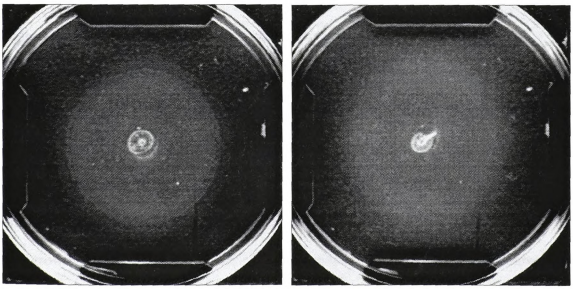


Figure 51. *Pseudomonas* KC responding to glucose (left) and aspartate(right).

7.1.2 DGC experiments

Experiments were performed in the DGC to test the response of the *Pseudomonas* KC to applied gradients of acetate and nitrate. Medium D-, containing 3.5 g/l K_2HPO_4, 1.24 g/l KH_2PO_4, and 1.0 g/l $(NH_4)_2SO_4$ was used as the minimal medium for the experiment. The pH of the medium was adjusted to 8.1 with KOH. After autoclaving, 2

ml/l of 1 M

to test the g

experiments

comparable

contained f

minimize t

chemotacti

degradatio

An

nitrate and

addition t

sink flask

and 0.25

coli grow

toward th

nearest t

appeared

source r

samples

the DG

15 µl a

ml/l of 1 M $MgSO_4$ was added to the medium. Shake flask experiments were performed to test the growth characteristics of *Pseudomonas* KC in this medium. The shake flask experiments showed that the maximum growth, as measured by optical density, was comparable to growth in the complete Medium D recipe. The Medium D- recipe contained fewer components, which was useful not only for simplicity, but also to minimize the number of consumable chemical components that could possibly elicit a chemotactic response. The effects of using Medium D- on carbon tetrachloride degradation were not studied as part of this work.

An initial DGC experiment was performed to test the chemotactic response to nitrate and acetate in Medium D- (results not shown). The source flask contained, in addition to 800 ml of Medium D-, 5 mM sodium acetate and 5 mM sodium nitrate. The sink flask contained 800 ml of Medium D-. The arena of the DGC contained Medium D- and 0.25% high-strength agar gel. After 28 hours of growth, a pattern similar to the *E. coli* growth patterns shown in Figure 13, had developed, with a definite bias in movement toward the acetate/nitrate source. A 0.5 ml sample was removed from the gel at the point nearest to the source reservoir. A similar sample was removed from the gel at what appeared to be the center of the chemotactic wave moving toward the acetate/nitrate source reservoir. The samples were injected into shake flasks and incubated. These samples were taken in an effort to isolate bacteria that were optimized for chemotaxis in the DGC environment.

A second DGC experiment was performed with the same parameters. This time, a 15 µl aliquot of the cell culture taken from the spot nearest the wall in the previous

experiment

22 hours b

Figure 52.

inoculation

experiment was used to inoculate the chamber. The gradient was allowed to initialize for 22 hours before inoculating the bacteria. The results of this experiment are shown in Figure 52. The times shown in the corners of the images correspond to the time after inoculation of the bacteria. The acetate/nitrate source is at the bottom in each image.

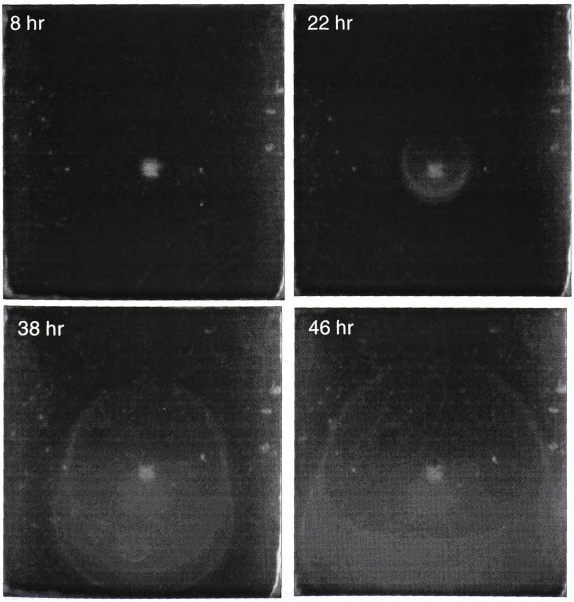


Figure 52. *Pseudomonas* KC responding to acetate and nitrate in a DGC.

7.1.3 Mo

A

otherwise

objects. T

experimer

sodium ac

KOH. Th

connector

solidify,

Figure 53

across th

ends of t

the wave

7.1.3 Movement of wave around obstacles

A first logical step in understanding the effect of solids on chemotaxis in an otherwise homogenous medium would be to study a medium which has one or a few objects. To test the response of the chemotactic wave to obstacles in its path, swarm plate experiments were performed (Laura Booms' work). Medium D- supplemented with 5 g/L sodium acetate, 3 g/L sodium nitrate, and 0.25% agar was adjusted to a pH of 8.2 with KOH. The plates were poured, and before the agar solidified, two sterilized tubing connectors with sealed ends objects were placed in the gel. The gel was allowed to solidify, and then 20 µl of Pseudomonas KC culture were inoculated in the center. In Figure 53A, taken 26 hours after inoculation, the wave has moved approximately halfway across the tubing connectors. In Figure 53B (42 hours), the wave has moved beyond the ends of the connectors. The objects did not appear to perceptibly inhibit the movement of the wave.

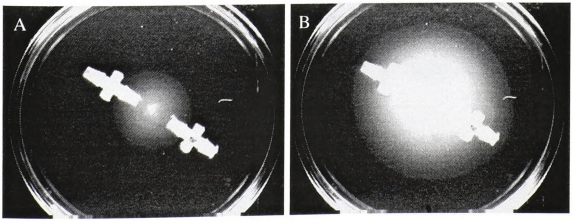


Figure 53. Movement of wave around obstacles in swarm plates.

To

performed

diameter

the path o

Ps

and the s

in a micr

15 µl of t

the gradi

A

The time

the botto

In Figur

experim

will not

To test how a large object would affect the chemotactic wave, an experiment was performed under the following conditions in a DGC. A cut-off glass test tube 1 cm in diameter was inserted into the gel, close to the source reservoir, to serve as an object in the path of the chemotactic wave.

Pseudomonas KC were grown for 3 days in a 100 ml flask containing Medium D- and the same ratio of acetate and nitrate as in the source flask. 6 ml of culture were spun in a microcentrifuge, and the resulting pellets were resuspended in 0.5 ml of fresh media. 15 µl of the resuspended culture were inoculated into the center of the arena 5 hours after the gradient was initiated.

A time-sequence of image captures from the experiment is shown in Figure 54. The time increases from left to right, and top to bottom. The source reservoir is located at the bottom in each picture. The edge of the wave just encountered the tube in Figure 54A. In Figure 54D the cells have passed the object and have rejoined on the other side. This experiment indicates that, at least in this very simple single-object situation, an obstacle will not catastrophically disrupt the chemotactic movement of the cell population.

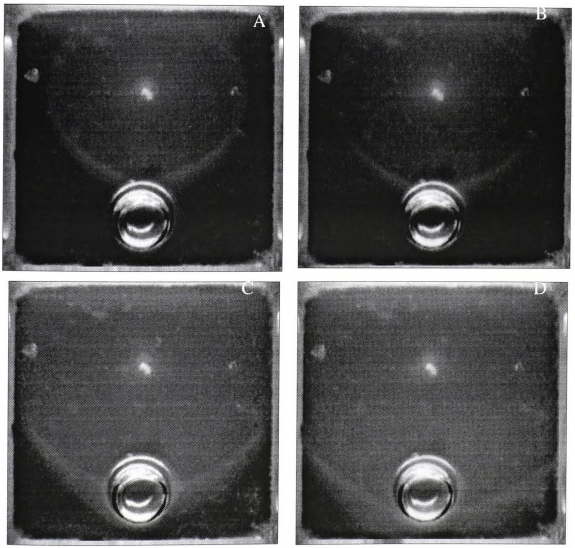


Figure 54. Chemotaxis of *Pseudomonas* KC around an object.

7.2 Fut

Fu

hypothesi

focus on

a wall of

DGC wa

cm from

into the

allowed

solidifie

Experim

designe

glass be

structur

could b

7.2 Future work on bioremediation

Further experimental and modeling work will be necessary to prove the hypothesis that chemotaxis can enhance a bioremediation effort. First, studies should focus on the effects of a porous medium on the chemotactic response. A method to create a wall of glass beads, sand, or soil in the DGC has been developed (see Figure 55). A DGC was modified to allow two specially cut microscope slides to be placed 1 cm and 2 cm from one wall. Glass beads were placed between the slides, and the gel was poured into the chamber. Small gaps between the slides and the bottom cover of the DGC allowed the gel to penetrate under the slides and up into the beads. After the gel solidified, the slides were carefully removed, and the three gel regions combined. Experiments to test how the bacteria are transported through the porous wall should be designed and performed with and without chemoattractant gradients present. Initially, glass beads of known diameter could be used to give a system with well-defined pore structure. Next, a more environmentally realistic wall material, such as aquifer sands, could be used.

Chemo
source

contami

had a te

used to

model v

model

Worden

porous

carbon

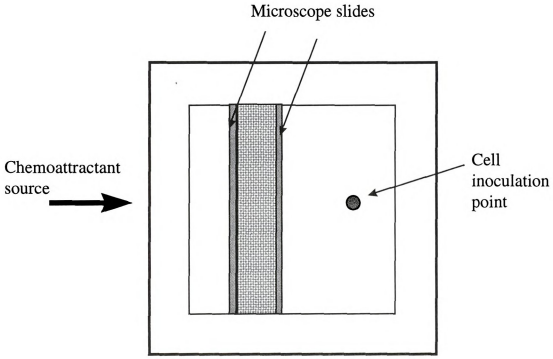


Figure 55. Set-up to make glass-bead wall in DGC.

A version of the mathematical model was developed that included a balance for a contaminant. The balance was similar to the nutrient and chemoattractant balances, but had a term that accounted for disappearance of the contaminant. This model could be used to predict the enhanced contaminant degradation when chemotaxis occurs. The model was never fully tested and debugged, and suitable parameters for the contaminant model were not found. A new cellular dynamics model has been proposed by Drs. Worden and Lastoskie which would allow for predictions of the behavior of cells in a porous medium.

For the *Pseudomonas* KC system specifically, a method to quantify the amount of carbon tetrachloride in a DGC will need to be developed. A control experiment with no

applied c

degradatic

movemen

compare t

growth sh

independe

O

bioremed

cells wit

injection

possible

uncontan

pulse che

how the

(Section

maximiz

optimize

zones. T

bioreme

money c

applied chemoattractant gradient should be performed to measure the amount of degradation of carbon tetrachloride that takes place in the absence of chemotactic movement. Then an experiment with an applied chemoattractant gradient should be run to compare the degradation rate to the control experiment. Factors such as the amount of growth should be accounted for in the analysis, so that the effect of chemotaxis can be independently evaluated.

Other experiments to discover ways in which chemotaxis could improve a bioremediation effort could focus on the concept of controlling the dissemination of the cells with chemoattractant gradients. One hypothesis is that by carefully placing the injection wells through which the chemoattractant is introduced into the soil, it may be possible to direct the spreading of the bacteria so as to minimize the losses into uncontaminated zones. A relatively simple experiment to test this hypothesis would be to pulse chemoattractant into different reservoirs of the DGC at timed intervals, and observe how the cell population responded. In the section of this thesis on microbial competition (Section 5), the model predicted that optimal chemoattractant concentrations exist that maximize the chemotactic wave properties, such as the carrying capacity, and could optimize the delivery protocols to obtain the best transport of cells to the contaminated zones. The mathematical model should prove to be an extremely useful tool to explore bioremediation, and, through modeling simulations, considerable experimental time and money can be saved.

8. SUM

Th

engineeri

response.

microbio

diffusion

chemoat

mathem

having t

predicti

involve

respons

mathem

the pre

advant

technic

compe

exploi

could

inhibi

8. SUMMARY

The focus of this research was on microbial chemotaxis, and how it could benefit engineering applications. Tools were developed to aid in the study of the chemotactic response. Experimental tools included the diffusion gradient chamber, microsensors and microbiosensors, and the laser diffusion capillary assay. A mathematical model of the diffusion gradient chamber was also developed that allowed two cell balances, two chemoattractant compounds, and a nutrient compound to be modeled simultaneously. The mathematical model allowed predictions of chemotactic responses to be made without having to run time consuming experiments. The model was validated by comparing its predictions to experimental data from DGC experiments. The validation experiments involved *Escherichia coli* responding to gradients of aspartate.

The applications that were identified that could benefit from the chemotactic response were microbial competition, selection of mutants, and bioremediation. The mathematical model was used to explore the effects of several modeling parameters on the predicted competitive outcomes. Chemotaxis was shown to give a competitive advantage to a bacterial population in certain cases in non-mixed environments. Several techniques to analyze the modeling predictions were developed, including the dynamic competition factor. For the selection of mutants topic, the chemotactic response was exploited to draw mutants of a bacterial species into an area in which the parent strain could not grow well. Again, the mathematical model was used to more fully explore the inhibition and chemotaxis interaction. The model predicted that the selection method

should wo

bioremed

found to

obstacles

Prelimina

were per

between

as an aqu

should work well in various scenarios. Finally, the response of an important bacterium for bioremediation, *Pseudomonas stutzeri* strain KC, was studied. *Pseudomonas* KC was found to be chemotactic to nitrate and acetate. The reaction of the chemotactic wave to obstacles also studied as a first step in examining chemotaxis through porous media. Preliminary competition experiments between *Pseudomonas* KC and *Escherichia coli* were performed. Further experiments could lead to a better understanding of the interplay between chemotaxis and competition in environments important to bioremediation, such as an aquifer.

APPENDICES

APPENDIX A

APPEN

Miscella

section

chemota

organis

require

aliquot

56A, g

gel. Th

be attr

oxyger

and 1

define

aspart

multi

howe

the ch

after

resea

APPENDIX A

Miscellaneous chemotaxis experiments

Appendix A includes selected experimental results that do not directly fit into a section of the thesis, but that are interesting or informative in the context of studying the chemotactic response.

Figure 56 shows three swarm plate experiments. In each of these experiments, the organism used was *E. coli* HCB 33. M63 minimal medium, supplemented with the required amino acids and streptomycin (see Section 4.2.3), was the base medium. A 10 µl aliquot of actively growing culture was inoculated into the center of each plate. In Figure 56A, glucose was initially present at a constant concentration of 3 mM throughout the gel. The image was taken 20 hours after inoculation. The multiple waves or bands could be attributed to chemotaxis to other chemoattractants, one of which would likely be oxygen. Figure 56B, also 20 hours after inoculation, shows the response to 1 mM glucose and 1 mM aspartate. In this case, a very uniform single wave formed, that is sharply defined on the outer edge. Since at least two known chemoattractants (glucose and aspartate) were present, and possibly a third with oxygen, it might be assumed that multiple bands would occur. This did not occur experimentally in this particular case, however. It is possible that the consumption of the two chemoattractants was similar, and the chemotactic waves overlapped closely. The image in Figure 56C was taken 42 hours after inoculation. This swarm plate contained 1 mM glucose and 3 mM glycerol. Some research has reported that glycerol can be an inhibitor of chemotaxis (see Zhulin *et al.*,

1997 fo

conditio

Patter

format

1995;

symm

or *S.*

substa

and g

strept

inocu

M63,

1997 for a review), which might explain the slower swarming under the glycerol conditions.

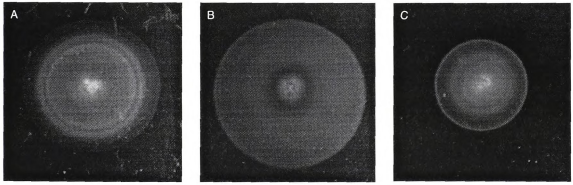


Figure 56. Swarm plates with various chemoattractants.

Pattern formation

An interesting phenomenon that is thought to be related to chemotaxis is the formation of patterns in bacterial populations (Agladze *et al.*, 1993; Woodward *et al.*, 1995; Budrene and Berg, 1995). In these studies, swarm plates were used to study the symmetrical patterns, sometimes described as looking like snowflakes, formed by *E. coli* or *S. typhimurium*. The current hypothesis is that the bacteria exude a chemoattractant substance, which causes conglomeration of bacteria in patterns.

A DGC experiment was set-up to test the response of *E. coli* HCB 33 to aspartate and glucose. M63 minimal medium was used with the required amino acids and streptomycin. The source concentrations were 3 mM aspartate and 5 mM glucose. The inoculation culture was grown in a shake flask with 10 mM glucose and supplemented M63. The gradients were allowed to initiate for 5 ½ hours prior to inoculation. A 10 ml

aliquot

the cent

until 16

DGC,

concen

hours)

produ

Howe

the c

resea

Move

the c

aliquot of cells that had been growing in the shake flask for 22 hours was inoculated in the center of the chamber. Because of a problem with the camera, no images were taken until 16 hours after inoculation. At that time, a very unique pattern had formed in the DGC, as shown in Figure 57A. This pattern includes spots indicating local high concentrations of *E. coli*. The pattern continued to evolve, as shown in Figure 57B (18 hours) and Figure 57C (20 hours). Attempts to reproduce these results failed.

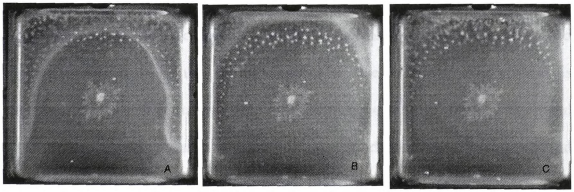


Figure 57. Pattern formation in the DGC.

A version of the mathematical was developed to incorporate a chemoattractant produced by the bacteria. This model was tested, and appeared to be working properly. However, no patterns were observed in the simulations. Work may be necessary to find the correct parameter combinations that allow for pattern formation, if this line of research is continued.

Movement around objects

A group of swarm plate experiments was performed to analyze the movement of the chemotactic wave of *Pseudomonas* KC around objects. The experimental conditions

are des

object

channe

around

was th

the co

cells w

gel sur

hypoth

approx

be har

in cap

capilla

are described in Section 7.1.3. In Figure 58, three objects were placed in the gel. The object in the lower right-hand section of the gel was a tubing connector with an open channel through it. It appeared that the cells swam more quickly through the opening than around the object. Two hypotheses were formed to explain this phenomenon. The first was that the tubing connector was full of water remaining from the autoclave cycle. When the connector was immersed in the gel, the gel may not have displaced the water. The cells would then encounter less resistance to swimming in the water as compared to the gel surrounding the connector, and could move through the connector faster. The second hypothesis was that, in the narrow confines of the connector (the channel is approximately 0.4 mm in diameter), the cells tumbling would be confined, and it would be harder for the cells to change swimming direction. Evidence of this has been observed in capillary assays (Liu and Papadopoulos, 1995; Liu *et al.*, 1996), although their capillaries had much smaller diameters.

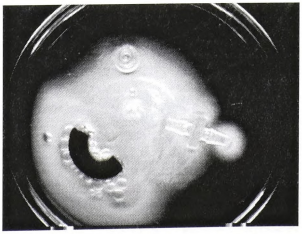


Figure 58. Chemotaxis through a hollow object.

APPENDIX B

APPE

The Alt

and fro

tridiago

time st

y_i

The

4) is

APPENDIX B

The Alternating-Direction Implicit Method

This discussion and the figures have been adapted from Chapra and Canale (1988) and from Carnahan *et al.* (1969). The alternating-direction implicit (ADI) scheme uses tridiagonal matrices to solve parabolic equations in two or more spatial dimensions. Each time step is broken into two steps, as shown in .

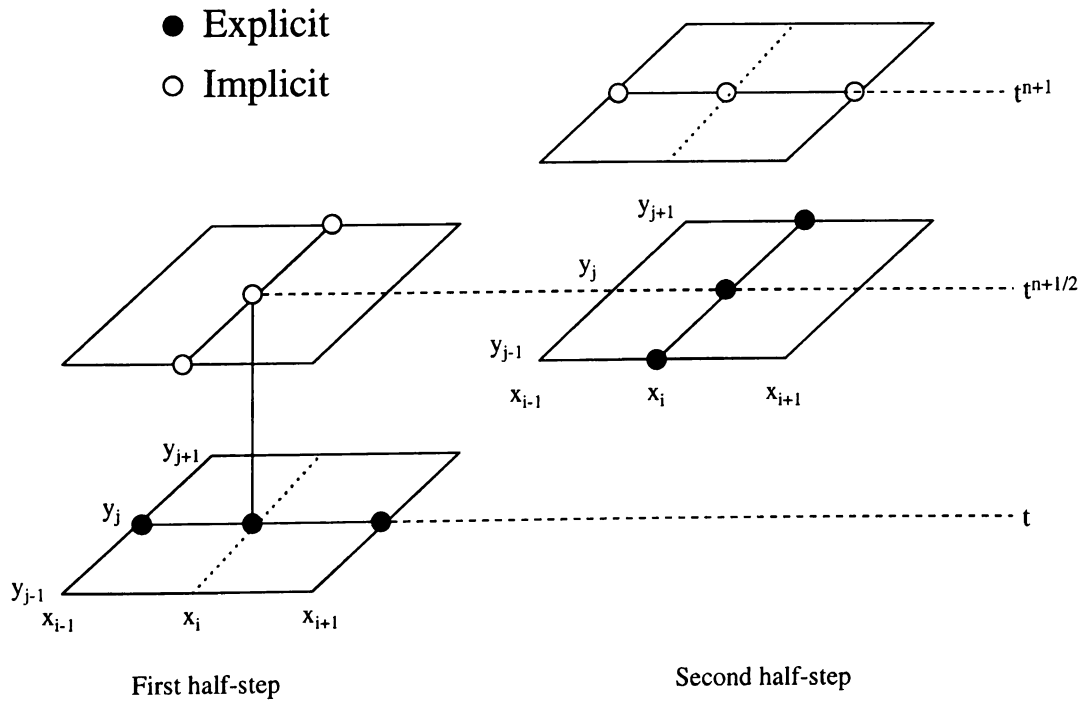


Figure 59. The alternating-direction implicit ethod.

The first step to solve the simple two-dimensional diffusion equation given in Equation (4) is written

such th

implici

Equati

where

tridiag

approx

so tha

explic

Again

$$\frac{S_{i,j}^{n+1/2} - S_{i,j}^{n}}{\Delta t / 2} = D_S \left(\frac{S_{i+1,j}^{n} - 2S_{i,j}^{n} + S_{i-1,j}^{n}}{(\Delta x)^2} + \frac{S_{i,j+1}^{n+1/2} - 2S_{i,j}^{n+1/2} + S_{i,j-1}^{n+1/2}}{(\Delta y)^2} \right) \quad (53)$$

such that the approximation of $\partial^2 S/\partial x^2$ is explicit, while the approximation of $\partial^2 S/\partial y^2$ is implicit. For the model presented in Section 4.2, the grid was square, and thus $\Delta x=\Delta y$, so Equation (53) may be rearranged as

$$-\lambda S_{i,j-1}^{n+1/2} + 2(1+\lambda) S_{i,j}^{n+1/2} - \lambda S_{i,j+1}^{n+1/2} = \lambda S_{i-1,j}^{n} + 2(1-\lambda) S_{i,j}^{n} + \lambda S_{i+1,j}^{n} \quad (54)$$

where $\lambda=D_S\Delta t/(\Delta x)^2$. When Equation (54) is written for each point on the grid, a tridiagonal set of simultaneous equations results. For the second half step, Equation (4) is approximated by

$$\frac{S_{i,j}^{n+1} - S_{i,j}^{n+1/2}}{\Delta t / 2} = D_S \left(\frac{S_{i+1,j}^{n+1} - 2S_{i,j}^{n+1} + S_{i-1,j}^{n+1}}{(\Delta x)^2} + \frac{S_{i,j+1}^{n+1/2} - 2S_{i,j}^{n+1/2} + S_{i,j-1}^{n+1/2}}{(\Delta y)^2} \right) \quad (55)$$

so that now the approximation of $\partial^2 S/\partial x^2$ is implicit, and the approximation of $\partial^2 S/\partial y^2$ is explicit. Equation (55) can be rearranged to yield

$$-\lambda S_{i-1,j}^{n+1} + 2(1+\lambda) S_{i,j}^{n+1} - \lambda S_{i+1,j}^{n+1} = \lambda S_{i,j-1}^{n+1/2} + 2(1-\lambda) S_{i,j}^{n+1/2} + \lambda S_{i,j+1}^{n+1/2} \quad (56)$$

Again, when Equation (56) is written for the entire grid, a tridiagonal matrix results.

APPENDIX C

APPE

Instruc

Th

(ua and

chamb

that the

- Th
- Fil
- Ea

On a S

1. M

T

diffus

T

spaci

the g

grow

expe

outp

APPENDIX C

Instructions for FORTRAN model

The program currently solves five coupled PDE'S. These include two cell balances (ua and ub), two chemoattractants (s and q) and a growth nutrient (h). The sides of the chamber are identified by compass directions (n,s,e,w). For the version of the program that these instructions pertain to, the chemoattractant q should be thought of as oxygen.

- The unit basis for all variables is mass=grams, length=centimeters, time=hours.
- Filenames are *italicized,* while variable names are in "quotes".
- Each input file is formatted, so be sure to keep numbers in the proper positions.

On a Sun or HP:

1. Modify the *param file.*

This file contains the parameters of the model, including chemotactic sensitivities, diffusion coefficients, mass transport coefficients, etc.

The param file also contains information about the simulation run-time and grid spacing. The value of "m" sets the grid size (m x m matrix). "Timeinit" sets the time that the gradients establish before cell inoculation. "Timeinoc" sets the time that the cells grow in the chamber. "Timeinit" plus "timeinoc" gives the total simulated time of the experiment. Param also contains instructions for the simulation that specify a movie output or single graph output.

A list of variables contained in the *param* file and their definition is included at the end of this section.

2. Modify the *init* file.

This file sets the initial conditions, including initial cell concentration, chemoattractant boundary conditions, and nutrient boundary conditions. The parameters beginning with r (rsn, for example) correspond to the reservoir concentrations. In the example, rsn is reservoir, s-chemoattractant, north side. The initial concentration in the gel is also set in *init.* For example, "stini" sets the concentration of s in the gel at the beginning of the simulation. The nutrient concentration is usually set equal to the nutrient concentration in the four reservoirs. For a chemoattractant that will make a gradient, the initial concentration in the gel is usually 0. The cell concentration is initialized by *"uaint"* and *"ubini"*. Note that these are not the initial concentrations, but are a parameter for the cell initialization subroutine.

3. Modify the "time" file.

"Numti" tells the program how many matrices to print. For example, for "numti" =04, four graphs would be produced at times specified by "timel ",

"time2", etc. The times are time after inoculation. Note that the variable "timeinoc" in *param* specifies how long the simulation will run, so any time longer than "timeinoc" will not print a graph.

4. If necessary, modify the FORTRAN program itself.

Subroutine uaO sets the initial cell distribution. Currently, an exponential peak is used.

The subroutines f and fs set the nutrient and chemoattractant uptake functions. **Note: both f and fs must have identical functionalities.** If one is changed, the other must also be changed. Both functions correspond to the same term in the equations, but are slightly different because of the finite differencing algorithm.

The subroutine chemofl sets the functionality of the chemotaxis term.

Other subroutines set the windows of the reservoirs, calculate velocities, set boundary conditions, and calculate certain other values.

5. To run the program, a Sun computer must be used. If an HP is the current computer, open a window to a Sun.

To run the program in batch mode, type:

batch

An at> prompt should appear. Type the following commands (↵ = return, underlined words should be replaced with appropriate name, and bold indicates pressing control and D at the same time):

at> f77 filename.f ↵

at> a.out ↵

at> mailx -s "subject" mail-address ↵

at> **ctrl-D**

The simulation is complete when mail bearing the *"subject"* is received. At this time, files called lcella.m, lcellb.m, latts.m, lattq.m, and lnut.m should be in the directory from which the program ran. These files are formatted for Matlab.

Matlab

1. A

m

2. A

c

3. Ty

exam

4. T

For e

shoul

5. Tv

Thre

plot,

Over

will

High

give

Mat

help

Matlab instructions:

1. At a Sun or HP, start Matlab by typing:

   matlab

2. At the Matlab prompt, switch to the directory containing the simulation files:

   cd path\directory

3. Type the name of the file you wish to work with, without the m extension. For example, to look at the cell Population a matrix, type lcella

4. To check if the matrices have been properly loaded, type who

For example, if "numti" = 4, and if individual graphs were specified, then there should be 4 matrices named ual, ua2, ua3, and ua4.

5. Two types of graphs are useful for viewing:

Three dimensional plots are created with the surf command. surf(ual) will yield a 3-D plot, with Matlab default view angle and axis.

Overhead view type plots are created with the appmult command.

   appmult(ual,3,'pcolor')

will give a good overhead plot. The value 3 can be changed to give better or worse plots. Higher numbers give smoother graphs, but take longer to view and print. Lower numbers give coarser plots, but are faster.

Matlab has a fairly good help feature. Typing help will list general help topics, or type help command to get specific help.

APPENDIX D

APP

The I

can b

Popu

APPENDIX D

The FORTRAN model

This is the FORTRAN model used to solve the competition model. This model can be reduced to the one-cell population model simply by setting the initial condition for Population B to zero in the initialization file (see Appendix E).

```
implicit double precision (a-h,o-z)
C
    parameter (nmax=100,kmax=100)
C
    dimension x(0:nmax),y(0:nmax)
    dimension ua1(0:nmax,0:nmax),ua2(0:nmax,0:nmax)
    dimension ub1(0:nmax,0:nmax),ub2(0:nmax,0:nmax)
    dimension H1(0:nmax,0:nmax),H2(0:nmax,0:nmax)
    dimension S1(0:nmax,0:nmax),S2(0:nmax,0:nmax)
    dimension q1(0:nmax,0:nmax),q2(0:nmax,0:nmax)
    dimension aa(0:nmax),bb(0:nmax),cc(0:nmax)
    dimension funua(0:nmax),funs(0:nmax),funh(0:nmax)
    dimension rhsua(0:nmax),rhss(0:nmax),rhsh(0:nmax)
    dimension funub(0:nmax)
    dimension rhsq(0:nmax),funq(0:nmax),rhsub(0:nmax)
    dimension uua(0:nmax),us(0:nmax),uh(0:nmax),uq(0:nmax),uub(0:nmax)
    dimension mcount3(0:kmax), time(0:kmax)
    dimension velm(0:500),vell(0:500)
    dimension distm(0:500),distl(0:500),timm(0:500),timl(0:500)
C
    open(16,file='lcella.m')
    open(32,file='lcellb.m')
    open(17,file='latts.m')
    open(23,file='lattq.m')
    open(18,file='lnut.m')
    open(19,file='time')
    open(24,file='init')
    open(22,file='param')
    open(30,file='distmax')
    open(31,file='distlim')
```

```
C
      read(22,101)m
      read(22,103)timinit
      read(22,103)timinoc
      read(22,102)numcel
      read(22,102)numvel
      read(22,102)kstep
      read(22,100)dmin
      read(22,102)movc
      read(22,100)R
      read(22,100)ayoff
      read(22,100)axoff
      read(22,100)byoff
      read(22,100)bxoff
      read(22,100)awid
      read(22,100)bwid
      read(22,100)Dh
      read(22,100)thn
      read(22,100)ths
      read(22,100)the
      read(22,100)thw
      read(22,100)Ds
      read(22,100)tsn
      read(22,100)tss
      read(22,100)tse
      read(22,100)tsw
      read(22,100)Dacs
      read(22,100)Dbcs
      read(22,100)DKas
      read(22,100)DKbs
      read(22,100)cas
      read(22,100)vas
      read(22,100)Yas
      read(22,100)cbs
      read(22,100)vbs
      read(22,100)Ybs
      read(22,100)Dq
      read(22,100)tqn
      read(22,100)tqs
      read(22,100)tqe
      read(22,100)tqw
      read(22,100)Dacq
      read(22,100)Dbcq
      read(22,100)DKaq
```

```
      read(22,100)DKbq
      read(22,100)caq
      read(22,100)vaq
      read(22,100)Yaq
      read(22,100)cbq
      read(22,100)vbq
      read(22,100)Ybq
      read(22,100)Dua
      read(22,100)va
      read(22,100)ca
      read(22,100)Ya
      read(22,100)Dub
      read(22,100)vb
      read(22,100)cb
      read(22,100)Yb
      read(24,100)uaini
      read(24,100)ubini
      read(24,100)rsn
      read(24,100)rss
      read(24,100)rse
      read(24,100)rsw
      read(24,100)rqn
      read(24,100)rqs
      read(24,100)rqe
      read(24,100)rqw
      read(24,100)rhn
      read(24,100)rhs
      read(24,100)rhe
      read(24,100)rhw
      read(24,100)stini
      read(24,100)qtini
      read(24,100)htini
  100 format(6x,d13.10)
  101 format(4x,i4)
  102 format(7x,i7)
  103 format(8x,d13.10)
      if(movc.eq.2)then
        write(16,*)'ua=['
        write(32,*)'ub=['
        write(17,*)'s=['
        write(23,*)'q=['
        write(18,*)'h=['
      endif
C
```

```
      hr=2.0d0*R/m
      ht=0.15625d0*hr*hr
      mcount1=timinit/ht+1
      mcount2=timinoc/ht+1
      write(*,*)'hr =',hr
      write(*,*)'ht =',ht
      write(*,*)'mcount1=',mcount1
      write(*,*)'mcount2=',mcount2
C
      read(19,201)numti
      do 555 i=1,numti
        read(19,202)time(i)
        mcount3(i)=time(i)/ht
        time(i)=mcount3(i)*ht
        write(*,*)'Real time',i,'=',time(i)
 555  continue
 201  format(6x,i2)
 202  format(6x,d6.2)
C  Calculate values of hr and ht
C
C set up the initial gradient of attractant and nutrient
C
C set up the initial chemoattractant concentration in chamber
      do 10 i=0,m
        do 5 j=0,m
         S1(i,j)=stini
         q1(i,j)=qtini
            H1(i,j)=htini
         ua1(i,j)=0.0d0
         ub1(i,j)=0.0d0
 5     continue
 10    continue
C
       D0=Dh*(ht/2.0d0)/(hr**2)
       D1=Ds*(ht/2.0d0)/(hr**2)
       D2=Dua*(ht/2.0d0)/(hr**2)
       D3=Dacs*(ht/2.0d0)/(hr**2)
       D4=Dq*(ht/2.0d0)/(hr**2)
       D5=Dacq*(ht/2.0d0)/(hr**2)
       D6=Dub*(ht/2.0d0)/(hr**2)
       D7=Dbcs*(ht/2.0d0)/(hr**2)
       D8=Dbcq*(ht/2.0d0)/(hr**2)
C
C solving the ode
```

```
      do 1000 k=1,mcount1
C
C take care of boundary points:
C
          j=0
        do 20 i=0,m
          t=tr(i,hr,tss,Ds)
            call rhbound(m,S1(i,0),S1(i,1),rss,D1,hr,t,rhss(i))
          funs(i)=1.0d0
          t=tr(i,hr,tqs,Dq)
            call rhbound(m,q1(i,0),q1(i,1),rqs,D4,hr,t,rhsq(i))
          funq(i)=1.0d0
          t=tr(i,hr,ths,Dh)
            call rhbound(m,H1(i,0),H1(i,1),rhs,D0,hr,t,rhsh(i))
          funh(i)=1.0d0
 20     continue
C
          call PDMTRIX(m,D1,funs,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhss,us)
          call PDMTRIX(m,D4,funq,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhsq,uq)
          call PDMTRIX(m,D0,funh,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhsh,uh)
          do 25 i=0,m
            S2(i,0)=us(i)
            q2(i,0)=uq(i)
            h2(i,0)=uh(i)
 25       continue
C
          j=m
        do 30 i=0,m
          t=tr(i,hr,tsn,Ds)
            call rhbound(m,S1(i,m),S1(i,m-1),rsn,D1,hr,t,rhss(i))
          funs(i)=1.0d0
          t=tr(i,hr,tqn,Dq)
            call rhbound(m,q1(i,m),q1(i,m-1),rqn,D4,hr,t,rhsq(i))
          funq(i)=1.0d0
          t=tr(i,hr,thn,Dh)
            call rhbound(m,H1(i,m),H1(i,m-1),rhn,D0,hr,t,rhsh(i))
          funh(i)=1.0d0
 30     continue
C
          call PDMTRIX(m,D1,funs,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhss,us)
```

```
      call PDMTRIX(m,D4,funq,aa,bb,cc)
      call TRISOLV(m,aa,bb,cc,rhsq,uq)
      call PDMTRIX(m,D0,funh,aa,bb,cc)
      call TRISOLV(m,aa,bb,cc,rhsh,uh)
      do 35 i=0,m
        S2(i,m)=us(i)
        q2(i,m)=uq(i)
        h2(i,m)=uh(i)
35    continue

C take care of interior points:
C
      do 50 j=1,m-1
         t=tr(j,hr,tsw,Ds)
         call rh2(m,S1(0,j),S1(0,j-1),S1(0,j+1),rsw,D1,hr,t,rhss(0))
       funs(0)=1.0d0+D1*hr*tr(j,hr,tsw,Ds)
         t=tr(j,hr,tqw,Dq)
         call rh2(m,q1(0,j),q1(0,j-1),q1(0,j+1),rqw,D4,hr,t,rhsq(0))
       funq(0)=1.0d0+D4*hr*tr(j,hr,tqw,Dq)
         t=tr(j,hr,thw,Dh)
         call rh2(m,h1(0,j),h1(0,j-1),h1(0,j+1),rhw,D0,hr,t,rhsh(0))
       funh(0)=1.0d0+D0*hr*tr(j,hr,thw,Dh)
C
         t=tr(j,hr,tse,Ds)
         call rh2(m,S1(m,j),S1(m,j-1),S1(m,j+1),rse,D1,hr,t,rhss(m))
       funs(m)=1.0d0+D1*hr*tr(j,hr,tse,Ds)
         t=tr(j,hr,tqe,Dq)
         call rh2(m,q1(m,j),q1(m,j-1),q1(m,j+1),rqe,D4,hr,t,rhsq(m))
       funq(m)=1.0d0+D4*hr*tr(j,hr,tqe,Dq)
         t=tr(j,hr,the,Dh)
         call rh2(m,h1(m,j),h1(m,j-1),h1(m,j+1),rhe,D0,hr,t,rhsh(m))
       funh(m)=1.0d0+D0*hr*tr(j,hr,the,Dh)
C
         do 40 i=1,m-1
        rhss(i)=S1(i,j)+D1*(S1(i,j+1)-2.0d0*S1(i,j)+S1(i,j-1))
        funs(i)=1.0d0
        rhsq(i)=q1(i,j)+D4*(q1(i,j+1)-2.0d0*q1(i,j)+q1(i,j-1))
        funq(i)=1.0d0
        rhsh(i)=h1(i,j)+D0*(h1(i,j+1)-2.0d0*h1(i,j)+h1(i,j-1))
        funh(i)=1.0d0
40      continue
         call PDMTRIX(m,D1,funs,aa,bb,cc)
         call TRISOLV(m,aa,bb,cc,rhss,us)
         call PDMTRIX(m,D4,funq,aa,bb,cc)
```

```
            call TRISOLV(m,aa,bb,cc,rhsq,uq)
            call PDMTRIX(m,D0,funh,aa,bb,cc)
            call TRISOLV(m,aa,bb,cc,rhsh,uh)
            do 45 i=0,m
              S2(i,j)=us(i)
              q2(i,j)=uq(i)
              h2(i,j)=uh(i)
 45       continue
 50       continue
C
C implicit in y direction:
C
            do 55 i=0,m
            do 56 j=0,m
              S1(i,j)=S2(i,j)
              q1(i,j)=q2(i,j)
              h1(i,j)=h2(i,j)
 56         continue
 55         continue
C
C take care of boundary points:
C
      i=0
        do 60 j=0,m
            t=tr(j,hr,tsw,Ds)
            call rhbound(m,S1(0,j),S1(1,j),rsw,D1,hr,t,rhss(j))
          funs(0)=1.0d0
            t=tr(j,hr,tqw,Dq)
            call rhbound(m,q1(0,j),q1(1,j),rqw,D4,hr,t,rhsq(j))
          funq(0)=1.0d0
            t=tr(j,hr,thw,Dh)
            call rhbound(m,H1(0,j),H1(1,j),rhw,D0,hr,t,rhsh(j))
          funh(0)=1.0d0
 60       continue
C
            call PDMTRIX(m,D1,funs,aa,bb,cc)
            call TRISOLV(m,aa,bb,cc,rhss,us)
            call PDMTRIX(m,D4,funq,aa,bb,cc)
            call TRISOLV(m,aa,bb,cc,rhsq,uq)
            call PDMTRIX(m,D0,funh,aa,bb,cc)
            call TRISOLV(m,aa,bb,cc,rhsh,uh)
            do 65 j=0,m
              S2(0,j)=us(j)
              q2(0,j)=uq(j)
```

```
            h2(0,j)=uh(j)
65        continue
C
        i=m
          do 70 j=0,m
            t=tr(j,hr,tse,Ds)
            call rhbound(m,S1(m,j),S1(m-1,j),rse,D1,hr,t,rhss(j))
          funs(0)=1.0d0
            t=tr(i,hr,tqe,Dq)
            call rhbound(m,q1(m,j),q1(m-1,j),rqe,D4,hr,t,rhsq(j))
          funq(0)=1.0d0
            t=tr(i,hr,the,Dh)
            call rhbound(m,H1(m,j),H1(m-1,j),rhe,D0,hr,t,rhsh(j))
          funh(0)=1.0d0
70        continue
C
            call PDMTRIX(m,D1,funs,aa,bb,cc)
            call TRISOLV(m,aa,bb,cc,rhss,us)
            call PDMTRIX(m,D4,funq,aa,bb,cc)
            call TRISOLV(m,aa,bb,cc,rhsq,uq)
            call PDMTRIX(m,D0,funh,aa,bb,cc)
            call TRISOLV(m,aa,bb,cc,rhsh,uh)
            do 75 j=0,m
              S2(m,j)=us(j)
              q2(m,j)=uq(j)
              h2(m,j)=uh(j)
75          continue

C take care of interior points:
C
        do 90 i=1,m-1
C
            t=tr(i,hr,tss,Ds)
            call rh2(m,S1(i,0),S1(i-1,0),S1(i+1,0),rss,D1,hr,t,rhss(0))
          funs(0)=1.0d0+D1*hr*tr(i,hr,tss,Ds)
            t=tr(i,hr,tqs,Dq)
            call rh2(m,q1(i,0),q1(i-1,0),q1(i+1,0),rqs,D4,hr,t,rhsq(0))
          funq(0)=1.0d0+D4*hr*tr(i,hr,tqs,Dq)
            t=tr(i,hr,ths,Dh)
            call rh2(m,h1(i,0),h1(i-1,0),h1(i-1,0),rhs,D0,hr,t,rhsh(0))
          funh(0)=1.0d0+D0*hr*tr(i,hr,ths,Dh)
C
            t=tr(i,hr,tsn,Ds)
            call rh2(m,S1(i,m),S1(i-1,m),S1(i+1,m),rsn,D1,hr,t,rhss(m))
```

```
      funs(m)=1.0d0+D1*hr*tr(i,hr,tsn,Ds)
        t=tr(i,hr,tqn,Dq)
        call rh2(m,q1(i,m),q1(i-1,m),q1(i+1,m),rqn,D4,hr,t,rhsq(m))
      funq(m)=1.0d0+D4*hr*tr(i,hr,tqn,Dq)
        t=tr(i,hr,thn,Dh)
        call rh2(m,h1(i,m),h1(i-1,m),h1(i-1,m),rhn,D0,hr,t,rhsh(m))
      funh(m)=1.0d0+D0*hr*tr(i,hr,thn,Dh)
C
        do 80 j=1,m-1
      rhss(j)=S1(i,j)+D1*(S1(i+1,j)-2.0d0*S1(i,j)+S1(i-1,j))
      funs(j)=1.0d0
      rhsq(j)=q1(i,j)+D4*(q1(i+1,j)-2.0d0*q1(i,j)+q1(i-1,j))
      funq(j)=1.0d0
      rhsh(j)=h1(i,j)+D0*(h1(i+1,j)-2.0d0*h1(i,j)+h1(i-1,j))
      funh(j)=1.0d0
 80    continue
C
        call PDMTRIX(m,D1,funs,aa,bb,cc)
        call TRISOLV(m,aa,bb,cc,rhss,us)
        call PDMTRIX(m,D4,funq,aa,bb,cc)
        call TRISOLV(m,aa,bb,cc,rhsq,uq)
        call PDMTRIX(m,D0,funh,aa,bb,cc)
        call TRISOLV(m,aa,bb,cc,rhsh,uh)
        do 85 j=0,m
          S2(i,j)=us(j)
          q2(i,j)=uq(j)
          h2(i,j)=uh(j)
 85      continue
 90     continue
C
      do 95 j=0,m
      do 96 i=0,m
        S1(i,j)=S2(i,j)
        q1(i,j)=q2(i,j)
        h1(i,j)=h2(i,j)
 96     continue
 95     continue
C
 1000   continue
C
C  set up the initial cell condition (inoculation)
C
C
     axoff=axoff*hr
```

```
      ayoff=ayoff*hr
      bxoff=bxoff*hr
      byoff=byoff*hr
      do 1010 i=0,m
        x(i)=hr*i-R
      do 1005 j=0,m
        y(j)=hr*j-R
        ua1(i,j)=ua0(x(i),y(j),uaini,ayoff,axoff,awid)
        ub1(i,j)=ub0(x(i),y(j),ubini,byoff,bxoff,bwid)
 1005   continue
 1010   continue
C
C solving the ode
C
      kt0=-1
      kt20=-1
      mcount4=1
      lvel=1
      mvel=1
      distm(0)=0.
      distl(0)=0.
      kvel=1
C*****************************************************************
*******
      do 2000 k=1,mcount2+1
C
C track the steps
C
        kt1=k/mcount2
C
C       if(kt1.eq.1) then
C         kt0=kt1
      if(k.eq.mcount3(mcount4)) then
C
C print out the results
        if(movc.eq.1)then
          if(mcount4.lt.10)then
            write(16,360)'ua',mcount4,' = ['
            write(32,360)'ub',mcount4,' = ['
            write(17,360)'s',mcount4,' = ['
            write(23,360)'q',mcount4,' = ['
            write(18,360)'h',mcount4,' = ['
  360 format(a2,i1,a4)
          elseif(mcount4.gt.9)then
```

```
          write(16,361)'ua',mcount4,' = ['
          write(32,361)'ub',mcount4,' = ['
          write(17,361)'s',mcount4,' = ['
          write(23,361)'q',mcount4,' = ['
          write(18,361)'h',mcount4,' = ['
  361 format(a2,i2,a4)
        endif
      do 1210 j=0,m
        write(16,575)(ua1(i,j),i=0,m)
        write(32,575)(ub1(i,j),i=0,m)
        write(17,575)(S1(i,j),i=0,m)
        write(23,575)(q1(i,j),i=0,m)
        write(18,575)(h1(i,j),i=0,m)
 1210    continue
        write(16,*)'];'
        write(32,*)'];'
        write(17,*)'];'
        write(23,*)'];'
        write(18,*)'];'
 575      format(2x,101(e10.4,2x))
        mcount4=mcount4+1
C

      elseif(movc.eq.2)then
C
C print out the results

        do 1211 j=0,m
          write(16,575)(ua1(i,j),i=0,m)
          write(32,575)(ub1(i,j),i=0,m)
          write(17,575)(S1(i,j),i=0,m)
          write(23,575)(q1(i,j),i=0,m)
          write(18,575)(h1(i,j),i=0,m)
 1211     continue
         mcount4=mcount4+1
C
       else
       endif
      endif
C
C kt2: index for status of the program
C
       kt2=k/(mcount2/10)
       if(kt2.gt.kt20) then
```

```
          kt20=kt2
          write(6,*) 'kt2=',kt2
          else
            endif
C
C implicit in x direction:
C
C take care of boundary points:
C
C
      rhss(0)=s1(0,0)+D1*(S1(0,1)-S1(0,0))
      funs(0)=1.0d0+(ht/2.0d0)*fs(cas,vas,s1(0,0))*ua1(0,0)/Yas
      funs(0)=funs(0)+(ht/2.0d0)*fs(cbs,vbs,s1(0,0))*ub1(0,0)/Ybs
      rhsq(0)=q1(0,0)+D4*(q1(0,1)-q1(0,0))
      funq(0)=1.0d0+(ht/2.0d0)*fs(caq,vaq,q1(0,0))*ua1(0,0)/Yaq
      funq(0)=funq(0)+(ht/2.0d0)*fs(cbq,vbq,q1(0,0))*ub1(0,0)/Ybq
      rhsh(0)=h1(0,0)+D0*(h1(0,1)-h1(0,0))
      funh(0)=1.0d0+(ht/2.0d0)*fs(ca,va,h1(0,0))*ua1(0,0)/Ya
      funh(0)=funh(0)+(ht/2.0d0)*fs(cb,vb,h1(0,0))*ub1(0,0)/Yb
      rhss(m)=S1(m,0)+D1*(s1(m,1)-s1(m,0))
      funs(m)=1.0d0+(ht/2.0d0)*fs(cas,vas,s1(m,0))*ua1(m,0)/Yas
      funs(m)=funs(m)+(ht/2.0d0)*fs(cbs,vbs,s1(m,0))*ub1(m,0)/Ybs
      rhsq(m)=q1(m,0)+D4*(q1(m,1)-q1(m,0))
      funq(m)=1.0d0+(ht/2.0d0)*fs(caq,vas,q1(m,0))*ua1(m,0)/Yaq
      funq(m)=funq(m)+(ht/2.0d0)*fs(cbq,vbs,q1(m,0))*ub1(m,0)/Ybq
      rhsh(m)=h1(m,0)+D0*(h1(m,1)-h1(m,0))
      funh(m)=1.0d0+(ht/2.0d0)*fs(ca,va,h1(m,0))*ua1(m,0)/Ya
      funh(m)=funh(m)+(ht/2.0d0)*fs(cb,vb,h1(m,0))*ub1(m,0)/Yb
C
      j=0
C Cell group a
            a12=0.5d0*chemof1(DKas,S1(1,0))*ua1(1,0)
            a12=a12+0.5d0*chemof1(DKas,S1(0,0))*ua1(0,0)
            b12=0.5d0*chemof1(DKas,S1(0,1))*ua1(0,1)
            b12=b12+0.5d0*chemof1(DKas,S1(0,0))*ua1(0,0)
            c12=0.5d0*chemof1(DKaq,q1(1,0))*ua1(1,0)
            c12=c12+0.5d0*chemof1(DKaq,q1(0,0))*ua1(0,0)
            d12=0.5d0*chemof1(DKaq,q1(0,1))*ua1(0,1)
            d12=d12+0.5d0*chemof1(DKaq,q1(0,0))*ua1(0,0)
      rhsua(0)=ua1(0,0)+D2*(ua1(0,1)-ua1(0,0))
      rhsua(0)=rhsua(0)+(ht/2.0d0)*f(ca,va,h1(0,0))*ua1(0,0)
              rhsua(0)=rhsua(0)-D3*a12*(s1(1,0)-s1(0,0))
                rhsua(0)=rhsua(0)-D3*b12*(S1(0,1)-s1(0,0))
              rhsua(0)=rhsua(0)-D5*c12*(q1(1,0)-q1(0,0))
```

```
          rhsua(0)=rhsua(0)-D5*d12*(q1(0,1)-q1(0,0))
        funua(0)=1.0d0
C
              a21=0.5d0*chemof1(DKas,S1(m,0))*ua1(m,0)
              a21=a21+0.5d0*chemof1(DKas,S1(m-1,0))*ua1(m-1,0)
              b12=0.5d0*chemof1(DKas,S1(m,1))*ua1(m,1)
              b12=b12+0.5d0*chemof1(DKas,S1(m,0))*ua1(m,0)
              c21=0.5d0*chemof1(DKaq,q1(m,0))*ua1(m,0)
              c21=c21+0.5d0*chemof1(DKaq,q1(m-1,0))*ua1(m-1,0)
              d12=0.5d0*chemof1(DKaq,q1(m,1))*ua1(m,1)
              d12=d12+0.5d0*chemof1(DKaq,q1(m,0))*ua1(m,0)
       rhsua(m)=ua1(m,0)+D2*(ua1(m,1)-ua1(m,0))
         rhsua(m)=rhsua(m)+(ht/2.0d0)*f(ca,va,h1(m,0))*ua1(m,0)
              rhsua(m)=rhsua(m)-D3*a21*(s1(m-1,0)-s1(m,0))
                rhsua(m)=rhsua(m)-D3*b12*(s1(m,1)-s1(m,0))
              rhsua(m)=rhsua(m)-D5*c21*(q1(m-1,0)-q1(m,0))
                rhsua(m)=rhsua(m)-D5*d12*(q1(m,1)-q1(m,0))
       funua(m)=1.0d0
C Cell group b
              a12=0.5d0*chemof1(DKbs,S1(1,0))*ub1(1,0)
              a12=a12+0.5d0*chemof1(DKbs,S1(0,0))*ub1(0,0)
              b12=0.5d0*chemof1(DKbs,S1(0,1))*ub1(0,1)
              b12=b12+0.5d0*chemof1(DKbs,S1(0,0))*ub1(0,0)
              c12=0.5d0*chemof1(DKbq,q1(1,0))*ub1(1,0)
              c12=c12+0.5d0*chemof1(DKbq,q1(0,0))*ub1(0,0)
              d12=0.5d0*chemof1(DKbq,q1(0,1))*ub1(0,1)
              d12=d12+0.5d0*chemof1(DKbq,q1(0,0))*ub1(0,0)
       rhsub(0)=ub1(0,0)+D6*(ub1(0,1)-ub1(0,0))
       rhsub(0)=rhsub(0)+(ht/2.0d0)*f(cb,vb,h1(0,0))*ub1(0,0)
              rhsub(0)=rhsub(0)-D7*a12*(s1(1,0)-s1(0,0))
                rhsub(0)=rhsub(0)-D7*b12*(S1(0,1)-s1(0,0))
              rhsub(0)=rhsub(0)-D8*c12*(q1(1,0)-q1(0,0))
         rhsub(0)=rhsub(0)-D8*d12*(q1(0,1)-q1(0,0))
        funub(0)=1.0d0
C
              a21=0.5d0*chemof1(DKbs,S1(m,0))*ub1(m,0)
              a21=a21+0.5d0*chemof1(DKbs,S1(m-1,0))*ub1(m-1,0)
              b12=0.5d0*chemof1(DKbs,S1(m,1))*ub1(m,1)
              b12=b12+0.5d0*chemof1(DKbs,S1(m,0))*ub1(m,0)
              c21=0.5d0*chemof1(DKbq,q1(m,0))*ub1(m,0)
              c21=c21+0.5d0*chemof1(DKbq,q1(m-1,0))*ub1(m-1,0)
              d12=0.5d0*chemof1(DKbq,q1(m,1))*ub1(m,1)
              d12=d12+0.5d0*chemof1(DKbq,q1(m,0))*ub1(m,0)
        rhsub(m)=ub1(m,0)+D6*(ub1(m,1)-ub1(m,0))
```

```
          rhsub(m)=rhsub(m)+(ht/2.0d0)*f(cb,vb,h1(m,0))*ub1(m,0)
               rhsub(m)=rhsub(m)-D7*a21*(s1(m-1,0)-s1(m,0))
                  rhsub(m)=rhsub(m)-D7*b12*(s1(m,1)-s1(m,0))
               rhsub(m)=rhsub(m)-D8*c21*(q1(m-1,0)-q1(m,0))
                  rhsub(m)=rhsub(m)-D8*d12*(q1(m,1)-q1(m,0))
       funub(m)=1.0d0
C
       do 1020 i=1,m-1
C
          t=tr(i,hr,tss,Ds)
             call rhbound(m,S1(i,0),S1(i,1),rss,D1,hr,t,rhss(i))
          funs(i)=1.0d0+(ht/2.0d0)*fs(cas,vas,s1(i,0))*ua1(i,0)/Yas
          funs(i)=funs(i)+(ht/2.0d0)*fs(cbs,vbs,s1(i,0))*ub1(i,0)/Ybs
          t=tr(i,hr,tqs,Dq)
             call rhbound(m,q1(i,0),q1(i,1),rqs,D4,hr,t,rhsq(i))
          funq(i)=1.0d0+(ht/2.0d0)*fs(caq,vaq,q1(i,0))*ua1(i,0)/Yaq
          funq(i)=funq(i)+(ht/2.0d0)*fs(cbq,vbq,q1(i,0))*ub1(i,0)/Ybq
          t=tr(i,hr,ths,Dh)
             call rhbound(m,H1(i,0),H1(i,1),rhs,D0,hr,t,rhsh(i))
          funh(i)=1.0d0+(ht/2.0d0)*fs(ca,va,h1(i,0))*ua1(i,0)/Ya
          funh(i)=funh(i)+(ht/2.0d0)*fs(cb,vb,h1(i,0))*ub1(i,0)/Yb
C Cell group a
               a12=0.5d0*chemof1(DKas,S1(i+1,0))*ua1(i+1,0)
               a12=a12+0.5d0*chemof1(DKas,S1(i,0))*ua1(i,0)
               a21=0.5d0*chemof1(DKas,S1(i,0))*ua1(i,0)
               a21=a21+0.5d0*chemof1(DKas,S1(i-1,0))*ua1(i-1,0)
               b12=0.5d0*chemof1(DKas,S1(i,1))*ua1(i,1)
               b12=b12+0.5d0*chemof1(DKas,S1(i,0))*ua1(i,0)
               c12=0.5d0*chemof1(DKaq,q1(i+1,0))*ua1(i+1,0)
               c12=c12+0.5d0*chemof1(DKaq,q1(i,0))*ua1(i,0)
               c21=0.5d0*chemof1(DKaq,q1(i,0))*ua1(i,0)
               c21=c21+0.5d0*chemof1(DKaq,q1(i-1,0))*ua1(i-1,0)
               d12=0.5d0*chemof1(DKaq,q1(i,1))*ua1(i,1)
               d12=d12+0.5d0*chemof1(DKaq,q1(i,0))*ua1(i,0)
         rhsua(i)=ua1(i,0)+D2*(ua1(i,1)-ua1(i,0))
            rhsua(i)=rhsua(i)+(ht/2.0d0)*f(ca,va,h1(i,0))*ua1(i,0)
                 rhsua(i)=rhsua(i)-D3*a12*(s1(i+1,0)-s1(i,0))
                 rhsua(i)=rhsua(i)-D3*a21*(s1(i-1,0)-s1(i,0))
                    rhsua(i)=rhsua(i)-D3*b12*(S1(i,1)-S1(i,0))
                 rhsua(i)=rhsua(i)-D5*c12*(q1(i+1,0)-q1(i,0))
                 rhsua(i)=rhsua(i)-D5*c21*(q1(i-1,0)-q1(i,0))
                    rhsua(i)=rhsua(i)-D5*d12*(q1(i,1)-q1(i,0))
            funua(i)=1.0d0
C Cell group b
```

```
            a12=0.5d0*chemof1(DKbs,S1(i+1,0))*ub1(i+1,0)
            a12=a12+0.5d0*chemof1(DKbs,S1(i,0))*ub1(i,0)
            a21=0.5d0*chemof1(DKbs,S1(i,0))*ub1(i,0)
            a21=a21+0.5d0*chemof1(DKbs,S1(i-1,0))*ub1(i-1,0)
            b12=0.5d0*chemof1(DKbs,S1(i,1))*ub1(i,1)
            b12=b12+0.5d0*chemof1(DKbs,S1(i,0))*ub1(i,0)
            c12=0.5d0*chemof1(DKbq,q1(i+1,0))*ub1(i+1,0)
            c12=c12+0.5d0*chemof1(DKbq,q1(i,0))*ub1(i,0)
            c21=0.5d0*chemof1(DKbq,q1(i,0))*ub1(i,0)
            c21=c21+0.5d0*chemof1(DKbq,q1(i-1,0))*ub1(i-1,0)
            d12=0.5d0*chemof1(DKbq,q1(i,1))*ub1(i,1)
            d12=d12+0.5d0*chemof1(DKbq,q1(i,0))*ub1(i,0)
      rhsub(i)=ub1(i,0)+D6*(ub1(i,1)-ub1(i,0))
        rhsub(i)=rhsub(i)+(ht/2.0d0)*f(cb,vb,h1(i,0))*ub1(i,0)
            rhsub(i)=rhsub(i)-D7*a12*(s1(i+1,0)-s1(i,0))
            rhsub(i)=rhsub(i)-D7*a21*(s1(i-1,0)-s1(i,0))
              rhsub(i)=rhsub(i)-D7*b12*(S1(i,1)-S1(i,0))
            rhsub(i)=rhsub(i)-D8*c12*(q1(i+1,0)-q1(i,0))
            rhsub(i)=rhsub(i)-D8*c21*(q1(i-1,0)-q1(i,0))
              rhsub(i)=rhsub(i)-D8*d12*(q1(i,1)-q1(i,0))
         funub(i)=1.0d0
 1020      continue
C
          call PDMTRIX(m,D1,funs,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhss,us)
          call PDMTRIX(m,D4,funq,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhsq,uq)
          call PDMTRIX(m,D0,funh,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhsh,uh)
          call PDMTRIX(m,D2,funua,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhsua,uua)
          call PDMTRIX(m,D6,funub,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhsub,uub)
          do 1025 i=0,m
            S2(i,0)=us(i)
            q2(i,0)=uq(i)
            h2(i,0)=uh(i)
            ua2(i,0)=uua(i)
            ub2(i,0)=uub(i)
 1025     continue
C
        j=m
C
         rhss(0)=S1(0,m)+D1*(s1(0,m-1)-s1(0,m))
```

```
      funs(0)=1.0d0+(ht/2.0d0)*fs(cas,vas,s1(0,m))*ua1(0,m)/Yas
      funs(0)=funs(0)+(ht/2.0d0)*fs(cbs,vbs,s1(0,m))*ub1(0,m)/Ybs
      rhsq(0)=q1(0,m)+D4*(q1(0,m-1)-q1(0,m))
      funq(0)=1.0d0+(ht/2.0d0)*fs(caq,vaq,q1(0,m))*ua1(0,m)/Yaq
      funq(0)=funq(0)+(ht/2.0d0)*fs(cbq,vbq,q1(0,m))*ub1(0,m)/Ybq
      rhsh(0)=h1(0,m)+D0*(h1(0,m-1)-h1(0,m))
      funh(0)=1.0d0+(ht/2.0d0)*fs(ca,va,h1(0,m))*ua1(0,m)/Ya
      funh(0)=funh(0)+(ht/2.0d0)*fs(cb,vb,h1(0,m))*ub1(0,m)/Yb
      rhss(m)=S1(m,m)+D1*(s1(m,m-1)-s1(m,m))
      funs(m)=1.0d0+(ht/2.0d0)*fs(cas,vas,s1(m,m))*ua1(m,m)/Yas
      funs(m)=funs(m)+(ht/2.0d0)*fs(cbs,vbs,s1(m,m))*ub1(m,m)/Ybs
      rhsq(m)=q1(m,m)+D4*(q1(m,m-1)-q1(m,m))
      funq(m)=1.0d0+(ht/2.0d0)*fs(caq,vaq,q1(m,m))*ua1(m,m)/Yaq
      funq(m)=funq(m)+(ht/2.0d0)*fs(cbq,vbq,q1(m,m))*ub1(m,m)/Ybq
      rhsh(m)=h1(m,m)+D0*(h1(m,m-1)-h1(m,m))
      funh(m)=1.0d0+(ht/2.0d0)*fs(ca,va,h1(m,m))*ua1(m,m)/Ya
      funh(m)=funh(m)+(ht/2.0d0)*fs(cb,vb,h1(m,m))*ub1(m,m)/Yb
C Cell group a
        a12=0.5d0*chemof1(DKas,S1(1,m))*ua1(1,m)
        a12=a12+0.5d0*chemof1(DKas,S1(0,m))*ua1(0,m)
        b21=0.5d0*chemof1(DKas,S1(0,m))*ua1(0,m)
        b21=b21+0.5d0*chemof1(DKas,S1(0,m-1))*ua1(0,m-1)
        c12=0.5d0*chemof1(DKaq,q1(1,m))*ua1(1,m)
        c12=c12+0.5d0*chemof1(DKaq,q1(0,m))*ua1(0,m)
        d21=0.5d0*chemof1(DKaq,q1(0,m))*ua1(0,m)
        d21=d21+0.5d0*chemof1(DKaq,q1(0,m-1))*ua1(0,m-1)
      rhsua(0)=ua1(0,m)+D2*(ua1(0,m-1)-ua1(0,m))
       rhsua(0)=rhsua(0)+(ht/2.0d0)*f(ca,va,h1(0,m))*ua1(0,m)
        rhsua(0)=rhsua(0)-D3*a12*(s1(1,m)-s1(0,m))
        rhsua(0)=rhsua(0)-D3*b21*(S1(0,m-1)-s1(0,m))
        rhsua(0)=rhsua(0)-D5*c12*(q1(1,m)-q1(0,m))
        rhsua(0)=rhsua(0)-D5*d21*(q1(0,m-1)-q1(0,m))
      funua(0)=1.0d0
C
C
        a21=0.5d0*chemof1(DKas,S1(m,m))*ua1(m,m)
        a21=a21+0.5d0*chemof1(DKas,S1(m-1,m))*ua1(m-1,m)
        b21=0.5d0*chemof1(DKas,S1(m,m))*ua1(m,m)
        b21=b21+0.5d0*chemof1(DKas,S1(m,m-1))*ua1(m,m-1)
        c21=0.5d0*chemof1(DKaq,q1(m,m))*ua1(m,m)
        c21=c21+0.5d0*chemof1(DKaq,q1(m-1,m))*ua1(m-1,m)
        d21=0.5d0*chemof1(DKaq,q1(m,m))*ua1(m,m)
        d21=d21+0.5d0*chemof1(DKaq,q1(m,m-1))*ua1(m,m-1)
      rhsua(m)=ua1(m,m)+D2*(ua1(m,m-1)-ua1(m,m))
```

```
            rhsua(m)=rhsua(m)+(ht/2.0d0)*f(ca,va,h1(m,m))*ua1(m,m)
              rhsua(m)=rhsua(m)-D3*a21*(s1(m-1,m)-s1(m,m))
                rhsua(m)=rhsua(m)-D3*b21*(S1(m,m-1)-s1(m,m))
              rhsua(m)=rhsua(m)-D5*c21*(q1(m-1,m)-q1(m,m))
                rhsua(m)=rhsua(m)-D5*d21*(q1(m,m-1)-q1(m,m))
           funua(m)=1.0d0
C
C Cell group b
            a12=0.5d0*chemof1(DKbs,S1(1,m))*ub1(1,m)
            a12=a12+0.5d0*chemof1(DKbs,S1(0,m))*ub1(0,m)
            b21=0.5d0*chemof1(DKbs,S1(0,m))*ub1(0,m)
            b21=b21+0.5d0*chemof1(DKbs,S1(0,m-1))*ub1(0,m-1)
            c12=0.5d0*chemof1(DKbq,q1(1,m))*ub1(1,m)
            c12=c12+0.5d0*chemof1(DKbq,q1(0,m))*ub1(0,m)
            d21=0.5d0*chemof1(DKbq,q1(0,m))*ub1(0,m)
            d21=d21+0.5d0*chemof1(DKbq,q1(0,m-1))*ub1(0,m-1)
         rhsub(0)=ub1(0,m)+D6*(ub1(0,m-1)-ub1(0,m))
           rhsub(0)=rhsub(0)+(ht/2.0d0)*f(cb,vb,h1(0,m))*ub1(0,m)
            rhsub(0)=rhsub(0)-D7*a12*(s1(1,m)-s1(0,m))
            rhsub(0)=rhsub(0)-D7*b21*(S1(0,m-1)-s1(0,m))
            rhsub(0)=rhsub(0)-D8*c12*(q1(1,m)-q1(0,m))
            rhsub(0)=rhsub(0)-D8*d21*(q1(0,m-1)-q1(0,m))
         funub(0)=1.0d0
C
C
            a21=0.5d0*chemof1(DKbs,S1(m,m))*ub1(m,m)
            a21=a21+0.5d0*chemof1(DKbs,S1(m-1,m))*ub1(m-1,m)
            b21=0.5d0*chemof1(DKbs,S1(m,m))*ub1(m,m)
            b21=b21+0.5d0*chemof1(DKbs,S1(m,m-1))*ub1(m,m-1)
            c21=0.5d0*chemof1(DKbq,q1(m,m))*ub1(m,m)
            c21=c21+0.5d0*chemof1(DKbq,q1(m-1,m))*ub1(m-1,m)
            d21=0.5d0*chemof1(DKbq,q1(m,m))*ub1(m,m)
            d21=d21+0.5d0*chemof1(DKbq,q1(m,m-1))*ub1(m,m-1)
         rhsub(m)=ub1(m,m)+D6*(ub1(m,m-1)-ub1(m,m))
           rhsub(m)=rhsub(m)+(ht/2.0d0)*f(cb,vb,h1(m,m))*ub1(m,m)
            rhsub(m)=rhsub(m)-D7*a21*(s1(m-1,m)-s1(m,m))
              rhsub(m)=rhsub(m)-D7*b21*(S1(m,m-1)-s1(m,m))
            rhsub(m)=rhsub(m)-D8*c21*(q1(m-1,m)-q1(m,m))
              rhsub(m)=rhsub(m)-D8*d21*(q1(m,m-1)-q1(m,m))
         funub(m)=1.0d0
C
       do 1030 i=1,m-1
C
      t=tr(i,hr,tsn,Ds)
```

```
            call rhbound(m,S1(i,m),S1(i,m-1),rsn,D1,hr,t,rhss(i))
          funs(i)=1.0d0+(ht/2.0d0)*fs(cas,vas,s1(i,m))*ua1(i,m)/Yas
          funs(i)=funs(i)+(ht/2.0d0)*fs(cbs,vbs,s1(i,m))*ub1(i,m)/Ybs
          t=tr(i,hr,tqn,Dq)
            call rhbound(m,q1(i,m),q1(i,m-1),rqn,D4,hr,t,rhsq(i))
          funq(i)=1.0d0+(ht/2.0d0)*fs(caq,vaq,q1(i,m))*ua1(i,m)/Yaq
          funq(i)=funq(i)+(ht/2.0d0)*fs(cbq,vbq,q1(i,m))*ub1(i,m)/Ybq
          t=tr(i,hr,thn,Dh)
            call rhbound(m,H1(i,m),H1(i,m-1),rhn,D0,hr,t,rhsh(i))
          funh(i)=1.0d0+(ht/2.0d0)*fs(ca,va,h1(i,m))*ua1(i,m)/Ya
          funh(i)=funh(i)+(ht/2.0d0)*fs(cb,vb,h1(i,m))*ub1(i,m)/Yb
C Cell group a
            a12=0.5d0*chemof1(DKas,S1(i+1,m))*ua1(i+1,m)
            a12=a12+0.5d0*chemof1(DKas,S1(i,m))*ua1(i,m)
            a21=0.5d0*chemof1(DKas,S1(i,m))*ua1(i,m)
            a21=a21+0.5d0*chemof1(DKas,S1(i-1,m))*ua1(i-1,m)
            b21=0.5d0*chemof1(DKas,S1(i,m))*ua1(i,m)
            b21=b21+0.5d0*chemof1(DKas,S1(i,m-1))*ua1(i,m-1)
            c12=0.5d0*chemof1(DKaq,q1(i+1,m))*ua1(i+1,m)
            c12=c12+0.5d0*chemof1(DKaq,q1(i,m))*ua1(i,m)
            c21=0.5d0*chemof1(DKaq,q1(i,m))*ua1(i,m)
            c21=c21+0.5d0*chemof1(DKaq,q1(i-1,m))*ua1(i-1,m)
            d21=0.5d0*chemof1(DKaq,q1(i,m))*ua1(i,m)
            d21=d21+0.5d0*chemof1(DKaq,q1(i,m-1))*ua1(i,m-1)
          rhsua(i)=ua1(i,m)+D2*(ua1(i,m-1)-ua1(i,m))
            rhsua(i)=rhsua(i)+(ht/2.0d0)*f(ca,va,h1(i,m))*ua1(i,m)
             rhsua(i)=rhsua(i)-D3*a12*(s1(i+1,m)-s1(i,m))
             rhsua(i)=rhsua(i)-D3*a21*(s1(i-1,m)-s1(i,m))
               rhsua(i)=rhsua(i)-D3*b21*(S1(i,m-1)-S1(i,m))
             rhsua(i)=rhsua(i)-D5*c12*(q1(i+1,m)-q1(i,m))
             rhsua(i)=rhsua(i)-D5*c21*(q1(i-1,m)-q1(i,m))
               rhsua(i)=rhsua(i)-D5*d21*(q1(i,m-1)-q1(i,m))
          funua(i)=1.0d0
C Cell group b
            a12=0.5d0*chemof1(DKbs,S1(i+1,m))*ub1(i+1,m)
            a12=a12+0.5d0*chemof1(DKbs,S1(i,m))*ub1(i,m)
            a21=0.5d0*chemof1(DKbs,S1(i,m))*ub1(i,m)
            a21=a21+0.5d0*chemof1(DKbs,S1(i-1,m))*ub1(i-1,m)
            b21=0.5d0*chemof1(DKbs,S1(i,m))*ub1(i,m)
            b21=b21+0.5d0*chemof1(DKbs,S1(i,m-1))*ub1(i,m-1)
            c12=0.5d0*chemof1(DKbq,q1(i+1,m))*ub1(i+1,m)
            c12=c12+0.5d0*chemof1(DKbq,q1(i,m))*ub1(i,m)
            c21=0.5d0*chemof1(DKbq,q1(i,m))*ub1(i,m)
            c21=c21+0.5d0*chemof1(DKbq,q1(i-1,m))*ub1(i-1,m)
```

```
          d21=0.5d0*chemof1(DKbq,q1(i,m))*ub1(i,m)
          d21=d21+0.5d0*chemof1(DKbq,q1(i,m-1))*ub1(i,m-1)
        rhsub(i)=ub1(i,m)+D6*(ub1(i,m-1)-ub1(i,m))
          rhsub(i)=rhsub(i)+(ht/2.0d0)*f(cb,vb,h1(i,m))*ub1(i,m)
            rhsub(i)=rhsub(i)-D7*a12*(s1(i+1,m)-s1(i,m))
            rhsub(i)=rhsub(i)-D7*a21*(s1(i-1,m)-s1(i,m))
              rhsub(i)=rhsub(i)-D7*b21*(S1(i,m-1)-S1(i,m))
            rhsub(i)=rhsub(i)-D8*c12*(q1(i+1,m)-q1(i,m))
            rhsub(i)=rhsub(i)-D8*c21*(q1(i-1,m)-q1(i,m))
              rhsub(i)=rhsub(i)-D8*d21*(q1(i,m-1)-q1(i,m))
         funub(i)=1.0d0
C
 1030      continue
C
          call PDMTRIX(m,D1,funs,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhss,us)
          call PDMTRIX(m,D4,funq,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhsq,uq)
          call PDMTRIX(m,D0,funh,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhsh,uh)
          call PDMTRIX(m,D2,funua,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhsua,uua)
          call PDMTRIX(m,D6,funub,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhsub,uub)
          do 1035 i=0,m
            S2(i,m)=us(i)
            q2(i,m)=uq(i)
            h2(i,m)=uh(i)
            ua2(i,m)=uua(i)
            ub2(i,m)=uub(i)
 1035     continue

C take care of interior points:
C
        do 1050 j=1,m-1
C
          t=tr(j,hr,tsw,Ds)
          call rh2(m,S1(0,j),S1(0,j-1),S1(0,j+1),rsw,D1,hr,t,rhss(0))
        funs(0)=1.0d0+(ht/2.0d0)*fs(cas,vas,s1(0,j))*ua1(0,j)/Yas
        funs(0)=funs(0)+(ht/2.0d0)*fs(cbs,vbs,s1(0,j))*ub1(0,j)/Ybs
        funs(0)=funs(0)+D1*hr*tr(j,hr,tsw,Ds)
          t=tr(j,hr,tqw,Dq)
          call rh2(m,q1(0,j),q1(0,j-1),q1(0,j+1),rqw,D4,hr,t,rhsq(0))
        funq(0)=1.0d0+(ht/2.0d0)*fs(caq,vaq,q1(0,j))*ua1(0,j)/Yaq
```

```
      funq(0)=funq(0)+(ht/2.0d0)*fs(cbq,vbq,q1(0,j))*ub1(0,j)/Ybq
      funq(0)=funq(0)+D4*hr*tr(j,hr,tqw,Dq)
        t=tr(j,hr,thw,Dh)
        call rh2(m,h1(0,j),h1(0,j-1),h1(0,j+1),rhw,D0,hr,t,rhsh(0))
      funh(0)=1.0d0+(ht/2.0d0)*fs(ca,va,h1(0,j))*ua1(0,j)/Ya
      funh(0)=funh(0)+(ht/2.0d0)*fs(cb,vb,h1(0,j))*ub1(0,j)/Yb
      funh(0)=funh(0)+D0*hr*tr(j,hr,thw,Dh)
        t=tr(j,hr,tse,Ds)
        call rh2(m,S1(m,j),S1(m,j-1),S1(m,j+1),rse,D1,hr,t,rhss(m))
      funs(m)=1.0d0+(ht/2.0d0)*fs(cas,vas,s1(m,j))*ua1(m,j)/Yas
      funs(m)=funs(m)+(ht/2.0d0)*fs(cbs,vbs,s1(m,j))*ub1(m,j)/Ybs
      funs(m)=funs(m)+D1*hr*tr(j,hr,tsw,Ds)
        t=tr(j,hr,tqe,Dq)
        call rh2(m,q1(m,j),q1(m,j-1),q1(m,j+1),rqe,D4,hr,t,rhsq(m))
      funq(m)=1.0d0+(ht/2.0d0)*fs(caq,vaq,q1(m,j))*ua1(m,j)/Yaq
      funq(m)=funq(m)+(ht/2.0d0)*fs(cbq,vbq,q1(m,j))*ub1(m,j)/Ybq
      funq(m)=funq(m)+D4*hr*tr(j,hr,tqw,Dq)
        t=tr(j,hr,the,Dh)
        call rh2(m,h1(m,j),h1(m,j-1),h1(m,j+1),rhe,D0,hr,t,rhsh(m))
      funh(m)=1.0d0+(ht/2.0d0)*fs(ca,va,h1(m,j))*ua1(m,j)/Ya
      funh(m)=funh(m)+(ht/2.0d0)*fs(cb,vb,h1(m,j))*ub1(m,j)/Yb
      funh(m)=funh(m)+D0*hr*tr(j,hr,thw,Dh)
C
       i=0
C Cell group a
          a12=0.5d0*chemof1(DKas,S1(i+1,j))*ua1(i+1,j)
          a12=a12+0.5d0*chemof1(DKas,S1(i,j))*ua1(i,j)
          b12=0.5d0*chemof1(DKas,S1(i,j+1))*ua1(i,j+1)
          b12=b12+0.5d0*chemof1(DKas,S1(i,j))*ua1(i,j)
          b21=0.5d0*chemof1(DKas,S1(i,j))*ua1(i,j)
          b21=b21+0.5d0*chemof1(DKas,S1(i,j-1))*ua1(i,j-1)
          c12=0.5d0*chemof1(DKaq,q1(i+1,j))*ua1(i+1,j)
          c12=c12+0.5d0*chemof1(DKaq,q1(i,j))*ua1(i,j)
          d12=0.5d0*chemof1(DKaq,q1(i,j+1))*ua1(i,j+1)
          d12=d12+0.5d0*chemof1(DKaq,q1(i,j))*ua1(i,j)
          d21=0.5d0*chemof1(DKaq,q1(i,j))*ua1(i,j)
          d21=d21+0.5d0*chemof1(DKaq,q1(i,j-1))*ua1(i,j-1)
        rhsua(0)=ua1(0,j)+D2*(ua1(0,j+1)-ua1(0,j))
     *              +D2*(ua1(0,j-1)-ua1(0,j))
          rhsua(0)=rhsua(0)+(ht/2.0d0)*f(ca,va,h1(0,j))*ua1(0,j)
           rhsua(0)=rhsua(0)-D3*a12*(s1(1,j)-s1(0,j))
             rhsua(0)=rhsua(0)-D3*b12*(s1(0,j+1)-s1(0,j))
             rhsua(0)=rhsua(0)-D3*b21*(s1(0,j-1)-s1(0,j))
           rhsua(0)=rhsua(0)-D5*c12*(q1(1,j)-q1(0,j))
```

```
            rhsua(0)=rhsua(0)-D5*d12*(q1(0,j+1)-q1(0,j))
            rhsua(0)=rhsua(0)-D5*d21*(q1(0,j-1)-q1(0,j))
          funua(0)=1.0d0
C
            i=m
C
            a21=0.5d0*chemof1(DKas,S1(i,j))*ua1(i,j)
            a21=a21+0.5d0*chemof1(DKas,S1(i-1,j))*ua1(i-1,j)
            b12=0.5d0*chemof1(DKas,S1(i,j+1))*ua1(i,j+1)
            b12=b12+0.5d0*chemof1(DKas,S1(i,j))*ua1(i,j)
            b21=0.5d0*chemof1(DKas,S1(i,j))*ua1(i,j)
            b21=b21+0.5d0*chemof1(DKas,S1(i,j-1))*ua1(i,j-1)
            c21=0.5d0*chemof1(DKaq,q1(i,j))*ua1(i,j)
            c21=c21+0.5d0*chemof1(DKaq,q1(i-1,j))*ua1(i-1,j)
            d12=0.5d0*chemof1(DKaq,q1(i,j+1))*ua1(i,j+1)
            d12=d12+0.5d0*chemof1(DKaq,q1(i,j))*ua1(i,j)
            d21=0.5d0*chemof1(DKaq,q1(i,j))*ua1(i,j)
            d21=d21+0.5d0*chemof1(DKaq,q1(i,j-1))*ua1(i,j-1)
          rhsua(m)=ua1(m,j)+D2*(ua1(m,j+1)-ua1(m,j))
     *                +D2*(ua1(m,j-1)-ua1(m,j))
            rhsua(m)=rhsua(m)+(ht/2.0d0)*f(ca,va,h1(m,j))*ua1(m,j)
             rhsua(m)=rhsua(m)-D3*a21*(s1(m-1,j)-s1(m,j))
               rhsua(m)=rhsua(m)-D3*b12*(s1(m,j+1)-s1(m,j))
               rhsua(m)=rhsua(m)-D3*b21*(s1(m,j-1)-s1(m,j))
             rhsua(m)=rhsua(m)-D5*c21*(q1(m-1,j)-q1(m,j))
               rhsua(m)=rhsua(m)-D5*d12*(q1(m,j+1)-q1(m,j))
               rhsua(m)=rhsua(m)-D5*d21*(q1(m,j-1)-q1(m,j))
          funua(i)=1.0d0
C
C Cell group b
            a12=0.5d0*chemof1(DKbs,S1(i+1,j))*ub1(i+1,j)
            a12=a12+0.5d0*chemof1(DKbs,S1(i,j))*ub1(i,j)
            b12=0.5d0*chemof1(DKbs,S1(i,j+1))*ub1(i,j+1)
            b12=b12+0.5d0*chemof1(DKbs,S1(i,j))*ub1(i,j)
            b21=0.5d0*chemof1(DKbs,S1(i,j))*ub1(i,j)
            b21=b21+0.5d0*chemof1(DKbs,S1(i,j-1))*ub1(i,j-1)
            c12=0.5d0*chemof1(DKbq,q1(i+1,j))*ub1(i+1,j)
            c12=c12+0.5d0*chemof1(DKbq,q1(i,j))*ub1(i,j)
            d12=0.5d0*chemof1(DKbq,q1(i,j+1))*ub1(i,j+1)
            d12=d12+0.5d0*chemof1(DKbq,q1(i,j))*ub1(i,j)
            d21=0.5d0*chemof1(DKbq,q1(i,j))*ub1(i,j)
            d21=d21+0.5d0*chemof1(DKbq,q1(i,j-1))*ub1(i,j-1)
          rhsub(0)=ub1(0,j)+D6*(ub1(0,j+1)-ub1(0,j))
     *                +D6*(ub1(0,j-1)-ub1(0,j))
```

```
      rhsub(0)=rhsub(0)+(ht/2.0d0)*f(cb,vb,h1(0,j))*ub1(0,j)
       rhsub(0)=rhsub(0)-D7*a12*(s1(1,j)-s1(0,j))
         rhsub(0)=rhsub(0)-D7*b12*(s1(0,j+1)-s1(0,j))
         rhsub(0)=rhsub(0)-D7*b21*(s1(0,j-1)-s1(0,j))
       rhsub(0)=rhsub(0)-D8*c12*(q1(1,j)-q1(0,j))
         rhsub(0)=rhsub(0)-D8*d12*(q1(0,j+1)-q1(0,j))
         rhsub(0)=rhsub(0)-D8*d21*(q1(0,j-1)-q1(0,j))
      funub(0)=1.0d0
C
      i=m
C
      a21=0.5d0*chemof1(DKbs,S1(i,j))*ub1(i,j)
      a21=a21+0.5d0*chemof1(DKbs,S1(i-1,j))*ub1(i-1,j)
      b12=0.5d0*chemof1(DKbs,S1(i,j+1))*ub1(i,j+1)
      b12=b12+0.5d0*chemof1(DKbs,S1(i,j))*ub1(i,j)
      b21=0.5d0*chemof1(DKbs,S1(i,j))*ub1(i,j)
      b21=b21+0.5d0*chemof1(DKbs,S1(i,j-1))*ub1(i,j-1)
      c21=0.5d0*chemof1(DKbq,q1(i,j))*ub1(i,j)
      c21=c21+0.5d0*chemof1(DKbq,q1(i-1,j))*ub1(i-1,j)
      d12=0.5d0*chemof1(DKbq,q1(i,j+1))*ub1(i,j+1)
      d12=d12+0.5d0*chemof1(DKbq,q1(i,j))*ub1(i,j)
      d21=0.5d0*chemof1(DKbq,q1(i,j))*ub1(i,j)
      d21=d21+0.5d0*chemof1(DKbq,q1(i,j-1))*ub1(i,j-1)
     rhsub(m)=ub1(m,j)+D6*(ub1(m,j+1)-ub1(m,j))
     *              +D6*(ub1(m,j-1)-ub1(m,j))
      rhsub(m)=rhsub(m)+(ht/2.0d0)*f(cb,vb,h1(m,j))*ub1(m,j)
       rhsub(m)=rhsub(m)-D7*a21*(s1(m-1,j)-s1(m,j))
         rhsub(m)=rhsub(m)-D7*b12*(s1(m,j+1)-s1(m,j))
         rhsub(m)=rhsub(m)-D7*b21*(s1(m,j-1)-s1(m,j))
       rhsub(m)=rhsub(m)-D8*c21*(q1(m-1,j)-q1(m,j))
         rhsub(m)=rhsub(m)-D8*d12*(q1(m,j+1)-q1(m,j))
         rhsub(m)=rhsub(m)-D8*d21*(q1(m,j-1)-q1(m,j))
      funub(i)=1.0d0
C
      do 1040 i=1,m-1
C
     rhss(i)=S1(i,j)+D1*(S1(i,j+1)-2.0d0*S1(i,j)+S1(i,j-1))
     funs(i)=1.0d0+(ht/2.0d0)*fs(cas,vas,s1(i,j))*ua1(i,j)/Yas
    funs(i)=funs(i)+(ht/2.0d0)*fs(cbs,vbs,s1(i,j))*ub1(i,j)/Ybs
     rhsq(i)=q1(i,j)+D4*(q1(i,j+1)-2.0d0*q1(i,j)+q1(i,j-1))
    funq(i)=1.0d0+(ht/2.0d0)*fs(caq,vaq,q1(i,j))*ua1(i,j)/Yaq
    funq(i)=funq(i)+(ht/2.0d0)*fs(cbq,vbq,q1(i,j))*ub1(i,j)/Ybq
     rhsh(i)=h1(i,j)+D0*(h1(i,j+1)-2.0d0*h1(i,j)+h1(i,j-1))
     funh(i)=1.0d0+(ht/2.0d0)*fs(ca,va,h1(i,j))*ua1(i,j)/Ya
```

```
          funh(i)=funh(i)+(ht/2.0d0)*fs(cb,vb,h1(i,j))*ub1(i,j)/Yb
C Cell group a
            a12=0.5d0*chemof1(DKas,S1(i+1,j))*ua1(i+1,j)
            a12=a12+0.5d0*chemof1(DKas,S1(i,j))*ua1(i,j)
            a21=0.5d0*chemof1(DKas,S1(i,j))*ua1(i,j)
            a21=a21+0.5d0*chemof1(DKas,S1(i-1,j))*ua1(i-1,j)
            b12=0.5d0*chemof1(DKas,S1(i,j+1))*ua1(i,j+1)
            b12=b12+0.5d0*chemof1(DKas,S1(i,j))*ua1(i,j)
            b21=0.5d0*chemof1(DKas,S1(i,j))*ua1(i,j)
            b21=b21+0.5d0*chemof1(DKas,S1(i,j-1))*ua1(i,j-1)
            c12=0.5d0*chemof1(DKaq,q1(i+1,j))*ua1(i+1,j)
            c12=c12+0.5d0*chemof1(DKaq,q1(i,j))*ua1(i,j)
            c21=0.5d0*chemof1(DKaq,q1(i,j))*ua1(i,j)
            c21=c21+0.5d0*chemof1(DKaq,q1(i-1,j))*ua1(i-1,j)
            d12=0.5d0*chemof1(DKaq,q1(i,j+1))*ua1(i,j+1)
            d12=d12+0.5d0*chemof1(DKaq,q1(i,j))*ua1(i,j)
            d21=0.5d0*chemof1(DKaq,q1(i,j))*ua1(i,j)
            d21=d21+0.5d0*chemof1(DKaq,q1(i,j-1))*ua1(i,j-1)
          rhsua(i)=ua1(i,j)+D2*(ua1(i,j+1)-ua1(i,j))
     *                +D2*(ua1(i,j-1)-ua1(i,j))
            rhsua(i)=rhsua(i)+(ht/2.0d0)*f(ca,va,h1(i,j))*ua1(i,j)
             rhsua(i)=rhsua(i)-D3*a12*(s1(i+1,j)-s1(i,j))
        rhsua(i)=rhsua(i)-D3*a21*(s1(i-1,j)-s1(i,j))
               rhsua(i)=rhsua(i)-D3*b12*(s1(i,j+1)-s1(i,j))
               rhsua(i)=rhsua(i)-D3*b21*(s1(i,j-1)-s1(i,j))
              rhsua(i)=rhsua(i)-D5*c12*(q1(i+1,j)-q1(i,j))
               rhsua(i)=rhsua(i)-D5*c21*(q1(i-1,j)-q1(i,j))
               rhsua(i)=rhsua(i)-D5*d12*(q1(i,j+1)-q1(i,j))
               rhsua(i)=rhsua(i)-D5*d21*(q1(i,j-1)-q1(i,j))
          funua(i)=1.0d0
C Cell group b
            a12=0.5d0*chemof1(DKbs,S1(i+1,j))*ub1(i+1,j)
            a12=a12+0.5d0*chemof1(DKbs,S1(i,j))*ub1(i,j)
            a21=0.5d0*chemof1(DKbs,S1(i,j))*ub1(i,j)
            a21=a21+0.5d0*chemof1(DKbs,S1(i-1,j))*ub1(i-1,j)
            b12=0.5d0*chemof1(DKbs,S1(i,j+1))*ub1(i,j+1)
            b12=b12+0.5d0*chemof1(DKbs,S1(i,j))*ub1(i,j)
            b21=0.5d0*chemof1(DKbs,S1(i,j))*ub1(i,j)
            b21=b21+0.5d0*chemof1(DKbs,S1(i,j-1))*ub1(i,j-1)
            c12=0.5d0*chemof1(DKbq,q1(i+1,j))*ub1(i+1,j)
            c12=c12+0.5d0*chemof1(DKbq,q1(i,j))*ub1(i,j)
            c21=0.5d0*chemof1(DKbq,q1(i,j))*ub1(i,j)
            c21=c21+0.5d0*chemof1(DKbq,q1(i-1,j))*ub1(i-1,j)
            d12=0.5d0*chemof1(DKbq,q1(i,j+1))*ub1(i,j+1)
```

```
            d12=d12+0.5d0*chemof1(DKbq,q1(i,j))*ub1(i,j)
            d21=0.5d0*chemof1(DKbq,q1(i,j))*ub1(i,j)
            d21=d21+0.5d0*chemof1(DKbq,q1(i,j-1))*ub1(i,j-1)
          rhsub(i)=ub1(i,j)+D6*(ub1(i,j+1)-ub1(i,j))
     *                +D6*(ub1(i,j-1)-ub1(i,j))
            rhsub(i)=rhsub(i)+(ht/2.0d0)*f(cb,vb,h1(i,j))*ub1(i,j)
             rhsub(i)=rhsub(i)-D7*a12*(s1(i+1,j)-s1(i,j))
         rhsub(i)=rhsub(i)-D7*a21*(s1(i-1,j)-s1(i,j))
               rhsub(i)=rhsub(i)-D7*b12*(s1(i,j+1)-s1(i,j))
               rhsub(i)=rhsub(i)-D7*b21*(s1(i,j-1)-s1(i,j))
             rhsub(i)=rhsub(i)-D8*c12*(q1(i+1,j)-q1(i,j))
               rhsub(i)=rhsub(i)-D8*c21*(q1(i-1,j)-q1(i,j))
               rhsub(i)=rhsub(i)-D8*d12*(q1(i,j+1)-q1(i,j))
               rhsub(i)=rhsub(i)-D8*d21*(q1(i,j-1)-q1(i,j))
          funub(i)=1.0d0
C
 1040       continue
C
            call PDMTRIX(m,D1,funs,aa,bb,cc)
            call TRISOLV(m,aa,bb,cc,rhss,us)
            call PDMTRIX(m,D4,funq,aa,bb,cc)
            call TRISOLV(m,aa,bb,cc,rhsq,uq)
            call PDMTRIX(m,D0,funh,aa,bb,cc)
            call TRISOLV(m,aa,bb,cc,rhsh,uh)
            call PDMTRIX(m,D2,funua,aa,bb,cc)
            call TRISOLV(m,aa,bb,cc,rhsua,uua)
            call PDMTRIX(m,D6,funub,aa,bb,cc)
            call TRISOLV(m,aa,bb,cc,rhsub,uub)
            do 1045 i=0,m
              S2(i,j)=us(i)
              q2(i,j)=uq(i)
              h2(i,j)=uh(i)
              ua2(i,j)=uua(i)
              ub2(i,j)=uub(i)
 1045        continue
 1050        continue
C
C implicit in y direction:
C
            do 1055 i=0,m
            do 1056 j=0,m
                 S1(i,j)=S2(i,j)
                 q1(i,j)=q2(i,j)
                 h1(i,j)=h2(i,j)
```

```
                ua1(i,j)=ua2(i,j)
                ub1(i,j)=ub2(i,j)
 1056       continue
 1055       continue
C
C take care of boundary points:
C
          i=0
C
         rhss(0)=S1(0,0)+D1*(S1(1,0)-S1(0,0))
            rhss(0)=rhss(0)-(ht/2.0d0)*f(cas,vas,s1(0,0))*ua1(0,0)/Yas
            rhss(0)=rhss(0)-(ht/2.0d0)*f(cbs,vbs,s1(0,0))*ub1(0,0)/Ybs
         funs(0)=1.0d0
         rhsq(0)=q1(0,0)+D4*(q1(1,0)-q1(0,0))
            rhsq(0)=rhsq(0)-(ht/2.0d0)*f(caq,vaq,q1(0,0))*ua1(0,0)/Yaq
            rhsq(0)=rhsq(0)-(ht/2.0d0)*f(cbq,vbq,q1(0,0))*ub1(0,0)/Ybq
         funq(0)=1.0d0
         rhsh(0)=h1(0,0)+D0*(h1(1,0)-h1(0,0))
            rhsh(0)=rhsh(0)-(ht/2.0d0)*f(ca,va,h1(0,0))*ua1(0,0)/Ya
            rhsh(0)=rhsh(0)-(ht/2.0d0)*f(cb,vb,h1(0,0))*ub1(0,0)/Yb
         funh(0)=1.0d0
         rhss(m)=S1(0,m)+D1*(S1(1,m)-S1(0,m))
            rhss(m)=rhss(m)-(ht/2.0d0)*f(cas,vas,s1(0,m))*ua1(0,m)/Yas
            rhss(m)=rhss(m)-(ht/2.0d0)*f(cbs,vbs,s1(0,m))*ub1(0,m)/Ybs
         funs(m)=1.0d0
         rhsq(m)=q1(0,m)+D4*(q1(1,m)-q1(0,m))
            rhsq(m)=rhsq(m)-(ht/2.0d0)*f(caq,vaq,q1(0,m))*ua1(0,m)/Yaq
            rhsq(m)=rhsq(m)-(ht/2.0d0)*f(cbq,vbq,q1(0,m))*ub1(0,m)/Ybq
         funq(m)=1.0d0
         rhsh(m)=h1(0,m)+D0*(h1(1,m)-h1(0,m))
            rhsh(m)=rhsh(m)-(ht/2.0d0)*f(ca,va,h1(0,m))*ua1(0,m)/Ya
            rhsh(m)=rhsh(m)-(ht/2.0d0)*f(cb,vb,h1(0,m))*ub1(0,m)/Yb
         funh(m)=1.0d0
C Cell group a
             a12=0.5d0*chemof1(DKas,S1(1,0))*ua1(1,0)
             a12=a12+0.5d0*chemof1(DKas,S1(0,0))*ua1(0,0)
             b12=0.5d0*chemof1(DKas,S1(0,1))*ua1(0,1)
             b12=b12+0.5d0*chemof1(DKas,S1(0,0))*ua1(0,0)
             c12=0.5d0*chemof1(DKaq,q1(1,0))*ua1(1,0)
             c12=c12+0.5d0*chemof1(DKaq,q1(0,0))*ua1(0,0)
             d12=0.5d0*chemof1(DKaq,q1(0,1))*ua1(0,1)
             d12=d12+0.5d0*chemof1(DKaq,q1(0,0))*ua1(0,0)
         rhsua(0)=ua1(0,0)+D2*(ua1(1,0)-ua1(0,0))
            rhsua(0)=rhsua(0)+(ht/2.0d0)*f(ca,va,h1(0,0))*ua1(0,0)
```

```
            rhsua(0)=rhsua(0)-D3*a12*(s1(1,0)-s1(0,0))
            rhsua(0)=rhsua(0)-D3*b12*(S1(0,1)-s1(0,0))
            rhsua(0)=rhsua(0)-D5*c12*(q1(1,0)-q1(0,0))
            rhsua(0)=rhsua(0)-D5*d12*(q1(0,1)-q1(0,0))
          funua(0)=1.0d0
C
            a12=0.5d0*chemof1(DKas,S1(1,m))*ua1(1,m)
            a12=a12+0.5d0*chemof1(DKas,S1(0,m))*ua1(0,m)
            b21=0.5d0*chemof1(DKas,S1(0,m-1))*ua1(0,m-1)
            b21=b21+0.5d0*chemof1(DKas,S1(0,m))*ua1(0,m)
            c12=0.5d0*chemof1(DKaq,q1(1,m))*ua1(1,m)
            c12=c12+0.5d0*chemof1(DKaq,q1(0,m))*ua1(0,m)
            d21=0.5d0*chemof1(DKaq,q1(0,m-1))*ua1(0,m-1)
            d21=d21+0.5d0*chemof1(DKaq,q1(0,m))*ua1(0,m)
          rhsua(m)=ua1(0,m)+D2*(ua1(1,m)-ua1(0,m))
            rhsua(m)=rhsua(m)+(ht/2.0d0)*f(ca,va,h1(0,m))*ua1(0,m)
              rhsua(m)=rhsua(m)-D3*a12*(s1(1,m)-s1(0,m))
              rhsua(m)=rhsua(m)-D3*b21*(s1(0,m-1)-s1(0,m))
              rhsua(m)=rhsua(m)-D5*c12*(q1(1,m)-q1(0,m))
              rhsua(m)=rhsua(m)-D5*d21*(q1(0,m-1)-q1(0,m))
          funua(m)=1.0d0
C
C Cell group b
            a12=0.5d0*chemof1(DKbs,S1(1,0))*ub1(1,0)
            a12=a12+0.5d0*chemof1(DKbs,S1(0,0))*ub1(0,0)
            b12=0.5d0*chemof1(DKbs,S1(0,1))*ub1(0,1)
            b12=b12+0.5d0*chemof1(DKbs,S1(0,0))*ub1(0,0)
            c12=0.5d0*chemof1(DKbq,q1(1,0))*ub1(1,0)
            c12=c12+0.5d0*chemof1(DKbq,q1(0,0))*ub1(0,0)
            d12=0.5d0*chemof1(DKbq,q1(0,1))*ub1(0,1)
            d12=d12+0.5d0*chemof1(DKbq,q1(0,0))*ub1(0,0)
          rhsub(0)=ub1(0,0)+D6*(ub1(1,0)-ub1(0,0))
            rhsub(0)=rhsub(0)+(ht/2.0d0)*f(cb,vb,h1(0,0))*ub1(0,0)
              rhsub(0)=rhsub(0)-D7*a12*(s1(1,0)-s1(0,0))
              rhsub(0)=rhsub(0)-D7*b12*(S1(0,1)-s1(0,0))
              rhsub(0)=rhsub(0)-D8*c12*(q1(1,0)-q1(0,0))
              rhsub(0)=rhsub(0)-D8*d12*(q1(0,1)-q1(0,0))
          funub(0)=1.0d0
C
            a12=0.5d0*chemof1(DKbs,S1(1,m))*ub1(1,m)
            a12=a12+0.5d0*chemof1(DKbs,S1(0,m))*ub1(0,m)
            b21=0.5d0*chemof1(DKbs,S1(0,m-1))*ub1(0,m-1)
            b21=b21+0.5d0*chemof1(DKbs,S1(0,m))*ub1(0,m)
            c12=0.5d0*chemof1(DKbq,q1(1,m))*ub1(1,m)
```

```
          c12=c12+0.5d0*chemof1(DKbq,q1(0,m))*ub1(0,m)
          d21=0.5d0*chemof1(DKbq,q1(0,m-1))*ub1(0,m-1)
          d21=d21+0.5d0*chemof1(DKbq,q1(0,m))*ub1(0,m)
        rhsub(m)=ub1(0,m)+D6*(ub1(1,m)-ub1(0,m))
          rhsub(m)=rhsub(m)+(ht/2.0d0)*f(cb,vb,h1(0,m))*ub1(0,m)
            rhsub(m)=rhsub(m)-D7*a12*(s1(1,m)-s1(0,m))
            rhsub(m)=rhsub(m)-D7*b21*(s1(0,m-1)-s1(0,m))
            rhsub(m)=rhsub(m)-D8*c12*(q1(1,m)-q1(0,m))
            rhsub(m)=rhsub(m)-D8*d21*(q1(0,m-1)-q1(0,m))
         funub(m)=1.0d0
C
         do 1060 j=1,m-1
C
            t=tr(j,hr,tsw,Ds)
            call rhbound(m,S1(0,j),S1(1,j),rsw,D1,hr,t,rhss(j))
          rhss(j)=rhss(j)-(ht/2.0d0)*f(cas,vas,s1(0,j))*ua1(0,j)/Yas
          rhss(j)=rhss(j)-(ht/2.0d0)*f(cbs,vbs,s1(0,j))*ub1(0,j)/Ybs
          funs(0)=1.0d0
            t=tr(j,hr,tqw,Dq)
            call rhbound(m,q1(0,j),q1(1,j),rqw,D4,hr,t,rhsq(j))
          rhsq(j)=rhsq(j)-(ht/2.0d0)*f(caq,vaq,q1(0,j))*ua1(0,j)/Yaq
          rhsq(j)=rhsq(j)-(ht/2.0d0)*f(cbq,vbq,q1(0,j))*ub1(0,j)/Ybq
          funq(0)=1.0d0
            t=tr(j,hr,thw,Dh)
            call rhbound(m,h1(0,j),h1(1,j),rhw,D0,hr,t,rhsh(j))
          rhsh(j)=rhsh(j)-(ht/2.0d0)*f(ca,va,h1(0,j))*ua1(0,j)/Ya
          rhsh(j)=rhsh(j)-(ht/2.0d0)*f(cb,vb,h1(0,j))*ub1(0,j)/Yb
          funh(0)=1.0d0
C Cell group a
            a12=0.5d0*chemof1(DKas,S1(1,j))*ua1(1,j)
            a12=a12+0.5d0*chemof1(DKas,S1(0,j))*ua1(0,j)
            b12=0.5d0*chemof1(DKas,S1(0,j+1))*ua1(0,j+1)
            b12=b12+0.5d0*chemof1(DKas,S1(0,j))*ua1(0,j)
            b21=0.5d0*chemof1(DKas,S1(0,j-1))*ua1(0,j-1)
            b21=b21+0.5d0*chemof1(DKas,S1(0,j))*ua1(0,j)
            c12=0.5d0*chemof1(DKaq,q1(1,j))*ua1(1,j)
            c12=c12+0.5d0*chemof1(DKaq,q1(0,j))*ua1(0,j)
            d12=0.5d0*chemof1(DKaq,q1(0,j+1))*ua1(0,j+1)
            d12=d12+0.5d0*chemof1(DKaq,q1(0,j))*ua1(0,j)
            d21=0.5d0*chemof1(DKaq,q1(0,j-1))*ua1(0,j-1)
            d21=d21+0.5d0*chemof1(DKaq,q1(0,j))*ua1(0,j)
          rhsua(j)=ua1(0,j)+D2*(ua1(1,j)-ua1(0,j))
            rhsua(j)=rhsua(j)+(ht/2.0d0)*f(ca,va,h1(0,j))*ua1(0,j)
             rhsua(j)=rhsua(j)-D3*a12*(s1(1,j)-s1(0,j))
```

```
            rhsua(j)=rhsua(j)-D3*b12*(S1(0,j+1)-s1(0,j))
            rhsua(j)=rhsua(j)-D3*b21*(S1(0,j-1)-s1(0,j))
            rhsua(j)=rhsua(j)-D5*c12*(q1(1,j)-q1(0,j))
            rhsua(j)=rhsua(j)-D5*d12*(q1(0,j+1)-q1(0,j))
            rhsua(j)=rhsua(j)-D5*d21*(q1(0,j-1)-q1(0,j))
          funua(j)=1.0d0
C
C Cell group b
            a12=0.5d0*chemof1(DKbs,S1(1,j))*ub1(1,j)
            a12=a12+0.5d0*chemof1(DKbs,S1(0,j))*ub1(0,j)
            b12=0.5d0*chemof1(DKbs,S1(0,j+1))*ub1(0,j+1)
            b12=b12+0.5d0*chemof1(DKbs,S1(0,j))*ub1(0,j)
            b21=0.5d0*chemof1(DKbs,S1(0,j-1))*ub1(0,j-1)
            b21=b21+0.5d0*chemof1(DKbs,S1(0,j))*ub1(0,j)
            c12=0.5d0*chemof1(DKbq,q1(1,j))*ub1(1,j)
            c12=c12+0.5d0*chemof1(DKbq,q1(0,j))*ub1(0,j)
            d12=0.5d0*chemof1(DKbq,q1(0,j+1))*ub1(0,j+1)
            d12=d12+0.5d0*chemof1(DKbq,q1(0,j))*ub1(0,j)
            d21=0.5d0*chemof1(DKbq,q1(0,j-1))*ub1(0,j-1)
            d21=d21+0.5d0*chemof1(DKbq,q1(0,j))*ub1(0,j)
          rhsub(j)=ub1(0,j)+D6*(ub1(1,j)-ub1(0,j))
            rhsub(j)=rhsub(j)+(ht/2.0d0)*f(cb,vb,h1(0,j))*ub1(0,j)
             rhsub(j)=rhsub(j)-D7*a12*(s1(1,j)-s1(0,j))
             rhsub(j)=rhsub(j)-D7*b12*(S1(0,j+1)-s1(0,j))
             rhsub(j)=rhsub(j)-D7*b21*(S1(0,j-1)-s1(0,j))
             rhsub(j)=rhsub(j)-D8*c12*(q1(1,j)-q1(0,j))
             rhsub(j)=rhsub(j)-D8*d12*(q1(0,j+1)-q1(0,j))
             rhsub(j)=rhsub(j)-D8*d21*(q1(0,j-1)-q1(0,j))
          funub(j)=1.0d0
 1060       continue
C
           call PDMTRIX(m,D1,funs,aa,bb,cc)
           call TRISOLV(m,aa,bb,cc,rhss,us)
           call PDMTRIX(m,D4,funq,aa,bb,cc)
           call TRISOLV(m,aa,bb,cc,rhsq,uq)
           call PDMTRIX(m,D0,funh,aa,bb,cc)
           call TRISOLV(m,aa,bb,cc,rhsh,uh)
           call PDMTRIX(m,D2,funua,aa,bb,cc)
           call TRISOLV(m,aa,bb,cc,rhsua,uua)
           call PDMTRIX(m,D6,funub,aa,bb,cc)
           call TRISOLV(m,aa,bb,cc,rhsub,uub)
           do 1065 j=0,m
             S2(0,j)=us(j)
             q2(0,j)=uq(j)
```

```
            h2(0,j)=uh(j)
            ua2(0,j)=uua(j)
            ub2(0,j)=uub(j)
 1065     continue
C
          i=m
C
        rhss(0)=S1(m,0)+D1*(S1(m-1,0)-S1(m,0))
          rhss(0)=rhss(0)-(ht/2.0d0)*f(cas,vas,s1(m,0))*ua1(m,0)/Yas
          rhss(0)=rhss(0)-(ht/2.0d0)*f(cbs,vbs,s1(m,0))*ub1(m,0)/Ybs
        funs(0)=1.0d0
        rhsq(0)=q1(m,0)+D4*(q1(m-1,0)-q1(m,0))
          rhsq(0)=rhsq(0)-(ht/2.0d0)*f(caq,vaq,q1(m,0))*ua1(m,0)/Yaq
          rhsq(0)=rhsq(0)-(ht/2.0d0)*f(cbq,vbq,q1(m,0))*ub1(m,0)/Ybq
        funq(0)=1.0d0
        rhsh(0)=h1(m,0)+D0*(h1(m-1,0)-h1(m,0))
          rhsh(0)=rhsh(0)-(ht/2.0d0)*f(ca,va,h1(m,0))*ua1(m,0)/Ya
          rhsh(0)=rhsh(0)-(ht/2.0d0)*f(cb,vb,h1(m,0))*ub1(m,0)/Yb
        funh(0)=1.0d0
        rhss(m)=S1(m,m)+D1*(S1(m-1,m)-S1(m,m))
          rhss(m)=rhss(m)-(ht/2.0d0)*f(cas,vas,s1(m,m))*ua1(m,m)/Yas
          rhss(m)=rhss(m)-(ht/2.0d0)*f(cbs,vbs,s1(m,m))*ub1(m,m)/Ybs
        funs(m)=1.0d0
        rhsq(m)=q1(m,m)+D4*(q1(m-1,m)-q1(m,m))
          rhsq(m)=rhsq(m)-(ht/2.0d0)*f(caq,vaq,q1(m,m))*ua1(m,m)/Yaq
          rhsq(m)=rhsq(m)-(ht/2.0d0)*f(cbq,vbq,q1(m,m))*ub1(m,m)/Ybq
        funq(m)=1.0d0
        rhsh(m)=h1(m,m)+D0*(h1(m-1,m)-h1(m,m))
          rhsh(m)=rhsh(m)-(ht/2.0d0)*f(ca,va,h1(m,m))*ua1(m,m)/Ya
          rhsh(m)=rhsh(m)-(ht/2.0d0)*f(cb,vb,h1(m,m))*ub1(m,m)/Yb
        funh(m)=1.0d0
C Cell group a
            a21=0.5d0*chemof1(DKas,S1(m-1,0))*ua1(m-1,0)
            a21=a21+0.5d0*chemof1(DKas,S1(m,0))*ua1(m,0)
            b12=0.5d0*chemof1(DKas,S1(m,1))*ua1(m,1)
            b12=b12+0.5d0*chemof1(DKas,S1(m,0))*ua1(m,0)
            c21=0.5d0*chemof1(DKaq,q1(m-1,0))*ua1(m-1,0)
            c21=c21+0.5d0*chemof1(DKaq,q1(m,0))*ua1(m,0)
            d12=0.5d0*chemof1(DKaq,q1(m,1))*ua1(m,1)
            d12=d12+0.5d0*chemof1(DKaq,q1(m,0))*ua1(m,0)
        rhsua(0)=ua1(m,0)+D2*(ua1(m-1,0)-ua1(m,0))
          rhsua(0)=rhsua(0)+(ht/2.0d0)*f(ca,va,h1(m,0))*ua1(m,0)
            rhsua(0)=rhsua(0)-D3*a21*(s1(m-1,0)-s1(m,0))
            rhsua(0)=rhsua(0)-D3*b12*(S1(m,1)-s1(m,0))
```

```
            rhsua(0)=rhsua(0)-D5*c21*(q1(m-1,0)-q1(m,0))
            rhsua(0)=rhsua(0)-D5*d12*(q1(m,1)-q1(m,0))
          funua(0)=1.0d0
C
C
            a21=0.5d0*chemof1(DKas,S1(m-1,m))*ua1(m-1,m)
            a21=a21+0.5d0*chemof1(DKas,S1(m,m))*ua1(m,m)
            b21=0.5d0*chemof1(DKas,S1(m,m-1))*ua1(m,m-1)
            b21=b21+0.5d0*chemof1(DKas,S1(m,m))*ua1(m,m)
            c21=0.5d0*chemof1(DKaq,q1(m-1,m))*ua1(m-1,m)
            c21=c21+0.5d0*chemof1(DKaq,q1(m,m))*ua1(m,m)
            d21=0.5d0*chemof1(DKaq,q1(m,m-1))*ua1(m,m-1)
            d21=d21+0.5d0*chemof1(DKaq,q1(m,m))*ua1(m,m)
          rhsua(m)=ua1(m,m)+D2*(ua1(m-1,m)-ua1(m,m))
            rhsua(m)=rhsua(m)+(ht/2.0d0)*f(ca,va,h1(m,m))*ua1(m,m)
            rhsua(m)=rhsua(m)-D3*a21*(s1(m-1,m)-s1(m,m))
            rhsua(m)=rhsua(m)-D3*b21*(s1(m,m-1)-s1(m,m))
            rhsua(m)=rhsua(m)-D5*c21*(q1(m-1,m)-q1(m,m))
            rhsua(m)=rhsua(m)-D5*d21*(q1(m,m-1)-q1(m,m))
          funua(m)=1.0d0
C
C Cell group b
            a21=0.5d0*chemof1(DKbs,S1(m-1,0))*ub1(m-1,0)
            a21=a21+0.5d0*chemof1(DKbs,S1(m,0))*ub1(m,0)
            b12=0.5d0*chemof1(DKbs,S1(m,1))*ub1(m,1)
            b12=b12+0.5d0*chemof1(DKbs,S1(m,0))*ub1(m,0)
            c21=0.5d0*chemof1(DKbq,q1(m-1,0))*ub1(m-1,0)
            c21=c21+0.5d0*chemof1(DKbq,q1(m,0))*ub1(m,0)
            d12=0.5d0*chemof1(DKbq,q1(m,1))*ub1(m,1)
            d12=d12+0.5d0*chemof1(DKbq,q1(m,0))*ub1(m,0)
          rhsub(0)=ub1(m,0)+D6*(ub1(m-1,0)-ub1(m,0))
            rhsub(0)=rhsub(0)+(ht/2.0d0)*f(cb,vb,h1(m,0))*ub1(m,0)
            rhsub(0)=rhsub(0)-D7*a21*(s1(m-1,0)-s1(m,0))
            rhsub(0)=rhsub(0)-D7*b12*(S1(m,1)-s1(m,0))
            rhsub(0)=rhsub(0)-D8*c21*(q1(m-1,0)-q1(m,0))
            rhsub(0)=rhsub(0)-D8*d12*(q1(m,1)-q1(m,0))
          funub(0)=1.0d0
C
C
            a21=0.5d0*chemof1(DKbs,S1(m-1,m))*ub1(m-1,m)
            a21=a21+0.5d0*chemof1(DKbs,S1(m,m))*ub1(m,m)
            b21=0.5d0*chemof1(DKbs,S1(m,m-1))*ub1(m,m-1)
            b21=b21+0.5d0*chemof1(DKbs,S1(m,m))*ub1(m,m)
            c21=0.5d0*chemof1(DKbq,q1(m-1,m))*ub1(m-1,m)
```

```
            c21=c21+0.5d0*chemof1(DKbq,q1(m,m))*ub1(m,m)
            d21=0.5d0*chemof1(DKbq,q1(m,m-1))*ub1(m,m-1)
            d21=d21+0.5d0*chemof1(DKbq,q1(m,m))*ub1(m,m)
          rhsub(m)=ub1(m,m)+D6*(ub1(m-1,m)-ub1(m,m))
           rhsub(m)=rhsub(m)+(ht/2.0d0)*f(cb,vb,h1(m,m))*ub1(m,m)
             rhsub(m)=rhsub(m)-D7*a21*(s1(m-1,m)-s1(m,m))
             rhsub(m)=rhsub(m)-D7*b21*(s1(m,m-1)-s1(m,m))
             rhsub(m)=rhsub(m)-D8*c21*(q1(m-1,m)-q1(m,m))
             rhsub(m)=rhsub(m)-D8*d21*(q1(m,m-1)-q1(m,m))
          funub(m)=1.0d0
C
          do 1070 j=1,m-1
C
             t=tr(j,hr,tse,Ds)
             call rhbound(m,S1(m,j),S1(m-1,j),rse,D1,hr,t,rhss(j))
           rhss(j)=rhss(j)-(ht/2.0d0)*f(cas,vas,s1(m,j))*ua1(m,j)/Yas
           rhss(j)=rhss(j)-(ht/2.0d0)*f(cbs,vbs,s1(m,j))*ub1(m,j)/Ybs
           funs(0)=1.0d0
             t=tr(j,hr,tqe,Dq)
             call rhbound(m,q1(m,j),q1(m-1,j),rqe,D4,hr,t,rhsq(j))
           rhsq(j)=rhsq(j)-(ht/2.0d0)*f(caq,vaq,q1(m,j))*ua1(m,j)/Yaq
           rhsq(j)=rhsq(j)-(ht/2.0d0)*f(cbq,vbq,q1(m,j))*ub1(m,j)/Ybq
           funq(0)=1.0d0
             t=tr(j,hr,the,Dh)
             call rhbound(m,h1(m,j),h1(m-1,j),rhe,D0,hr,t,rhsh(j))
           rhsh(j)=rhsh(j)-(ht/2.0d0)*f(ca,va,h1(m,j))*ua1(m,j)/Ya
           rhsh(j)=rhsh(j)-(ht/2.0d0)*f(cb,vb,h1(m,j))*ub1(m,j)/Yb
           funh(0)=1.0d0
C Cell group a
             a21=0.5d0*chemof1(DKas,S1(m-1,j))*ua1(m-1,j)
             a21=a21+0.5d0*chemof1(DKas,S1(m,j))*ua1(m,j)
             b12=0.5d0*chemof1(DKas,S1(m,j+1))*ua1(m,j+1)
             b12=b12+0.5d0*chemof1(DKas,S1(m,j))*ua1(m,j)
             b21=0.5d0*chemof1(DKas,S1(m,j-1))*ua1(m,j-1)
             b21=b21+0.5d0*chemof1(DKas,S1(m,j))*ua1(m,j)
             c21=0.5d0*chemof1(DKaq,q1(m-1,j))*ua1(m-1,j)
             c21=c21+0.5d0*chemof1(DKaq,q1(m,j))*ua1(m,j)
             d12=0.5d0*chemof1(DKaq,q1(m,j+1))*ua1(m,j+1)
             d12=d12+0.5d0*chemof1(DKaq,q1(m,j))*ua1(m,j)
             d21=0.5d0*chemof1(DKaq,q1(m,j-1))*ua1(m,j-1)
             d21=d21+0.5d0*chemof1(DKaq,q1(m,j))*ua1(m,j)
          rhsua(j)=ua1(m,j)+D2*(ua1(m-1,j)-ua1(m,j))
           rhsua(j)=rhsua(j)+(ht/2.0d0)*f(ca,va,h1(m,j))*ua1(m,j)
            rhsua(j)=rhsua(j)-D3*a21*(s1(m-1,j)-s1(m,j))
```

```
            rhsua(j)=rhsua(j)-D3*b12*(S1(m,j+1)-s1(m,j))
            rhsua(j)=rhsua(j)-D3*b21*(S1(m,j-1)-s1(m,j))
            rhsua(j)=rhsua(j)-D5*c21*(q1(m-1,j)-q1(m,j))
            rhsua(j)=rhsua(j)-D5*d12*(q1(m,j+1)-q1(m,j))
            rhsua(j)=rhsua(j)-D5*d21*(q1(m,j-1)-q1(m,j))
          funua(j)=1.0d0
C
C Cell group b
            a21=0.5d0*chemof1(DKbs,S1(m-1,j))*ub1(m-1,j)
            a21=a21+0.5d0*chemof1(DKbs,S1(m,j))*ub1(m,j)
            b12=0.5d0*chemof1(DKbs,S1(m,j+1))*ub1(m,j+1)
            b12=b12+0.5d0*chemof1(DKbs,S1(m,j))*ub1(m,j)
            b21=0.5d0*chemof1(DKbs,S1(m,j-1))*ub1(m,j-1)
            b21=b21+0.5d0*chemof1(DKbs,S1(m,j))*ub1(m,j)
            c21=0.5d0*chemof1(DKbq,q1(m-1,j))*ub1(m-1,j)
            c21=c21+0.5d0*chemof1(DKbq,q1(m,j))*ub1(m,j)
            d12=0.5d0*chemof1(DKbq,q1(m,j+1))*ub1(m,j+1)
            d12=d12+0.5d0*chemof1(DKbq,q1(m,j))*ub1(m,j)
            d21=0.5d0*chemof1(DKbq,q1(m,j-1))*ub1(m,j-1)
            d21=d21+0.5d0*chemof1(DKbq,q1(m,j))*ub1(m,j)
          rhsub(j)=ub1(m,j)+D6*(ub1(m-1,j)-ub1(m,j))
            rhsub(j)=rhsub(j)+(ht/2.0d0)*f(cb,vb,h1(m,j))*ub1(m,j)
             rhsub(j)=rhsub(j)-D7*a21*(s1(m-1,j)-s1(m,j))
             rhsub(j)=rhsub(j)-D7*b12*(S1(m,j+1)-s1(m,j))
             rhsub(j)=rhsub(j)-D7*b21*(S1(m,j-1)-s1(m,j))
             rhsub(j)=rhsub(j)-D8*c21*(q1(m-1,j)-q1(m,j))
             rhsub(j)=rhsub(j)-D8*d12*(q1(m,j+1)-q1(m,j))
             rhsub(j)=rhsub(j)-D8*d21*(q1(m,j-1)-q1(m,j))
          funub(j)=1.0d0
 1070       continue
C
           call PDMTRIX(m,D1,funs,aa,bb,cc)
           call TRISOLV(m,aa,bb,cc,rhss,us)
           call PDMTRIX(m,D4,funq,aa,bb,cc)
           call TRISOLV(m,aa,bb,cc,rhsq,uq)
           call PDMTRIX(m,D0,funh,aa,bb,cc)
           call TRISOLV(m,aa,bb,cc,rhsh,uh)
           call PDMTRIX(m,D2,funua,aa,bb,cc)
           call TRISOLV(m,aa,bb,cc,rhsua,uua)
           call PDMTRIX(m,D6,funub,aa,bb,cc)
           call TRISOLV(m,aa,bb,cc,rhsub,uub)
           do 1075 j=0,m
             S2(m,j)=us(j)
             q2(m,j)=uq(j)
```

```
            h2(m,j)=uh(j)
            ua2(m,j)=uua(j)
            ub2(m,j)=uub(j)
1075      continue

C take care of interior points:
C
        do 1090 i=1,m-1
C
           t=tr(i,hr,tss,Ds)
           call rh2(m,S1(i,0),S1(i-1,0),S1(i+1,0),rss,D1,hr,t,rhss(0))
           rhss(0)=rhss(0)-(ht/2.0d0)*f(cas,vas,s1(i,0))*ua1(i,0)/Yas
           rhss(0)=rhss(0)-(ht/2.0d0)*f(cbs,vbs,s1(i,0))*ub1(i,0)/Ybs
         funs(0)=1.0d0+D1*hr*tr(i,hr,tss,Ds)
           t=tr(i,hr,tqs,Dq)
           call rh2(m,q1(i,0),q1(i-1,0),q1(i+1,0),rqs,D4,hr,t,rhsq(0))
           rhsq(0)=rhsq(0)-(ht/2.0d0)*f(caq,vaq,q1(i,0))*ua1(i,0)/Yaq
           rhsq(0)=rhsq(0)-(ht/2.0d0)*f(cbq,vbq,q1(i,0))*ub1(i,0)/Ybq
         funq(0)=1.0d0+D4*hr*tr(i,hr,tqs,Dq)
           t=tr(i,hr,ths,Dh)
           call rh2(m,h1(i,0),h1(i-1,0),h1(i+1,0),rhs,D0,hr,t,rhsh(0))
           rhsh(0)=rhsh(0)-(ht/2.0d0)*f(ca,va,h1(i,0))*ua1(i,0)/Ya
           rhsh(0)=rhsh(0)-(ht/2.0d0)*f(cb,vb,h1(i,0))*ub1(i,0)/Yb
         funh(0)=1.0d0+D0*hr*tr(i,hr,ths,Dh)
           t=tr(i,hr,tsn,Ds)
           call rh2(m,S1(i,m),S1(i-1,m),S1(i+1,m),rsn,D1,hr,t,rhss(m))
           rhss(m)=rhss(m)-(ht/2.0d0)*f(cas,vas,s1(i,m))*ua1(i,m)/Yas
           rhss(m)=rhss(m)-(ht/2.0d0)*f(cbs,vbs,s1(i,m))*ub1(i,m)/Ybs
         funs(m)=1.0d0+D1*hr*tr(i,hr,tsn,Ds)
           t=tr(i,hr,tqn,Dq)
           call rh2(m,q1(i,m),q1(i-1,m),q1(i+1,m),rqn,D4,hr,t,rhsq(m))
           rhsq(m)=rhsq(m)-(ht/2.0d0)*f(caq,vaq,q1(i,m))*ua1(i,m)/Yaq
           rhsq(m)=rhsq(m)-(ht/2.0d0)*f(cbq,vbq,q1(i,m))*ub1(i,m)/Ybq
         funq(m)=1.0d0+D4*hr*tr(i,hr,tqn,Dq)
           t=tr(i,hr,thn,Dh)
           call rh2(m,h1(i,m),h1(i-1,m),h1(i+1,m),rhn,D0,hr,t,rhsh(m))
           rhsh(m)=rhsh(m)-(ht/2.0d0)*f(ca,va,h1(i,m))*ua1(i,m)/Ya
           rhsh(m)=rhsh(m)-(ht/2.0d0)*f(cb,vb,h1(i,m))*ub1(i,m)/Yb
         funh(m)=1.0d0+D0*hr*tr(i,hr,thn,Dh)
C
          j=0
C Cell group a
           a12=0.5d0*chemof1(DKas,S1(i+1,0))*ua1(i+1,0)
           a12=a12+0.5d0*chemof1(DKas,S1(i,0))*ua1(i,0)
```

```
          a21=0.5d0*chemof1(DKas,S1(i,0))*ua1(i,0)
          a21=a21+0.5d0*chemof1(DKas,S1(i-1,0))*ua1(i-1,0)
          b12=0.5d0*chemof1(DKas,S1(i,1))*ua1(i,1)
          b12=b12+0.5d0*chemof1(DKas,S1(i,0))*ua1(i,0)
          c12=0.5d0*chemof1(DKaq,q1(i+1,0))*ua1(i+1,0)
          c12=c12+0.5d0*chemof1(DKaq,q1(i,0))*ua1(i,0)
          c21=0.5d0*chemof1(DKaq,q1(i,0))*ua1(i,0)
          c21=c21+0.5d0*chemof1(DKaq,q1(i-1,0))*ua1(i-1,0)
          d12=0.5d0*chemof1(DKaq,q1(i,1))*ua1(i,1)
          d12=d12+0.5d0*chemof1(DKaq,q1(i,0))*ua1(i,0)
        rhsua(0)=ua1(i,0)+D2*(ua1(i+1,0)-ua1(i,0))
     *              +D2*(ua1(i-1,0)-ua1(i,0))
         rhsua(0)=rhsua(0)+(ht/2.0d0)*f(ca,va,h1(i,0))*ua1(i,0)
           rhsua(0)=rhsua(0)-D3*a12*(s1(i+1,0)-s1(i,0))
           rhsua(0)=rhsua(0)-D3*a21*(s1(i-1,0)-s1(i,0))
           rhsua(0)=rhsua(0)-D3*b12*(S1(i,1)-s1(i,0))
           rhsua(0)=rhsua(0)-D5*c12*(q1(i+1,0)-q1(i,0))
           rhsua(0)=rhsua(0)-D5*c21*(q1(i-1,0)-q1(i,0))
           rhsua(0)=rhsua(0)-D5*d12*(q1(i,1)-q1(i,0))
        funua(0)=1.0d0
C
          j=m
C
C
          a12=0.5d0*chemof1(DKas,S1(i+1,m))*ua1(i+1,m)
          a12=a12+0.5d0*chemof1(DKas,S1(i,m))*ua1(i,m)
          a21=0.5d0*chemof1(DKas,S1(i,m))*ua1(i,m)
          a21=a21+0.5d0*chemof1(DKas,S1(i-1,m))*ua1(i-1,m)
          b21=0.5d0*chemof1(DKas,S1(i,m))*ua1(i,m)
          b21=b21+0.5d0*chemof1(DKas,S1(i,m-1))*ua1(i,m-1)
          c12=0.5d0*chemof1(DKaq,q1(i+1,m))*ua1(i+1,m)
          c12=c12+0.5d0*chemof1(DKaq,q1(i,m))*ua1(i,m)
          c21=0.5d0*chemof1(DKaq,q1(i,m))*ua1(i,m)
          c21=c21+0.5d0*chemof1(DKaq,q1(i-1,m))*ua1(i-1,m)
          d21=0.5d0*chemof1(DKaq,q1(i,m))*ua1(i,m)
          d21=d21+0.5d0*chemof1(DKaq,q1(i,m-1))*ua1(i,m-1)
        rhsua(m)=ua1(i,m)+D2*(ua1(i+1,m)-ua1(i,m))
     *              +D2*(ua1(i-1,m)-ua1(i,m))
         rhsua(m)=rhsua(m)+(ht/2.0d0)*f(ca,va,h1(i,m))*ua1(i,m)
           rhsua(m)=rhsua(m)-D3*a12*(s1(i+1,m)-s1(i,m))
           rhsua(m)=rhsua(m)-D3*a21*(s1(i-1,m)-s1(i,m))
           rhsua(m)=rhsua(m)-D3*b21*(S1(i,m-1)-s1(i,m))
           rhsua(m)=rhsua(m)-D5*c12*(q1(i+1,m)-q1(i,m))
           rhsua(m)=rhsua(m)-D5*c21*(q1(i-1,m)-q1(i,m))
```

```
            rhsua(m)=rhsua(m)-D5*d21*(q1(i,m-1)-q1(i,m))
          funua(m)=1.0d0
C
C Cell group b
            a12=0.5d0*chemof1(DKbs,S1(i+1,0))*ub1(i+1,0)
            a12=a12+0.5d0*chemof1(DKbs,S1(i,0))*ub1(i,0)
            a21=0.5d0*chemof1(DKbs,S1(i,0))*ub1(i,0)
            a21=a21+0.5d0*chemof1(DKbs,S1(i-1,0))*ub1(i-1,0)
            b12=0.5d0*chemof1(DKbs,S1(i,1))*ub1(i,1)
            b12=b12+0.5d0*chemof1(DKbs,S1(i,0))*ub1(i,0)
            c12=0.5d0*chemof1(DKbq,q1(i+1,0))*ub1(i+1,0)
            c12=c12+0.5d0*chemof1(DKbq,q1(i,0))*ub1(i,0)
            c21=0.5d0*chemof1(DKbq,q1(i,0))*ub1(i,0)
            c21=c21+0.5d0*chemof1(DKbq,q1(i-1,0))*ub1(i-1,0)
            d12=0.5d0*chemof1(DKbq,q1(i,1))*ub1(i,1)
            d12=d12+0.5d0*chemof1(DKbq,q1(i,0))*ub1(i,0)
          rhsub(0)=ub1(i,0)+D6*(ub1(i+1,0)-ub1(i,0))
     *              +D6*(ub1(i-1,0)-ub1(i,0))
            rhsub(0)=rhsub(0)+(ht/2.0d0)*f(cb,vb,h1(i,0))*ub1(i,0)
            rhsub(0)=rhsub(0)-D7*a12*(s1(i+1,0)-s1(i,0))
            rhsub(0)=rhsub(0)-D7*a21*(s1(i-1,0)-s1(i,0))
            rhsub(0)=rhsub(0)-D7*b12*(S1(i,1)-s1(i,0))
            rhsub(0)=rhsub(0)-D8*c12*(q1(i+1,0)-q1(i,0))
            rhsub(0)=rhsub(0)-D8*c21*(q1(i-1,0)-q1(i,0))
            rhsub(0)=rhsub(0)-D8*d12*(q1(i,1)-q1(i,0))
          funub(0)=1.0d0
C
            j=m
C
C
            a12=0.5d0*chemof1(DKbs,S1(i+1,m))*ub1(i+1,m)
            a12=a12+0.5d0*chemof1(DKbs,S1(i,m))*ub1(i,m)
            a21=0.5d0*chemof1(DKbs,S1(i,m))*ub1(i,m)
            a21=a21+0.5d0*chemof1(DKbs,S1(i-1,m))*ub1(i-1,m)
            b21=0.5d0*chemof1(DKbs,S1(i,m))*ub1(i,m)
            b21=b21+0.5d0*chemof1(DKbs,S1(i,m-1))*ub1(i,m-1)
            c12=0.5d0*chemof1(DKbq,q1(i+1,m))*ub1(i+1,m)
            c12=c12+0.5d0*chemof1(DKbq,q1(i,m))*ub1(i,m)
            c21=0.5d0*chemof1(DKbq,q1(i,m))*ub1(i,m)
            c21=c21+0.5d0*chemof1(DKbq,q1(i-1,m))*ub1(i-1,m)
            d21=0.5d0*chemof1(DKbq,q1(i,m))*ub1(i,m)
            d21=d21+0.5d0*chemof1(DKbq,q1(i,m-1))*ub1(i,m-1)
          rhsub(m)=ub1(i,m)+D6*(ub1(i+1,m)-ub1(i,m))
     *              +D6*(ub1(i-1,m)-ub1(i,m))
```

```
          rhsub(m)=rhsub(m)+(ht/2.0d0)*f(cb,vb,h1(i,m))*ub1(i,m)
             rhsub(m)=rhsub(m)-D7*a12*(s1(i+1,m)-s1(i,m))
             rhsub(m)=rhsub(m)-D7*a21*(s1(i-1,m)-s1(i,m))
             rhsub(m)=rhsub(m)-D7*b21*(S1(i,m-1)-s1(i,m))
             rhsub(m)=rhsub(m)-D8*c12*(q1(i+1,m)-q1(i,m))
             rhsub(m)=rhsub(m)-D8*c21*(q1(i-1,m)-q1(i,m))
             rhsub(m)=rhsub(m)-D8*d21*(q1(i,m-1)-q1(i,m))
        funub(m)=1.0d0
C
          do 1080 j=1,m-1
C
         rhss(j)=S1(i,j)+D1*(S1(i+1,j)-2.0d0*S1(i,j)+S1(i-1,j))
            rhss(j)=rhss(j)-(ht/2.0d0)*f(cas,vas,s1(i,j))*ua1(i,j)/Yas
            rhss(j)=rhss(j)-(ht/2.0d0)*f(cbs,vbs,s1(i,j))*ub1(i,j)/Ybs
         funs(j)=1.0d0
         rhsq(j)=q1(i,j)+D4*(q1(i+1,j)-2.0d0*q1(i,j)+q1(i-1,j))
            rhsq(j)=rhsq(j)-(ht/2.0d0)*f(caq,vaq,q1(i,j))*ua1(i,j)/Yaq
            rhsq(j)=rhsq(j)-(ht/2.0d0)*f(cbq,vbq,q1(i,j))*ub1(i,j)/Ybq
         funq(j)=1.0d0
         rhsh(j)=h1(i,j)+D0*(h1(i+1,j)-2.0d0*h1(i,j)+h1(i-1,j))
            rhsh(j)=rhsh(j)-(ht/2.0d0)*f(ca,va,h1(i,j))*ua1(i,j)/Ya
            rhsh(j)=rhsh(j)-(ht/2.0d0)*f(cb,vb,h1(i,j))*ub1(i,j)/Yb
         funh(j)=1.0d0
C Cell group a
             a12=0.5d0*chemof1(DKas,S1(i+1,j))*ua1(i+1,j)
             a12=a12+0.5d0*chemof1(DKas,S1(i,j))*ua1(i,j)
             a21=0.5d0*chemof1(DKas,S1(i,j))*ua1(i,j)
             a21=a21+0.5d0*chemof1(DKas,S1(i-1,j))*ua1(i-1,j)
             b12=0.5d0*chemof1(DKas,S1(i,j+1))*ua1(i,j+1)
             b12=b12+0.5d0*chemof1(DKas,S1(i,j))*ua1(i,j)
             b21=0.5d0*chemof1(DKas,S1(i,j))*ua1(i,j)
             b21=b21+0.5d0*chemof1(DKas,S1(i,j-1))*ua1(i,j-1)
             c12=0.5d0*chemof1(DKaq,q1(i+1,j))*ua1(i+1,j)
             c12=c12+0.5d0*chemof1(DKaq,q1(i,j))*ua1(i,j)
             c21=0.5d0*chemof1(DKaq,q1(i,j))*ua1(i,j)
             c21=c21+0.5d0*chemof1(DKaq,q1(i-1,j))*ua1(i-1,j)
             d12=0.5d0*chemof1(DKaq,q1(i,j+1))*ua1(i,j+1)
             d12=d12+0.5d0*chemof1(DKaq,q1(i,j))*ua1(i,j)
             d21=0.5d0*chemof1(DKaq,q1(i,j))*ua1(i,j)
             d21=d21+0.5d0*chemof1(DKaq,q1(i,j-1))*ua1(i,j-1)
          rhsua(j)=ua1(i,j)+D2*(ua1(i+1,j)-ua1(i,j))
     *                  +D2*(ua1(i-1,j)-ua1(i,j))
            rhsua(j)=rhsua(j)+(ht/2.0d0)*f(ca,va,h1(i,j))*ua1(i,j)
             rhsua(j)=rhsua(j)-D3*a12*(s1(i+1,j)-s1(i,j))
```

```
            rhsua(j)=rhsua(j)-D3*a21*(s1(i-1,j)-s1(i,j))
            rhsua(j)=rhsua(j)-D3*b12*(s1(i,j+1)-s1(i,j))
            rhsua(j)=rhsua(j)-D3*b21*(s1(i,j-1)-s1(i,j))
            rhsua(j)=rhsua(j)-D5*c12*(q1(i+1,j)-q1(i,j))
            rhsua(j)=rhsua(j)-D5*c21*(q1(i-1,j)-q1(i,j))
            rhsua(j)=rhsua(j)-D5*d12*(q1(i,j+1)-q1(i,j))
            rhsua(j)=rhsua(j)-D5*d21*(q1(i,j-1)-q1(i,j))
          funua(j)=1.0d0
C Cell group b
            a12=0.5d0*chemof1(DKbs,S1(i+1,j))*ub1(i+1,j)
            a12=a12+0.5d0*chemof1(DKbs,S1(i,j))*ub1(i,j)
            a21=0.5d0*chemof1(DKbs,S1(i,j))*ub1(i,j)
            a21=a21+0.5d0*chemof1(DKbs,S1(i-1,j))*ub1(i-1,j)
            b12=0.5d0*chemof1(DKbs,S1(i,j+1))*ub1(i,j+1)
            b12=b12+0.5d0*chemof1(DKbs,S1(i,j))*ub1(i,j)
            b21=0.5d0*chemof1(DKbs,S1(i,j))*ub1(i,j)
            b21=b21+0.5d0*chemof1(DKbs,S1(i,j-1))*ub1(i,j-1)
            c12=0.5d0*chemof1(DKbq,q1(i+1,j))*ub1(i+1,j)
            c12=c12+0.5d0*chemof1(DKbq,q1(i,j))*ub1(i,j)
            c21=0.5d0*chemof1(DKbq,q1(i,j))*ub1(i,j)
            c21=c21+0.5d0*chemof1(DKbq,q1(i-1,j))*ub1(i-1,j)
            d12=0.5d0*chemof1(DKbq,q1(i,j+1))*ub1(i,j+1)
            d12=d12+0.5d0*chemof1(DKbq,q1(i,j))*ub1(i,j)
            d21=0.5d0*chemof1(DKbq,q1(i,j))*ub1(i,j)
            d21=d21+0.5d0*chemof1(DKbq,q1(i,j-1))*ub1(i,j-1)
          rhsub(j)=ub1(i,j)+D6*(ub1(i+1,j)-ub1(i,j))
     *                  +D6*(ub1(i-1,j)-ub1(i,j))
            rhsub(j)=rhsub(j)+(ht/2.0d0)*f(cb,vb,h1(i,j))*ub1(i,j)
             rhsub(j)=rhsub(j)-D7*a12*(s1(i+1,j)-s1(i,j))
             rhsub(j)=rhsub(j)-D7*a21*(s1(i-1,j)-s1(i,j))
             rhsub(j)=rhsub(j)-D7*b12*(s1(i,j+1)-s1(i,j))
             rhsub(j)=rhsub(j)-D7*b21*(s1(i,j-1)-s1(i,j))
             rhsub(j)=rhsub(j)-D8*c12*(q1(i+1,j)-q1(i,j))
             rhsub(j)=rhsub(j)-D8*c21*(q1(i-1,j)-q1(i,j))
             rhsub(j)=rhsub(j)-D8*d12*(q1(i,j+1)-q1(i,j))
             rhsub(j)=rhsub(j)-D8*d21*(q1(i,j-1)-q1(i,j))
          funub(j)=1.0d0
 1080       continue
C
           call PDMTRIX(m,D1,funs,aa,bb,cc)
           call TRISOLV(m,aa,bb,cc,rhss,us)
           call PDMTRIX(m,D1,funq,aa,bb,cc)
           call TRISOLV(m,aa,bb,cc,rhsq,uq)
           call PDMTRIX(m,D0,funh,aa,bb,cc)
```

```
          call TRISOLV(m,aa,bb,cc,rhsh,uh)
          call PDMTRIX(m,D2,funua,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhsua,uua)
          call PDMTRIX(m,D6,funub,aa,bb,cc)
          call TRISOLV(m,aa,bb,cc,rhsub,uub)
          do 1085 j=0,m
            S2(i,j)=us(j)
            q2(i,j)=uq(j)
            h2(i,j)=uh(j)
            ua2(i,j)=uua(j)
            ub2(i,j)=uub(j)
1085      continue
1090      continue
C
        do 1095 j=0,m
          do 1096 i=0,m
            S1(i,j)=S2(i,j)
            q1(i,j)=q2(i,j)
            h1(i,j)=h2(i,j)
            ua1(i,j)=ua2(i,j)
            ub1(i,j)=ub2(i,j)
1096            continue
1095            continue
C
2000    continue
C
      call veloc(mvel,lvel,distm,timm,distl,timl,velm,vell)
      write(30,*)'timemax,distmax,velm'
      write(31,*)'timelim,distlim,vell'
      kvel2=mcount2/numvel
        do 2060 i=1,mvel-1
          write(30,*)timm(i),',',distm(i),',',velm(i)
2060    continue
        do 2062 i=1,lvel-1
          write(31,*)timl(i),',',distl(i),',',vell(i)
2062    continue
     if(movc.eq.2)then
       write(16,*)'];'
       write(17,*)'];'
       write(23,*)'];'
       write(18,*)'];'
     endif
     stop
     end
```

```
C
C
C The following four functions give the initial conditions
C
C
      function ua0(x,y,uaini,yoff,xoff,awid)
      implicit double precision (a-h,o-z)
 301  format(6x,d10.9)
       r=sqrt((x-xoff)**2+(y-yoff)**2)
       if(r.le.0.3d0) then
        ua0=uaini/dexp(awid*r)
       else
        ua0=0.0d0
       endif
      return
      end
C
      function ub0(x,y,ubini,yoff,xoff,bwid)
      implicit double precision (a-h,o-z)
 301  format(6x,d10.9)
       r=sqrt((x-xoff)**2+(y-yoff)**2)
       if(r.le.0.3d0) then
        ub0=ubini/dexp(bwid*r)
       else
        ub0=0.0d0
       endif
      return
      end
C
C
C The function f(x) is the uptake of nutrient
C
      function f(ca,v,x)
      implicit double precision (a-h,o-z)
      if(x.ge.0.0d0) then
      f=v*x/(ca+x)
          else
            f=0.0d0
          endif
      return
      end
C
C fs=f(x)/x
      function fs(c,v,x)
```

```
      implicit double precision (a-h,o-z)
      if(x.ge.0.0d0) then
      fs=v/(c+x)
          else
           fs=0.0d0
          endif
      return
      end
C
C chemof1*chemof2: non-linear chemotaxis coefficient function
C chemof1 is defined by Keller-Segel model
C
      function chemof1(DK,x)
      implicit double precision (a-h,o-z)
      if(x.ge.0.0d0) then
        chemof1=DK/(DK+x)**2
         else
          chemof1=0.0d0
         endif
      return
      end
C
C function diffu is defined by Monod's law
C
      function diffu(v2,c2,eps,x)
      implicit double precision (a-h,o-z)
      if(x.ge.0.0d0) then
        diffu=v2*x/(c2+x)+eps
         else
           diffu=eps
         endif
      return
      end
C tr(i): sets tr to 0 if outside window, or to desired value inside window
      function tr(i,h,tk,dxy)
      implicit double precision (a-h,o-z)
      x=-2.5d0+i*h
      y=abs(x)
          if(y.le.2.0d0) then
            tr=tk/dxy
          else
            tr=0.0d0
          endif
      return
```

```
      end
C
C rhbound: subroutine to calculate boundary conditions at
C         sink or reservoir.
C
      subroutine rhbound(n,a,b,c,d,hr,t,rh)
      implicit double precision (a-h,o-z)
C     dimension a(0:n),b(0:n),rh(0:n)
      rh=a+d*(b-a-hr*t*(a-c))
      return
      end
C rh2: subroutine to calculate boundary conditions at
C         sink or reservoir.
C
      subroutine rh2(n,a,b,e,c,d,hr,t,rh)
      implicit double precision (a-h,o-z)
C     dimension a(0:n),b(0:n),rh(0:n)
      rh=a+d*(b-2*a+e+hr*t*c)
      return
      end
C
C maxcalc: finds maximum density on wave
C
      subroutine maxcalc(ua1,k,m,ht,hr,umax,pmax,t)
      implicit double precision (a-h,o-z)
      dimension ua1(0:100,0:100)
      mcent=m/2
      imax=0
 5    if(ua1(mcent,imax+1).ge.ua1(mcent,imax))then
           imax=imax+1
           goto 5
      endif
      umax=ua1(mcent,imax)
      pmax=2.5d0-float(imax)*hr
C     write(*,*)'pmax=',pmax
      t=k*ht
C     write(*,*)'t=',t
      return
      end
C
C limcalc: finds first point where minimum cell density occurs
C
      subroutine limcalc(ua1,k,m,ht,hr,dmin,ulim,plim,t)
      implicit double precision (a-h,o-z)
```

```
      dimension ua1(0:100,0:100)
      mcent=m/2
      imax=0
 5    if(ua1(mcent,imax+1).lt.dmin)then
           imax=imax+1
           goto 5
      endif
      ulim=ua1(mcent,imax)
      plim=2.5d0-float(imax)*hr
C     write(*,*)'plim=',plim
      t=k*ht
C     write(*,*)'t=',t
      return
      end
C
      subroutine veloc(mvel,lvel,distm,timm,distl,timl,velm,vell)
      implicit double precision (a-h,o-z)
      dimension distm(0:100),distl(0:100)
      dimension timm(0:100),timl(0:100)
      dimension velm(0:100),vell(0:100)
      do 2089 i=2,mvel-1
           velm(i-1)=(distm(i)-distm(i-1))/(timm(i)-timm(i-1))
 2089 continue
      do 2099 i=2,lvel-1
           vell(i-1)=(distl(i)-distl(i-1))/(timl(i)-timl(i-1))
 2099 continue
      return
      end
C
C SOLVES THE TRIDIAGONAL SYSTEM OF LINEAR EQUATIONS
C
      SUBROUTINE TRISOLV(N,A,B,C,FUN,X)
      implicit double precision (a-h,o-z)
      DIMENSION A(0:N),B(0:N),C(0:N),FUN(0:N),X(0:N)
      DIMENSION ARFA(0:500),BATA(0:500),GAMA(0:500),GUN(0:500)
C
      B(0)=0.0d0
      C(N)=0.0d0
C
      ARFA(0)=A(0)
      GAMA(0)=C(0)
      DO 4000 I=1,N
           GAMA(I)=C(I)
        BATA(I)=B(I)/ARFA(I-1)
```

```
      ARFA(I)=A(I)-BATA(I)*C(I-1)
4000  CONTINUE
      GUN(0)=FUN(0)
      DO 4999 I=1,N
        GUN(I)=FUN(I)-BATA(I)*GUN(I-1)
4999  CONTINUE
      X(N)=GUN(N)/ARFA(N)
      DO 4888 I=1,N
        X(N-I)=(GUN(N-I)-GAMA(N-I)*X(N-I+1))/ARFA(N-I)
4888  CONTINUE
C
      RETURN
      END
C
C SET UP THE TRIDIAGONAL MATRIX for constant coefficient terms
C
      SUBROUTINE PDMTRIX(n,dxy,fu,aa,bb,cc)
      implicit double precision (a-h,o-z)
      DIMENSION fu(0:n),aa(0:n),bb(0:n),cc(0:n)
          bb(0)=0.0
          aa(0)=fu(0)+1.0d0*dxy
          cc(0)=-1.0d0*dxy
          bb(n)=-1.0d0*dxy
          aa(n)=fu(n)+1.0d0*dxy
          cc(n)=0.0
C
          do 5000 i=1,n-1
            bb(i)=-dxy
            cc(i)=-dxy
            aa(i)=fu(i)+2.0d0*dxy
5000      continue
C
          return
          end
```

APPENDIX E

APPENDIX E

Input files for model

Sample in put files for the program given in Appendix E. The "param" file sets the values of all parameters, and the time for solution. The "init" file sets the initial conditions for each population and chemical component. The "time" file specifies which time-points should be printed to the output files.

Sample param file:

```
m=   41
timinit= 06.0d0
timinoc= 30.0d0
numcel=    2
numvel=   100
kstep =    20
dmin =  0.000001d0
movc=      1      1=individual matrices, 2=movie
R   = 2.5d0
ayoff=  0.0d0
axoff=  0.0d0
byoff=  0.0d0
bxoff=  0.0d0
awid =  1.0d0
bwid =  1.0d0
Dh   = 0.01d0
thn  = 0.155d0
ths  = 0.155d0
the  = 0.000d0
thw  = 0.000d0
Ds   = 0.033d0
tsn  = 0.1071d0
tss  = 0.1071d0
tse  = 0.000d0
tsw  = 0.000d0
Dacs =  0.03d0
Dbcs =  0.00d0
DKas =  0.00000200d0
```

```
DKbs = 0.00000200d0
cas  = 0.0000550d0
vas  = 0.600d0
Yas  = 1.0000d0
cbs  = 0.0000550d0
vbs  = 0.000d0
Ybs  = 1.0000d0
Dq   = 0.033d0
tqn  = 0.891d0
tqs  = 0.891d0
tqe  = 0.000d0
tqw  = 0.000d0
Dacq = 0.080d0
Dbcq = 0.00d0
DKaq = 0.0000330d0
DKbq = 0.0000330d0
caq  = 0.000000067d0
vaq  = 0.02d0
Yaq  = 1.0000d0
cbq  = 0.000000067d0
vbq  = 0.0d0
Ybq  = 1.0000d0
Dua  = 0.00100d0
va   = 0.35d0
ca   = 0.00000408d0
Ya   = 0.350d0
Dub  = 0.0010d0
vb   = 0.50d0
cb   = 0.00000408d0
Yb   = 0.50d0
```

Sample init file:

```
uaini= 0.000005d0
ubini= 0.000005d0
rsn  = 0.000132d0
rss  = 0.00000d0
rse  = 0.0000d0
rsw  = 0.0000d0
rqn  = 0.0000132d0
rqs  = 0.0000132d0
rqe  = 0.0000132d0
rqw  = 0.0000132d0
```

```
rhn  =  0.00046d0
rhs  =  0.00046d0
rhe  =  0.00046d0
rhw  =  0.00046d0
stini=  0.00000d0
qtini=  0.0000132d0
htini=  0.00046d0
```

Sample time file:

```
numti=04
time1= 10.0d0
time2= 15.0d0
time3= 20.0d0
time4= 30.0d0
```

APPENDIX F

APPENDIX F

Matlab program files

The following Matlab program files were created for use with Matlab 4.2c for Windows. No guarantees are made that they will work with more recent versions. Commands enclosed in quotes should be entered into Matlab exactly as typed.

The file appmult.m plots the cell matrix as a density plot. The simulation plots in Figure 13 were created using appmult.m, for example. The syntax is "appmult(u,#,'pcolor')" where u is the name of the matrix to be plotted, and # is the amount of interpolation; typically 2 or 3 works well. (Note: I did not write this file. It is modified from the version found at the Mathworks site, www.mathworks.com.)

```
function hfi = app_int(x,y,z,s,dc)
% This function approximates interpolated shading by interpolating
% data and using flat shading.  The inputs are the data that was used
% to generate the original object, a scale factor, and a string
% that contains the drawing command. The
% function returns a handle to the new
% object.
%
% Syntax 1:For just Z-Data (e.g., surf(z) => app_int(z,s,dc))
% z = zdata, s = scaling factor and dc = drawing command used to
% create the plot
%
% Syntax 2: For x,y,z Data ( e.g., surf(x,y,z) ) => app_int
% (x,y,z,s,dc)
%
% Example 1: [x,y,z] = peaks;surf(z);shading interp
% app_int(z,3,'surf')   interpolate by a factor of 3
%
% Example 2: [x,y,z ] = peaks; surf(x,y,z); shading interp
% app_int(x,y,z,3,'surf')interpolate by a factor of 3
if ( nargin == 3 )
  dc = z;
```

```
 z = x;
 s = y;
 [m n] = size(z);
 x = 1:n; y = (1:m)';
 mscal = m*s;
 nscal = n*s;
 xi = linspace(1,n,nscal);
 yi = linspace(1,m,mscal)';
 zi = interp2(x,y,z,xi,yi);
% figure;
 cmd = ['hfi=' dc '(xi,yi,zi);'];
 eval(cmd);
 shading flat;
elseif ( nargin == 5 )
 [m n] = size(z);
 mscal = m*s;
 nscal = n*s;
 mint = min(min(x));
 maxt = max(max(x));
 xi = linspace(mint,maxt,mscal);
 mint = min(min(y));
 maxt = max(max(y));
 yi = linspace(mint,maxt,nscal)';
 zi = interp2(x,y,z,xi,yi);
 figure;
 cmd = ['hfi=' dc '(xi,yi,zi);'];
 eval(cmd);
 shading flat;
end
 colormap(gray)
 axis('square')
```

The program surftwo.m allows two cell matrices to be plotted on the same figure, with different colormaps. Examples of this program are Figure 33C and Figure 33D. The syntax is "surftwo(u1,u2)" where u1 and u2 are the two cell matrices.

```
%Program to plot two cell populations on one figure
%with different colormaps.
function[h]=surftwo(ua,ub)
h(1)=surf(ua),axis equal
```

```
hold on
h(2)=surf(ub);
hold off
m=64;
cmin=min(ua(:));
cmax=max(ua(:));
c1=min(m,round((m-1)*(ua-cmin)/(cmax-cmin))+1);
cmin=min(ub(:));
cmax=max(ub(:));
c2=min(m,round((m-1)*(ub-cmin)/(cmax-cmin))+1)+64;
set(h(1),'cdata',c1);
set(h(2),'cdata',c2);
caxis([min(c1(:)) max(c2(:))]);
view(2);
shading interp;
colormap([bone(64);copper(64)]);
```

The file centmass.m calculates the average mass for two cell matrices. An example is Figure 35F. The syntax is "[cma,cmb]=centmass(u1,u2)" where u1 and u2 are the cell matrices. The returned variables cma and cmb may be plotted with "plot(cma), hold, plot(cmb)".

```
function[cma,cmb]=centmass(ua,ub)
%calculate the total mass along line perpendicular to gradient,
%by summing across each line parallel to gradient.
[r,c]=size(ua);
centline=round(r/2);
for i=1:r;
cma(i)=sum(ua(i,:))/(r-1);
cmb(i)=sum(ub(i,:))/(r-1);
end
```

The file competit.m computes the dynamic competition factor for two cell populations. Examples are Figure 35A and Figure 35B. The syntax is "competit", where

the value for num (the number of timepoints) and the cell matrices, have already been entered into the Matlab variable space. The returned variable rab is the dynamic competition factor, and is the ratio of the total mass of Population A to the total mass of Population B. A plot can be created using "plot(rab)".

```
%Computes total mass of a and b and ratio a/b for uai, ubi where i=1:num
%To use, first specify value for num, then type competit
clear ta,clear tb,clear rab
for i=1:num;
[ta(i)]=totmass(eval(['ua',num2str(i)]),1.6);
[tb(i)]=totmass(eval(['ub',num2str(i)]),1.6);
rab(i)=ta(i)/tb(i);
end
```

The file totmass.m is a subroutine called by the program competit.m.

```
function[tmass]=totmass(u,ht)
% Calculates total mass under cell profile
% Assumes length and width of DGC = 5 cm
% ht is the height of the gel, usually 1.6 cm
% Find number of increments
[r,c]=size(u);
% Compute area under 2-D plot, multiply by l*w*h for mass
tmass=trapz(trapz(u))*5/(r-1)*5/(c-1)*ht;
```

The program totvel.m calculates the mass fluxes due to chemotaxis to S, chemotaxis to Q, random motility, and the total mass flux. Examples are Figure 21-Figure 24. The syntax is

"[uxs,uys,uxq,uyq,uxu,uyu,uxt,uyt,mag,dir,dxc,dyc,aoverh]=totvel(kds,xos,kdq,xoq,du,s, q,u,minv)" where kds and kdq are the dissociation constants for the receptor-S and

receptor-Q complexes, respectively; xos and xoq are the chemotactic sensitivity coefficients for S and Q; du is the random motility coefficient; s, q, and u are the S-chemoattractant, Q-chemoattractant, and cell matrices, respectively; and minv is a minimum flux below which no values are reported. The returned variables uxs and uys are the x and y components of the flux due to chemotaxis to S; uxq and uyq are the x and y components of the flux due to chemotaxis to Q; uxu and uyu are the x and y components of the flux due to random motility; and uxt and uyt are the x and y components of the total mass flux. The fluxes can be plotted with the quiver command, as illustrated near the end of the program.

```
function[uxs,uys,uxq,uyq,uxu,uyu,uxt,uyt,mag,dir,dxc,dyc,aoverh]=totvel(kds,xos
        ,kdq,xoq,du,s,q,u,minv)
%This program calculates the flux matrices for chemotaxis to S and Q,
%(uxs,uys,uxq,uyq), random motility (uxu,uyu), and the total flux (uxt,uyt).
[m,n]=size(u);
dir=zeros(m,n);
%Calls to the subroutine chemo, which is where intermediate fluxes actually
        calculated.
[uxs,uys]=chemo(kds,xos,s,u,minv);
[uxq,uyq]=chemo(kdq,xoq,q,u,minv);
[uxu,uyu]=randmot(du,u,minv);
%Total flux found by summing components of flux.
uxt=uxs+uxq+uxu;
uyt=uys+uyq+uyu;
%Magnitude of maximum flux calculated.
mag=sqrt(uxt.*uxt+uyt.*uyt);
%Maximum flux magnitude found.
maxvel=max(max(abs(mag)));
%A minimum flux value to be graphed is calculated. Minv=5 works well. (80% of
        fluxes graphed).
minvel=maxvel/minv;
%Loop to remove fluxes whose magnitudes are lower than minvel.
for i = 1:m,
        for j = 1:n;
                if abs(mag(i,j)) < minvel
                        uxt(i,j)=0;
```

receptor-Q complexes, respectively; xos and xoq are the chemotactic sensitivity coefficients for S and Q; du is the random motility coefficient; s, q, and u are the S-chemoattractant, Q-chemoattractant, and cell matrices, respectively; and minv is a minimum flux below which no values are reported. The returned variables uxs and uys are the x and y components of the flux due to chemotaxis to S; uxq and uyq are the x and y components of the flux due to chemotaxis to Q; uxu and uyu are the x and y components of the flux due to random motility; and uxt and uyt are the x and y components of the total mass flux. The fluxes can be plotted with the quiver command, as illustrated near the end of the program.

```
function[uxs,uys,uxq,uyq,uxu,uyu,uxt,uyt,mag,dir,dxc,dyc,aoverh]=totvel(kds,xos
        ,kdq,xoq,du,s,q,u,minv)
%This program calculates the flux matrices for chemotaxis to S and Q,
%(uxs,uys,uxq,uyq), random motility (uxu,uyu), and the total flux (uxt,uyt).
[m,n]=size(u);
dir=zeros(m,n);
%Calls to the subroutine chemo, which is where intermediate fluxes actually
        calculated.
[uxs,uys]=chemo(kds,xos,s,u,minv);
[uxq,uyq]=chemo(kdq,xoq,q,u,minv);
[uxu,uyu]=randmot(du,u,minv);
%Total flux found by summing components of flux.
uxt=uxs+uxq+uxu;
uyt=uys+uyq+uyu;
%Magnitude of maximum flux calculated.
mag=sqrt(uxt.*uxt+uyt.*uyt);
%Maximum flux magnitude found.
maxvel=max(max(abs(mag)));
%A minimum flux value to be graphed is calculated. Minv=5 works well. (80% of
        fluxes graphed).
minvel=maxvel/minv;
%Loop to remove fluxes whose magnitudes are lower than minvel.
for i = 1:m,
        for j = 1:n;
                if abs(mag(i,j)) < minvel
                        uxt(i,j)=0;
```

```
                    uyt(i,j)=0;
                end
            end
        end
        %subplot(2,2,1),quiver(uxs,uys,3,'w'),title('chemo s'),axis square
        %subplot(2,2,2),quiver(uxq,uyq,3,'w'),title('chemo q'),axis square
        %subplot(2,2,3),quiver(uxu,uyu,3,'w'),title('randmot'),axis square
        %subplot(2,2,4),quiver(uxt,uyt,3,'w'),title('total flux'),axis square
        dxc=zeros(m,n);
        dyc=zeros(m,n);
        for i = 1:m,
            for j = 1:n;
                if mag(i,j)~=0
                    dxc(i,j)=uxt(i,j)/mag(i,j);
                    dyc(i,j)=uyt(i,j)/mag(i,j);
                end
            end
        end
```

The file chemo.m is called as a subroutine in totvel.m.

```
function[ux,uy,satx,saty]=chemo(kd,xo,c,u,minv)
%calculate the partial derivatives of c in the x and y directions
[m,n]=size(c);
dx=5/m;
dy=5/n;
[px,py]=gradient(c,dx,dy);
%calculate the matrix kd/(kd+c)^2
for i = 1:m,
    for j = 1:n;
        sat(i,j)=kd/(kd+c(i,j))^2;
    end
end
%calculate the velocity coefficient matrix kd/(kd+c)^2*px(or py)
%the .* operator causes only the corresponding two matrix positions to be
multiplied
%rather than real matrix multiplication
satx=xo*(sat.*px);
saty=xo*(sat.*py);
%clear all values of u below a minimum value
maxu=max(max(u));
```

```
minu=maxu/1000;
for i = 1:m,
        for j = 1:n;
                if u(i,j) < minu
                        u(i,j)=0;
                end
        end
end
%multiplies the cell density times the velocity coefficient
ux=u.*satx;
uy=u.*saty;
mag=sqrt(ux.*ux+uy.*uy);
maxvel=max(max(abs(mag)));
minvel=maxvel/minv;
%maxvely=max(max(abs(uy)));
%minvely=maxvely/minv;
for i = 1:m,
        for j = 1:n;
                if abs(mag(i,j)) < minvel
                        ux(i,j)=0;
                        uy(i,j)=0;
                end
        end
end
```

The file randmot.m is called as a subroutine in the program totvel.m

```
function[ux,uy]=randmot(du,u,minv)
%calculate the partial derivatives of u in the x and y directions
[px,py]=gradient(u);
%calculate the velocity coefficient matrix du*px(or py)
%the .* operator causes only the corresponding two matrix positions to be
        multiplied
%rather than real matrix multiplication
%clear all values of u below a minimum value
[m,n]=size(u);
maxu=max(max(u));
minu=maxu/1000;
for i = 1:m,
        for j = 1:n;
                if u(i,j) < minu
```

```
                    u(i,j)=0;
                end
            end
        end
        ux=du*u.*px;
        uy=du*u.*py;
        mag=sqrt(ux.*ux+uy.*uy);
        maxvel=max(max(abs(mag)));
        minvel=maxvel/minv;
        for i = 1:m,
            for j = 1:n;
                if abs(mag(i,j)) < minvel
                    ux(i,j)=0;
                    uy(i,j)=0;
                end
            end
        end
```

The program "retard4.m" calculates the global chemotactic response factor. An example is Figure 25. The syntax is "[phi,den,num,dent]=retard4(u,s,kd,perct)" where u is the cell matrix, s is the chemoattractant S matrix, kd is the dissociation constant for the receptor-S complex, and perct is the lowest percent value of the maximum cell concentration for which a value of the global chemotactic response factor will be calculated. The global chemotactic response factor is given by the returned variable phi.

```
function[phi,den,num,dent]=retard4(u,s,kd,perct)
% Calculation of retardation factor.
% Retard4 calculates factor along y-direction for all x.
% perct is the minimum percentage of the maximum cell concentration above
%       which the cell concentration will not be assumed to be 0.
% Right now, only works when gradient in direction of matrix columns.
% Calculation of s-gradient
[m,n]=size(u);
dy=5/m;
maxui=max(max(u));
minui=perct*maxui;
```

```
for i=1:n,
        grads=abs(gradient(s(:,i)',dy)');
        % Calculate (1+s/kd)) term.
        grp1=(1+s(:,i)/kd);
        % Calculate (1+s/kd)^(-2)*gradient(s)
        gsgrp1=grp1.^(-2).*grads;
        % Calculate integrand of numerator of retardation factor
        num=gsgrp1.*u(:,i);
        for j=1:m,
                if u(j,i) < minui
                        num(j)=0;
                end
        end
% Calculate integrand of denominator of retardation factor
        den=grads.*u(:,i);
% Evaluate integrals with trapezoid method. Calculate phi.
%       phi=zeros(size(pos));
%Make den a function of i, so average maximum flux can be reported.
        dent(i)=trapz(den);
        phi(i)=trapz(num)/trapz(den);
end
```

APPENDIX G

APPENDIX G

Dimensionless model

Over the course of the competition modeling studies, it became apparent that there was an almost infinite number of parameter combinations to be tested. Many times, it is possible to reduce the number of parameters in a model by making it dimensionless, and then varying the dimensionless groups.

The competition model was made dimensionless by employing the following scales (the ' indicates the dimensional quantity):

$$t = v_{aH} t' \tag{57}$$

$$u_i = \frac{u_i'}{u_{i0}} \text{ for i=a,b} \tag{58}$$

$$j = \frac{j'}{j_0} \text{ for j=H,S,Q} \tag{59}$$

$$\nabla = \nabla' \sqrt{\frac{D_H}{v_{aH}}} \tag{60}$$

When these scales were substituted into the dimensional equations and the equations were rearranged, the new dimensionless equations were:

$$\frac{\partial u_a}{\partial t} = \lambda_a \nabla^2 u_a - \lambda_a \delta_{aS} \nabla \cdot \left[\left(\frac{\theta_{aS}}{(\theta_{aS} + S)^2} \right) u_a \nabla S \right] - \lambda_a \delta_{aQ} \nabla \cdot \left[\left(\frac{\theta_{aQ}}{(\theta_{aQ} + Q)^2} \right) u_a \nabla Q \right] + \frac{H}{\psi_{aH} + H} u_a \quad (61)$$

$$\frac{\partial u_b}{\partial t} = \lambda_b \nabla^2 u_b - \lambda_b \delta_{bS} \nabla \cdot \left[\left(\frac{\theta_{bS}}{(\theta_{bS} + S)^2} \right) u_b \nabla S \right] - \lambda_b \delta_{bQ} \nabla \cdot \left[\left(\frac{\theta_{bQ}}{(\theta_{bQ} + Q)^2} \right) u_b \nabla Q \right] + \frac{\pi_H H}{\psi_{bH} + H} u_b \quad (62)$$

$$\frac{\partial S}{\partial t} = \sigma_S \nabla^2 S - \beta_{aS} \frac{S}{\psi_{aS} + S} u_a - \pi_H \beta_{bS} \frac{S}{\psi_{bS} + S} u_b \quad (63)$$

$$\frac{\partial Q}{\partial t} = \sigma_Q \nabla^2 Q - \beta_{aQ} \frac{Q}{\psi_{aQ} + Q} u_a - \pi_H \beta_{bQ} \frac{Q}{\psi_{bQ} + Q} u_b \quad (64)$$

$$\frac{\partial H}{\partial t} = \nabla^2 H - \phi_a \frac{H}{\psi_{aH} + H} u_a - \pi_H \phi_b \frac{H}{\psi_{bH} + H} u_b \quad (65)$$

where the dimensionless variables are defined as:

$$\lambda_i = \frac{\mu_i}{D_H} \text{ for i=a,b Measure of random motility to nutrient diffusion} \quad (66)$$

$$\delta_{ij} = \frac{\chi_{oij}}{\mu_{ij}} \text{ for i=a,b; j=S,Q Measure of chemotaxis to random motitility} \quad (67)$$

$$\theta_{ij} = \frac{K_{dij}}{j_0} \text{ for i=a,b} \quad (68)$$

$$\psi_{ij} = \frac{C_{ij}}{j_o} \text{ for i=a,b; j=S,Q Saturation constant over initial concentration} \quad (69)$$

$$\pi_j = \frac{v_{bj}}{v_{aj}} \text{ for j=S,Q,e,d Max. growth rate of b on j to max. growth rate of a on j} \quad (70)$$

$$\beta_{ij} = \frac{v_{ij}u_{io}}{v_{iH}j_0} \text{ for i=a,b; j=S,Q,H,e Compares max uptake of i on j to rate of i on H} \quad (71)$$

$$\sigma_j = \frac{D_j}{D_H} \text{ for j=S,Q Ratio of diffusivity of j to diffusivity of H} \quad (72)$$

$$\phi_i = \frac{u_{i0}}{H_0 Y_{iH}} \text{ for i=a,b; Stoichiometric concentration ratio} \quad (73)$$

By comparing the dimensionless equations (Equations (61)-(65)) to the dimensional equations (Equations (38)-(40)), it is obvious that the number of parameters has been reduced by only one. The introduction of the dimensionless groups has not significantly simplified the model.

$$\psi_{ij} = \frac{C_{ij}}{j_o} \text{ for i=a,b; j=S,Q Saturation constant over initial concentration} \quad (69)$$

$$\pi_j = \frac{v_{bj}}{v_{aj}} \text{ for j=S,Q,e,d Max. growth rate of b on j to max. growth rate of a on j} \quad (70)$$

$$\beta_{ij} = \frac{v_{ij}u_{io}}{v_{iH}j_0} \text{ for i=a,b; j=S,Q,H,e Compares max uptake of i on j to rate of i on H} \quad (71)$$

$$\sigma_j = \frac{D_j}{D_H} \text{ for j=S,Q Ratio of diffusivity of j to diffusivity of H} \quad (72)$$

$$\phi_i = \frac{u_{i0}}{H_0 Y_{iH}} \text{ for i=a,b; Stoichiometric concentration ratio} \quad (73)$$

By comparing the dimensionless equations (Equations (61)-(65)) to the dimensional equations (Equations (38)-(40)), it is obvious that the number of parameters has been reduced by only one. The introduction of the dimensionless groups has not significantly simplified the model.

LIST OF REFERENCES

LIST OF REFERENCES

Abu-Ashour, J., D. M. Joy, H. Lee, H. R. Whiteley, and S. Zelin. 1994. Transport of Microorganisms through Soil. Water, Air and Soil Pollution. **75**: 141-158.

Adler, J. 1966. Chemotaxis in Bacteria. Science. 153, 708-716.

Adler, J. 1972. A Method for Measuring Chemotaxis and Use of the Method to Determine Optimum Conditions for Chemotaxis by *Escherichia coli*. Journal of General Microbiology. **74**: 77-91.

Agladze, K., L. Budriene, G. Ivanitsky, V. Krinsky, V. Shakhbazyan, and M. Tsyganov. Wave mechanisms of pattern formation in microbial populations. Proc. R. Soc. Lond. 253: 131-135.

Ames, T. 1997. A Continuous Magnetofluidized Bed Bioreactor for Production of Plant-Cell Secondary Metabolites. Doctoral Dissertation. Michigan State University.

Bailey, J. E. and D. F. Ollis. 1986. Biochemical Engineering Fundamentals. 2nd edition. McGraw-Hill Inc., New York.

Barton, J. W. and R. M. Ford. 1997. Mathematical Model for Characterization of Bacterial Migration through Sand Cores. Biotechnology and Bioengineering. 53: 487-496.

Berg, H. C. 1988. A Physicist Looks at Bacterial Chemotaxis, p. 1-9. *In* Cold Spring Harbor Symposia on Quantitative Biology, Vol. III.

Berg, H. C. and P. M. Tedesco. 1975. Transient response to chemotactic stimuli in Escherichia coli. Proc. Nat. Acad. Sci. USA. **72**: 3235-3239.

Berg, H.C. and D. A. Brown. 1972. Chemotaxis in Escherichia coli Analysed by Three-Dimensional Tracking. Nature. **239**:500-504.

Boon, J., and B. Herpigny. 1986. Model for chemotactic bacteria bands. Bull. Math. Biol. **48**:1-19.

Bosma, T. N. P., J. L. Schnoor, G. Schraa, and A. J. B. Zehnder. 1988. Simulation Model for Biotransformation of Xenobiotics and Chemotaxis in Soil Columns. J. of Contaminant Hydrogeology. **2**: 225-236.

Budrene, E. O. and H. C. Berg. 1995. Dynamics of formation of symmetrical patterns by chemotactic bacteria. Nature. 376: 49-53.

Caldwell, D. E. and P. Hirsch. 1972. Growth of microorganisms in two-dimensional steady-state diffusion gradients. Can. J. Microbiol. 19: 53-58.

Carnahan, B., H. A. Luther, and J. O. Wilkes. Approximation of the Solution of Partial Differential Equations, p. 429-464. *In* Applied Numerical Methods. John Wiley and Sons, Inc. 1969.

Chapra, S. C. and R. P. Canale. Parabolic Equations in Two Spatial Dimensions, p. 745-748. *In* Numerical Methods for Engineers. McGraw-Hill Book Company. 1988.

Chet, I. and R. Mitchell. 1976. Ecological Aspects of Microbial Chemotactic Behavior. Ann. Rev. Microbiol. 30:221-39.

Criddle, C. S., J. T. DeWitt, D. Grbic-Galic, and P. L. McCarty. 1990. Transformation of Carbon Tetrachloride by *Pseudomonas* sp. Strain KC under Denitrification Conditions. Appl. Environ. Microbiol. **56**:3240-3246.

Cussler, E. L. 1984. Diffusion: Mass transfer in fluid systems. 1st edition. Cambridge University Press, Cambridge.

Devare , M. and M. Alexander. 1995. Bacterial Transport and Phenanthrene Biodegradation in Soil and Aquifer Sand. Soil. Sci. Am. J. 59: 1316-1320.

Duffy, K. J., P. T. Cummings, and R. M. Ford. 1995. Random Walk Calculations for Bacterial Migration in Porous Media. Biophysical Journal. **68**:800-806.

Dybas, M.J., G.M. Tatara, and C.S. Criddle. 1995. Localization and Characterization of the Carbon Tetrachloride Transforming of *Pseudomonas* sp. strain KC. Appl. Environ. Microbiol. **61**:758-762.

Emerson, D., R. M. Worden, and J. A. Breznak. 1994. A Diffusion Gradient Chamber for Studying Microbial Behavior and Separating Microorganisms. *Appl. Environ. Microbiol.* 60,1269-1278.

Emerson, D., S. F. Peteu, and R. M. Worden. 1996a. A Catalase Microbiosensor for Detecting Hydrogen Peroxide. Biotechnology Techniques. 10:673-678.

Emerson, D., S. Peteu, M. Widman, and M. Worden. 1996b. A Microbiosensor for Sensing Glucose, Galactose, or Choline, and its Application for Measuring Glucose Gradients in Semi-solid Gels. The Sixth International Meeting on Chemical Sensors. NIST, Gaithersburg, MD. July 22-25, 1996.

Ford, R. M. 1992. Mathematical Modeling and Quantitative Characterization of Bacterial Motility and Chemotaxis, p. 177-215. *In* C. J. Hurst (ed.), Modeling the Metabolic and Physiologic Activities of Microorganisms. John Wiley & Sons, Inc. New York.

Ford, R. M. and D. A. Lauffenburger. 1990. Measurement of Bacterial Random Motility and Chemotaxis Coefficients: II. Application of Single-Cell-Based Mathematical Model. Biotech. Bioeng. **37**: 661-672.

Ford, R. M., B. R. Phillips, J. A. Quinn, and D. A. Lauffenburger. 1990. Measurement of Bacterial Random Motility and Chemotaxis Coefficients: I. Stopped-Flow Diffusion Chamber Assay. Biotech. Bioeng. **37**: 647-660.

Frymier, P. D., R. M. Ford and P. T. Cummings. 1992. Cellular Dynamics Simulations of Bacterial Chemotaxis. Chemical Engineering Science. **48**:687-699.

Frymier, P. D., R. M. Ford and P. T. Cummings. 1994. Analysis of Bacterial Migration: 1. Numerical Solution of Balance Equation. AIChe Journal. **40**: 704-715.

Hansen, S. R. and S. P. Hubbell. 1980. Single-Nutrient Microbial Competition: Qualitative Agreement Between Experimental and Theoretically Forecast Outcomes. Science. 207: 1491-1493.

Harwood, C. S., R. E. Porales, and M. Dispensa. 1990. Chemotaxis of *Pseudomonas putida* toward Chlorinated Benzoates. Appl. Environ. Microbiol. **56**: 1501-1503.

Holmes, E. E., M. A. Lewis, J. E. Banks, and R. R. Veit. Partial Differential Equations in Ecology: Spatial Interactions and Population Dynamics. Ecology. 75: 17-29.

Kato, J., A. Ito, T. Nikata and H. Ohtake. 1992. *Phosphate Taxis in Pseudomonas aeruginosa.* J. Bacteriol. **174**: 5149-5151.

Keller, E. F. and L. A. Segel. 1971. Travelling Bands of Chemotactic Bacteria: a Theoretical Analysis. J. Theor. Biol. **30**: 235-248.

Kelly, F. X., K. J. Dapsis, and D. A. Lauffenburger. 1988. Effect of Bacterial Chemotaxis on Dynamics of Microbial Competition. Microb. Ecol. **16**: 115-131.

Kennedy, M. J. and J. G. Lawless. 1985. Role of Chemotaxis in the Ecology of Denitrifiers. Appl. Environ. Microbiol. **49**: 109-114.

Lauffenburger, D. A., R. Aris, and K. Keller. 1982. Effects of Cell Motility and Chemotaxis on Microbial Population Growth. Biophys. J. **40**: 209-219.

Lauffenburger, D., and B. Calcagno P. 1983. Competition Between Two Microbial Populations in a Nonmixed Environment: Effect of Cell Random Motility. Biotech. and Bioeng. **25**: 2130-2125.

Lauffenburger, D., R. Aris, and K. H. Keller. 1981. Effects of random motility on growth of bacterial populations. Microb. Ecol. 7:207-227.

Liu, Z. and K. D. Papadopoulos. 1995. Unidirectional Motility of *Escherichia coli* in Restricitve Capillaries. Applied and Environmental Microbiology. 61: 3567-3572.

Liu, Z., W. Chen, and K. D. Papadopoulos. 1996. Bacterial Motility, Collisions, and Aggregation in a 3-μm-Diameter Capillary. Biotechnology and Bioengineering. 53: 1-4.

Macnab, R. M. 1987. Motility and Chemotaxis, p. 732-759. *In* F. C. Neidhardt, J. L. Ingraham, K. B. Low, B. Magasanik, M. Schaechter, and H. E. Umbarger (ed.), *Escherichia coli* and *Salmonella typhimurium:* Cellular and Molecular Biology, vol. 1. American Society for Microbiology, Washington, D. C.

Maniatis, T., E. F. Fritsch, and J. Sambrook. 1982. "Molecular Cloning: A Laboratory Manual." Cold Spring Harbor Laboratory, p. 68.

Mayotte, T.J., M.J. Dybas, and C.S. Criddle. 1996. Bench-Scale Evaluation of Bioaugmentation to Remediate Carbon-Tetrachloride Contaminated Aquifer Materials. Ground Water, **34**: 358-367.

Mesibov, R., G. W. Ordal, and J. Adler. 1973. The Range of Attractant Concentrations for Bacterial Chemotaxis and the Threshold and Size of Response over This Range. J. Gen. Physiol. **62**: 203-223.

Mikola, Mark. 1996. Optimization of Biocatalytic 3-Dehydroshikimic Acid Production from D-Glucose in *Escherichia coli*. Master's Thesis. Michigan State University.

Nikata, T., K. Sumida, J. Kato and H. Ohtake. 1992. Rapid Method for Analyzing Bacterial Behavioral Responses to Chemical Stimuli. Appl. Environ. Microbiol. **58**: 2250-2254.

Nossal, R. 1972. Boundary Movement of Chemotactic Bacterial Populations. Mathematical Biosciences. **13**: 397-406.

Perry, R. H. and D. Green (eds.). 1984. Perry's Chemical Engineers' Handbook, 6th Edition. McGraw-Hill Book Company, pp. 3-258 to 3-287.

Peteu, S. F., D. Emerson, and R. M. Worden. 1996. A Clark-type oxidase enzyme-based amperometric microbiosensor for sensing glucose, galactose, or choline. Biosensors and Bioelectronics. **11**:1059-1071.

Revsbech, N. P. 1989. An oxygen microsensor with a guard cathode. Limnol. Oceanogr. **34**: 474-478.

Rivero, M. A., R. T. Tranquillo, H. M. Buettner and D. A. Lauffenburger. 1989. Transport Models for Chemotactic Cell Populations Based on Individual Cell Behavior. Chemical Engineering Science. **44**: 2881-2897.

Rivero-Hudec, M. and D. A. Lauffenburger. 1986. Quantification of Bacterial Chemotaxis by Measurement of Model Parameters Using the Capillary Assay. Biotechnology and Bioengineering. **28**: 1178-1190.

Sarkar, A. K., G. Georgiou, and M. M. Sharma. 1994a. Transport of Bacteria in Porous Media: I. An Experimental Investigation. Biotech. and Bioeng. 44: 489-497.

Sarkar, A. K., G. Georgiou, and M. M. Sharma. 1994b. Transport of Bacteria in Porous Media: II. A Model for Convective Transport and Growth. Biotech. and Bioeng. **44**: 499-508.

Schmidt, S., M. T. Widman, and R. M. Worden. 1997. A Laser-Diffraction Capillary Assay for Measuring Microbial Random Motility. Biotech. Techniques. 11: 423-426.

Segel, L. A. 1977. A Theoretical Study of Receptor Mechanisms in Bacterial Chemotaxis. J. Appl. Math. **32**: 653-665.

Sitaraman, R. S., S. H. Ibrahim, and N. R. Kuloor. 1963. J. Chem. Eng. Data. **8**:198.

Soby, S. and K. Bergman. 1983. Motility and Chemotaxis of Rhizobium meliloti in Soil. Applied and Environmental Microbiology. **46**: 995-998.

Staffeld, P. O., D. A. Lauffenburger, and J. A. Quinn. 1987. Mathematical Analysis of Cell Transport Phenomena: Bacterial Chemotaxis in the Capillary Assay. Chem. Eng. Comm. **58**: 339-351.

Strauss, I, P. D. Frymier, C. M. Hahn, and R. M. Ford. 1995. Analysis of Bacterial Migration: II. Studies with Multiple Attractant Gradients. AIChE Journal. **41**: 402-414.

Tan, Y., J. T. Gannon, P. Baveye, and M. Alexander. 1994. Transport of bacteria in an aquifer sand: Experiments and model simulations. Water Resources Research. 30: 3243-3252.

Tatara, G. M., M. J. Dybas, and C. S. Criddle. 1993. Effects of Medium and Trace Metals on Kinetics of Carbon Tetrachloride Transformation by *Pseudomonas* sp. Strain KC. Appl. Environ. Microbiol. **49**: 109-114.

Tilman, D. 1994. Competition and Biodiversity in Spatially Structured Habitats. Ecology. 75: 2-16.

Widman, M. T., D. Emerson, C. C. Chiu, and R. M. Worden. 1996. Modeling Microbial Chemotaxis in a Diffusion Gradient Chamber. Biotech. and Bioeng. 55: 191-205.

Wilke, C. R. and P. Chang. 1955. Correlation of Diffusion Coefficients in Dilute Solutions. AIChE J. **1**: 264-270.

Witt, M.E. 1994. Development of a Laboratory-Scale Model Aquifer System to Monitor a Carbon-Tetrachloride-Transforming Zone by *Pseudomonas* sp. strain KC, M.S. Thesis, Michigan State University, East Lansing, MI.

Wolfe, A. J., and H. C. Berg. 1989. Migration of Bacteria in Semisolid Agar. Proc. Natl. Acad. Sci. USA **86**: 6973-6977.

Woodward, D. E., R. Tyson, M. R. Myerscough, J. D. Murray, E. O. Budrene, and H. C. Berg. Spatio-Temporal Patterns Generated by *Salmonella typhimurium*. Biophysical Journal. 68: 2181-2189.

Yamamoto, K. and Y. Imae. 1993. Cloning and Characterization of the *Salmonella typhimurium*-specific chemoreceptor Tcp for taxis to citrate from phenol. Proc. Natl. Acad. Sci. USA. **90**: 217-221.

Zhulin, I. B., E. H. Roswell, M. S. Johnson, and B. L. Taylor. 1997. Glycerol Elicits Energy Taxis of *Escherichia coli* and *Salmonella typhimurium*. J. Bacteriology. 179: 3196-3201.