

This is to certify that the

thesis entitled

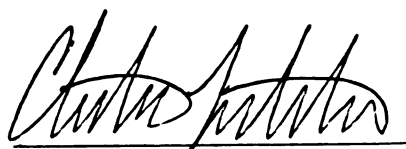
Cellular Dynamics Simulation of Microbial
Chemotaxis in Porous Media

presented by

Jason R. Mondro

has been accepted towards fulfillment
of the requirements for

Master's degree in Chemical Engineering



Major professor

Date

5/7/02



PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

**CELLULAR DYNAMICS SIMULATION OF MICROBIAL CHEMOTAXIS IN
POROUS MEDIA**

By

Jason R. Mondro

A THESIS

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

MASTER OF SCIENCE

Department of Chemical Engineering and Materials Science

2002

ABSTRACT

CELLULAR DYNAMICS SIMULATION OF MICROBIAL CHEMOTAXIS IN POROUS MEDIA

By

Jason R. Mondro

Chemotaxis refers to the directed migration of bacteria in response to a chemical gradient. Engineered applications of chemotaxis offer the promise of solving difficulties associated with in situ bioremediation for the removal of soil and groundwater pollutants. Chemotaxis also plays an important role in the human immune system, and control of chemotactic response may improve retention of surgically implanted medical devices. Thus, an understanding of the mechanisms of chemotaxis, and strategies for its manipulation is desired.

A *cellular dynamics* (CD) simulation method has been developed for the analysis of chemotactic cell migration in porous media in response to gradients of consumable chemoattractants. The simulation results are used to interpret experimental cell motility measurements in heterogeneous porous aquifer solids. In CD simulations, macroscopic cell population motility coefficients are calculated from the motion characteristics of individual cells (i.e. swimming speed, tumbling frequency, turn angle distribution).

Using the CD simulation method, the motility and displacement of bacteria has been investigated in bulk and porous environments, in response to consumable and nonconsumable chemoattractants. The simulation results provide validation of several experimental phenomena observed for migration of bacteria in porous aquifer solids.

DEDICATION

I would like to dedicate this thesis to my parents Thomas and Josephine for their support and encouragement and especially to my wife Melissa for her patience and love.

ACKNOWLEDGMENTS

I would like to express my appreciation to Dr. Christian Lastoskie for all the help and guidance in conducting this research. I would like to thank Dr. Mark Worden for his insight into the experimental behavior of bacteria cells.

I would like to thank Dr. Syed Hashsham for teaching me about the microbiology of bacteria in a class that I really enjoyed and to Dr. Melissa Baumann for her inspiration into the biomedical application of chemotaxis and implanted devices.

I would like to thank Ana Sarcheck for her help in running simulations and trying to keep me organized. I would like to thank Sydney Forrester for his help in programming the subroutine to place random particles in the model and for his expertise in using MatLab to make the movies showing cell movement in front of the changing chemoattractant concentration field.

I would also like to acknowledge Laura Booms and Caroline Roush for their experimental photographs of interesting bacteria behavior.

TABLE OF CONTENTS

LIST OF TABLES.....	vii
LIST OF FIGURES.....	viii
NOMENCLATURE	xii
1. INTRODUCTION	1
1.1 BIOREMEDIATION BACKGROUND	1
1.1.1 Motivation and Severity of problem.....	1
1.1.2 Definition of problem.....	2
1.1.3 Relation to chemotaxis solution	3
1.2 BIOMEDICAL BACKGROUND.....	5
1.2.1 Motivation and Severity of problem.....	5
1.2.2 Definition of problem	5
1.2.3 Relation to chemotaxis solution	6
1.3 OBJECTIVE.....	7
2. LITERATURE BACKGROUND	8
2.1 CHEMOTACTIC MECHANISM.....	8
2.2 CURRENT EXPERIMENTAL PROTOCOLS.....	11
2.3 CURRENT MODELING PROTOCOLS	11
2.4 IMPROVED SIMULATIONS USING CELLULAR DYNAMICS.....	12
3. SIMULATION METHODOLOGY	15
3.1 CD SIMULATION MECHANICS	15
3.2 GRID SPACING	19
3.3 STABILITY	21
3.4 ASSUMPTIONS	24
3.5 DIMENSIONALITY.....	25
4. LINEAR GRADIENT NON-CONSUMABLE	25
4.1 SINGLE CELL TRAJECTORY	26
4.2 EFFECTS OF VARYING CHEMOTACTIC SENSITIVITY COEFFICIENT.....	28
4.3 EFFECTS OF VARYING CELL SWIMMING VELOCITY	32
4.4 EFFECTS OF VARYING THE TUMBLING PROBABILITY	34
4.5 GAUSSIAN TURN ANGLE DISTRIBUTIONS	35
4.6 COMPARISON OF <i>E. COLI</i> AND <i>P. PUTIDA</i> MOTILITY	37
4.7 EFFECTS OF THE MEAN AND VARIANCE OF THE TURN ANGLE DISTRIBUTION.....	39

5. CONSUMABLE NON POROUS	43
5.1 INITIAL DISTRIBUTION.....	46
5.2 DIFFUSION.....	47
5.3 NO DIFFUSION.....	48
5.4 ERROR ANALYSIS	50
5.5 ALTERNATIVE METHOD OF GRADIENT DETECTION	50
5.6 DEPENDENCE OF CONCENTRATION ON CHEMOTAXIS.....	55
6. MIGRATION IN POROUS MEDIA	57
6.1 SLIT PORES.....	59
6.2 MULTISLIT BARRIER.....	64
6.2.1 Non-reflective barrier	68
6.2.2 Reflective barrier.....	76
6.3 HETEROGENEOUS	80
6.3.1 Migration into a strip of porous media	82
6.3.2 Migration in a bed of porous media	88
7. VISUALIZATION TOOLS	97
7.1 EXCEL	97
7.2 ARRAY VISUALIZER	98
7.3 MATLAB	98
8. CONCLUSIONS & FUTURE WORKS.....	99
8.1 CONCLUSIONS.....	99
8.2 SUGGESTED TOPICS FOR FUTURE RESEARCH.....	102
8.2.1 3-D gradient detection and 3-D heterogeneous porous.....	102
8.2.2 Multiple attractants/repellent consumption and or excretion	102
8.2.3 Adsorption and energy tabulation to determine growth / death	102
APPENDIX A.....	103
APPENDIX B.....	142
APPENDIX C.....	143
REFERENCES.....	145

LIST OF TABLES

Table 1. Values used in the simulations of a non-consumable linear gradient of α -methylaspartate for <i>E. coli</i>	27
Table 2. Mean, median, and peak values of bacteria turn angle distributions.	38
Table 3. Values used in the simulations starting with a uniform concentration of aspartate consumed by <i>E. coli</i>	44

LIST OF FIGURES

Figure 1. Schematic of flagella coordination by <i>E. coli</i> during Run and Tumble.	9
Figure 2. Biasing of run lengths in the presence of a chemoattractant gradient.	10
Figure 3. Simplified flow chart of model methodology.....	17
Figure 4. Schematic of the grid network with attractant being consumed based on cell locations.	18
Figure 5. Comparison of adjusting consumption based on grid spacing.	21
Figure 6. Cell trajectory comparison.....	27
Figure 7. Snapshots of chemotactic and non-chemotactic cells in a linear gradient.....	28
Figure 8. Snapshots of cell location along a linear non-consumable gradient.....	30
Figure 9. Cell density profile in a linear gradient at varying chemotactic sensitivity coefficients	31
Figure 10. Chemotactic Sensitivity versus Motility and Mean Distance.....	32
Figure 11. Velocity versus Motility and Mean Distance.	33
Figure 12. Tumbling Probability versus Motility and Mean Distance.....	34
Figure 13. Normalized turn angle distribution varying mean.	36
Figure 14. Normalized turn angle distribution varying variance.	37
Figure 15. Turn angle distribution comparison of <i>E. coli</i> and <i>P. putida</i>	38
Figure 16. Cell density profile snapshot of <i>E. coli</i> and <i>P. putida</i> at 30 minutes.....	39
Figure 17. Turn angle mean value versus motility and cell mean distance in a linear nonconsumable gradient.	40
Figure 18. Variance of the Turn Angle versus the motility and cell mean distance in a linear non-consumable gradient.	41
Figure 19. Comparison of the mean turn angle in 2D, 3D against theoretical equation...	42
Figure 20. Comparison of the turn angle variance in 2D, 3D against theoretical equation.	42
Figure 21. Multiple Chemotactic waves of bacteria due to consumable chemoattractant.	44

Figure 22. Snapshots with Consumption showing the cell density patterns.	45
Figure 23. Radial cell density at different consumption rates.....	46
Figure 24. Effect of the consumption coefficient on motility with diffusion.	48
Figure 25. Motility with diffusion and no diffusion on a logarithmic scale.	49
Figure 26. Cell density profile where chemoattractant is depleted at center.	51
Figure 27. Cell density profile where chemoattractant is not depleted at center.	51
Figure 28. Comparison of different chemotactic sensitivities on the cell density profile.	52
Figure 29. Relative cell density profiles of gradient sensing methods with χ_o of 3.5E-5.	54
Figure 30. Relative cell density profiles of gradient sensing methods with χ_o of 105E-4	55
Figure 31. Cell density profile and concentration profile due to consumption.....	56
Figure 32. The effect of concentration on cell density in a decaying step gradient.....	57
Figure 33. Schematic of cell trajectory through a bed of spheres.....	58
Figure 34. Photograph of <i>Pseudomonas KC</i> migrating faster through sand core sample.	58
Figure 35. Simulation results of cells in a slit pore.....	59
Figure 36. Effect of pore width on motility at different consumption rates.....	60
Figure 37. Motility at different population consumption rates.	61
Figure 38. Effect of pore width on motility with different number of cells.....	62
Figure 39. Comparison of motilities in pores with consumption and no consumption. ...	63
Figure 40. Cell density profile for decaying step gradient in different pore widths.	64
Figure 41. Migration into multiple 100-micron slit pores with a porosity of 33%.	65
Figure 42. Migration into multiple 100-micron slit pores with a porosity of 50%.	65
Figure 43. Cell migration into multislits with a uniform initial concentration.	69
Figure 44. Radial cell density profile and concentration profile with a uniform initial concentration.....	69
Figure 45. Cell migration into multislits with a lower initial concentration in the pores.	71

Figure 46. Radial cell density profile and concentration profile with a lower initial concentration in the pores.	71
Figure 47. Effect of chemoattractant concentration on penetration ratio.....	72
Figure 48. Effect of chemoattractant concentration on the motility coefficient.	72
Figure 49. Enhanced migration with initial lower concentration in multiple slit pores....	73
Figure 50. Enhanced migration with initial uniform concentration in multiple slit pores.	74
Figure 51. Comparison of the penetration ratio on short and long multislits.	74
Figure 52. Comparison of the motility coefficient on short and long multislits.	75
Figure 53. The effect of porosity on the penetration ratio and motility in multislits.....	75
Figure 54. Comparison of reflective and non-reflective walls of a 200-micron multislit.	76
Figure 55. Simulation results of reflective 900-micron multislit pores.	77
Figure 56. Simulation result of 100-micron pore width multislit with reflective walls....	78
Figure 57. Comparison of 2D and 3D motion on penetration ratio.	79
Figure 58. Comparison of 2D and 3D motion on motility coefficient.....	79
Figure 59. Cell trajectory through packed bed of 2D disks	80
Figure 60. Cell trajectory through packed bed of cylinders	81
Figure 61. Cell trajectory through a packed bed of spheres.....	81
Figure 62. Motility of different shape 500-micron particles	82
Figure 63. Simulation results showing cells migrating into porous region.....	83
Figure 64. Simulation result of cells starting out in a line along the porous region.	84
Figure 65. Cell density profile for initial cell location of point versus line.	85
Figure 66. Motility results for different particle diameter in strip.	85
Figure 67. Penetration ratio result for different particle diameter in strip.	86
Figure 68. Motility coefficient and Penetration ratio of 2D simulations with a porous strip.....	87
Figure 69. Migration in ordered 400-micron diameter 2D particles with 63% porosity. .	88

Figure 70. Migration in random 400-micron diameter 2D particles with 64.6% porosity.	89
Figure 71. Migration in random 400-micron diameter 2D particles with 38.5% porosity.	90
Figure 72. Migration in random 400-micron diameter 3D particles with 51% porosity.	91
Figure 73. Effect of particle size on the motility coefficient in a bed of pores.....	92
Figure 74. Effect of porosity on the motility coefficient in 400-micron particle bed.	93
Figure 75. Effect of porosity on motility coefficient in 500-micron particle bed.....	94
Figure 76. Cell density profiles for 2D and 3D motion with non-porous consumption. ..	96
Figure 77. Cell density profiles for 2D and 3D motion in pores with consumption.....	97
Figure 78. MatLab output of tracking cell location and concentration in porous media. .	99

NOMENCLATURE

NP	Number of cells
t	Time step
X	Length along positive X axis
Y	Length along positive Y axis
DG	Grid spacing
p_o	Tumbling probability (Berg and Brown 1972)
v	Cell swimming velocity (Strauss et al. 1995)
C	Chemoattractant concentration
K_d	Dissociation constant (Rivero and Lauffenburger 1986)
χ_o, χ_o^{3D}	Chemotactic sensitivity (Rivero and Lauffenburger 1986)
$\Omega(t)$	Mean displacement of the cells
μ_o	Random motility coefficient
μ	Chemotactic motility coefficient
$r_1(t)$	Position of cell at time t
D	Diffusion coefficient (Berg and Brown 1972)
ν	Differential tumbling probability
N_T	Number of chemo receptor sites per cell
ϕ	Turn angle resulting from a cell tumble
S	Direction vector of the cell
k_o	Consumption rate
k	Consumption rate
ΔX	Length between nodal points
W	Weight distance of cell to nodal point
σ	Variance
θ	Turn angle
θ_{Mean}	Mean turn angle
D_{eff}	Effective diffusion coefficient
ε	Porosity or void fraction
τ	Tortuosity

1. INTRODUCTION

Two main areas for applications of chemotaxis include bioremediation of contaminated water and soil and in the biomedical area such as in medical implants. Chemotaxis has the potential to overcome several difficulties associated with *in situ* bioremediation. By enhancing the dissemination of cells throughout a contaminated aquifer, chemotaxis can help overcome slow advective transport and reduce fouling near injection wells. By increasing microbial competitiveness, the chemotactic trait improves the likelihood that an introduced microorganism will colonize the subsurface. Also, by increasing cell motility, chemotaxis can extend the scope of the colonized region, thus lowering capital costs and operating expenses by effectively reducing the number of wells needed for delivery of cells and nutrients at the field site.

Chemotaxis also plays an important role in the recovery and successfulness of surgery involving medical implants. Both bacteria and tissue cells utilize chemotaxis and race to colonize surfaces of implants. The immune response and the healing process produce chemotactic factors that are attractants for tissue cells and may act as repellants to some bacteria. However, bacteria can migrate to the surface of an implant by different methods. By modeling the response of motile bacteria in the body to chemoattractants and chemorepellants, methods can be developed to repel bacteria from the implant site thereby minimizing the chance of infection.

1.1 BIOREMEDIATION BACKGROUND

1.1.1 Motivation and Severity of problem

Over the years, solvents have other organic pollutants have been indiscriminately dumped without consideration of the effects on the environment or on the people living in

the communities that are in close proximity to the dumpsites. According to a recent report, “About 72 Million U.S. citizens live within 4 miles of a USEPA Super fund National Priority List Site and 4.4 million people live within 1 mile.”(Penny and Haldeman, 1994) We can no longer live by the mottos “out of sight out of mind” or “dilution is the solution”. We have a moral obligation to dispose of chemicals properly as well as to clean up our past mistakes. Therefore, how important is it to live in a clean and safe environment? To what lengths should we go to achieve this goal?

It has been estimated that by using traditional decontamination methods, the cost to “clean up” or remediate these sites could approach \$1 trillion dollars (usgs.gov, 2000). This estimate includes only the direct costs of clean up and not the costs associated with the degradation of quality land, the associated medical problems, and the effects upon the quality of life of local residents. Bioremediation, on the other hand, has the potential to reduce costs, treat more sites, and remediate them faster than traditional methods. An excellent example of a case that would benefit from bioremediation is contamination by chlorinated solvents, because of the nature and toxicity of these solvents; priority in remediation should be given to these chemicals.

1.1.2 Definition of problem

Bioremediation involves the use of biological methods to remove a pollutant from an environment like soil or ground water. In this research we will be concerned with *in situ* bioremediation, which involves the addition of non-native bacteria that have the ability to break down the polluting contaminant. The inoculated bacteria need to compete with indigenous bacteria for nutrients. A trait that can give it a competitive advantage is chemotaxis. Chemotaxis helps overcome difficulty in bioremediation by enhancing dispersion of bacteria into the contaminated area. This will help reduce the amount of

fouling and clogging at the injection point. It can also reduce the number of injection wells and the overall cost.

An important aspect in bioremediation is the motility of the cells. If the cells are non-motile or swim very slowly, the cells cannot reach the contaminated areas. A second aspect is the cell diameter to the soil pore free space ratio. If the bacteria cell is too large or the pores are clogged then the motility of the cell is hindered. Third assuming that the cells can swim freely between the soil particles, how do we know which way they will go? A bioremediation project will not be a success if the bacteria avoid the contaminant. When choosing a bacteria cell for bioremediation a property of the cell that is needed, if not required, is that the cell does not see the contaminant as a repellent.

1.1.3 Relation to chemotaxis solution

In bioremediation projects, a novel and effective property would be for the bacteria to see the contaminant or a nutrient that is present in the contaminant as an attractant. This property is called chemotaxis. With chemotaxis, bacteria will stay in the areas of higher contaminant concentration for a longer period of time compared to a bacteria utilizing only random motion (Duffy et al., 1995).

For some types of bacteria, oxygen is a chemoattractant, which can cause problems with bioremediation (Widman et al., 1997). The oxygen concentration will be highest next to the injection well and will drop off as the water containing the oxygen diffuses into the soil. This gradient of oxygen will cause migration to the injection well, which can be in the opposite direction of the increasing contaminant concentration. This mass migration is also the cause of well plugging. This leads to the question “How do we know which concentration gradient the bacteria will more likely pursue?” Studies of chemotaxis on multiple concentration gradients show the model that best fits experiments is a simple

additivity of the gradients (Strauss, 1995). More recent research indicates that each receptor doesn't act independently and the activation of one receptor will increase the activity of another receptor by 100-fold and the whole array of receptors may be involved. (Gestwicki, 2002).

Highly chlorinated chemicals in anoxic conditions undergo reductive dechlorination, which involves removing chlorine atoms one at a time. Many bacteria in anoxic conditions can degrade highly chlorinated chemicals, but cannot degrade lightly chlorinated compounds. The problem with partial degradation is that the byproducts can be more toxic than the original compound. An example is the degradation of TCE to vinyl chloride. TCE (5 ppm) is a suspected carcinogen, while vinyl chloride (2 ppm) is a known carcinogen (EPA, 1997). Another example of this is carbon tetrachloride, which can be degraded to the suspected carcinogen chloroform (Tatera et al., 1993). Even though these lightly chlorinated chemicals can be degraded using aerobic bacteria, the contaminated environment is not oxic so it won't degrade. The addition of oxygen would help to degrade the lightly chlorinated compounds, however the degradation of the highly chlorinated compounds would stop.

Oxygen is a good electron acceptor and yields significant energy for cell growth, metabolism and motility. One of the problems is that oxygen is not very soluble in water. Only 9.1 mg/liter of oxygen is soluble in water at 20 °C (Newton, 1999). Since oxygen is not very soluble it cannot be transported longer distances from the injection well. To overcome the solubility problem more injection wells are needed with more additions of oxygen added more often which will raise the cost of this remediation process.

Cells in an oxic environment grow faster than in an anoxic environments. The logic, that bioremediation will be enhanced by a larger number of cells from faster growth may have some faults. The faster rate of growth of cells has the possibility of clogging the pore spaces between the soil particles, resulting in a sequestered cell population that can't get to the contaminant.

1.2 BIOMEDICAL BACKGROUND

1.2.1 Motivation and Severity of problem

Biomaterials and artificial prosthetics have been used by ancient Egyptians and Greeks and were made out of plant material, wood and metals (Jacoby, 2001). Modern devices including orthopedic joints, catheters, pacemakers, and cosmetic enhancements are all made from a variety of different materials from metals to polymers and transplanted tissues. A major cause of complication with implanted devices is infection, which can result in sickness and death. While the infection rate in the 1960's for people undergoing a hip implant was 6.8%, it was still around 1% in the 1990's. The infection rate for knee and elbow implants is 2 to 9%, however it approaches 100% for indwelling urethral catheters (Strickler and Mclean, 1995). The average cost for a knee prosthesis in 1991 dollars is \$8,600 for a non-infected patient and \$62,100 for an infected patient (Strickler and Mclean, 1995). Since replacement of the implant is expensive and has risks for the patient, more research needs to be focused on keeping the bacteria from getting to the surface.

1.2.2 Definition of problem

Bacteria can get to the surface of implants through several different methods. First, they can be initially present from improperly sterilized material or a breach in sterile technique during operation. Second, airborne bacteria or indigenous bacteria from

the surrounding skin can enter the wound, even after the operation. Once bacteria invade the body, they are carried by the flow of the tissue's interstitial fluid and Brownian motion. In this floating planktonic form the bacteria are more susceptible to antibiotics and can be killed easier by the immune system (Gristina and Costerton, 1985). Bacteria like *Escherichia coli* (*E. coli*) and *Pseudomonas* have flagella and actively swim to implant surfaces. Motile strains of *Streptococci* have also been isolated contrary to the belief that cocci bacteria are not motile (Van Der Drift et al., 1975). Once a thin layer of bacteria or biofilm has formed on an implant surface, in most cases the only effective treatment is removal of the implant (Chang and Merritt, 1994).

Bacteria alone in the body may not create an infection, but they become more virulent once they adhere to a surface (Gristina et al., 1989). Bacteria in a biofilm can grow faster and their metabolic activity can increase or decrease depending on the nutrient availability (Hamilton, 1987). In this state, they are more resistant to antibiotics and can have micro niches of multiple bacteria (Gristina et al., 1989). Over 99% of bacterial biomass in nature exists as a biofilm, rather than free bacteria, indicating that they would rather adhere to whatever solid surface they find (Gristina and Costerton, 1985). Once colonized, bacterial biofilms can spread and kill surrounding tissue. A weakened immune system can increase the probability of infection so patients on immuno-suppressant drugs are at high risk. Fortunately surfaces that are initially colonized by healthy tissue tend to be resistant to bacterial infection (Ratner et al., 1996).

1.2.3 Relation to chemotaxis solution

Changes in extracellular pH from inflammation can affect the motility for bacteria with flagella due to intracellular pH changes. High or low pH is a chemorepellant for *Escherichia coli* and other bacteria. Around a pH of 3-4, flagella disintegrate into

subunits, while a pH between 6 and 8 doesn't affect *E. coli* motility. Even at pH 7, when in the presence of glucose *Escherichia coli* are not motile due to loss of flagella (Adler and Templeton, 1967). The effect on motility of pH in the presence of different attractants is not the same due to use of different receptors. The presence of different chemicals can affect chemotaxis without binding to receptor proteins. Repellants, membrane-permeant weak acids such as acetate and benzoate do not bind chemoreceptors but shunt protons into the cytoplasm lowering intercellular pH affecting chemotaxis (Slonczewski et al., 1982).

Heavy metal ions at very low concentration such as from metal implants inhibit motility of *E. coli*, while amino acids act as chelating agents to stimulate motility (Adler and Templeton, 1967). Unfortunately, build up of metal ions in surrounding tissue has an undesirable cytotoxic effect and can lead to chronic inflammation and accelerated corrosion of the implant leading to device failure. The presence of iron makes bacteria resistant to killing by leukocytes indicating, its use as a biomaterial is not advisable (Weinberg, 1974).

Both bacteria and bone cells utilize chemotaxis, thus methods should be exploited to optimize the specificity in the design of implant devices to benefit tissue cells and inhibit bacteria. For example, the porous structure of hydroxyapatite (HA) can be used to dispense antibiotics, chemoattractants for osteoblast, osteoclast and tissue cells, and chemorepellants for bacteria.

1.3 OBJECTIVE

The objective of this research is to develop stochastic cellular dynamics (CD) simulation calculations for cell motility coefficients from knowledge of the swimming characteristics (swimming speed, tumbling probability, turn angle distribution) of

peritrichious microbes. Random motility simulations will be compared to theoretical relations.

The effects of consumption on migration are determined and discussed in this paper. Different mechanisms for gradient detection will be compared and analyzed. The CD simulation will help determine the effects of porous media on the migration of bacteria in a consumable chemoattractant. The boundary conditions involving the interface with solid objects will be discussed and simulated to help explain observations seen in experiments.

2. LITERATURE BACKGROUND

2.1 CHEMOTACTIC MECHANISM

Chemotaxis refers to the directed migration of organisms in response to chemical concentration gradients. Chemotactic microorganisms migrate toward regions of high chemoattractant concentrations at much faster rates than are realized through random migration or convective transport alone (Witt et al., 1999). Chemotactic microorganisms possess a significant ecological advantage over non-chemotactic competitors, in that chemotactic species are more readily able to secure nutrients needed for cell growth and avoid potential toxins. Chemotaxis can therefore play an important role in engineered, *in situ*, bioremediation systems for the treatment of soil/groundwater contaminant plumes.

Peritrichious bacteria cells such as *E. coli* and *Salmonella typhimurium* move in a series of runs and tumbles. Bacteria swim by rotating their flagella clockwise, which propels them forward. Bacteria change direction by reversing their rotation of their flagella causing a tumble and resulting in a new swimming direction. Figure 1 shows the flagella bundled during cell swimming on the left and unbundled flagella during a tumble

on the right. This run and tumble motion can be simulated by a random walk (Berg and Brown, 1972). The chemotactic effect is a result of the bacteria swimming for a longer period of time up a perceived chemoattractant gradient before tumbling as shown in Figure 2. The movement of bacteria unlike eukaryotic cells is actually a klinokinetic response (Rivero et al., 1989; Ford, 1992). The overall directed motion of the bacteria population results in a chemotactic effect. In chemoklinotactic simulations by Bornbush, the optimum turn angle that requires the least amount of energy to reach the attractant source is between 40 and 100 degrees (Bornbush and Conner, 1986). *E. coli* strain AW405 has a mean turn angle of 68 degrees which is in the middle of the optimum range (Berg and Brown, 1972).

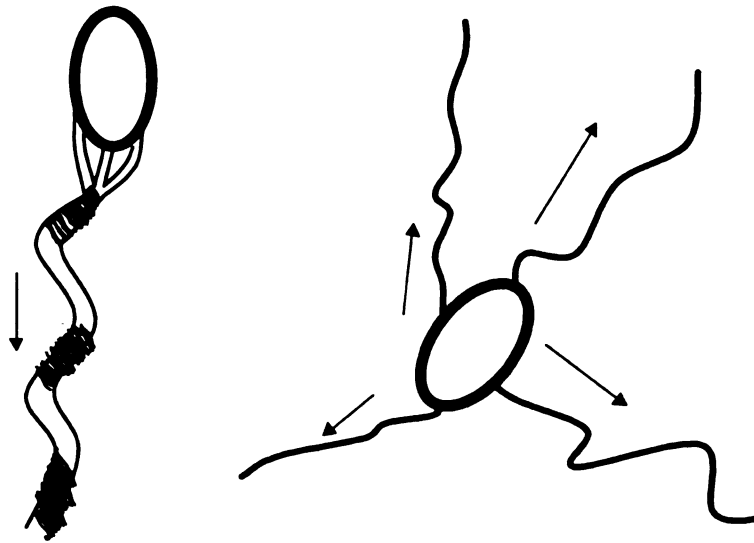


Figure 1. Schematic of flagella coordination by *E. coli* during Run and Tumble.

Bacteria cells, which are on the order of 1 to 2 microns, are too small to sense a chemical gradient along the length of its body so it can only bias its movement by

extending its run length. Bacteria use specialized proteins on their membranes called receptors to measure the concentration of chemicals around them such as nutrients, electron acceptors and toxins. Bacteria compare the concentration of a chemical in its environment with the concentration sensed over the previous few seconds. As the cell swims the temporal variations can be interpreted as a spatial gradient. The tumbling probability is modulated according to the gradient detected.

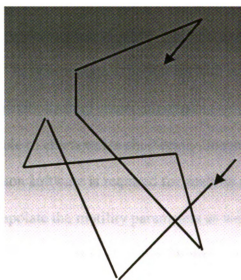


Figure 2. Biasing of run lengths in the presence of a chemoattractant gradient.

Both aspartate and α -methylaspartate are chemoattractants for *E. coli*. Aspartate can be metabolized and consumed by *E. coli*, while α -methylaspartate acts as an attractant incapable of being metabolized. These two chemicals have similar structures enabling them to use the same receptor site on the cell. Even though their binding disassociation constants (K_d) are different, they are the best candidates for chemotaxis studies (Tso and Adler, 1974; Strauss et al., 1995; Mesibov et al., 1973). Furthermore, aspartate is essential to biological systems and literature values of K_d have been reported.

2.2 CURRENT EXPERIMENTAL PROTOCOLS

Traditionally chemotaxis has been studied experimentally using capillary assays and swarm plate experiments. Capillary assays are beneficial for screening bacteria to determine if a particular chemical is an attractant or a repellent. The gradient at the mouth of the capillary is difficult to characterize and control, thus making it difficult to get accurate estimates of the cell motility and chemotactic sensitivity coefficients (Berg and Turner, 1990; Ford, 1992; Marx, 2000; Vickers, 1981). Swarm plate experiments provide a decent visual representation of where and how quickly the bacteria migrate from an initial position. In swarm plate experiments with sustained chemical gradients or non-consumable (non-metabolizable) chemoattractants can't easily be done, but this type of experiment is amenable to characterize chemotaxis due to gradients produced by cell metabolism. Visualization software is required for analysis of the cell density as a function of time to extrapolate the motility parameters as well as fitting the mathematical models. (Nikata et al., 1992; Strauss et al., 1995; Widman et al., 1997; Biondi et al., 1998) An innovative method to analyze the behavior of bacteria in a fixed gradient has been done using a Diffusion Gradient Chamber (DGC). In the DGC, constant concentrations of chemoattractant is recirculated at one end and a buffer solution without chemoattractant is recirculated at the opposite end of the chamber creating a fixed gradient for studying consumable and non-consumable chemoattractants or chemorepellants (Widman et al., 1997).

2.3 CURRENT MODELING PROTOCOLS

A number of mathematical models exist that have quantified motility parameters and chemotactic sensitivity for simplified physical conditions (Ford et al., 1991; Brosilow et al., 1996; Rivero and Lauffenburger, 1986). One model uses a decaying step gradient,

which simulates a large 1-Dimensional (1-D) chemoattractant gradient (Frymier et al., 1994). Most of these models use different methods to solve different variations of the cell balance equations, derived by Alt, to compute the unknown parameters. See Alt's publications for more details on the cell balance equations (1980). To further prove the tractability of the CD method, Ford's group used Monte Carlo method to solve Alt's cell balance equations (Frymier et al., 1993). Studies on consumable chemoattractants have been done using cell balance equations or other methods, which don't always quantify the cell motion parameters (Marx and Aitken, 2000; Dillon et al., 1995; Widman et al., 1997).

The motile behavior of a population of cells is characterized by both the mean square distance the cell moves (motility) and the drift velocity (mean distance) along a gradient. The motility is analogous to a diffusion coefficient indicating how dispersed the cell population gets. The mean distance a cell population moves along a gradient is a measure of its chemotactic response (Berg, 1983).

2.4 IMPROVED SIMULATIONS USING CELLULAR DYNAMICS

An alternative to solving cell balance equations is the cellular dynamics (CD) method, which utilizes the cell motion parameters and independently calculates the new positions of each cell. In addition, the algorithm considers the duration of time necessary for a cell to tumble. The cell motion can be calculated using either two dimensions or three dimensions, with the ability to calculate chemoattractant gradients in two dimensions only. The CD model requires the chemotactic sensitivity as input. Therefore, this parameter must be determined experimentally via a different method or taken from reported literature values.

Three main independent scenarios were examined using the CD method. The first case in focus is a fixed linear gradient with no consumption, which correlates with the simplified chemotactic models previously done. This method is used to characterize how the cell motion parameters affect chemotaxis. After first determining the optimal chemotactic sensitivity to visually see displacement for the 30 minute simulation, several of the cell motion parameters (i.e. cell velocity, mean turn angle, tumbling probability, etc.) were adjusted to determine their effects on motility.

The second case modeled consumption starting with a uniform concentration of chemoattractant. A significant advantage to the CD model is that the chemoattractant concentration gradients can be independently measured in two dimensions. This is required to model the chemotactic movement of bacteria that create gradients by consumption.

The third area of investigation involved migration in porous media. Pore widths and particle size were varied as well as the configuration of the porous region. The model was adapted to simulate migration in slit and heterogeneous pores with consumption and diffusion of a chemoattractant and the simultaneous detection of the gradient produced. Experiments with bacteria in both porous and non-porous media show patterns and other interesting phenomena. In an experiment with *Pseudomonas KC* responding to nitrate, bacteria swam further through a porous sand core sample than through the bulk phase (Figure 34). So we wanted to study migration in pores to determine what was causing the accelerated migration. We also get some interesting behavior without having pores. Other experiments show migration away from the origin

in waves or bands of higher cell concentration resembling concentric rings. Both these phenomena can be explained using our model for consumable chemoattractants.

The simulation model uses dimensionless variables in its calculations requiring the input parameters to be scaled to a dimensionless number. The unit of length is scaled with the value of 1 micron, which is the approximate length of an *E. coli* cell. The scaling unit for time was selected to be 0.1 seconds; corresponding to the length of time a bacterium tumbles (Berg and Turner, 1990). Concentration is scaled against the dissociation constant to yield a dimensionless concentration. The consumption rate is scaled with time from the input value of dimensionless concentration over time per cell.

Dillion uses a stochastic chemotactic model for the modeling of cells with adsorption on to biofilms in porous media. These simulations track a few cells for only a few seconds. Our CD simulations involve a large population of cells over 30 minutes and longer giving longer-term dynamic effects of cell populations. Chemoattractant concentration, individual cells and trajectories are discretely monitored in our model rather than solving equations of motion. Dillion et al. states that simplified chemotactic models that ignore fluid effects give poor predictions of nutrient consumption (1995). Nevertheless, our model is for non-flow systems and the chemoattractant does not have a replenishing source or boundary to have significant fluid effects. We do however account for diffusion of the chemoattractant. The hydrodynamic effect causing lower concentration next to the cells can be accounted for by using a thin film mass transfer coefficient with chemoattractant diffusing through to yield a surface concentration.

3. SIMULATION METHODOLOGY

3.1 CD SIMULATION MECHANICS

The bacteria runs are relatively straight and have a constant run time in the absence of chemoattractant gradients (Brown and Berg, 1974). The basal tumbling probability (p_0) is the reciprocal of the straight line run time. When the cell is moving in the direction of increasing concentration the run lengths are extended. The basal tumbling probability is adjusted based on the dot product of the direction vector (S) and the concentration vector according to Equation 1 (Brown and Berg, 1974; Strauss et al., 1995).

$$p = p_0 \exp \left(- \frac{\chi_o^{3D}}{v} \frac{K_d}{(K_d + C)^2} S \cdot \nabla C \right) \quad (1)$$

E. coli only biases its tumbling probability when moving (up gradient) toward a higher concentration of chemoattractant and returns to the basal tumbling probability when moving (down gradient) away from a higher concentration of chemoattractant. *Salmonella typhimurium* not only has a bias of longer runs when the concentration increases, but exhibits shorter runs when swimming down gradient of the chemoattractant as well (Macnab and Koshland, 1972).

The accepted expression for random motility with tumbles considered instantaneous is defined by equation 2. The random motility coefficient μ_0 is differentiated from the chemotactic motility coefficient by the subscript zero and is calculated from a simulation of no gradient or consumption was compared to $1.62\text{E-}6 \text{ cm}^2/\text{s}$ from the CD simulations. Using Equation 2 gives a random motility of $1.65\text{E-}6$

cm²/s resulting in a 1.85 % error compared to the random motility of 1.62E-6 cm²/s calculated by the CD simulation (Liu, 1996; Biondi et al., 1998). A smaller random motility coefficient of 1.4E-6 cm²/s was calculated from a simulation using 0.05 second time step and 0.1 second delay for a tumble. Neglecting the inclusion of the actual time of a tumble results in a difference of 15.1 % in the random motility coefficient compared to the actual random motility. Using a time step of 0.1 instead of 0.05 seconds yields a decreased random motility with the addition of a 5% error.

$$\mu_o = \frac{v^2}{2p_o(1 - \langle \cos(\phi) \rangle)} \quad (2)$$

In equation 2, the coefficient 2 refers to the dimensionality of the system. The average turn angle ϕ that is calculated from the turn angle distribution is 79.2° in the model. The motility coefficient is only invariant when the gradient is unidirectional and non-changing or linear. Consumption from a single point yields a mean square displacement slope that changes over time because the gradient also changes as the cells thin out in a circular wave from their initial location. At large consumption rates where cells get stranded behind, large deviations from a linear slope occur because part of the cell population is moving with random motility and the other portion of cells respond to a changing gradient as the cell wave front proceeds.

By using a cellular dynamics model, the stochastic nature allows the chemotactic response of individual cells to be randomly biased giving a realistic population behavior at larger population sizes. A simplified flow chart of the model methodology is shown in Figure 3.

A random number is generated to determine if the cell tumbles or runs. If the random number is less than the new tumbling probability, the cell tumbles, thus staying at its present location. A new direction is then picked based on its current direction and its turn angle distribution. If the cell runs, the direction stays constant and a new location is determined based on the cell velocity. This process is repeated for each cell for each time step.

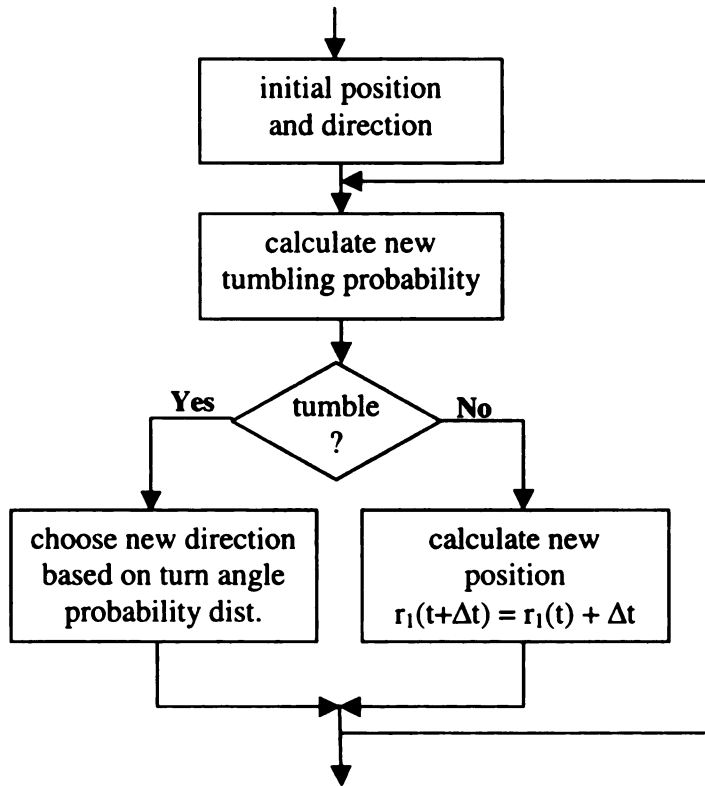


Figure 3. Simplified flow chart of model methodology.

The motility coefficient (μ) is obtained from the mean displacement of the cell population ($\Omega(t)$). The cell displacement is averaged over the number of cells (N) (Duffy, 1995).

$$\Omega(t) = \frac{1}{N} \sum_{i=1}^N \left[dx^2 + dy^2 \right] \quad (3)$$

Einstein Relationship for 2 dimensions

$$\begin{aligned} \Omega(t) &\rightarrow 4\mu t \\ \text{as } t &\rightarrow \infty \end{aligned} \quad (4)$$

Expression for the motility coefficient

$$\mu = \frac{1}{4} \left(\frac{\Omega(t)}{t} \right) \quad (5)$$

The concentration of chemoattractant is tracked at nodal points on a 2-Dimensional grid with the origin located at the center (Figure 4).

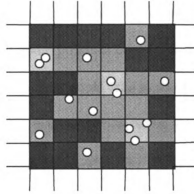


Figure 4. Schematic of the grid network with attractant being consumed based on cell locations.

The four nodal points around the cell are updated for consumption of each cell and are weighted (W) based upon its relative distance from each node. The consumption is modeled as a constant or zeroth order reaction rate. A Michaelis-Menten kinetic rate could be used as well for concentration dependent consumption of nutrients. The rate of consumption in these simulations is assumed constant where the change in concentration per time is defined as k.

$$\frac{dC}{dt} = k \quad (6)$$

The chemoattractant concentration field is updated at the end of each time step for diffusion. The edges of the grid can be chosen to be periodic or non-flux depending on the details of the simulation. Equation 7 shows how the concentration at each point is updated due to cell consumption.

$$C_{(x, y, t + \Delta t)} = C_{(x, y, t)} - W_{(x, y)} k \Delta t \quad (7)$$

3.2 GRID SPACING

The 2-dimensional grid works well for keeping track of the chemoattractant concentrations at fixed nodal points as shown in Figure 4. Grid spacing is extremely important in obtaining an accurate concentration profile. If the grid spacing is too small, the cell will swim past several nodal points at each time step, creating gaps in consumption along the cell path. This could lead to an erroneous concentration profile. Alternatively, if the grid spacing is too large the nodal square for consumption will be unrealistically large as well. As a result, the calculated gradient would be located far from the cell position. Furthermore, cell movement will approach random motility at high grid spacing values. An error analysis was completed on adjusting the grid spacing ranging from values of 24 μm to 40 μm . The error percentage, though small, is noticeably higher for the simulations completed using a grid spacing of 40 μm , and lower for a grid spacing of 24 μm .

The effects of the grid spacing on the chemotactic motility coefficient were determined for simulations with consumable chemoattractant. Grid spacing would appear to have a significant effect on the gradients calculated using the nodal points. When the total consumption on a concentration basis at the four closest nodal points is

kept constant and the grid spacing is changed, the shape of the motility trend is similar to the case of changing consumption rates because the consumption rate is effectively changed. The consumption rate needs to be constant per volume or area of the grid space, so adjustment of the changing area of the nodal square needs to be incorporated to conserve the rate of mass consumption. An area based on 20-micron grid spacing was assumed for adjusting the consumption rate in 2 dimensions at different grid spacing using the expression in equation 8. The units on the consumption rate k and k_o are (dimensionless concentration/(number of cells * seconds)). The consumption rate k has a constant mass basis, while k_o has a constant concentration basis for use at the 4 surrounding nodal points.

$$k = k_o \left(\frac{\Delta X_o}{\Delta X} \right)^2 \quad (8)$$

By adjusting the consumption rate based on a constant mass basis yields a constant motility as the grid space changes as shown in Figure 5. Diffusion is significantly larger than the consumption rate so the gradients smooth over the nodal points making the simulations less dependent on grid space. This allows us to have more flexibility in setting the size of the grid space when running simulations with consumption.

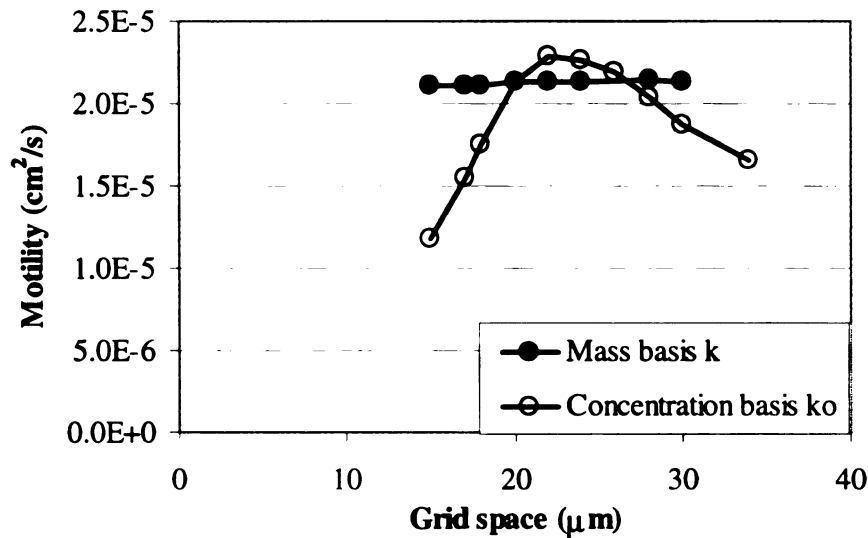


Figure 5. Comparison of adjusting consumption based on grid spacing.

3.3 STABILITY

The criteria for numerical stability were determined using the center finite differencing method to approximate diffusion. To increase the margin of stability, the grid spacing should be as large as possible, while keeping the time step to a minimum. With our algorithm, the time step was fixed to the length of time a bacteria takes to tumble (approximately 0.1 seconds), leaving only the grid spacing available to adjust. The grid spacing is used to determine the gradient that the bacterium senses. It is believed that bacteria sense a gradient temporally rather than spatially. Therefore, the grid spacing must correlate with the distance the bacteria swims during the time it takes to notice a change in the concentration of its surroundings (Macnab and Koshland, 1972; Brown and Berg, 1974). As in the case of consumption, the grid spacing cannot be arbitrarily set.

The minimum grid spacing required for numerical stability was calculated to be 21.1 μm using a time step of 0.1 seconds. This value was calculated by solving a

concentration balance to determine the grid size and time step that balance the diffusion in and out of a nodal point. This nodal point has a lower concentration than the four surrounding points. Using both the basal tumbling frequency and cell velocity, the approximate average distance a cell travels before it senses a gradient is 18.3 μm . Based upon the calculations and an error analysis, a grid spacing of 24 μm was chosen for our simulations to ensure stability as well as give a reasonable estimate of the cells ability to sense gradients.

Our simulations are based on a 2-D grid with concentration saved at the nodal points. The consumption at each nodal point is based on the number of cells present in each grid space and how close they are to the nodal point. Diffusion is accounted for by using an explicit finite difference scheme utilizing equations 9 and 10.

$$C(t + \Delta t) = C(t) + D\Delta t \left(\frac{\partial^2 C}{\partial X^2} + \frac{\partial^2 C}{\partial Y^2} \right) \quad (9)$$

$$\frac{\partial^2 C}{\partial X^2} = \frac{C_{j+1,k} - 2C_{j,k} + C_{j-1,k}}{DG^2} \quad (10)$$

A diffusion balance was done to determine the stability criteria for the grid spacing and time step giving equation 11.

$$C_{center} + dC_{center} \leq C_{surrounding} - dC_{surrounding} \quad (11)$$

The change in concentration due to diffusion at the center point and the 4 surrounding points are defined by the following expressions.

$$dC_{center} = 4dC_{surrounding} \quad (12)$$

$$\frac{dC_{center}}{\Delta t} = D \left(\frac{2C_{surrounding} - 2C_{center}}{DG^2} + \frac{2C_{surrounding} - 2C_{center}}{DG^2} \right) \quad (13)$$

We had to be careful in choosing the size of the grid spacing. The explicit method goes numerically unstable if the grid spacing to time step ratio is too small. Since we have already chosen a time step, we wanted to pick a grid spacing that was small enough but not be unstable. Substituting these expressions into equation 11 yields a result independent of the concentrations at the center and the 4 closest points.

The ratio of grid space squared over time step has to be greater than 5 times the diffusion coefficient. We determined that 24 micron grid spacing that we used was numerically stable by using equation 15.

$$\frac{5}{4} \frac{D\Delta t}{DG^2} (4) \leq 1 \quad (14)$$

$$DG \geq \sqrt{5D\Delta t} \quad (15)$$

For a 3D grid the criteria for numerical stability is expressed by equation 16. The coefficient under the square root is the number of nodal points surrounding one nodal point plus the nodal point at the center (i.e. 4+1 for 2D and 6+1 for 3D). The critical grid spacing for 3D diffusion is only 18% larger then for 2D diffusion. This expression will be helpful when the model is expanded for diffusion in 3D.

$$DG \geq \sqrt{7D\Delta t} \quad (16)$$

An alternative method of calculating the diffusion may be employed to maintain stability at smaller grid spacing. A method for calculating the gradient a bacteria sense's independent of the grid spacing may be implemented based on the time it takes for a cell to notice a concentration change. The time it takes a bacterium to react to a concentration change is on the order of a run duration, which for *E. coli* is around 4 seconds (Berg, 1993).

3.4 ASSUMPTIONS

A cell swims in a series of runs and tumbles and randomly picks a new direction based on the turn angle distribution. In this model cells are assumed to swim in a straight line between tumbles and during a tumble only the direction of the cell changes and the location remains the same.

Cell growth and death does not occur in the model. The number of cells remains constant, which reasonably models cells in the lag phase (no growth and no death), and the stationary phase (growth equals death). Migration for the exponential growth phase may be modeled for simulations shorter than the cell doubling time.

All the cells are assumed to have the same swimming velocity, basal tumbling probability, chemotactic sensitivity, and disassociation constant.

Multiple cells can occupy the same location on the grid and cells cannot sense the presence of other cells directly. Only through the gradients in the chemoattractant produced by the other cells can the neighboring cells be sensed.

Multiple gradients are not sensed, effects from other chemoattractants such as oxygen are not currently modeled. The substrate used for energy in the nonconsumable attractant simulations is assumed to be plentiful and not be a chemoattractant.

The availability of electron donors or acceptors and its effect on the consumption rate is not considered but assumed to be available in sufficient quantity.

3.5 DIMENSIONALITY

The input parameters for chemotactic sensitivity and single cell swimming velocity are three-dimensional values, which are scaled to two-dimensional values when studying two-dimensional migration with consumption.

Equations 17 and 18 scale the dimensionality of the cell motion. The chemotactic sensitivity is proportional to velocity squared. The differential tumbling frequency is denoted as v , and N_T is the total number of receptors (Frymier et al., 1993; Rivero et al., 1989). For simplicity the two-dimensional chemotactic sensitivity and velocity will be noted as χ_0 and v respectively in this paper.

$$v^{2D} = v^{3D} \left(\frac{\sqrt{2}}{\sqrt{3}} \right) \quad (17)$$

$$\chi_o^{2D} = v^2 v N_T = \chi_o^{3D} \left(\frac{\sqrt{2}}{\sqrt{3}} \right)^2 \quad (18)$$

4. LINEAR GRADIENT NON-CONSUMABLE

The simulation parameters used in initial simulations were 2000 cells which is small compared to the number encountered in experiments, but large enough to get realistic average results without having to use excessive computer time (some 1 hour simulations required 4 hours using an 800 MHZ Pentium).

Other authors have used a chemotactic sensitivity (χ_0) value of $4.1E-4 \text{ cm}^2/\text{s}$ for *E. coli* responding to α -methylaspartate (Strauss et al., 1995). A χ_0 value of $2E-2 \text{ cm}^2/\text{s}$

was selected for our simulations to give a noticeable chemotactic response with a small linear gradient of $1.25\text{E-}5 \text{ mM}/\mu\text{m}$ that could be experimentally obtained easily. The larger value of χ_0 is used solely to show the trend in motility when cell motion parameters are changed. Furthermore, variability in cell cultures can cause the χ_0 value to vary 20 % from the mean (Marx and Aitken, 2000). The chemoattractant response is also heavily dependent on temperature. Experiments by Adler showed that increasing temperature from 20 C to 30 C yielded a 20-fold increase in bacteria attracted and no chemotaxis below 15 C (Adler 1973). Since the χ_0 is dependent on the velocity squared, by only doubling of cell velocity could explain a 4-fold increase in the chemotactic response. Above 37 C the synthesis of the flagella stops which caused the movement of the bacteria to diminish (Adler and Templeton, 1967).

4.1 SINGLE CELL TRAJECTORY

Figure 6 shows the simulation trajectory results of a cell movement according to random motility compared to cell movements demonstrating chemotactic behavior using a fixed, linear, non-consumable gradient, increasing in the positive y-direction. The symbol χ represents the chemotactic sensitivity. The simulations for the trajectory results were completed using various chemotactic sensitivity coefficients. The velocity and tumbling probability of the cells were fixed at values of $22 \mu \text{ m/s}$ and 1.2 s^{-1} , respectively. Table 1 contains the input parameters used in the simulations of cells responding to a linear non-consumable gradient.

When χ_0 is at or approximately zero, the cell clearly demonstrates a typical random motion with very little migration away from the point of inoculation. While gradually increasing the value of χ_0 , the cell movement deviates from random motion to

that of a cell responding to a chemical gradient. If an unreasonably high value of χ_0 is used, such as $4.1 \text{ cm}^2/\text{s}$, chemotaxis is clearly evident because the cell displays a very strong, biased movement along the gradient without deviating from its path. The average distance a cell moves from its initial location is larger for larger values of the chemotactic sensitivity.

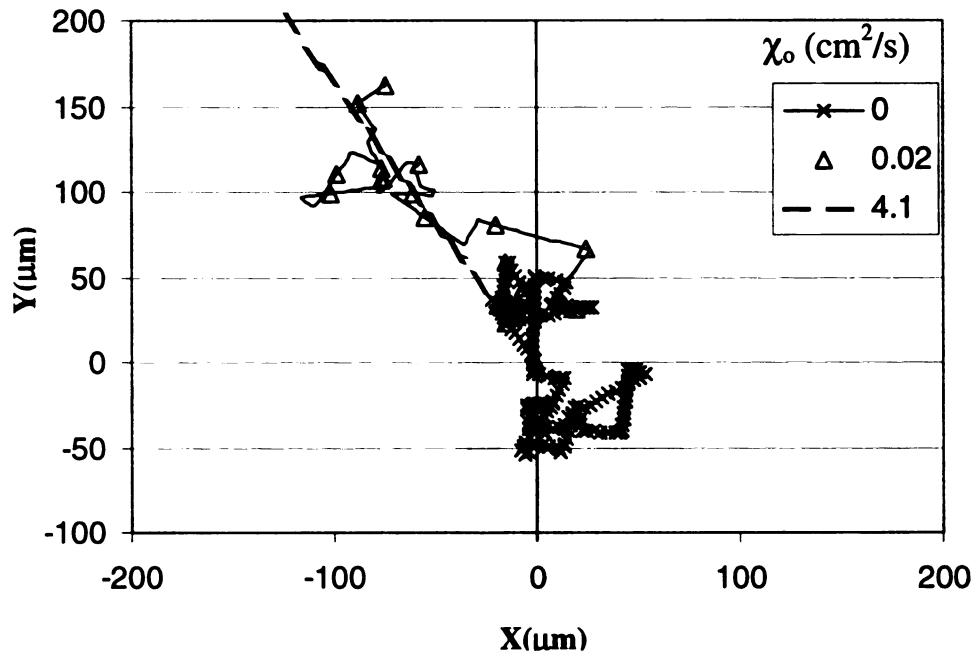


Figure 6. Cell trajectory comparison

Table 1. Values used in the simulations of a non-consumable linear gradient of α -methylaspartate for *E. coli*.

NP	2000
X	40,000 μm
Y	40,000 μm
DG	40 μm
p_0	1.2 s^{-1}
v	22 $\mu\text{m/s}$
C	1 mM
K_d	0.125 mM
χ_0^{3D}	$2\text{E-}2 \text{ cm}^2/\text{s}$

4.2 EFFECTS OF VARYING CHEMOTACTIC SENSITIVITY COEFFICIENT

Figure 7 shows the difference between cells with a moderate ($\chi_0 = 2\text{E-}2 \text{ cm}^2/\text{s}$) chemotactic sensitivity and those demonstrating no chemotaxis at all ($\chi_0 = 0 \text{ cm}^2/\text{s}$). The lack of a gradient within the environment will cause the cells to move in a random walk, thus denying any movement biased toward one direction. The cells will inhabit the location close to the point of inoculation. An introduction of a gradient will cause the cells to migrate away from their origin. Even at small values of χ_0 , the cells will demonstrate chemotaxis, biasing migration more with larger gradients and at longer time periods.

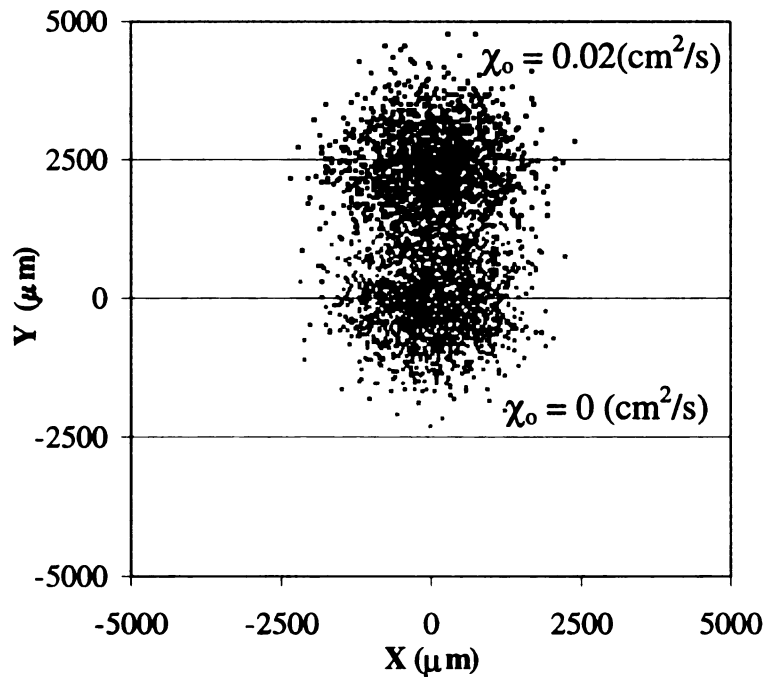


Figure 7. Snapshots of chemotactic and non-chemotactic cells in a linear gradient.

When the chemotactic sensitivity is small, they stay in a circular pattern and the population moves to areas of higher concentration. When the chemotactic sensitivity gets larger, they tumble less frequently and the shape of the colony flattens out to an oval

shape. At a very large chemotactic sensitivity a limiting value is reached where the cells don't tumble and they continue to move up gradient in the direction of their initial orientation. The cell density profile also shows this result. Figure 8 is a comparison of snapshots showing cell positioning at several values of the chemotactic sensitivity coefficient ($\chi_0 = 4.1, 4.1\text{E-}1, 0$) cm^2/s taken at 30 minutes after inoculation. While increasing χ_0 , a cell population will migrate along the y-axis in the positive direction. At low values of χ_0 , there is little, noticeable movement of cells in the positive y-direction, imitating non-chemotactic behavior at small time periods. If χ_0 becomes extremely large (e.g. $\chi_0 = 4.1 \text{ cm}^2/\text{s}$), the cells cease to frequently tumble. As a result, they move radially in the direction of the gradient producing a cell snapshot resembling a semicircle, where very little random motion occurs. The density of cells increases at locations of higher chemoattractant concentration, proving that chemotaxis displaces cells more dramatically than random motility.

To better illustrate the difference between the average displacement (drift velocity) and the motility coefficient (diffusive mean square displacement) in Figure 8 the motility coefficient is larger for the cell distribution with a χ_0 of $4.1 \text{ cm}^2/\text{s}$ then the cell distribution with a χ_0 of $0.41 \text{ cm}^2/\text{s}$, while the average displacement is larger for χ_0 of $0.41 \text{ cm}^2/\text{s}$ then χ_0 of $4.1 \text{ cm}^2/\text{s}$. In different applications the goal may be to disperse the cells more or it may be to transport the cell to a focused area so both the motility coefficient and the average displacement of the cell population are important in defining the behavior of the cell population.

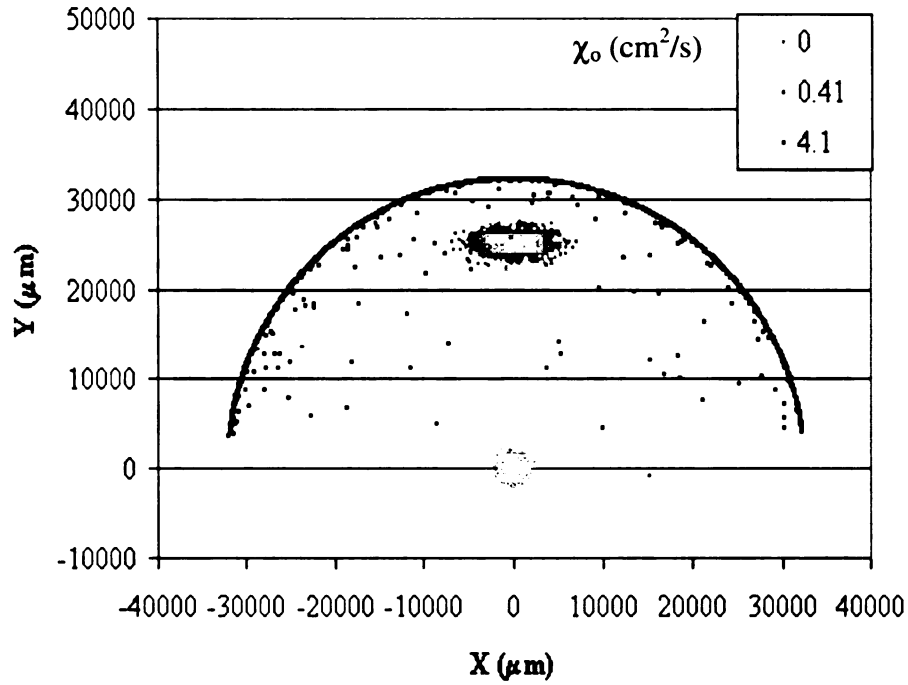


Figure 8. Snapshots of cell location along a linear non-consumable gradient.

Simulation results for the density profile of a 2000 cell population migrating in response to a fixed linear chemical gradient are shown in Figure 9. The cell density is plotted as a function of location in the y-direction for cells using a variety of χ_0 values. Again, chemotaxis can be seen from observing the density of cells with respect to the distance the cells traveled, as plotted on the x-axis of Figure 9. As the value of χ_0 increases from zero to $4.1 \text{ cm}^2/\text{s}$, the spatial location of the cells increases as well, clearly showing the behavior of chemotaxis. Beyond a χ_0 value of $4.1 \text{ cm}^2/\text{s}$, the motility ceases to increase, demonstrating a limit to how large the motility can get.

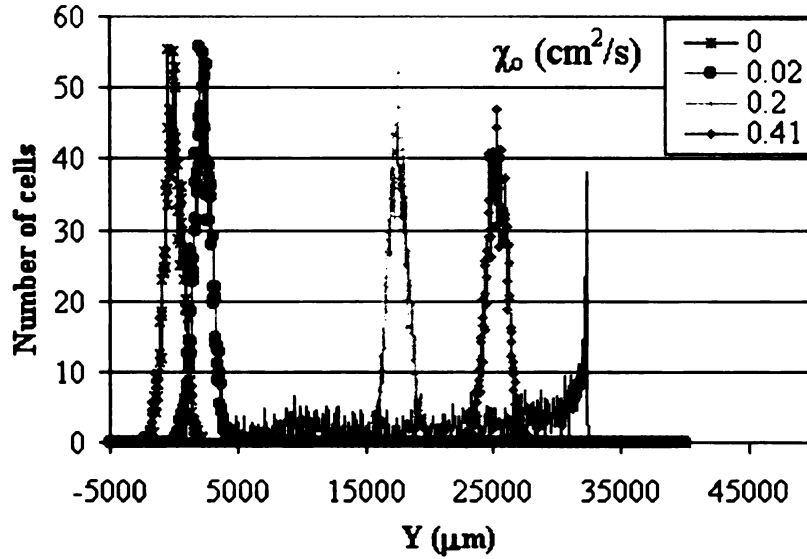


Figure 9. Cell density profile in a linear gradient at varying chemotactic sensitivity coefficients

Figure 10 shows the effect of the chemotactic sensitivity coefficient (χ_0) on the motility and the mean displacement of the cells. The dotted line corresponds to the mean distance up gradient at its respective χ_0 value and the solid line corresponds to the motility data (displacement from its original position). The motility coefficient increases as a sigmoid curve as plotted on a logarithmic scale, it starts to level out around χ_0 of $1\text{E}-1 \text{ cm}^2/\text{s}$ for simulations using a fixed linear gradient ($1.25\text{E}-5 \text{ mM}/\mu\text{m}$). The chemotactic response increases exponentially over a range of χ_0 for approximately 4 orders of magnitude. A dramatic increase in motility occurs at this critical sensitivity. A limiting motility coefficient, corresponding to straight-run cell motion, is attained at very large $\chi_0 = 4.1 \text{ cm}^2/\text{s}$ values. The calculated motility coefficients only correspond to the conditions and simulation time used for determining them. With a different χ_0 or a different concentration gradient, the motility will correspondingly change. The mean displacement of the cell population increases with respect to an increasing chemotactic sensitivity

coefficient and levels off at higher chemotactic sensitivities are reached where cells in frequently tumble.

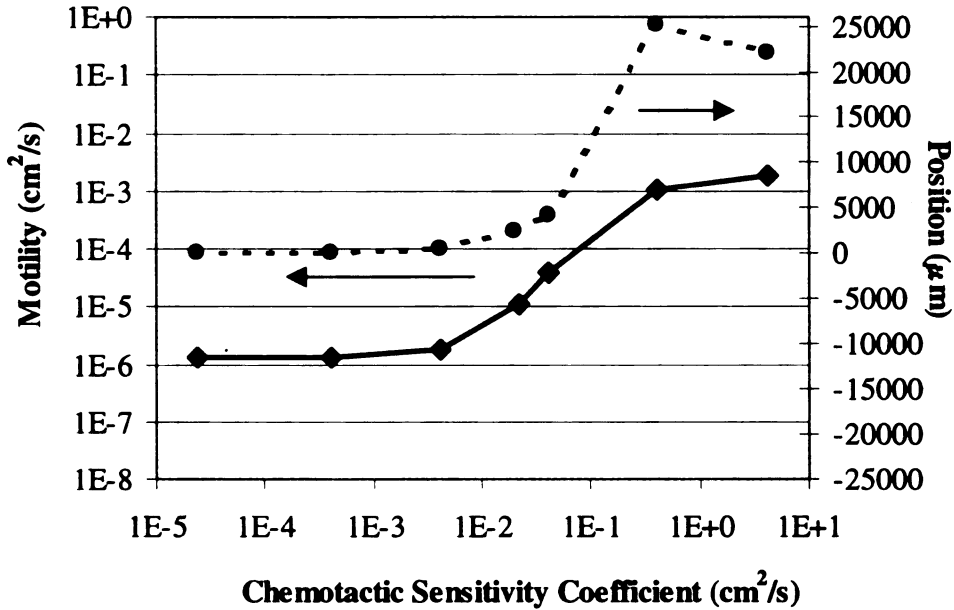


Figure 10. Chemotactic Sensitivity versus Motility and Mean Distance.

4.3 EFFECTS OF VARYING CELL SWIMMING VELOCITY

The cell swimming velocity parameter appears in equation 1 to suppress tumbling and calculate the distance a cell moves to each new location. Thus, the effect that velocity has on the motility coefficient is much more complex than the remaining parameters. According to equation 18, the chemotactic sensitivity is a function of velocity so it needs to change as the velocity changes and should not be kept constant. To determine the effect velocity has on motility the vN_T was kept constant. A typical value of vN_T is 75 s, however a value of 750 s was used in the simulation to obtain a more pronounced chemotactic effect (Ford, 1992). Simulations were completed at several cell velocity values for both the random motility and chemotactic scenarios, in the presence of a fixed, linear, methylaspartate gradient. Figure 11 gives the trend of the motility coefficient calculated for a few of the velocities.

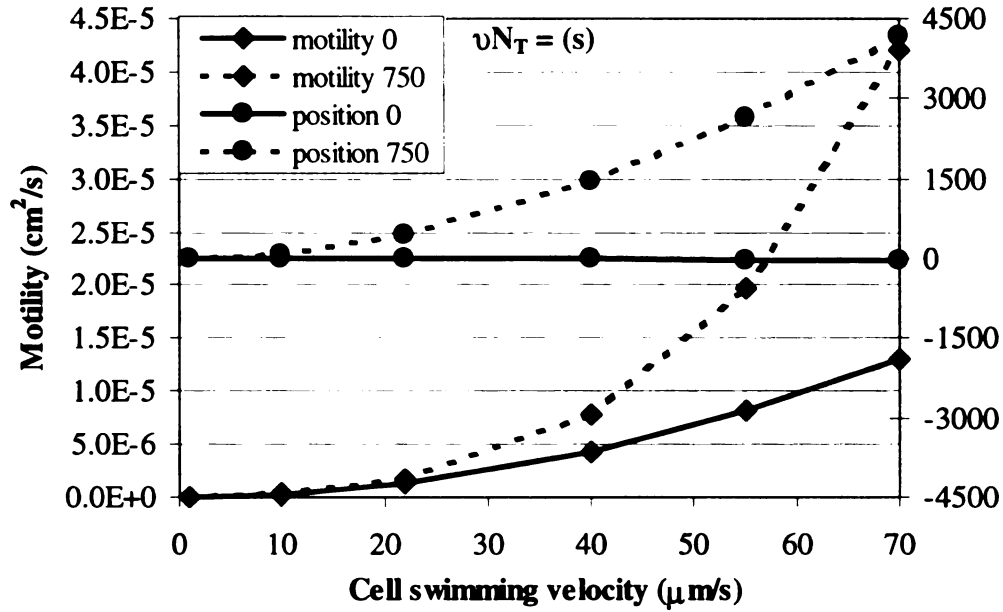


Figure 11. Velocity versus Motility and Mean Distance.

The velocity ranged from 1 $\mu\text{m/s}$ (chemotactic motility $2.65\text{E-}9 \text{ cm}^2/\text{s}$) to a much higher velocity of 70 $\mu\text{m/s}$ (chemotactic motility of $4.2\text{E-}5 \text{ cm}^2/\text{s}$). The circles represent mean distance results whereas diamonds represent the motility results, while dotted lines refer to chemotactic data and the solid lines refer to non-chemotactic.

The dotted line with diamonds displays the motility trend with respect to increasing velocity at a constant vN_T of 750 s. The motility increases in a quadratic order for both the random and chemotactic cases, as predicted by equation 2 for random motility. The mean displacement of the cell population from their initial location remained constant for non-chemotactic cells as velocity increased. For the chemotactic scenario, the mean cell displacement increases as a quadratic, analogous to the non-chemotactic case.

4.4 EFFECTS OF VARYING THE TUMBLING PROBABILITY

Figure 12 demonstrates the effect of adjusting the tumbling probability of the cells for both the chemotactic and non-chemotactic scenarios. The chemotactic sensitivity coefficient χ_0 of $2\text{E-}2 \text{ cm}^2/\text{s}$ was used with a constant cell swimming velocity of $22 \text{ }\mu\text{m}/\text{s}$. At 30 minutes of cell simulation, the center of mass displacement remains constant as the tumbling probability is increased for both chemotactic and non-chemotactic cells. The simulation results show that chemotactic and non-chemotactic populations will disperse more at lower tumbling probabilities. The chemotactic cells will be displaced up gradient, while the non-chemotactic cells will remain clustered around the origin. Increasing the tumbling probability will cause the cells to tumble frequently, resulting in little dispersal of the cell colony. For both chemotactic and non-chemotactic scenarios, the motility decreases dramatically until it reaches a tumbling probability of 1 s^{-1} , which is common for *E. coli*, and then it levels off.

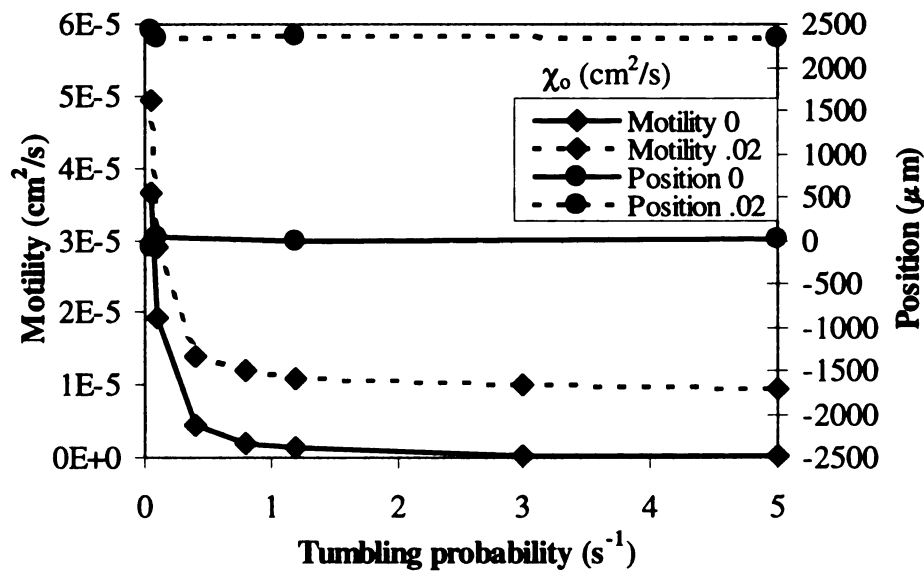


Figure 12. Tumbling Probability versus Motility and Mean Distance.

For simulations running at very short time intervals (e.g. a few seconds or less), authors have reported that cells with shorter run lengths (higher tumbling probabilities) show more chemotactic response, thus contradicting our simulations results (Dillon et al., 1995). We have found that chemotactic response (mean displacement) at longer time intervals (e.g. 30 min) was independent of the tumbling probability. It will take longer for a cell with extended run lengths to orient itself with the gradient, showing a chemotactic response from its initial location. A random orientation of the cell, if facing down gradient, will move with random motility until it readjusts itself in the direction of the gradient, thus allowing it to bias its motion. A longer amount of time is required for a cell with extended run lengths to complete enough tumbles to orient itself with the gradient.

At shorter time intervals, cells with high tumbling probability will often orient themselves more rapidly, allowing movement up gradient. A shorter length of time is then required for movement up gradient, with the cells oriented in a tight cluster. At longer run times, chemotactic response (mean displacement) of high and low tumble probability is indistinguishable, but our results conclude that cell motility decreases with higher tumbling probability due to the reduced dispersion of the cells (Figure 12). Simulation results show that the motility coefficient increases at low tumbling probability.

4.5 GAUSSIAN TURN ANGLE DISTRIBUTIONS

A set of gaussian turn angle distributions were calculated and normalized to determine the effect the mean turn angle and the variance of turn angle distribution would have on the motility coefficient. These distributions are symmetrical with both the mean

and the median at the peak of the distribution. The gaussian turn angle distribution was calculated using equation 19.

$$probability = Exp\left(-\frac{(\theta - \theta_{Mean})^2}{\sigma^2}\right) \quad (19)$$

Figure 13 and Figure 14 show the normalized turn angle distributions used in our simulations for the mean value and the variance, respectively. The turn angle distribution used for all the other simulations in this thesis is for *E. coli* strain NR50 reported by Frymier (Duffy and Ford, 1997).

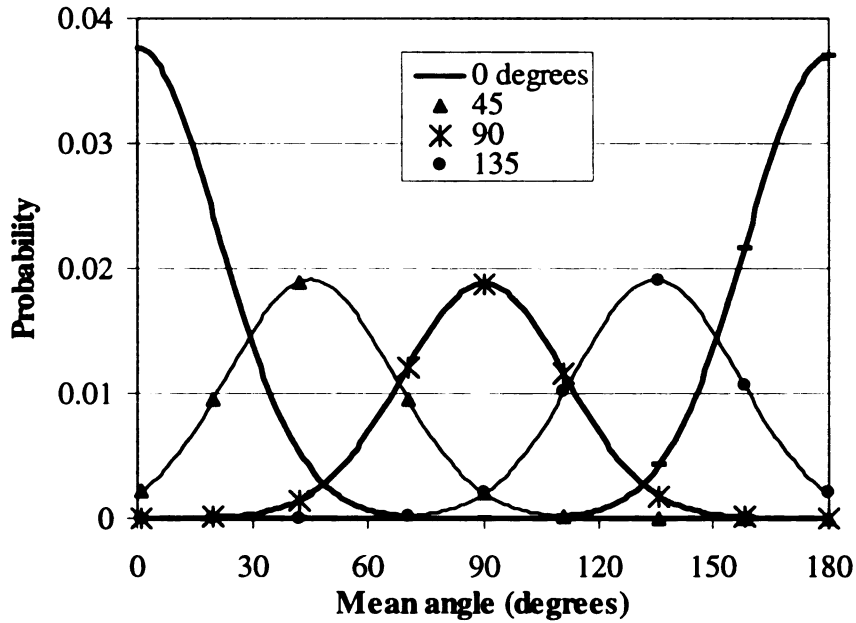


Figure 13. Normalized turn angle distribution varying mean.

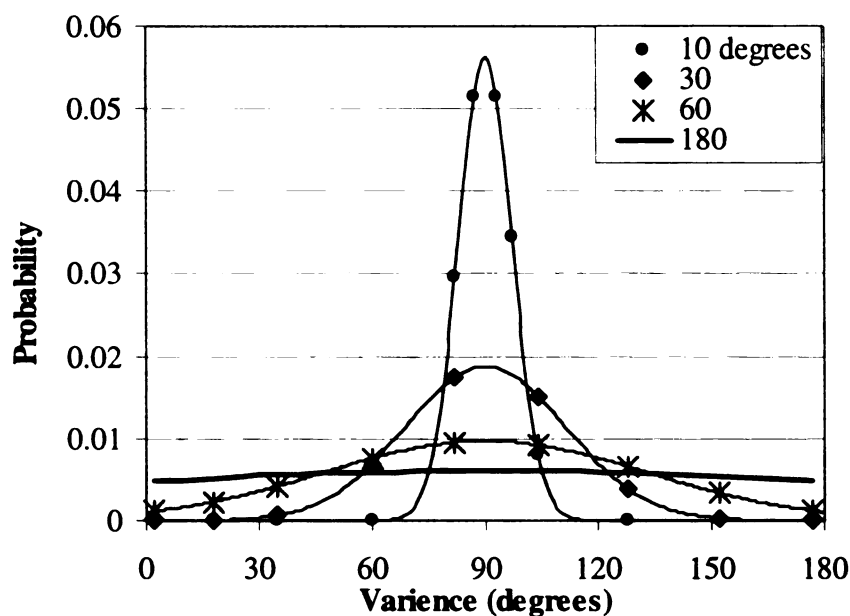


Figure 14. Normalized turn angle distribution varying variance.

4.6 COMPARISON OF *E. COLI* AND *P. PUTIDA* MOTILITY

The actual turn angle distributions are not gaussian but are skewed toward smaller turn angles as shown in Figure 15. The gaussian turn angle distributions are symmetric thus they have the same mean and median values that occur at the peak to the distribution. The turn angle distributions for *E. coli* have mean, median and peak values distinctive from each other, while *P. putida* has a bimodal distribution with two peaks. When correlating the behavior of the bacteria with different turn angle distributions, which value contributes most to the motility behavior of the bacteria? One method would be to look at the median turn angle, but the correlation would not be very smooth for skewed distributions. A better method would be to take the moments of the distribution, which have a strong statistical basis. The first moment would be the mean or average of the turn angles, while the second moment would be the variance. According to our

simulations, the mean of the turn angle is the most important mode in determining the motility coefficient. Table 2 shows the values of the mean, median and peak of the cell turn angle distributions.

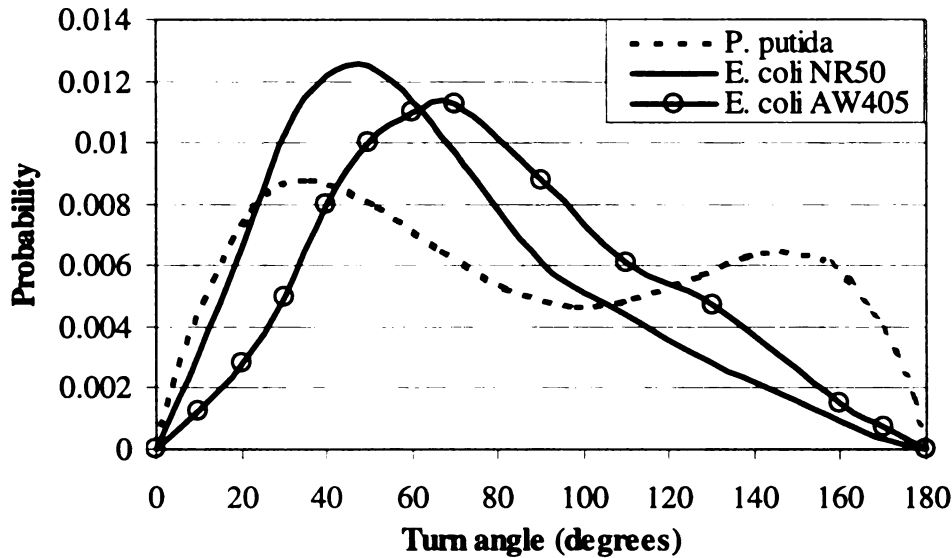


Table 2. Mean, median, and peak values of bacteria turn angle distributions.

Cell type	Mean	Median	Peak
<i>E. coli</i> strain NR50	79.2	75.7	68
<i>E. coli</i> strain AW405	65	59.2	50
<i>P. putida</i>	85	77.2	35, 145

Figure 16 shows a comparison snapshot of the cell density of *E. coli* and *P. putida*. The figure clearly shows that differences in the shape of the turn angle distribution do not significantly affect the motility. Even though our simulation method was developed for peritrichious bacteria similar to that of *E. coli* and *Salmonella*, it works equally as well for *Pseudomonas putida*. *Pseudomonas putida* has polar flagella, a bimodal turn angle distribution and swims twice as fast as *E. coli* (Duffy and Ford, 1997). Even though *P. putida* swims twice as fast as *E. coli*, all the parameters in the simulation

snapshot in Figure 16 where held constant except the turn angle distribution for comparison reasons. Comparisons of the turn angle distributions are in Figure 15.

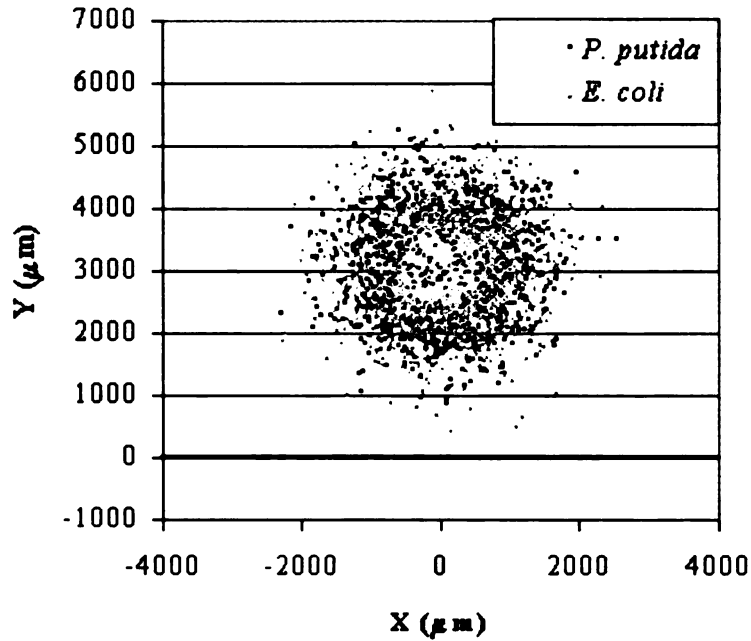


Figure 16. Cell density profile snapshot of *E. coli* and *P. putida* at 30 minutes.

4.7 EFFECTS OF THE MEAN AND VARIANCE OF THE TURN ANGLE DISTRIBUTION

The variance of the gaussian distribution was fixed to a value that mimics that of *E. coli* (30 degrees) and the turn angle was adjusted to values of 0, 45, 90, 135 and 180 degrees. A value of 0 degrees corresponds to the cell continuing to move in its present direction whereas 180 degrees refer to cells making a full reversal. While using a χ_0 value of $2\text{E-}2 \text{ cm}^2/\text{s}$, motility and mean displacement data was taken at these various turn angles. Figure 17 clearly shows that cells utilizing smaller turn angles will migrate farther than those with turn angles that are large. Thus, smaller turn angles are desirable.

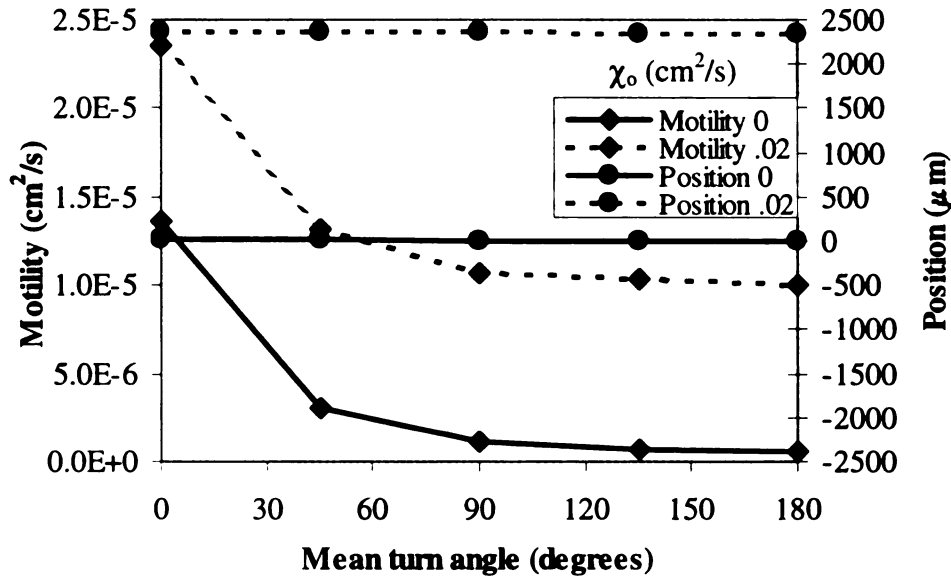


Figure 17. Turn angle mean value versus motility and cell mean distance in a linear nonconsumable gradient.

Figure 18 shows the effect of the turn angle distribution variance on both the motility and the mean displacement of the cells. Simulations were run for both the chemotactic and non-chemotactic scenarios. The variance was adjusted while maintaining a turn angle of 90 degrees (right-angle turns). Using a χ_0 value of $2.0\text{E}-2 \text{ cm}^2/\text{s}$ for the chemotactic case, the turn angle distribution variance was adjusted to 10, 30, 60 and 180 degrees. The process was then repeated for the non-chemotactic scenario. Adjustments in the turn angle distribution variance had no effect on both the motility and the mean displacement of the cells for both scenarios. That is, the motility remained constant at every angle tested as well as the mean displacement of the cells.

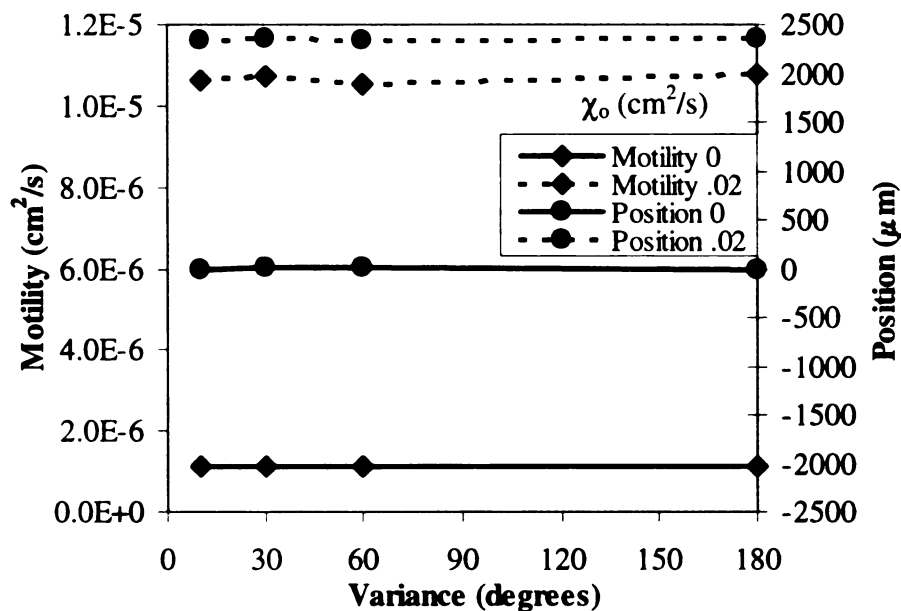


Figure 18. Variance of the Turn Angle versus the motility and cell mean distance in a linear non-consumable gradient.

The accuracy of our model was checked against the widely accepted theoretical equation for random motility (equation 2). This equation accounts for tumbles as instantaneous so simulations with bacteria swimming runs at each time step were performed to make a true comparison. The results of our simulation show excellent agreement in motility for both 2D and 3D cell movement with the motility calculated with the theoretical equation. Plots of motility vs. mean turn angle (Figure 19) and variance (Figure 20) show good agreement over the entire range of angles.

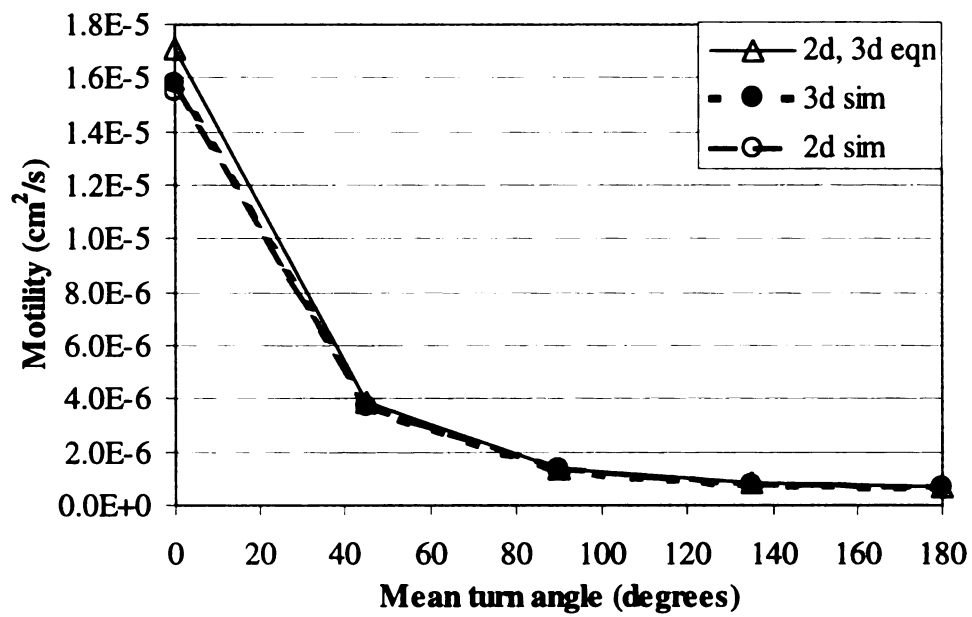


Figure 19. Comparison of the mean turn angle in 2D, 3D against theoretical equation.

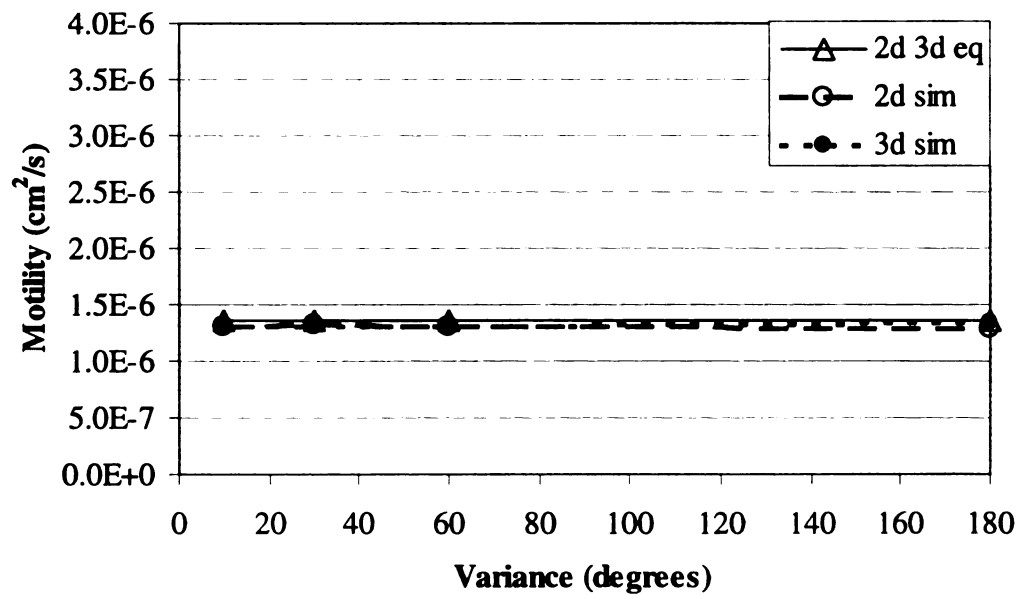


Figure 20. Comparison of the turn angle variance in 2D, 3D against theoretical equation.

5. CONSUMABLE NON POROUS

Knowledge about chemotaxis in a fixed linear gradient can be used to better understand chemotaxis in a dynamic chemoattractant concentration environment. The environment within the presence of a chemoattractant is constantly being depleted based on the location of the cells and then simultaneously replenished by diffusion. This results in a complex concentration profile, which can be approximated as a fixed linear gradient at each cell location for the duration of one time step, thus resulting in a variety of gradient directions. The consumption coefficient (k) is a measure of the rate of a cell's substance intake through its cell wall, which is carried out by transport proteins located in the cell wall. The rate of uptake is affected by the saturation of the receptors by the concentration around the cell (Madigan et al., 2000). The rate is also affected by the availability of the electron donors, acceptors and the cell metabolism (Dillon and Fauci, 2000).

Consumption of chemoattractants causes a band or wave of cells to migrate from the center. Multiple bands of cells have been observed from experiments (Figure 21) and are most likely the result of consumption of multiple chemoattractants (Emerson et al., 1994). Table 3 contains the parameters used in the simulations where cell react to gradients created from consumption of the chemoattractant.

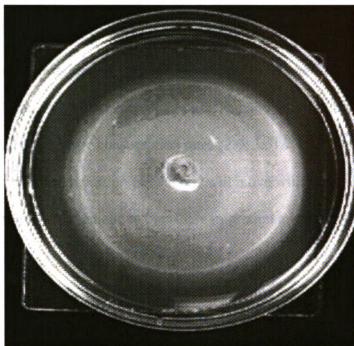


Figure 21. Multiple Chemotactic waves of bacteria due to consumable chemoattractant.

Table 3. Values used in the simulations starting with a uniform concentration of aspartate consumed by *E. coli*.

NP	2000
X	4000 μm
Y	4000 μm
DG	24 μm
P_0	1.2 s^{-1}
v	22 $\mu\text{m/s}$
C	.1 mM
K_d (aspartate)	0.005 mM
χ_0 (aspartate)	5E-4 cm^2/s
D (aspartate)	8.9E-6 cm^2/s
K_d (fuctose)	0.125 mM
χ_0 (fuctose)	3.5E-5 cm^2/s
D (fuctose)	8.6E-6 cm^2/s
k_0	.01 - 8 $\text{s}^{-1}\text{cell}^{-1}$

Figure 22 demonstrates how cells migrate outward from the point of inoculation in response to local attractant gradients formed by consumption with a chemotactic sensitivity of $1\text{E-}5 \text{ cm}^2/\text{s}$. In the snap shot of consumption rate of $0.1 \text{ s}^{-1}\text{cell}^{-1}$ the gradients are too small to bias all the cells away from the center. At the consumption rate of $1.5 \text{ s}^{-1}\text{cell}^{-1}$ all the cells will detect a large gradient and will bias their migration away from the center. At the larger consumption rates ($2.0, 3.5 \text{ s}^{-1}\text{cell}^{-1}$), the gradients are depleted at the center and a portion of the cells will move with random motility while the cells at the perimeter will continue to bias there migration away from the center.

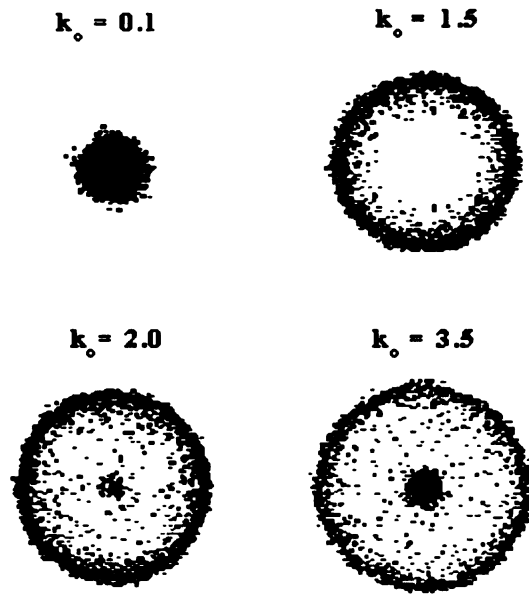


Figure 22. Snapshots with Consumption showing the cell density patterns.

The value of the consumption rate that each pattern happens is different for each set of experimental parameters. The number of cells, initial concentration of chemoattractant, the swimming parameters of the bacteria and the chemotactic sensitivity

and the consumption rate will effect when each pattern will be produced and do not necessarily correspond to values used in other parts of the paper.

The radial cell density shows that at higher consumption rates, cells can be stranded in areas of depleted chemoattractant and will move with random motion while the cells on the outer band will continue to consume the chemoattractant and migrate further away. Leaving cells behind lowers the motility of the population. This also explains the bands formed during experiments. Only 3 of the 4 cell density profiles are shown to enhance clarity. The 30-minute simulation results in the Figure 23 used a chemotactic sensitivity of $1\text{E-}3 \text{ cm}^2/\text{s}$ for consumption of aspartate.

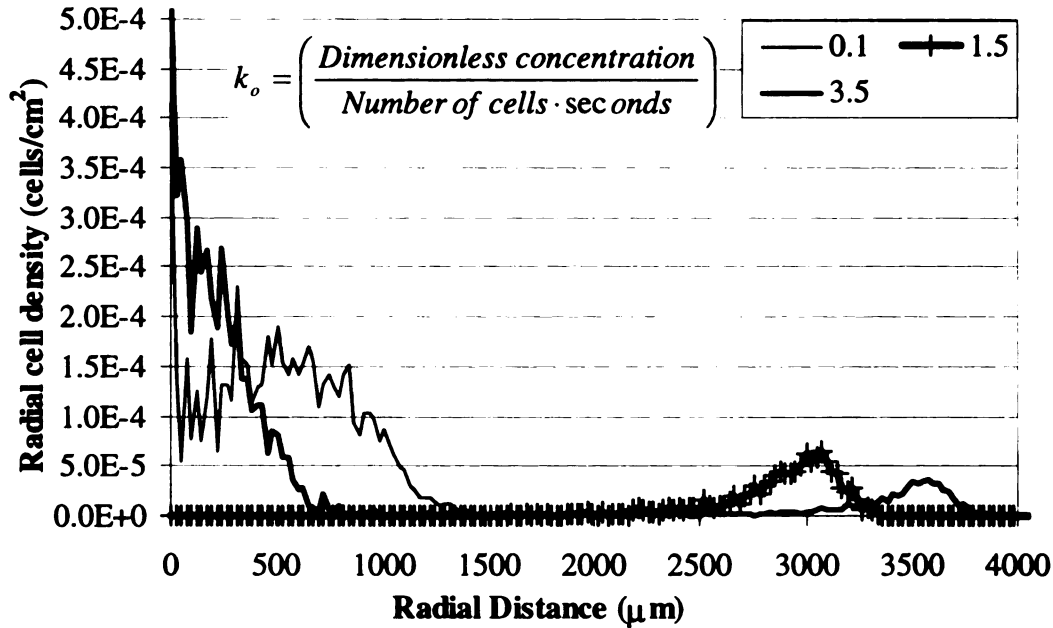


Figure 23. Radial cell density at different consumption rates.

5.1 INITIAL DISTRIBUTION

Simulations were completed with cells initially located at every grid point, thus forming an equal distribution of cells. Within the presence of a constant concentration of chemoattractant, an increase in the consumption coefficient or χ_0 does not affect the

motility. The concentration is constantly being depleted at every point in equal amounts. Therefore, no significant gradients will form. The random movement of the cells will cause small variations in cell density. As a result, small variations in chemoattractant concentration will occur with a random direction of the gradients.

5.2 DIFFUSION

Diffusion occurs naturally from a difference in concentration (concentration gradient) between two nodal points. The rate of diffusion is proportional to the gradient size. A linear gradient is achieved when the diffusion into and out of a point is equal. Inoculation of all the cells at one location produces a cooperative gradient in the radial direction from that location, giving different results than an equal distribution of cells. Since the gradient is radially symmetric, the cells will produce a gradient in the initial direction the cells are facing.

Figure 24 shows that there is an optimum consumption rate that maximizes motility. This corresponds to the point where the produced gradient is the largest without leaving cells behind. In Figure 24 the motility increases to a peak value, which occurs from balancing large gradients with maintaining gradients at the center. At low ($0.1 \text{ s}^{-1} \text{ cell}^{-1}$) and high ($8 \text{ s}^{-1} \text{ cell}^{-1}$) consumption coefficient values, the motility approaches that of random motion, increasing to a peak value, but has a slower drop off. The random motility coefficient calculated in the absence of a chemoattractant gradient using a 0.1 s time step is $1.32\text{E-}6 \text{ cm}^2/\text{s}$. The magnitude of the peak is proportional to the χ_0 value. Widman also observed that the carrying capacity of the gradient with consumption is larger at increased chemotactic sensitivity values (1997).

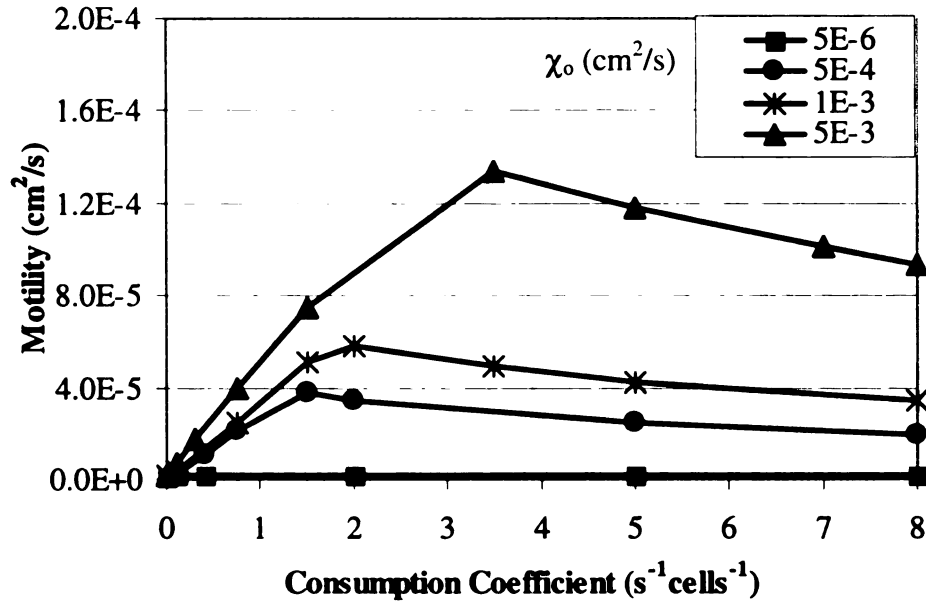


Figure 24. Effect of the consumption coefficient on motility with diffusion.

At consumption coefficients less than $2 \text{ s}^{-1} \text{ cell}^{-1}$, smaller gradients exist, but the chemoattractant is not completely depleted because diffusion occurs equally as fast as consumption. The cells subsequently can find an existing gradient and begin migration from the center of inoculation. At consumption coefficients greater than $4 \text{ s}^{-1} \text{ cell}^{-1}$, the existing gradients are larger, but the chemoattractant is completely depleted at the center, thus a majority of the cells are left behind and move with random motility. This causes a decrease in population motility. The consumption simulations correlate with the error for the linear gradient also yielding less than 3% error. The error bars are smaller than the graph symbols, so they were left off for clarity.

5.3 NO DIFFUSION

The most noticeable difference between the diffusion and non-diffusion simulations is the diffusion case has a higher peak value (Figure 25). At χ_0 of $5\text{E-}3 \text{ cm}^2/\text{s}$

the maximum motility coefficient calculated to be $1.34\text{E-}4 \text{ cm}^2/\text{s}$ for diffusion and $2.4\text{E-}5 \text{ cm}^2/\text{s}$ for the no diffusion simulation. The diffusion simulations have dark lines and no diffusion has gray lines.

The results of these simulations indicate that diffusion is important in cell migration with consumption and should not be ignored. For non-diffusion simulations, only the cells at the front of the chemotactic wave will experience gradients. These cells will depleted the chemoattractant completely leaving most of the cells at the center moving with random motion. Increasing the chemotactic sensitivity increases the motility for all the values of consumption coefficient that were simulated.

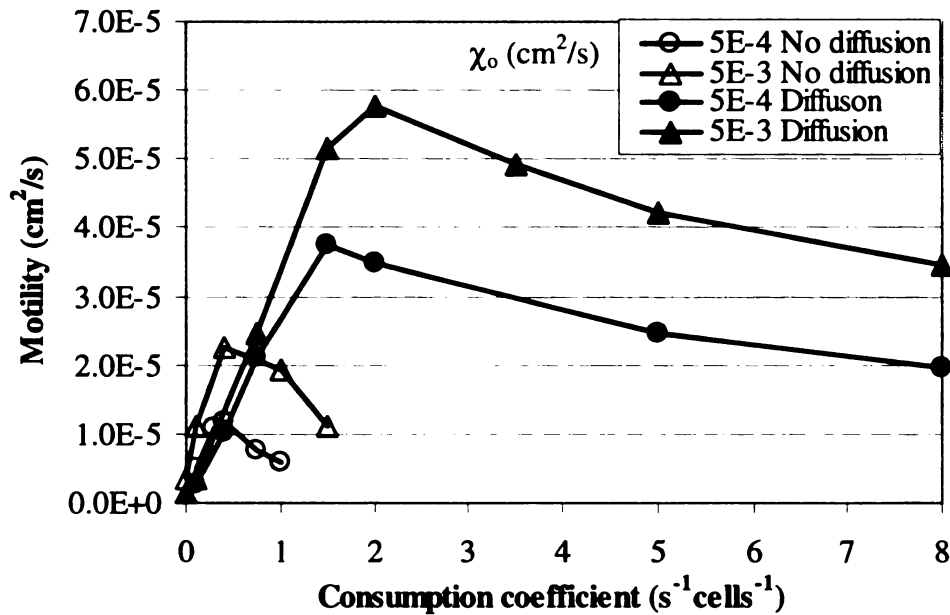


Figure 25. Motility with diffusion and no diffusion on a logarithmic scale.

An interesting phenomenon is at low consumption coefficients the non-diffusion case has a higher motility coefficient because when none of the cells get left behind in either case the only effect of diffusion is smoothing out the gradients. At higher consumption coefficients the diffusion case has a higher motility coefficient because the gradients are maintained and the cells don't get stranded. The effect of diffusion on the

motility coefficient for the equal distribution of cells doesn't have much of an effect because of the lack of gradients to smooth out.

5.4 ERROR ANALYSIS

Motility values were obtained by running replicate simulations, while altering the seed number for the random number subroutine and keeping all other parameters constant. A standard deviation was calculated for each scenario and used to determine an error percentage. The percent error for all simulation results, including the linear gradient and consumption cases, is at or below 3%. A notable trend with regard to the error analysis is the percent error increases with an increase in motility. At larger motility values, the cells will move farther from their initial positions, giving them more possible locations thus more variability.

5.5 ALTERNATIVE METHOD OF GRADIENT DETECTION

The decaying step function is ideal for validating chemotactic models with consumption of a chemoattractant. The decaying step function creates both changing temporal and spatial gradients and is easily defined. Comparisons of different parameters can be done knowing that the gradient will be the same for each case. The concentration profiles in each case are similar resulting in a common gradient detection response.

The simulations start with a uniform concentration of 0.1 mM aspartate and all the cells starting at the center yielding a plot of the cell density and the concentration profile due to consumption of the chemoattractant. The peak of the cell density is located at the lower concentrations of the formed gradient. In Figure 26, the chemoattractant concentration is completely diminished resulting in a second peak of cell concentration located at the origin that moves with random motility. In Figure 27 almost all the cells

are biased away from the center resulting in a wide peak compared to plot a where a narrow peak is separated from the center peak by an area of low cell concentration.

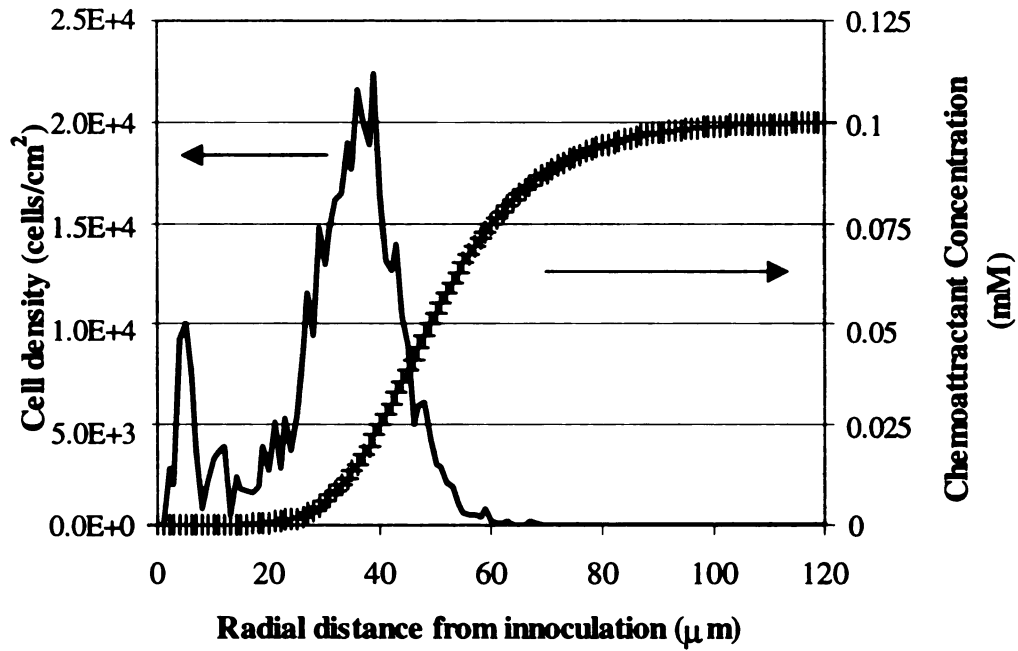


Figure 26. Cell density profile where chemoattractant is depleted at center.

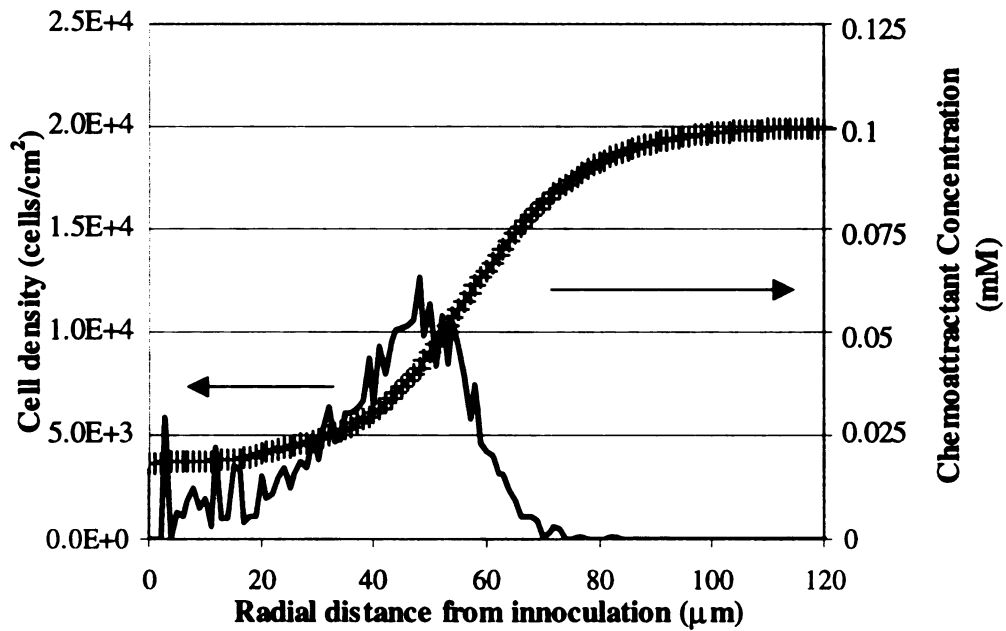


Figure 27. Cell density profile where chemoattractant is not depleted at center.

An alternate method of detecting the chemoattractant gradient has been developed to account for temporal gradients and compare to spatial gradients, since experiments have indicated that bacteria react to changes in concentration over the 4 previous seconds (Berg, 1993). In the case of the decaying step function, the center position of the concentration gradient is fixed and the cell density changes as it reacts to the gradient. The gradient also decreases with time. Hypothetically cells at the top of the gradient will detect a lower concentration temporally and may chemotactically go the wrong way down the gradient. Cells at the low end of the gradient will sense a higher concentration over time and may swim away from the gradient. This will not be a problem because simulations show cell density bias up gradient within 1 second. A comparison of different chemotactic sensitivities on the cell density in response to a fuctose gradient at 6 minutes is shown Figure 28. Initially the relative cell density is uniform at 1 all along the chamber.

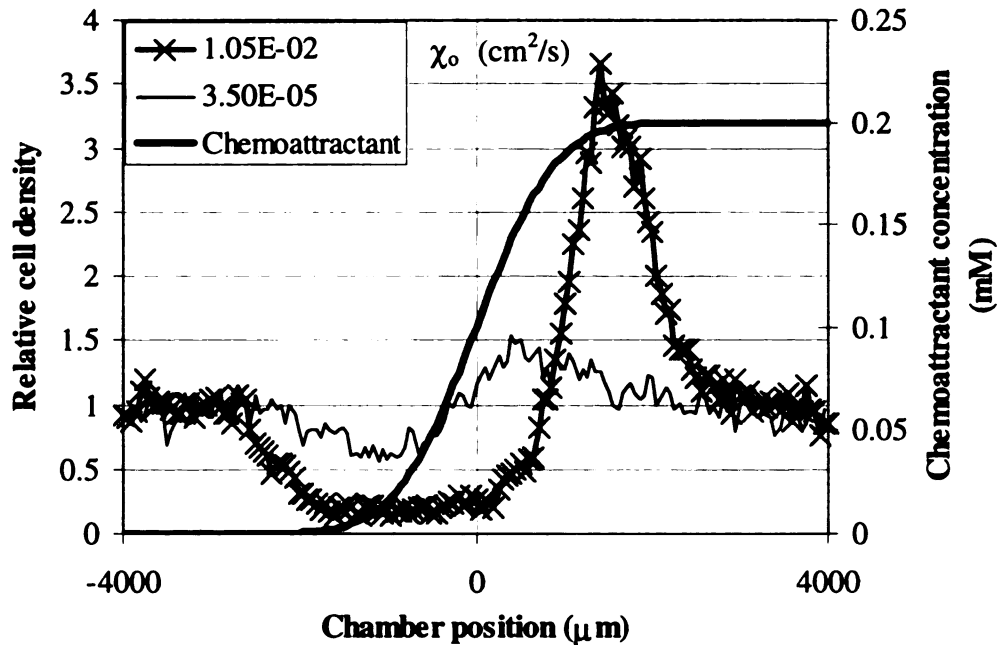


Figure 28. Comparison of different chemotactic sensitivities on the cell density profile.

Simulations using the values for fuctose were used to compare our simulation results to those obtained by Ford's research lab for cells responding to a decaying step gradient. The shape of our relative cell density profiles correlate well with those obtained by Ford's research group.

The original mathematical relation for the chemotactic biasing of the tumbling probability included a time derivative as well as the spatial dot product. Ford assumed that the time derivative was much smaller than the spatial derivative so it was dropped from the simplified equation.

$$p = p_o \exp \left(- \frac{\chi_o^{3D}}{v} \frac{K_d}{(K_d + C)^2} \left(\frac{1}{v} \frac{dC}{dt} + S \bullet \nabla C \right) \right) \quad (20)$$

Our simulations show that the time derivative is on the same order of magnitude as the spatial derivative for the simulations with a chemotactic sensitivity for fuctose of $3.5\text{E-}5 \text{ cm}^2/\text{s}$. The combined simulations with the spatial and time derivative resulted in a much enhanced cell density up gradient. Simulation using the full time derivative and dot product results, indicated by the line with x's, in a cell density profile much enhanced over either one separately.

An alternate method of detecting the gradient by the concentration change at each cell at 4 seconds as shown in equation 21 was developed.

$$p = p_o \exp \left(- \frac{\chi_o^{3D}}{v} \frac{K_d}{(K_d + C)^2} \left(\frac{dC}{\left(\sqrt{dX^2 + dY^2} \right)} \right)_{dt = 4} \right) \quad (21)$$

The simulation results in Figure 29 show a significantly lower motility for this method where the data series with the squares represents the temporal gradient converted to a spatial gradient.

Obviously this method under predicts the gradients that are seen using the regular spatial gradient depicted by the regular solid line. The simulations using only the time derivative result in a cell density profile depicted by circles that is slightly reduced from the regular spatial derivative. Fucose with a chemotactic sensitivity of $3.5\text{E-}5 \text{ cm}^2/\text{s}$ was used in the comparison simulations at 12 minutes.

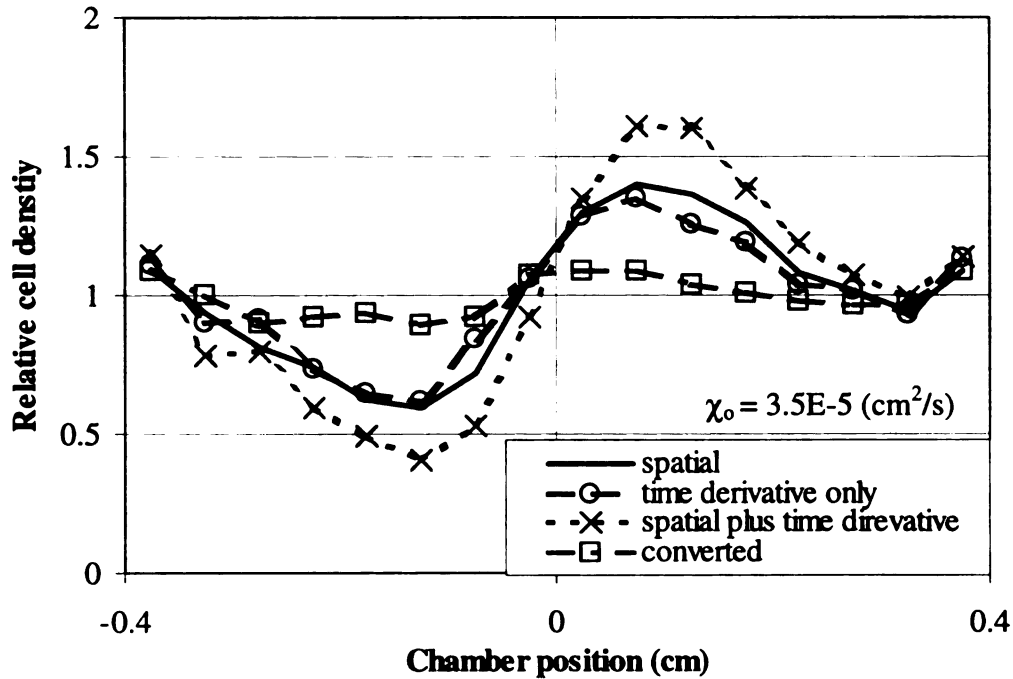


Figure 29. Relative cell density profiles of gradient sensing methods with χ_0 of $3.5\text{E-}5$.

A close correlation of cell density profiles resulted from the simulations using $105\text{E-}4 \text{ cm}^2/\text{s}$ for a chemotactic sensitivity for fucose at 6 minutes for the entire set of different gradient sensing methods except the simple temporal method, which didn't cause much biased migration as seen in Figure 30. The addition of the time derivative

does enhance the chemotactic migration of the bacteria at lower chemotactic sensitivities, but when the chemotactic sensitivity is very high, tumbling is already suppressed so the addition of the time derivative will not have a significant effect as seen in Figure 30.

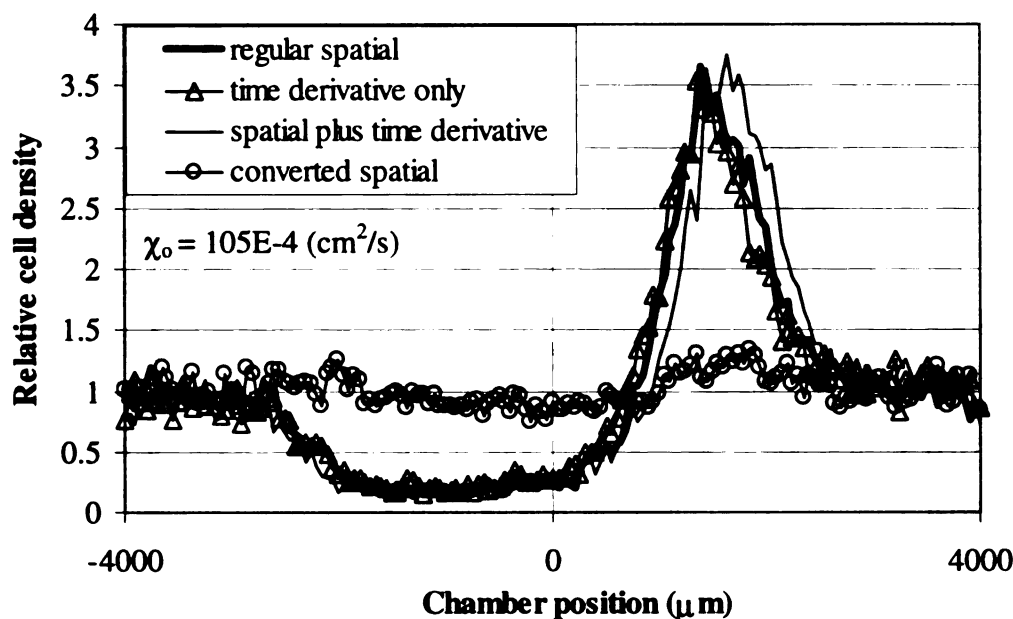


Figure 30. Relative cell density profiles of gradient sensing methods with χ_0 of 105E-4

5.6 DEPENDENCE OF CONCENTRATION ON CHEMOTAXIS

The consumable chemoattractant concentration profile resembles a decaying step function, but is shifted along with the bacteria as they migrate away from the center. The peak of the bacteria density in Figure 31 stays at the base of the gradient resulting in the continued sensing of a large gradient. The relation between the concentration profile and the cell density profile that we obtained from the CD model agree with the experimental profiles researchers have observed. Amperometric sensors have been used to measure the spatial concentration gradients of the chemoattractants glucose and oxygen, while simultaneously measuring the cell density with a CCD camera. In experiments with a

diffusion gradient chamber, the front of the chemotactic wave showed a high density of cells at the base of the gradients of glucose and oxygen that were produced from consumption (Peteu et al., 1998). In the simulation results without growth, the gradient decreases as the bacteria thin out from covering a larger area, and diffusion replenishes the area behind the wave front.

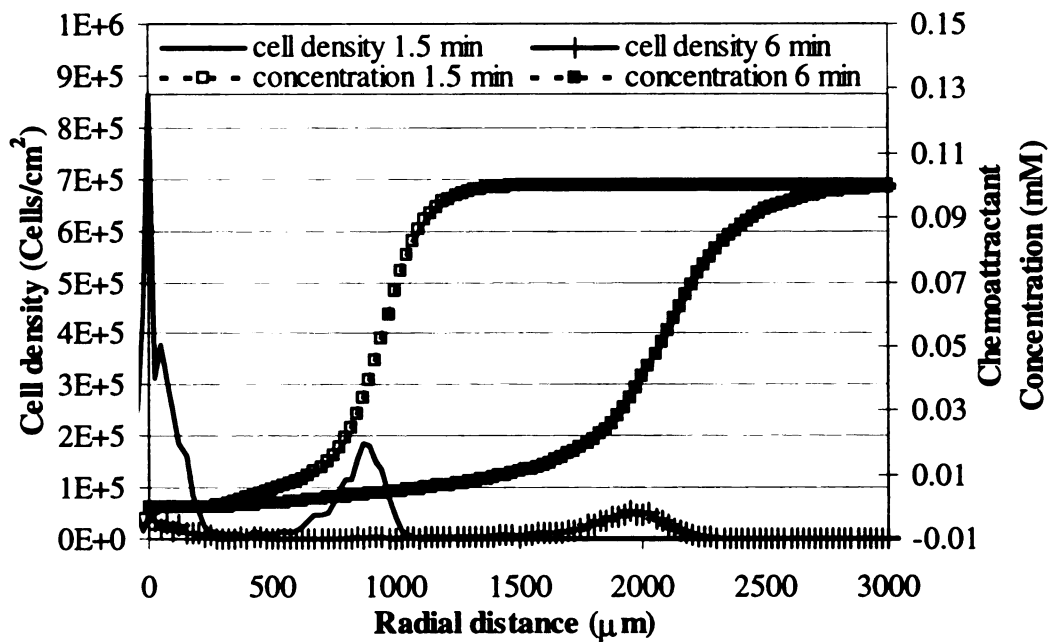


Figure 31. Cell density profile and concentration profile due to consumption.

The concentration that the cell senses has an effect on how strongly the cell population migration is biased when exposed to a gradient. A decaying step gradient of methylaspartate was simulated with 3000 cells starting at the midpoint of the step gradient. To determine the effect of the concentration, the concentration that the cell senses was varied but the magnitude of the methylaspartate gradient was kept constant by shifting the concentration up and starting with an initial step change in concentration of 0.2 mM. In Figure 32, cell density 1 responded to a concentration varying between 0 and

.2 mM, which shifted the peak up gradient 850 microns. Cell density 2 corresponds with cells that had a step gradient of between .2 and .4 mM and shifted the peak 400 microns up gradient. The peak of cell density profile 3 was still centered at the cells initial location responding to a gradient between .8 and 1 mM indicating saturated receptors.

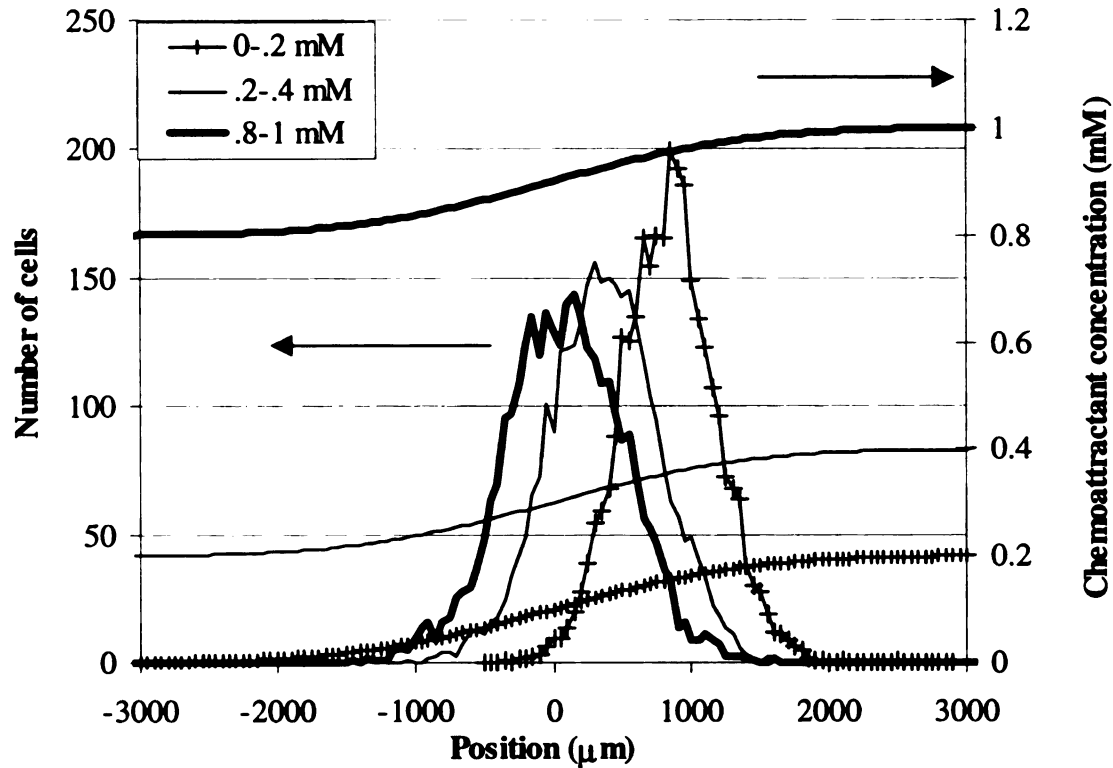


Figure 32. The effect of concentration on cell density in a decaying step gradient.

6. MIGRATION IN POROUS MEDIA

Migration of bacteria in pores is a widespread phenomenon and needs to be understood to apply them to applications in bioremediation in soil and infection of implants in the body. Porous media changes the dynamics of bacterial migration, hydrodynamic effects of solid surfaces and sorption to the solid surfaces needs to be taken into account. Figure 33 shows a schematic of the swimming path of a cell through a bed of spheres. Just as important and not as widely considered is the production and the

magnitude of chemoattractant gradients can influence the direction and rate of dispersal of bacteria.

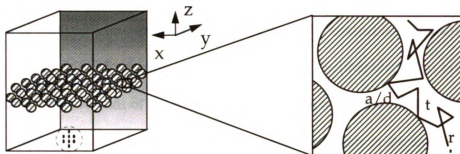


Figure 33. Schematic of cell trajectory through a bed of spheres.

Experiments with *Pseudomonas KC* reacting to a nitrate gradient showed farther migration when swimming through porous soil core samples than when swimming in bulk phase fluid (Figure 34). This result seemed surprising since one would speculate that the migration in pores would be hindered due to collision and possible adsorption with soil particles. The penetration ratio in this paper is defined as b/a .

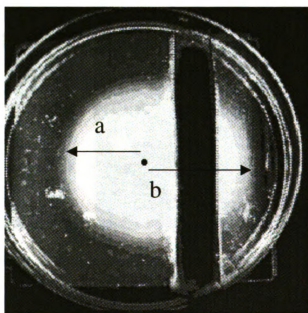


Figure 34. Photograph of *Pseudomonas KC* migrating faster through sand core sample.

6.1 SLIT PORES

The first set of simulations we did was look at the behavior of a population of cells in a simple slit pore with and without consumption. In Figure 35, we set the initial location of 2000 cells into the center of the slit pore of consumable chemoattractant. After simulations of 15 minutes, the cells indicated by the little black dots bias their migration to the two ends of the pore. The chemoattractant at the center of the pore is depleted indicated by the light gray color and a gradient points in the direction of the openings of the pore as the cells follow the gradient. The black area at the openings of the pores indicates the high concentration of chemoattractant while the black diamond indicates the average location of the cells.

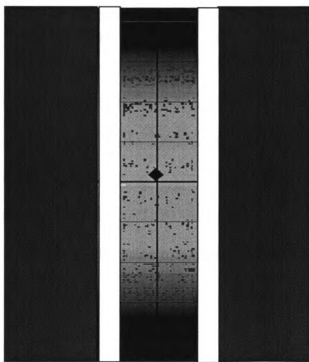


Figure 35. Simulation results of cells in a slit pore.

We found that with a linear non-consumable chemoattractant, the motility did not change very much as the pores got smaller. With the consumable chemoattractant, the

motility of the cells was more complicated. Under some conditions, the motility was actually lower than random motility where gradients are not sustained and collisions with the pore wall hindered movement as seen in Figure 36.

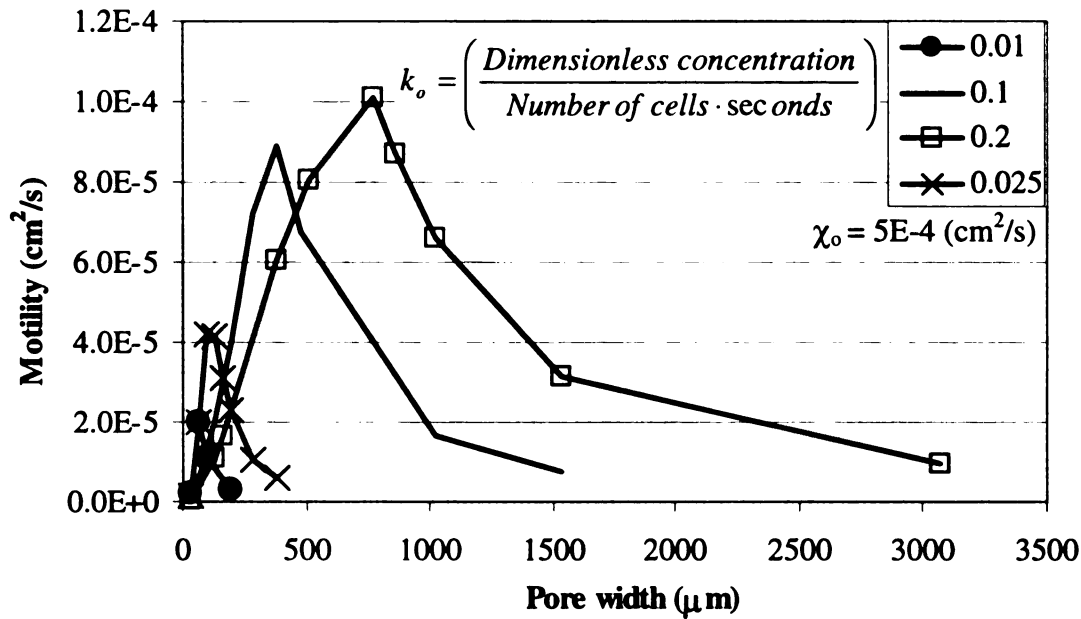


Figure 36. Effect of pore width on motility at different consumption rates.

Figure 37 shows that there is an optimum population consumption rate for each size pore. Cells in smaller pore sizes benefit from a smaller consumption rate. Larger gradients could be formed quicker as the cells are confined until group consumption outpaces diffusion so the chemoattractant is depleted and no gradient is sustained for other cells to detect.

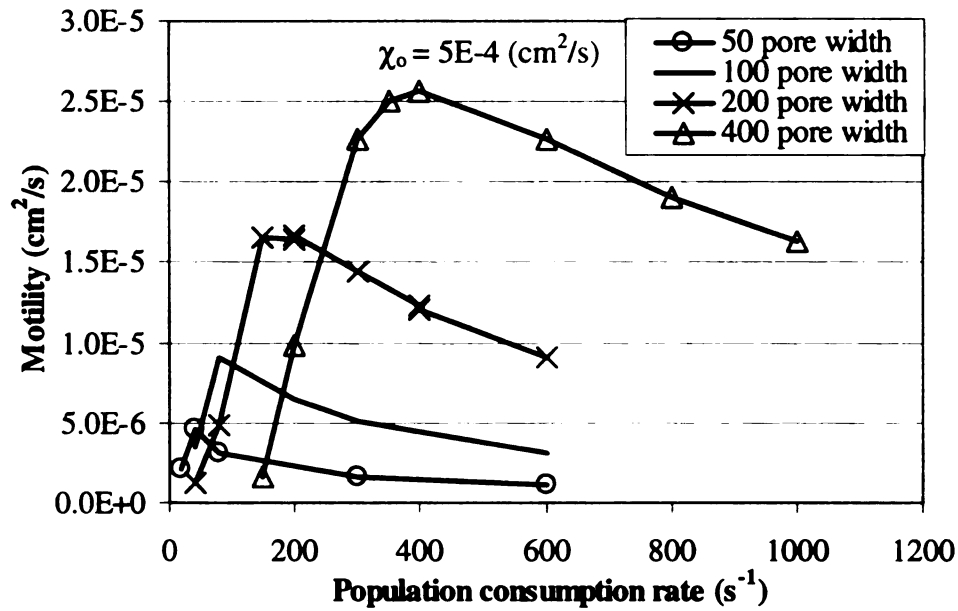


Figure 37. Motility at different population consumption rates.

The combined term of the consumption rate times the number of cells gives the population consumption rate which determines the chemoattractant gradient and thus determines the motility. Different consumption rates with different number of cells but the same product results in virtually the same motility indicating that a larger population of cells could be effectively modeled by using a larger consumption rate and a smaller number of cells. The motility for each population consumption rate rises to a peak almost linearly and then drops off with a quadratic order at larger pore sizes (Figure 37). The linear regime represents a sustained chemoattractant gradient and over time more and more cells move away from the center and begin to bias their motion up gradient.

The curvilinear regime represents the continued depletion of the chemoattractant at the center and only a fixed portion of the cells move chemotactically up the gradient and the rest move with random motion. Smaller population consumption rates had

correspondingly smaller motilities with peaks at smaller pore widths and a smaller range of chemotactic acceleration in pores. Larger cell populations have the ability to accelerate their motility in larger pores giving them a competitive advantage against smaller colonies of cells in the same size pore as shown in Figure 38. Smaller colonies only have a marginal advantage in small pores since the gradient is depleted and only a fixed amount of cells will sense the gradient.

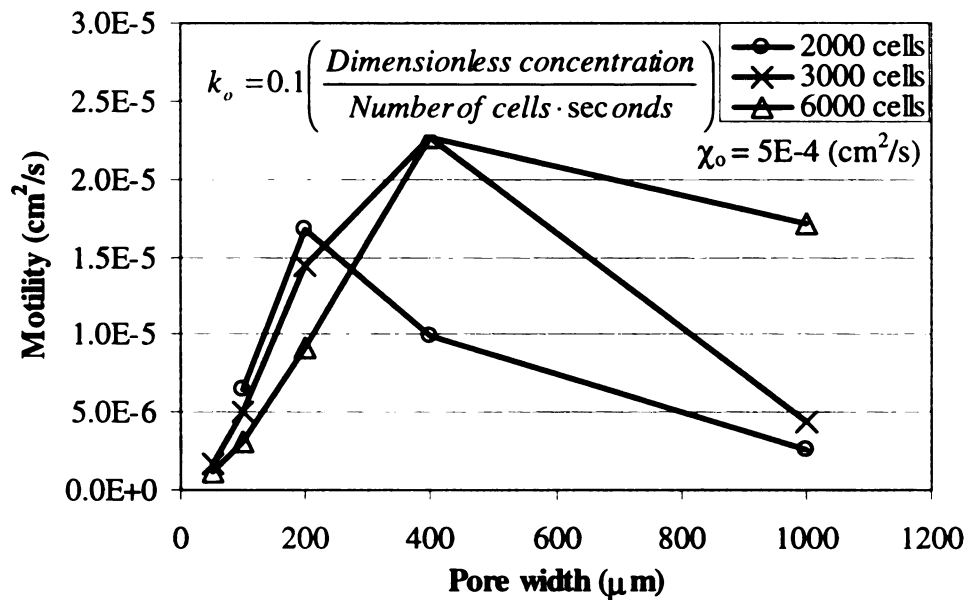


Figure 38. Effect of pore width on motility with different number of cells.

To determine what causes enhanced migration in pores, a decaying step gradient was simulated in different size slit pores (Figure 39). Three thousand cells responding to gradients in aspartate with a chemotactic sensitivity of $5\text{E}-4 \text{ cm}^2/\text{s}$ at 6 minutes were used for these comparisons and a consumption rate of $0.1 \text{ s}^{-1} \text{ cells}^{-1}$ was used for the simulations of cells creating their own gradient by consumption. Smaller pore sizes showed a small decrease in the motility for the random, and non-consumable step gradient. The presence of a consumable chemoattractant enables accelerated migration in

pores. The magnitude of the acceleration is constant for larger pore sizes and increases rapidly at moderate pore widths and then drops off at a critical pore width.

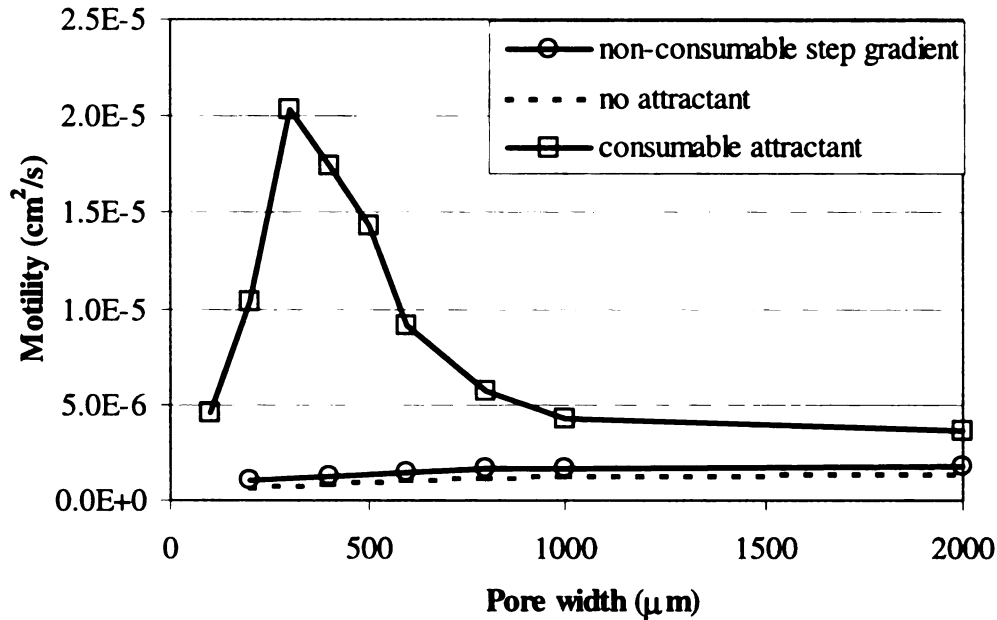


Figure 39. Comparison of motilities in pores with consumption and no consumption.

Small changes in the cell density profile in Figure 40 indicate that the interaction from the pore walls doesn't change very much in smaller pore sizes in the presence of a large gradient. Three thousand cells responding to gradients in methylaspartate with a chemotactic sensitivity of $4.1\text{E-}4 \text{ cm}^2/\text{s}$ at 6 minutes were used for these comparisons. Since the gradient is equally defined in the simulations, the enhanced migration through pores must result from the combination of a larger gradient and a lower consumable chemoattractant concentration.

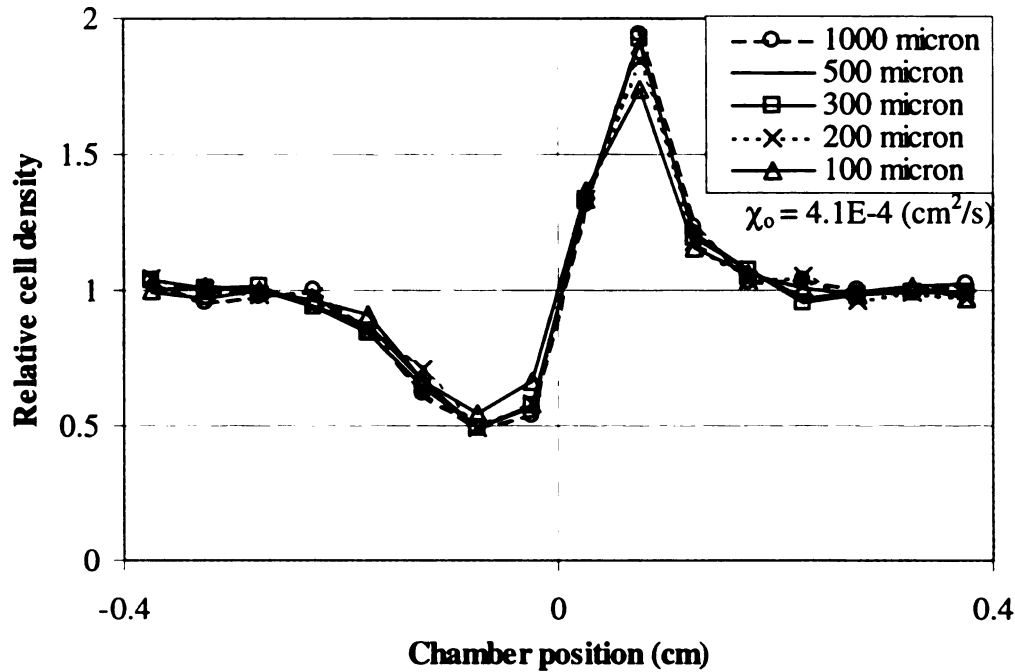


Figure 40. Cell density profile for decaying step gradient in different pore widths.

6.2 MULTISLIT BARRIER

Accelerated migration was shown to happen through tubing connectors with a pore width of 0.4 mm, both glass beads and sand also showed accelerated migration. To explain the results of the experiment with cells moving faster through the porous material then through the bulk phase, simulations were set up to model the experiment. A barrier of parallel horizontal slit pores were set up on the right side of the simulation space with the cells initiated at the center. These simulations will be used to explain the results of the experiment in Figure 34. The porosity in the multislit simulations had a small effect on the penetration ratio when the pore width was kept constant and only the solid region width was increased. Simulations with larger solid regions (Figure 41) did have lower penetration ratios then the smaller solid region (Figure 42) simulations.

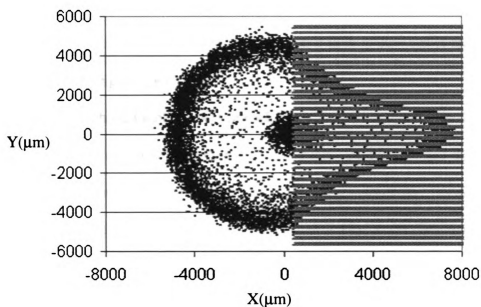


Figure 41. Migration into multiple 100-micron slit pores with a porosity of 33%.

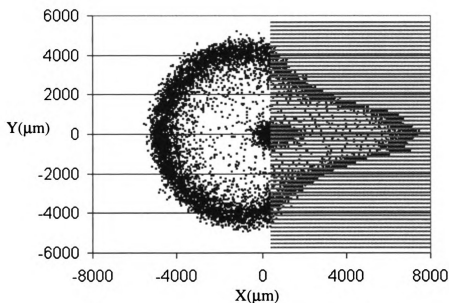


Figure 42. Migration into multiple 100-micron slit pores with a porosity of 50%.

We have established that once a sufficient number of cells are in pores they can collectively create large localized gradients causing accelerated migration in pores. Cells outside of a porous barrier moving with only random motility will be greatly impeded by the barrier and will barely penetrate into the pores. What causes the cells to be drawn

into the pores? Logic would state that there would have to be some kind of directed migration towards the pore openings.

It was noted that a lot of bubbles formed in and around the porous media when the cells migrated through them. *Pseudomonas KC* favorably consumes nitrate until it is depleted then it metabolizes nitrite, which is a weaker chemoattractant. The metabolism of nitrite results in the formation of nitrogen gas, which is most likely the cause of the gas formation. The absence of bubbles in the rest of the bulk phase indicates that the concentration of nitrate or nitrite is more plentiful then in the pores causing less chemotaxis in the bulk phase. Low concentrations of multiple chemoattractants in series may also cause accelerated migration in pores. Multiple gradients give rise to multiple chemotactic waves of cells, which would increase the number of cells that could chemotactically enhance their migration through the pores resulting in more cells on the other side of the pores or further from their initial location. Our model currently can only simulate one chemoattractant so enhanced gradients of a single chemoattractant was studied.

The hypothesis that maybe there was additional nitrate in the sand core sample that is drawing the cells in was tested by starting out with a higher concentration of chemoattractant in the slit pores. This may explain the experimental results for the soil but not the glass beads or tubing connector. This scenario seems plausible since in the capillary experiments, a gradient of chemoattractant diffusing out of the capillary draws the cells in to it.

As for the glass beads and tubing connector, what kind of concentration in the pores could cause accelerated migration? We know (from the tumbling probability

equation and results) that chemotaxis is enhanced at low chemoattractant concentration. If the concentration of chemoattractant is lower in the pores this could cause accelerated migration compared to the bulk even when going against the gradient at the mouth of the pore.

What would cause the chemoattractant in the porous material to have a lower concentration than the bulk concentration? Reviewing Laura Boom's research notebook indicated that the experiment that showed accelerated migration through the tubing connector was wet when it was placed in the agar. The water that was in the connector likely diluted the concentration of the chemoattractant in the connector or caused a dilute chemoattractant film against the agar for the cells to swim through. Mark Widman's proposed explanation was the bacteria could swim faster due to reduced resistance (1997).

These explanations seem reasonable since when the experiment was repeated with a dry connector, accelerated migration was not noticed. The lab notebook does not indicate if the beads and sand were dry after they were autoclaved. Residual water or condensation may have caused the chemoattractant to be diluted in the porous media.

The accelerated migration could also be explained without having a diluted chemoattractant in the porous media. The experiments showing accelerated migration in the porous media lasted between 63 to 91 hours with a porous media thickness of 10.5 mm to 15.5 mm. It takes 2 hours for the number of bacteria in pores to double, which is twice as long compared to growth in the liquid phase (Sharma, 1993). Even if a few cells enter the porous media, the number of bacteria would double 45 times producing a lot of

bacteria in the pores. Results from the slit pore simulation indicate that a large amount of cells in a pore will cause accelerated migration.

The results of our simulations indicate that increased swimming velocity or cell growth and division is not required to accelerate migration through pores but may enhance it. Modulation of the turn angle distribution caused by confinement of pore walls is not needed either.

6.2.1 Non-reflective barrier

The first simulations were done with a uniform concentration of 0.1mM aspartate and non-reflective boundary conditions when a cell encounters a wall. The consumption rate used in these simulations is $0.1 \text{ s}^{-1} \text{ cells}^{-1}$. Different pore widths and slit wall thicknesses were tried with different consumption rates to see if the cells would move into the pores and move farther compared to the cells swimming in the opposite direction in the bulk phase. In the simulations of cells entering multislit pores with widths of 900 microns in Figure 43 showed a penetration ratio of 2.06. The penetration ratios obtained from experiments reported in Laura Boom's lab notebook ranged from 1.2 to 1.6. The radial cell density plot in Figure 44 shows a narrow chemotactic wave at its farthest point and two other high cell concentration peaks were located at the mouth of the pore entrance. Only one cell concentration peak developed for migration into the bulk area. The concentration gradient is steeper in the porous area than in the bulk.

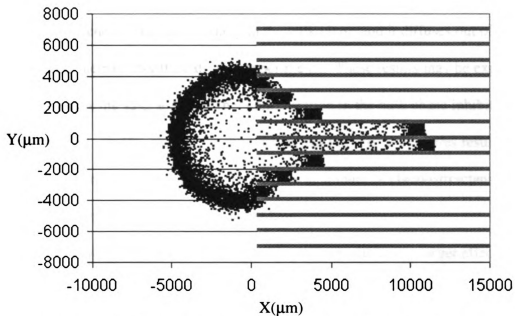


Figure 43. Cell migration into multislits with a uniform initial concentration.

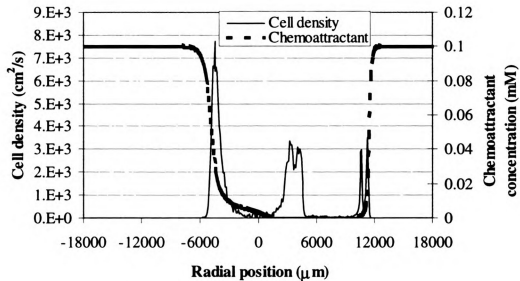


Figure 44. Radial cell density profile and concentration profile with a uniform initial concentration.

Simulations with higher concentration of chemoattractant in the slit pores actually show a decreased penetration into the pores. Although increased penetration occurs when there is no chemoattractant initially in the bulk phase and it diffuses out of the pores, which is the basis for the capillary experiments. These results may be explained by the receptors being saturated in higher concentration in the bulk phase inhibiting chemotaxis. The simulations with no chemoattractant initially in the pores resulted in a lower motility into the pores indicating that a critical amount of chemoattractant is needed in the pores to create an adequate gradient.

At the lower concentration in the pore, a gradient will have a larger effect on the tumbling probability as equation 1 predicts leading to the increased motility through the pore that we see in the simulations. Simulations in multiple parallel pores were done with pore concentrations at half to a fifth of the bulk concentration. The solid pore walls had a thickness of 100 microns. Pore openings ranging from 50 microns to 1800 microns all showed an increase in penetration at 30 minutes compared to 15 minutes. A lower chemoattractant concentration in the pores resulted in further penetration into the pores, while a higher chemoattractant concentration decreased the penetration. Cells will swim into the pore by random motility even though there is a gradient decreasing into the mouth of the pore. Once they are inside they will create a gradient by consuming the chemoattractant that will bias their migration through the pore. Figure 45 compares the results of the simulations of cells entering multislit pores with widths of 900 microns giving a penetration ratio of 2.24, which is 8 % larger than the in the simulations of uniform initial chemoattractant concentration. The radial cell density plot in Figure 46 shows a smaller leading edge cell density peak and a large peak at the pore entrance. The

concentration profile was shallower due to diffusion of chemoattractant from the opposite side of the pore openings.

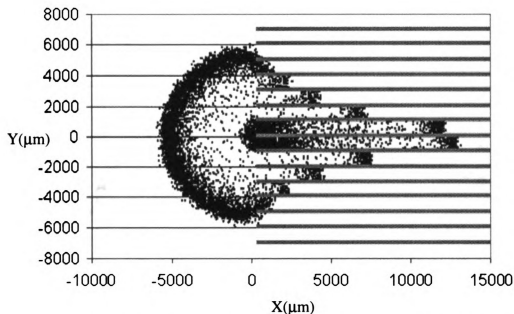


Figure 45. Cell migration into multislits with a lower initial concentration in the pores.

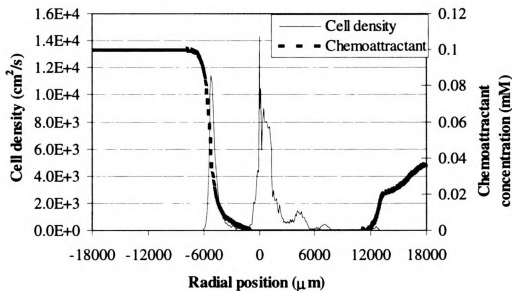


Figure 46. Radial cell density profile and concentration profile with a lower initial concentration in the pores.

The motility and penetration ratio both peak at a pore size of around 700 microns and decreases at smaller pore widths and at larger pore widths as shown in Figure 47 and Figure 48. The simulations with lower chemoattractant concentration in the pores always resulted in a higher value.

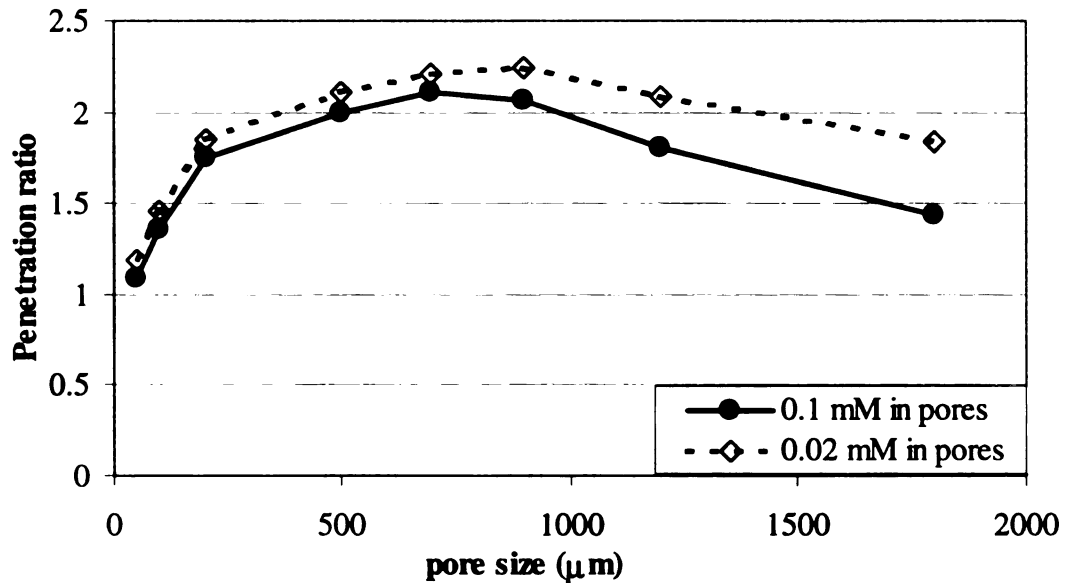


Figure 47. Effect of chemoattractant concentration on penetration ratio.

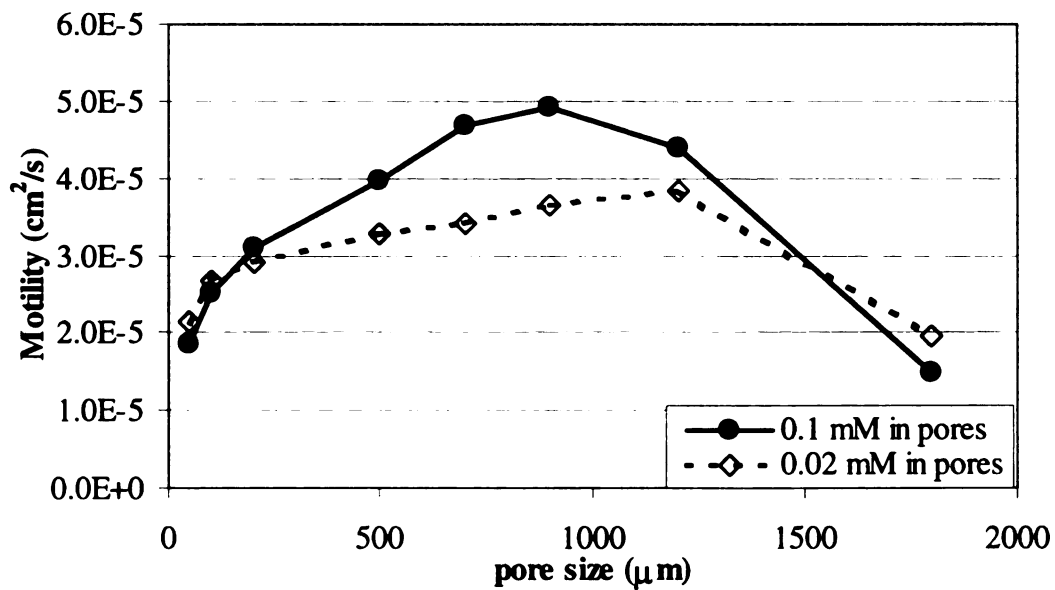


Figure 48. Effect of chemoattractant concentration on the motility coefficient.

Simulations were done using shorter pore lengths with a 100-micron pore width to observe the behavior when the cells exit a porous region. The acceleration of migration continues when the cells are in the pores, but when the porous region stops the migration is no longer enhanced. In Figure 49 the cells stay in the area against the outside of the pores and are more likely to swim back into the pores when the concentration is lower in the pores.

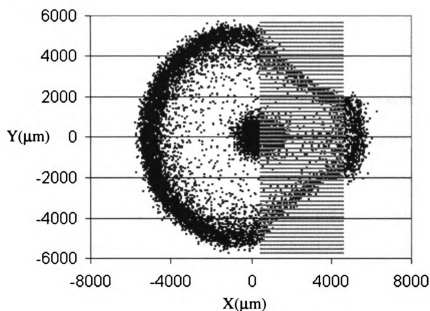


Figure 49. Enhanced migration with initial lower concentration in multiple slit pores.

In Figure 50 with uniform initial chemoattractant concentration when the cells exit the pores they fan out in a semicircular pattern and a chemotactic wave forms.

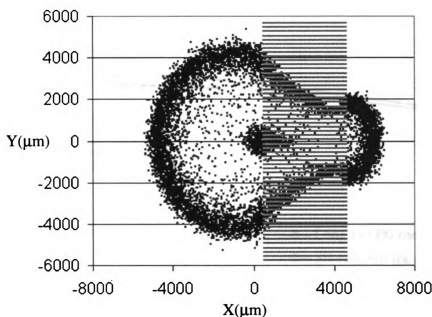


Figure 50. Enhanced migration with initial uniform concentration in multiple slit pores.

In the short pores where the cells swim out the opposite end the highest penetration ratio is when the concentration is the same in the pores as in the bulk solution. The penetration ratio decreases at both higher and lower chemoattractant concentration in the pores as shown in Figure 51, while the motility continues to increase with decreasing concentration for the range of concentration simulated in Figure 52.

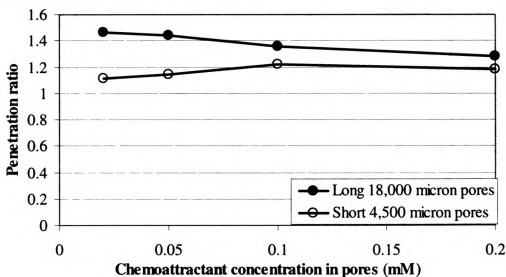


Figure 51. Comparison of the penetration ratio on short and long multislits.

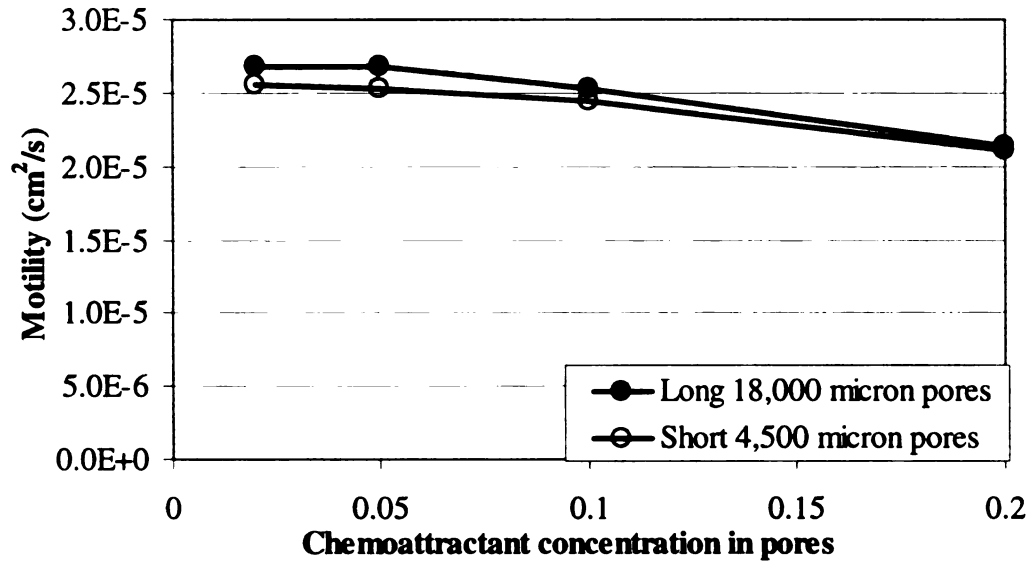


Figure 52. Comparison of the motility coefficient on short and long multislips.

The effect of porosity of the porous region was determined by simulating 100-micron pore widths with various sizes of solid pore wall between them. The penetration ratio seemed to decrease then level off at higher porosity shown in Figure 53, while the motility increased and then leveled off.

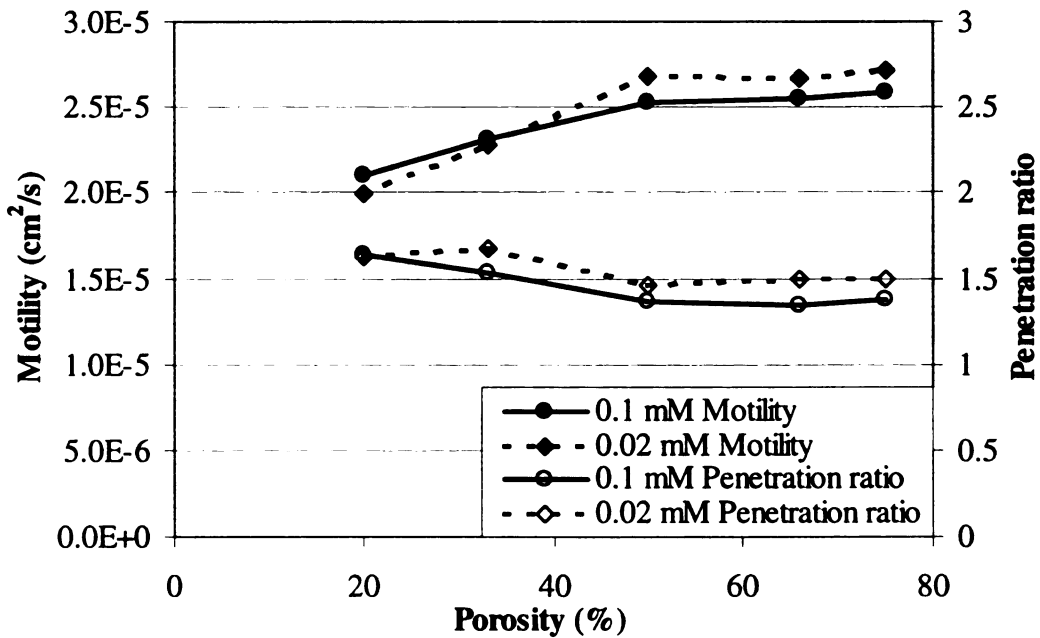


Figure 53. The effect of porosity on the penetration ratio and motility in multislips.

These simulations only use one chemoattractant so when it is diminished biased migration is inhibited. In experiments and natural conditions, bacteria are able to detect multiple chemoattractant gradients and create new ones by metabolism of a primary chemoattractant (i.e. nitrate to nitrite to N_2).

6.2.2 Reflective barrier

For the case of reflective pore walls, the motility through the pores from our simulation is higher than through the bulk when in the presence of a consumable chemoattractant. In the cases where the chemoattractant is not consumable, the motility through the pores corresponds with the motility in the bulk since movement is not denied when it collides with a wall it is just repositioned. The reflective pores always resulted in further penetration into the pores than the non-reflective simulations. The cell density profiles for 200-micron pores in Figure 54 show that cells migrate farther into the pores when the pore walls are reflective.

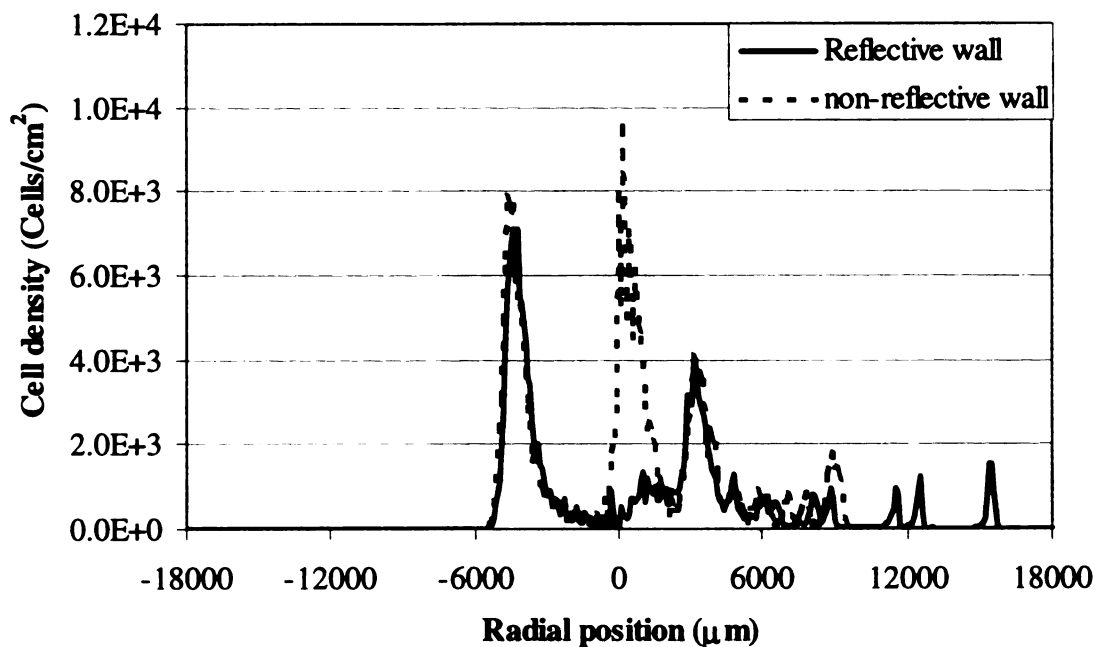


Figure 54. Comparison of reflective and non-reflective walls of a 200-micron multislit.

The mode found in natural environments is somewhere between the reflective and non-reflective case. The type of soil and the adsorption properties of the bacteria would determine the amount of retention of the bacteria in the pores. Figure 55 shows the cell snap shot of cells moving through multislit pores with 900-micron width with reflective pore walls. The penetration ratio is 2.33 which compares to the non-reflective simulations where the penetration ratio was 2.06, yielding a difference of 11.6 %

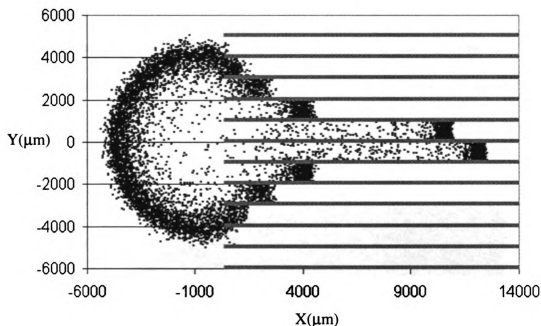


Figure 55. Simulation results of reflective 900-micron multislit pores.

The motility into the pores compared to bulk motility increased with time for pore sizes from 100 micron and larger for all the concentration ratios simulated with 0.02 mM, 0.1 mM, and 0.2 mM in the pores and a uniform concentration of 0.1 mM everywhere else. The simulation using 50-micron pore size and concentration in the pores a fifth (0.02 mM) of the bulk initially showed increased penetration ratio at 15 minutes, but at 30 minutes the penetration ratio decreased but was still larger than in the bulk. Since the boundaries are reflective the decrease is not more resistance with the pore wall. This

enhancement of the motility is only temporary and only lasts as long as the cells are in the pore. Once the cells reach the end of the pore the concentration increases and the biased migration decreases. In the absence of growth, there are relatively few cells at the end of the pore, which takes longer to produce sufficient local gradients and a lower concentration.

The 100-micron pore with equal Figure 56 had the largest penetration ratio of 3.29. For the reflective wall boundaries, the case with higher and lower concentration in the pores resulted in a lower penetration ratio. The simulation with 0.02 mM chemoattractant concentration in the pores is a lot lower then the bulk and results in a higher penetration ratio at 15 minutes and a lower penetration ratio at 30 minutes compared to the constant chemoattractant concentration.

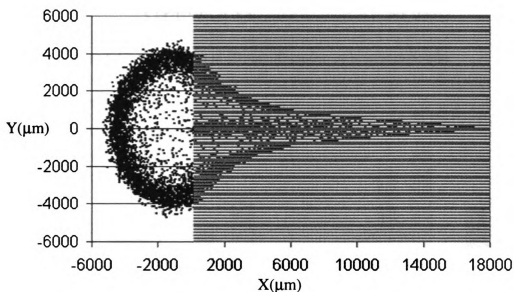


Figure 56. Simulation result of 100-micron pore width multislit with reflective walls.

Figure 57 and Figure 58 show the relation between the 2D and 3D motion on the simulations in the slit pores. The 3D simulations showed that the motility is lower then the 2D simulations for different pore sizes for both the reflective and non-reflective

migration through the pores with a difference of about 60%. The penetration ratio for the 2D and 3D simulations correlated pretty well for both the reflective and non-reflective simulations with a difference of around 6%.

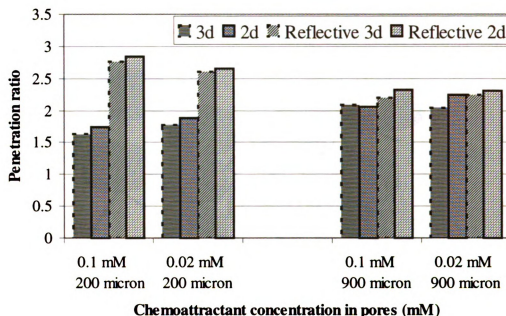


Figure 57. Comparison of 2D and 3D motion on penetration ratio.

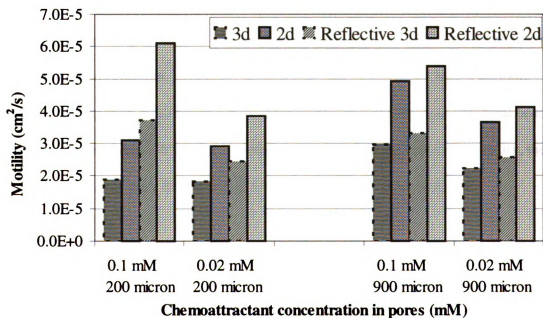


Figure 58. Comparison of 2D and 3D motion on motility coefficient.

6.3 HETEROGENEOUS

Three different effective shapes of particles were simulated. The first is circle disks (Figure 59) with 2D motion, which is completely packed and closes off migration at 22% porosity.

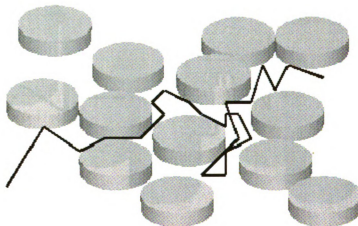


Figure 59. Cell trajectory through packed bed of 2D disks

The other two shapes use 3D motion and are a cylinder and a sphere. The highest porosity possible using spheres on our square lattice grid is 48%, which corresponds to a face centered configuration. In the 3D motion simulations the z-axis uses a periodic boundary so migration is through a bed of long (Figure 60) or strings of spherical beads (Figure 61) respectively. When the cylinders are completely packed 22% porosity, the cell can still move in the z direction but is hindered in the x and y direction.

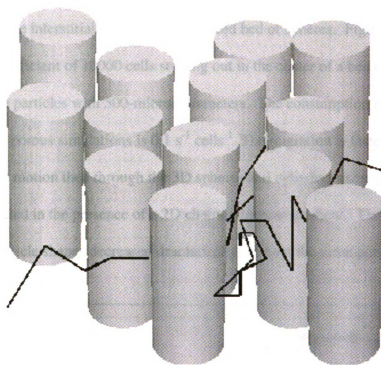


Figure 60. Cell trajectory through packed bed of cylinders

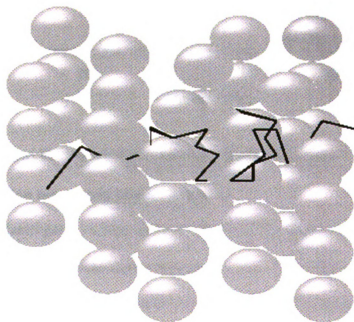


Figure 61. Cell trajectory through a packed bed of spheres.

Where as the spherical string of beads can be completely packed and the cells can move through the interstitial spaces as in a packed bed of spheres. Figure 62 compares the motility coefficient of 10000 cells starting out in the center of a bed of the three different shape particles with 500-micron diameters. The consumption rate used in all the heterogeneous porous simulations is $0.1 \text{ s}^{-1} \text{ cells}^{-1}$. The migration is faster through the 2D disks using 2D motion then through the 3D spheres and cylinders using 3D motion, which is expected in the presence of a 2D chemoattractant gradient. The motility for all three of the particle shapes decreased dramatically at about the same porosity of 50 %.

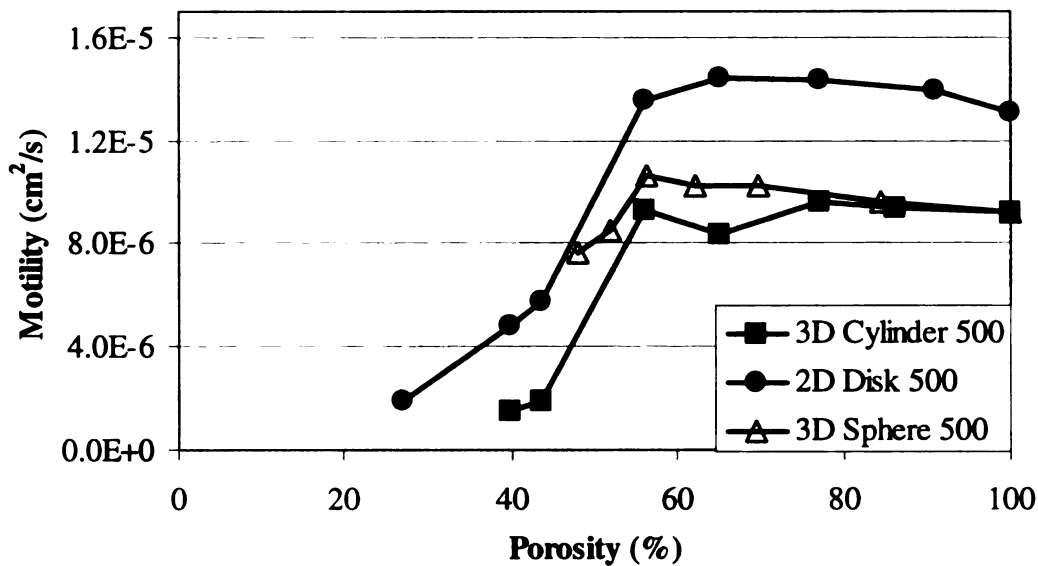


Figure 62. Motility of different shape 500-micron particles

6.3.1 Migration into a strip of porous media

Randomly placed particles were added to the simulation grid to simulate migration into porous media. Simulations using a strip of porous media on the right side of the simulation space were used to correlate with the results of the multislit simulations as well as previous experimental results. Figure 63 shows results of a simulation of

10000 cells starting at the center and moving into a strip of 400-micron particles with 3D motion. The black dots are the cells and the gray circles are the particles with a porosity of 57%.

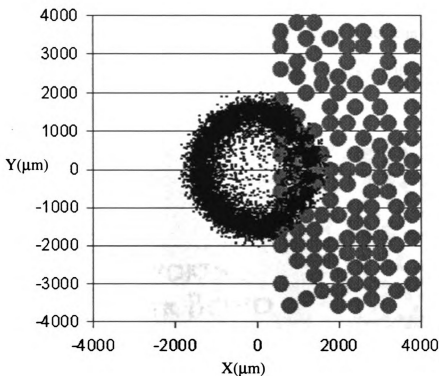


Figure 63. Simulation results showing cells migrating into porous region.

The porosity was calculated from the area not taken up by the particles. The average porosity of the Schoolcraft bioremediation site sand is 40.2 % (Criddle, 1997). Experimental tortuosities for particle sizes of 100 to 800 ranged from 4 to .5 (Barton, 1995). The tortuosity increases as the particle size decreases because of the smaller effective pore diameter and the curvature of the pore is larger and causes a collision with the pore wall before its basal run length. When cells react to a chemoattractant gradient, the run length is increased effectively increasing the tortuosity (Barton, 1995). We observed that the motility through the pores decreased with decreasing particle size.

Experimental results indicate that the particle size has an effect on the motility independent of the porosity (Duffy, 1995).

In Figure 64 the cells were started in a line along the edge of the porous region instead of a point to see how the initial location of the cells affected migration into porous regions. The cell movement and chemoattractant diffusion is periodic in these simulations. Two bands of high cell density form, one migrates into the porous region and the other migrates away from the porous region.

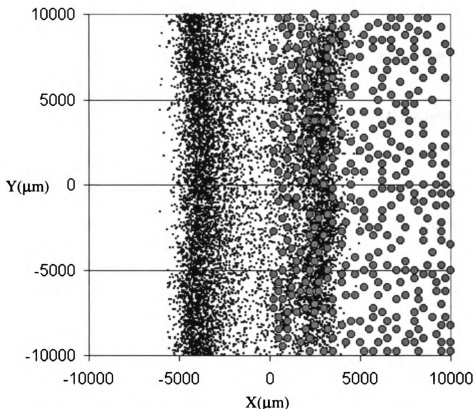


Figure 64. Simulation result of cells starting out in a line along the porous region.

Starting the cells out in a line along the edge of the particles versus starting them out at a point resulted in a lower motility and penetration ratio as seen from the cell density profile in Figure 65. The rest of the simulation results are done with all the cells starting out at a point.

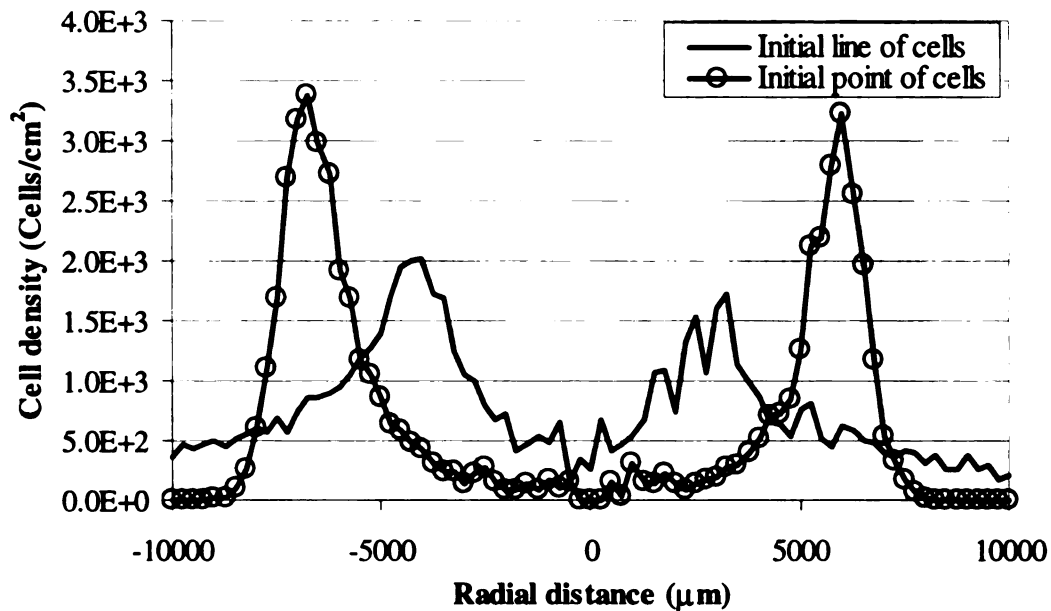


Figure 65. Cell density profile for initial cell location of point versus line.

Simulations were done using particle diameters ranging from 200 microns to 1600 microns. The motility coefficient for the cells moving into the pores and the cells moving away from the particles was calculated separately. Using a porosity of 65% at 2 hours, the motility coefficient changes very little over the range of particle sizes from 200 to 1600 microns as shown in Figure 66.

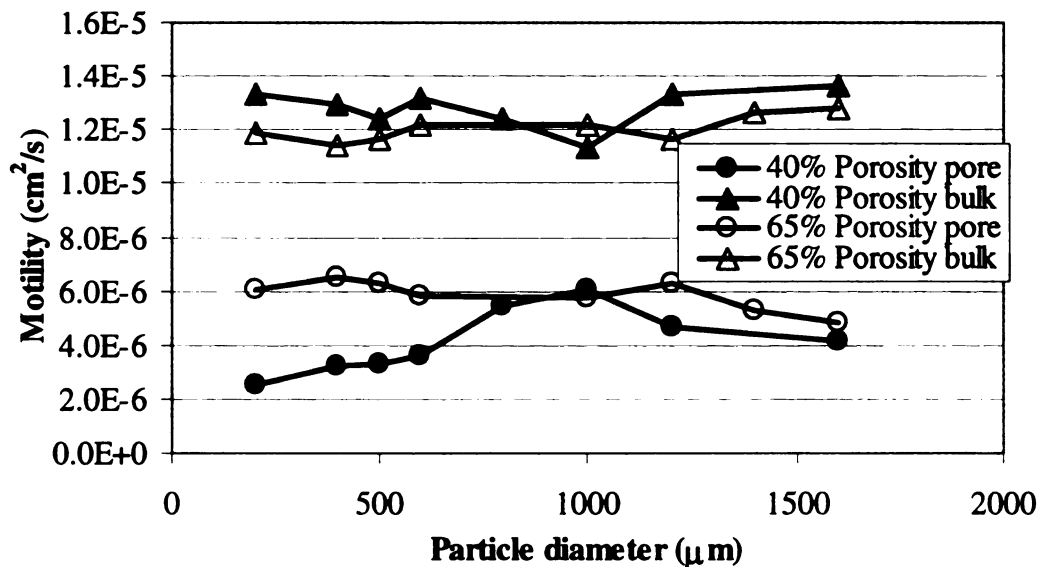


Figure 66. Motility results for different particle diameter in strip.

While at a porosity of 40 %, the motility coefficient for the cells moving into the particles decrease at smaller diameter particles and the motility coefficient of the cells away from the particles increases slightly since more cells move away from the particles due to obstruction in the opposite direction by the particles. In Figure 67 the penetration ratio seemed to peak before decreasing more at smaller diameter particles. At a porosity of 65% the peak in penetration ratio was around a particle size of 500 microns, while a porosity of 40 % peaked around 1000 microns.

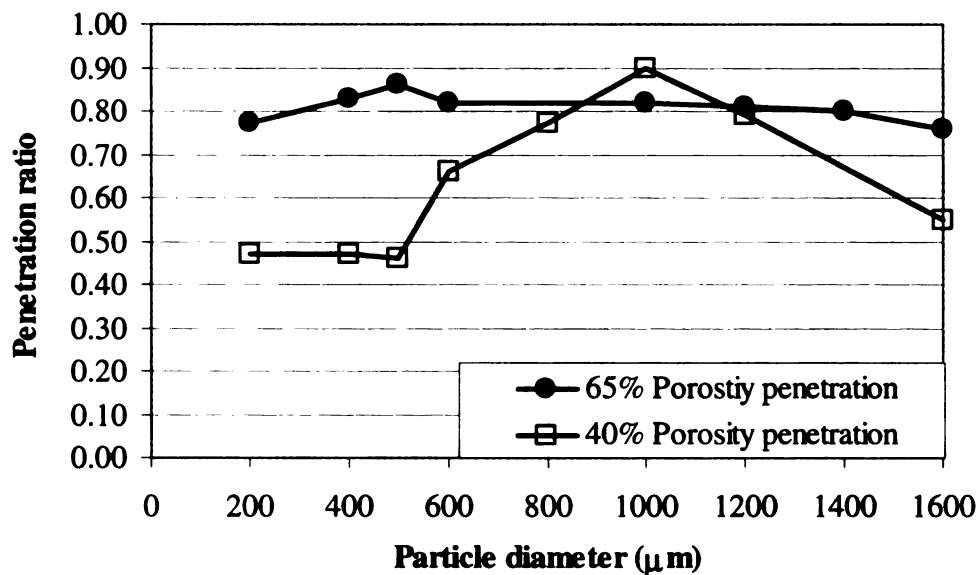


Figure 67. Penetration ratio result for different particle diameter in strip.

The penetration ratio and the motility coefficient of cells swimming into the particles both decreased at lower porosity, while the motility coefficient for the cells swimming away from the particles increased at lower porosity as seen in Figure 68.

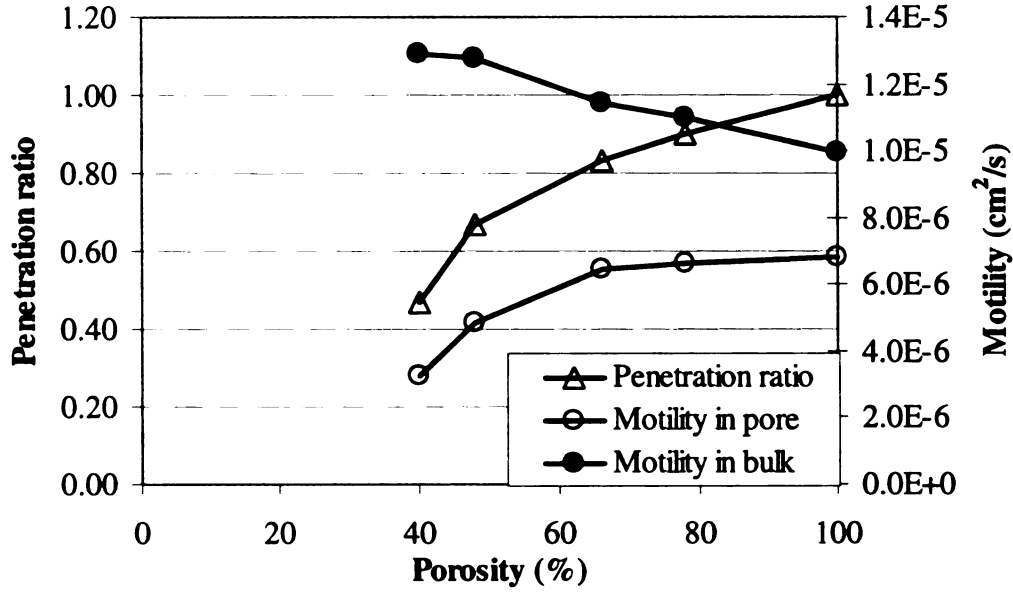


Figure 68. Motility coefficient and Penetration ratio of 2D simulations with a porous strip.

The effective diffusion coefficient in the porous material was incorporated into the simulation to account for chemoattractant diffusion in a porous bed.

$$D_{eff} = D \frac{\varepsilon}{\tau} \quad (22)$$

A lower diffusion coefficient causes increased motility in nonporous media, but decreased motility in porous media under the conditions simulated. For the simulations in a porous strip and the normal diffusion coefficient in the bulk, using a lower diffusion coefficient in the strip did not significantly change the motility from a uniform diffusion coefficient everywhere. The effect of including the effective diffusivity for diffusion in the porous region in the simulations using a tortuosity set at 3.25 resulted in a decrease in motility of 2.9% and a decrease in penetration ratio of 4.1%.

6.3.2 Migration in a bed of porous media

Simulations with the cells initiated in the center of the porous media were also done. Figure 69 shows results of a simulation of 10000 cells starting at the center in a bed of 400-micron particles after 2 hours. The black dots are the cells and the gray circles are the particles. In an ordered symmetric bed of particles the cellular dynamics simulations predict symmetric cell locations. The ordered array of 400-micron diameter 2D disks in Figure 69 has a porosity of 63%. When the particles are all evenly spaced a symmetric cell distribution is achieved.

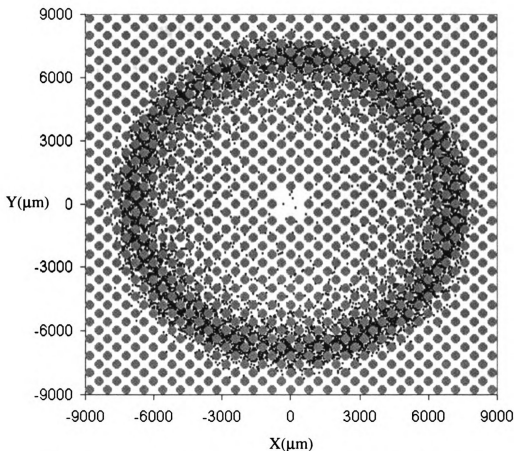


Figure 69. Migration in ordered 400-micron diameter 2D particles with 63% porosity.

When particles are placed together they effectively make a larger particle blocking the way of the cells and the random distributions of the particles placed together causes an irregular distribution of cells. Figure 70 has a porosity of 64.6%, which is roughly the same porosity as the ordered particles and has the same particle diameter of 400 microns but the cells are starting to get irregularly dispersed. A few clusters of cells are hindered by particles placed next to each other causing cells to find alternate paths to continue migrating from the center.

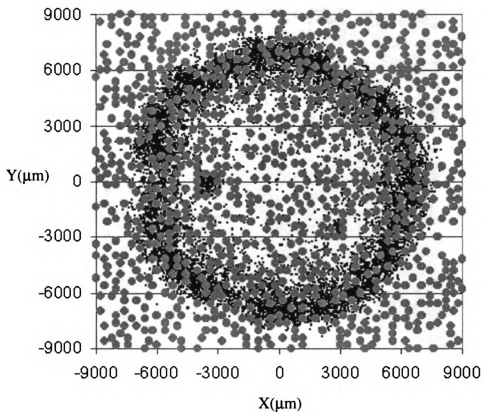


Figure 70. Migration in random 400-micron diameter 2D particles with 64.6% porosity.

As the porosity decreases the cell swimming paths will become less symmetrical in the simulations of 2D motion around 2D disks. The porosity of the simulation in Figure 72 was 38.5 % using 2D motion. The position of the cells is irregular making it difficult to predict where the cells will migrate to without performing a cellular dynamics simulation.

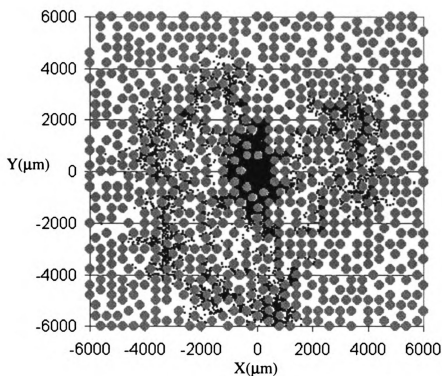


Figure 71. Migration in random 400-micron diameter 2D particles with 38.5% porosity.

Random packing does not always cause cells to be irregularly distributed. In the 3D simulation shown in Figure 72 a cluster is left in the center and a band of cells spread out radially around spheres with a porosity of 51 %. The ability to move in 3D around spherical particles prevents the cells from being trapped even when particles are touching each other.

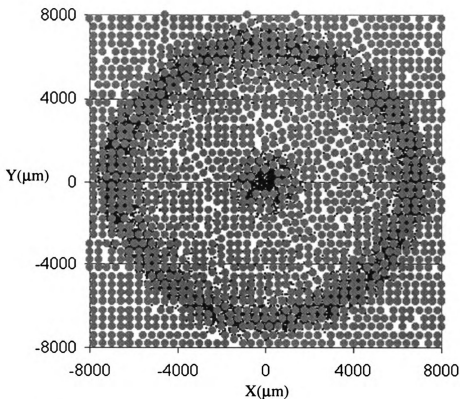


Figure 72. Migration in random 400-micron diameter 3D particles with 51% porosity.

At a porosity of 56.3 % the 2D simulations yielded a motility coefficient significantly larger than the 3D simulations of cylinders and spheres (Figure 73). The motility is higher for the 3D spheres than the 3D cylinders because there are more spaces for the cells to pass through without colliding with a solid particle. In 2D simulations with porosity corresponding to soil of 40%, the motility was significantly lower than the motility at higher porosity. We expect the motility to decrease at smaller particle sizes

because of the increase in tortuosity. It is difficult to determine if the motility goes through a peak around 500 microns because the motility calculated from simulations varies widely at different seed numbers for the same porosity. More simulations need to be done to calculate error bars to determine the statistically significant effect.

Chemotaxis does enhance migration in pores under certain conditions. Since the porosity and particle size of soil can't be effectively manipulated for bioremediation of soil, other parameters that optimize the migration of cells in a defined set of porous media will greatly improve the prediction of efficacy in bioremediation projects.

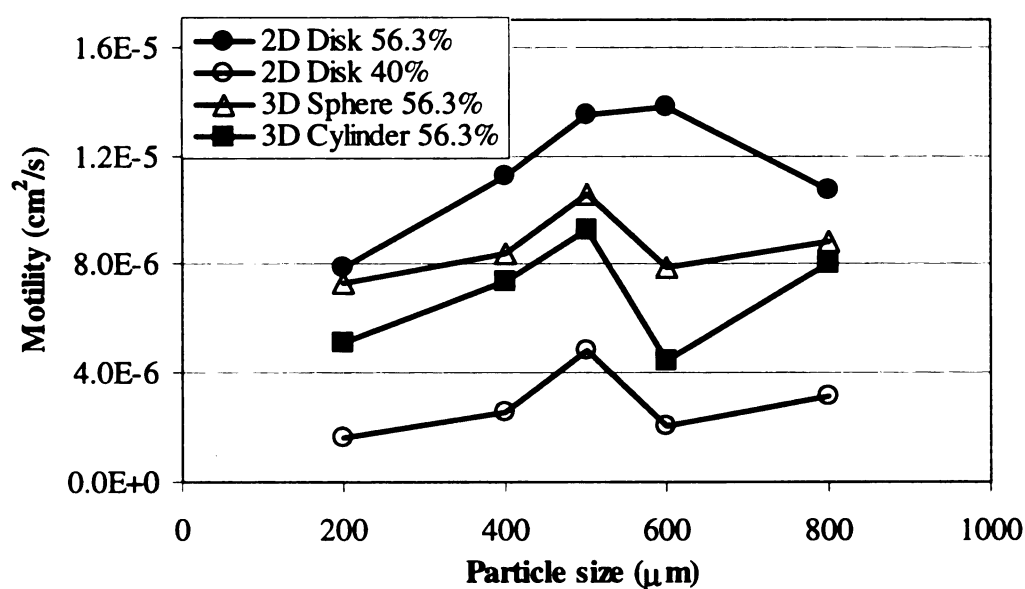


Figure 73. Effect of particle size on the motility coefficient in a bed of pores.

In Figure 74 the motility coefficient is shown for 400-micron diameter particles at 30 minutes, 2 hours, and 12 hours as porosity changes. At higher porosity above 80% different time lengths produced virtually the same motility coefficient. At lower porosity the motility through the pores increases over time, but not significantly after 2 hours. Larger particle size and longer simulation time seem to extend the peak in motility to

lower porosities as seen for Figure 75 with 500-micron spherical particles with 3D motion.

The increase in motility over time seems counter intuitive and there may be a few possible reasons for this effect. The first hypothesis involves the use of 3D motion in a 2D gradient. Some of the cells initial orientation is in the direction out of the plane of the 2D gradient and it takes them a while to orient themselves with the gradient. The other hypothesis involves a changing of the gradient and concentration that the cells sense. Over time the concentration gradient precedes the wave front of cells and the lower local concentration in the pores causes enhanced migration through the pores. To determine the reason more for the increase in motility simulations would have to be done in 2D as well as a comparison of the local concentration that the cell senses at different times.

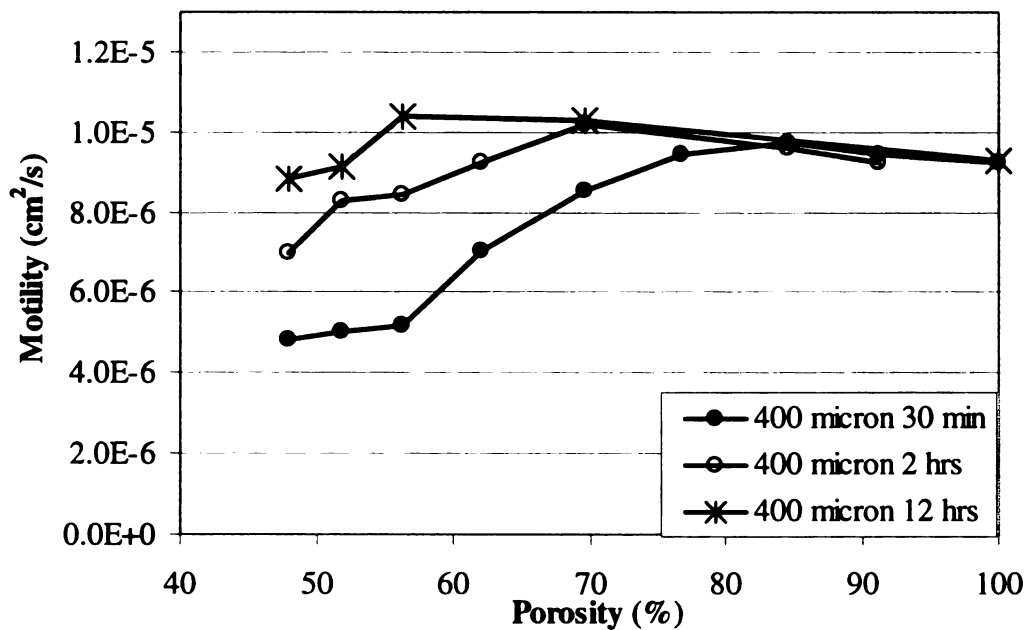


Figure 74. Effect of porosity on the motility coefficient in 400-micron particle bed.

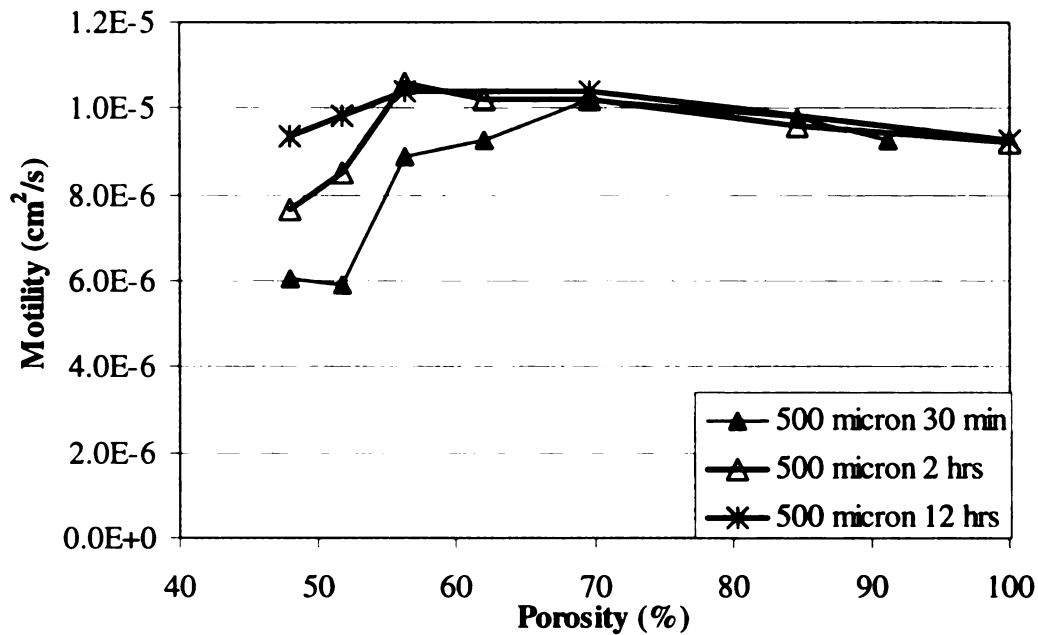


Figure 75. Effect of porosity on motility coefficient in 500-micron particle bed.

From the slit pore simulations we know that the amount of localized consumption determines if chemotactic migration will occur in a given pore width. It is likely that the changing effective pore widths and frequent change in direction caused the reduced motility. The direction of the chemoattractant gradient doesn't follow the pore direction and frequently biases the cell migration into the wall of the particles. A 3D chemoattractant gradient would follow the pore direction and not lead directly into a particle like a 2D gradient. In our simulations the cell stays at its previous location if it will collide with a particle wall. Enhanced migration will more likely be encountered if cells reflectively bounce off of a particle wall.

In the 3D simulations of migration through spherical particles the z direction is periodic which works well for characterizing where bacteria can travel based on particle collision, but can't differentiate consumption far along the z-axis with consumption at

low values of z . Cells all along the z -axis use the same nodal points to account for chemoattractant consumption.

The model of 3D migration with a 2D gradient is most valid when the chemoattractant profile is longer in 2 directions and shorter in the third. In the application of bioremediation a layer of chemoattractant or pollutant that is a chemoattractant that forms a plume that is approximated as a thin plane. For example the plume may spread quicker in the vertical direction with the entrance into the ground along a line. The plume may spread out horizontally when it reaches an impenetrable clay layer creating a thin plane of chemoattractant. A layer of decomposed organic material may consist of higher concentration of chemoattractant such as nitrate. In the 3D simulations, the initial location of cells is better characterized as a line then a point source because of the periodic boundary conditions in the z direction, which is also more likely to occur when physically inoculating cells.

The motility coefficient for 3D motion and a 2D gradient does not correlate directly to the motility coefficient of 3D motion in a 3D gradient, while the cell density profiles for 2D and 3D match for migration in non-porous media in Figure 76. The motility coefficient for the 30-minute 3D simulation with aspartate consumption is $9.24\text{E-}6 \text{ cm}^2/\text{s}$ and $1.3\text{E-}5 \text{ cm}^2/\text{s}$ for the 2D simulation. Migration will be biased in 2 directions and the third will have random motility leading to an anisotropic distribution of cells that is not symmetric. In the case of no pores the correlation factor to translate the 3D nonsymmetrical motility to a symmetric 2D motility is the square root of 2. Simulations using 2D motion with a 2D gradient should correlate directly with 3D motion in a 3D gradient because it can be considered symmetric.

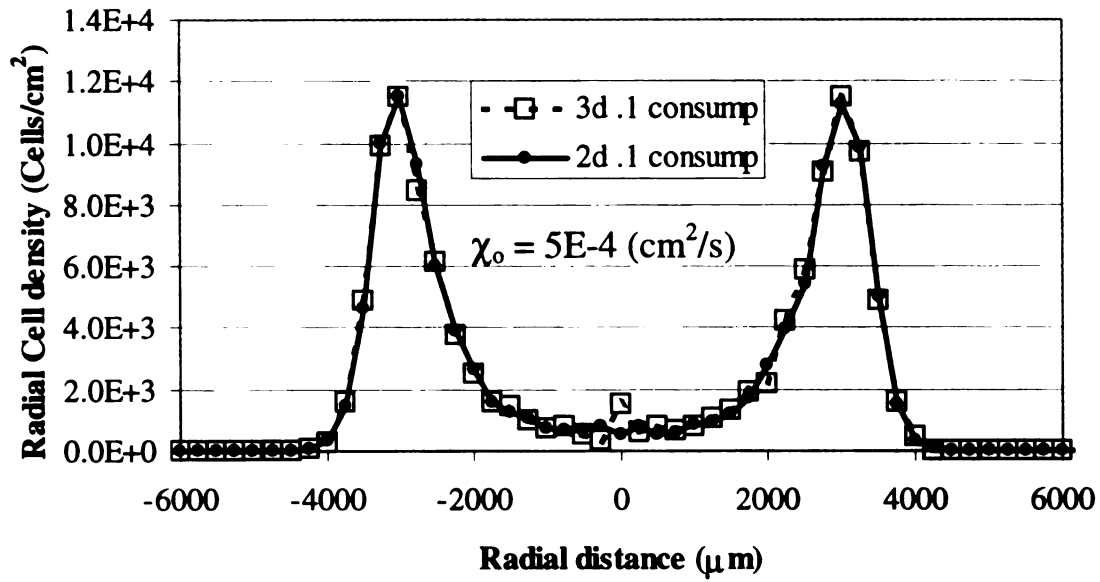


Figure 76. Cell density profiles for 2D and 3D motion with non-porous consumption.

The cell density in the porous bed had roughly the same behavior for the 2D and 3D simulations in Figure 77 at the same porosity but does not correlate as well as the nonporous density profiles. Unlike the non-porous simulations, the motility coefficients are significantly closer together with a less than 3 % difference. The 2D motility coefficient is $8.02\text{E-}6 \text{ cm}^2/\text{s}$ and the 3D motility coefficient is $8.26\text{E-}6 \text{ cm}^2/\text{s}$. The 3D motility coefficient is larger than the 2D motility coefficient for simulations where particles are closely packed, which is opposite of the non-porous simulations.

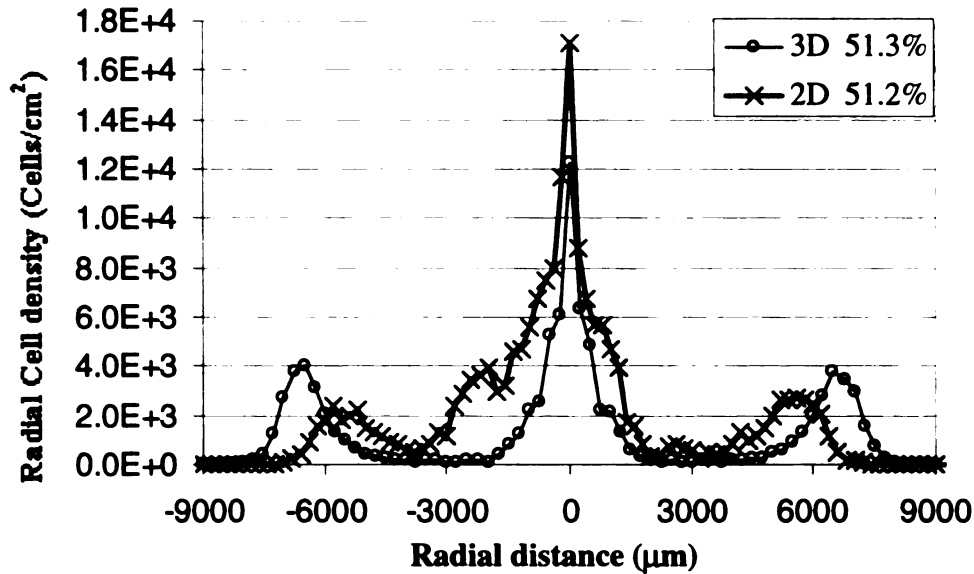


Figure 77. Cell density profiles for 2D and 3D motion in pores with consumption.

Changing the seed number of the randomly placed particles resulted in approximately the same porosity for a given set of size probabilities. The motility varied over a 100% difference using different seed numbers even though the porosity was the same. Changes in porosity with the same seed number caused only a small change in motility. This indicates that how the particles are packed has a bigger effect than how close they are together.

7. VISUALIZATION TOOLS

7.1 EXCEL

Various tools were used to visualize the results of the simulations. Excel was the most frequently used, which easily gives snapshots of cell positions at different time steps. It is also good at producing line plots to picture trends as parameters are varied. The gaussian turn angle distributions were calculated and converted to an accumulative

probability curve using Excel. A hundred probabilities at equal angle measurements were used in the turn angle probability input file.

7.2 ARRAY VISUALIZER

The Compaq Array Visualizer that comes with the Compaq Visual FORTRAN Professional addition has been incorporated into the simulation code to display the concentration field at predefined time steps. Changes in the concentration field color indicate the degree of consumption of the chemoattractant corresponding to the location of the cell positions. The Array Visualizer was used to make snap shots of concentration variations in the slit pore simulations.

7.3 MATLAB

The visualization of the simulations for the migration through heterogeneous porous media was accomplished by using MatLab. The cell positions, concentration values at the nodal points and location of the randomly placed particles were written to a text file at several time steps. These files were read into MatLab to produce a slide show of the changes concentration and cell position around particles. Adding the mpgwrite capability to MatLab yielded a movie file in mpg format that can be played on a wide variety of operating systems. In order to change the speed of the movie, it was converted to an avi format with a timing of 2 frames per second versus 30 frames per second. The code used to produce the slide show movies are in the appendix. Figure 78 is a frame of the Movie from MatLab that shows the migration of cells in porous media as well as the change in concentration due to consumption.

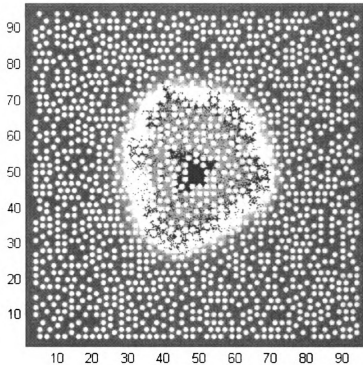


Figure 78. MatLab output of tracking cell location and concentration in porous media.

8. CONCLUSIONS & FUTURE WORKS

8.1 CONCLUSIONS

The trend of the motility or diffusive dispersion of the cells is affected by the cell characteristic parameters. The motility increases with velocity, and decreases with tumbling probability and mean turn angle. The variance of the turn angle had no effect on the motility of the cells. The chemotactic sensitivity's effect on the motility resulted in a sigmoid curve that increased exponentially and then leveled off.

The only parameter that gave a relevant change in the chemotactic response (mean displacement) was the chemotactic sensitivity χ_0 . The drift velocity or the mean distance moved from the origin did not change with changes in the mean turn angle, turn

angle variance, or tumbling probability. The diffusiveness of the cell distribution or motility coefficient increased with lower tumbling probabilities and lower average turn angles, while changes in the variance of the turn angle distribution didn't effect either the average displacement or the diffusiveness of the cell distribution. Simulation results indicate that a distribution of turn angles is not needed only the average turn angle making it easier to model different bacteria without knowledge of their turn angle distributions. The mean distance increased with increased velocity. The mean distance up gradient increases exponentially as the chemotactic sensitivity increases, however it levels off when the cells stop tumbling at very high chemotactic sensitivity.

Chemotaxis with coupled consumption and diffusion results in migration patterns different from a fixed linear gradient of chemoattractant. It is clear that consumption can increase motility by increasing the magnitude of the attractant gradient. It should also be noted that motility would decrease if the concentration of attractants is limited and a gradient is not sustained. A colony of cells can migrate faster from the point of inoculation by the cooperative gradient production of multiple cells compared to a single cell or an even distribution of cells.

Consumption of a chemoattractant causes enhanced gradient production, which explains both the high cell density bands and accelerated migration in pores. A moderate consumption coefficient leads to increased motility where cells are not left behind. Experiments that show multiple bands are responding to multiple chemoattractants. The cells move chemotactically up the step gradient as they do with the gradient produced from consumption. The necessity of a cell to be in a low concentration to bias its migration in a new direction makes ecological sense, since the cell would want to stay

in areas of high nutrients or resources. In the case with consumption, the chemotactic response is determined by how fast the produced decaying step function gets shifted due to consumption.

Chemotaxis does enhance migration in pores under certain conditions. Since the porosity and particle size of soil can't be effectively manipulated for bioremediation of soil, other parameters that optimize the migration of cells in a defined set of porous media will greatly improve the prediction of efficacy in bioremediation projects. The cellular dynamics method can predict irregular cell distribution where previous work with cell balance equations can only give an average cell distribution.

Enhanced migration was shown using multiple parallel slit pores. A lower concentration of chemoattractant in pores enhances penetration into the pores more than a uniform concentration when cells do not reflect off the walls.

The configuration, porosity and particle size affected the migration in heterogeneous pores. The randomness and placement of the particles has a dominant effect on where the cells will migrate and the resulting cell density profile. There is an optimum particle size that has the optimal pore width that causes accelerated migration for the given population consumption rate. The porosity had a shallow peak then a sharp drop off for the conditions simulated. Larger particle size and longer simulation time seem to extend the peak to lower porosities. When the chemoattractant is consumable, the motility through the pores increases over time, but not significantly after 2 hours

The cell density profiles for the 2D and 3D cases correlated well when the same porosity was used and not the same number of particles. For the 2D and 3D simulations, the motility coefficients appear to have a square root 2 correlation in the absence of pores

with the 2D motility larger, but in high density of center faced particles, the 3D motility dominates.

8.2 SUGGESTED TOPICS FOR FUTURE RESEARCH

8.2.1 3-D gradient detection and 3-D heterogeneous porous

Future work includes detection and consumption of a 3-D attractant gradient. Simulations of migration through heterogeneous porous networks in 3-D can have gradients in 3-D and follow the pore direction more exactly. Methods to pack the particles closer together need to be developed to approximate the porosity usually found in soil.

8.2.2 Multiple attractants/repellent consumption and or excretion

More interesting and innovative results could potentially developed by including other interactions into the model. Modeling response to multiple chemoattractant gradients could give a better fit of models to experimental observations, since the receptors do not work independently. Simulations where the bacteria excrete and respond to metabolites that are either an attractant or repellant will help determine under what conditions patterns such as spots or fractal occur.

8.2.3 Adsorption and energy tabulation to determine growth / death

The adsorption of bacteria during collision with pore walls can be simulated using a sticking coefficient. Methods for including cell growth and death would need to be developed by allocating extra memory for future cells and an algorithm to place new cells into the array variable to take place of the dead ones. An energy term can be developed to simulate the energy gained from nutrients and expended from swimming and can be used to determine if the cell will divide or die.

APPENDIX A

Fortran cellular dynamics code:

PROGRAM CELLSVID

*This program simulates three dimensional, cellular chemotaxis.

*

* Simulation Code Written by JASON MONDRO with some work by

*Justin Jorgensen, John A. Nievierowski, and Professor C. M. Lastoskie

*

*Department of Chemical Engineering

*Michigan State University

*

*DEFINITIONS

*Parameters:

* NPMAX- Maximum number of cells allowed (for array sizing)

* NXMAX,NYMAX- Maximum number of grid spaces along length of axis

*Variables:

* NP- Number of cells

* ISEED- Large negative value to start random number generator

* DT- Time step increment value

* V- Cell swimming velocity

* PT- Tumbling Frequency

* FILENM- The filename for the sequential output files

* DG- Grid spacing

* YLENGTH- Length of the positive y axis

* XLENGTH- Length of the positive x axis

* CONC- Concentration of chemoattractant or high level of linear grad

* KD- Dissociation constant for chemoattractant

* CHI- Chemotactic Sensitivity

* D- Diffusion Coefficient for chemoattractant

* CONSUMP- Consumption rate coefficient

*

*Arrays:

* BOXCORNER- Array holding the solid box postions

* PORECODE- Array used to modify diffusion eqn at particle interface

* BIN- Array holding the turn angle distribution function

* X,Y,Z- Current x,y,z positions of cells

* XO,YO,ZO- Initial x,y,z positions of cells

* XVEL,YVEL,ZVEL- x,y,z velocity components

* XG,YG- x,y position of the grid

* CG- concentration at grid position

*Functions:

* RAN1- Random number generator

* TUNANG- Turn angle distribution function (phi)

* ERF- Calculates error function for step gradient

*Subroutines:

* INITVEL- Initiates velocity vectors

* VELN- Calculates the velocity components

* TUMBLE- Get turn angle and resultant velocity compents after tumble

* NOTUMBLE- Calculates new cellular position if no tumble occurs

* ROOTMEAN- Calculates the motility and outputs both the motility and
position at given time increments

* READINFO- Reads the inital conditions and input files

* CONCDIFF- Calculates the diffusion using center differencing

* GRID- Sets up points on a grid

```

* GRADIENT- Calculates Concentration gradients and new turning prob
*
*Files:
* cells.prn- initial positions of cells
* points.prn- values for the turn angle distribution function
* motility.prn- output of motility calculations throughout simulation
* names.prn- contains the sequential filenames
* chXXXX.dat- sequential files containing cell position output
*
* MAIN PROGRAM MODULE -----
*
* The main program module is responsible for execution of the primary do
* loop as well as the opening and closing of all I/O files. The primary
* loop (50) executes over the number of timesteps while the secondary
* loop executes over the number of cells. A random number is compared
* to the turn probability (PT) to determine if the cell will TUMBLE or
* NOTUMBLE. The ROOTMEAN routine will be called incrementally (INCR)
* to calculate and output data.
*
* Declaration of Variables

    use AVDef
    use DFLib
    use AVViewer
    use change
    PARAMETER (NXMAX=1000, NYMAX=1000)
    PARAMETER (NPMAX=20000)
    PARAMETER (NKMAX=1)
    PARAMETER (NDENMAX=1000)
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    INTEGER I,NFILES,DIFFTYPE,NXBND,NYBND
    INTEGER K,N,NMOTDIM,NUMCAP
    INTEGER GTYPE,NGRAPH1,XSPC,YSPC,XCAP,YCAP
    INTEGER A,B,STAT
    INTEGER SEED1,XLEFTBOUND,XRIGHTBOUND
    INTEGER TUMBDEL,nprint,NDIRECT
    INTEGER NTUMB(npmax,NKMAX)
    integer porecode(-NXMAX:NXMAX,-NYMAX:NYMAX)
    integer boxcorner(-NXMAX:NXMAX,-NYMAX:NYMAX)
    REAL BIN(0:100)
    DOUBLE PRECISION X(NPMAX),Y(NPMAX),Z(NPMAX)
    DOUBLE PRECISION XO(NPMAX),YO(NPMAX),ZO(NPMAX)
    DOUBLE PRECISION XVEL(NPMAX),YVEL(NPMAX),ZVEL(NPMAX)
    DOUBLE PRECISION DENCELL(-NDENMAX:NDENMAX)
    DOUBLE PRECISION DENCELL1(-NDENMAX:NDENMAX)
    DOUBLE PRECISION RHO(-NDENMAX:NDENMAX)
*   DOUBLE PRECISION PRN(0:NXMAX)
*   DOUBLE PRECISION NEGORN(0:NXMAX)
*   !DEC$ATTRIBUTES array_visualizer :: CG
    DOUBLE PRECISION KD,CONSUMPCOF,DOTP,PTN,CHI,DIFF
    DOUBLE PRECISION HALFCDIM,CDIM,ERRTIME,TIME
    DOUBLE PRECISION DCX,DCY,XPOINT,YPOINT,ZLENGTH,fraction2
    DOUBLE PRECISION XLENGTH,YLENGTH,SLIT,DG,DGSQ,CAPLENGTH,YTOTAL,COLCNT
    DOUBLE PRECISION PT,CONSUMP,ONE,ZERO, adjust,FRACTION, DIFFPORE
    CHARACTER FILENM*10
    COMMON/CONST/PI,PHI,THETA,NP,NTIME,V,DT,NMOTDIM,K

```

```

COMMON/CONST2/XLENGTH,YLENGTH,SLIT,DG,DGSQ,COLCNT
COMMON/INTEG/GTYPE,NGRAPH1,XSPC,YSPC,XCAP,YCAP,N,NUMCAP
COMMON/CGRAD/KD,CONSUMPCOF,DOTP,PTN,CHI,ZLENGTH
COMMON/DENSE/DENCELL,DENCELL1,RHO,NPRINT,NDIRECT
COMMON/GRAPH/HALFCDIM,CDIM,ERRTIME,TIME
COMMON/NUMB/ONE,ZERO
COMMON/VELS/VX,VY,VZ
COMMON/ARRS/BIN
COMMON/PORES/PORECODE,BOXCORNER
COMMON/POROUS/FRACTION,XLEFTBOUND,XRIGHTBOUND,fraction2
COMMON/INVARENT/TUMBDEL,NTUMB
* COMMON/MAPPING/PRN,NEGORN
  common/info/avgta,tunang1,PT,SEED1
COMMON/temp1/xtemp,ytemp,ZTEMP,TEMP2,TEMP3

* Definition of Pi
  PI=4.0*ATAN(1.0)

* Open Excel DATA FILES. NOTICE--- "STATUS" in UNIT 3,
*   OPEN statement must be changed to "NEW" when this program
*   is initialized.
  OPEN (UNIT=1, FILE='cellsvidpore.prn', STATUS='OLD')
* OPEN (UNIT=2, FILE='points.prn', STATUS='OLD')
  OPEN (UNIT=2, FILE='EcoliNR50B.PRN', STATUS='OLD')
* OPEN (UNIT=2, FILE='EcoliAW405B.PRN', STATUS='OLD')
  OPEN (UNIT=3, FILE='motility.prn')
  OPEN (UNIT=4, FILE='names.prn')

* READINFO -----*
*
* Read value for random number generator, number of cells,
* time step increment value, the number of time steps,
* REALless velocity and the tumbling frequency.
* Read initial cell positions into arrays.
* Read BIN values for turn angle distribution
* into array "BIN".
*
  READ (1,*) ISEED,NP,DT,NTIME,INCR,VD,PT
  READ (1,*) CELLDIA,XLENGTH,YLENGTH,DG,CONC,KD,CHIO,D,CONSUMP
  READ (1,*) SLIT,CAPLENGTH,ZLENGTH

* Linear Gradient
* Select Grid Type
* Gtype=1 for linear
* Gtype=2 for step
* Gtype=3 for constant
* Gtype=4 for capillary

  GTYPE=3

* TUMDEL=1 FOR INCLUDING DELAY AT A TUMBLE TUMDEL=0 FOR NO DELAY
* TUMDEL=2 FOR DELAY EQUAL TO TIME STEP

  TUMBDEL=2

* NGRAPH1=1 FOR PLOTING

```

NGRAPH1=0

NPRINT=1

* NDIRECT=0 FOR VERTICAL DENSITY NDIRECT=1 FOR HORIZONTAL DENSITY
NDIRECT=1

* nsim=1 for including initial time step in data files
nsim=0

* CALCULATE WITH DIFFUSION

* 1 FOR YES 0 FOR NO

DIFFTYPE=1

* SET OUTPUT FORMAT N=1 FOR A NEW FILE AT EACH TIME STEP

* N=2 AT EA PARTICLE N=3 FOR ALL ON ONE FILE

N=1

* Pick dimension of motion

* NMOTDIM=2 for 2-D NMOTDIM=3 for 3-D

NMOTDIM=2

* Pick boundary conditions for x direction

* NXBND=0 for cell stops if hits wall

* NXBND=1 for period boundary NXBND=2 for reflective boundary

NXBND=0

* Pick boundary conditions for Y direction

* NYDNS=0 for cell stops if hits wall

* NYBND=1 for period boundary

NYBND=0

* Pick initial cell positions

* cellpos=1 for input file step cellpos=2 for all at origin

* cellpos=3 for one at each grid point

CELLPOS=2

IF (CELLPOS.EQ.1) THEN

DO I=1,NP

READ (1,*) X(I),Y(I),Z(I)

XO(I)=X(I)

YO(I)=Y(I)

ZO(I)=Z(I)

end do

END IF

IF (CELLPOS.EQ.2) THEN

DO I=1,NP

X(I)=0

Y(I)=0

Z(I)=0

XO(I)=X(I)

YO(I)=Y(I)

ZO(I)=Z(I)

```

        end do
    END IF

    IF (CELLPOS.EQ.3) THEN
        I=0
        XSPC=XLENGTH/30
        YSPC=YLENGTH/30
        DO M=1-YSPC,YSPC-1
            DO L=1-XSPC,XSPC-1
                I=I+1
                X(I)=30*L
                Y(I)=30*M
                XO(I)=X(I)
                YO(I)=Y(I)
            END DO
        END DO
        NP=(2*(XSPC-1))*(2*(YSPC))+1
    ENDIF

    DO 15 I=0,100
        READ (2,*) BIN(I)
15  CONTINUE

    DIFF=D*DT*100000000/(CELLDIA**2) !converts to dimensionless micron
    CHI=CHIO*DT*100000000/(CELLDIA**2) !converts to dimensionless micron
    V=VD*DT !converts to dimensionless

    IF (NMOTDIM.EQ.2) THEN
        V=V*(2**.5)/(3**.5)
        CHI=CHI/1.5 ! converts 3-d chi to 2-d chi
    *   V=V/(2**.5)
    *   CHI=CHI/(2) ! converts 3-d chi to 2-d chi
    ** velocity independent chi
    *   chi=750*(v**2)/dt

    END IF
    SEED1=ISEED
    CDIM=CONC/KD
    XSPC=XLENGTH/DG
    YSPC=YLENGTH/DG
    YCAP=CAPLENGTH/DG+YSPC
    XCAP=SLIT/DG
    YTOTAL=CAPLENGTH+YLENGTH
    DGSQ=DG*DG
    TIME=0.0000000001
    ERRTIME=(4*DIFF*TIME)**.5
    HALFCDIM=CDIM/2
    CONSUMPCOF=CONSUMP*DT *((20/DG)**2)
    DIFFPORE=DIFF !*(1-(FRACTION/100))/3.25
    nxspc=-xspc
    nyspc=-yspc
    ONE=1
    ZERO=0

    call AllocateArray(XSPC,YSPC,YCAP,status)
    *****

```

```

      CALL BOXSUB(xspc,yspc,DG)
*****
      CALL INITVEL(XVEL,YVEL,ZVEL,ISEED)

      IF (N.eq.1) THEN
        NFILES=NTIME
      ENDIF
      IF (N.eq.2) THEN
        NFILES=NP
      ENDIF
      IF (N.eq.3) THEN
        NFILES=INCR
      ENDIF

      DO 100 I=5,(NFILES/INCR+4)+6+3*(NFILES/INCR)
        READ(4,*) FILENM
        FILENM(7:10)='.dat'
        OPEN (UNIT=I, FILE=FILENM)
100  CONTINUE
        OPEN (UNIT=9998, FILE='allconc.prn')
        OPEN (UNIT=9999, FILE='allcells.prn')

      CALL GRID(CG,DG)
77  FORMAT (T6,I5)
120  Format (20(1x,f8.4))
121  Format (600(1x,f6.2))
      M1=6+NFILES/INCR
      M2=M1+NFILES/INCR+1
      M3=M1-1

      DO 50 K=1,NTIME
        print *, k
        TIME=REAL(K)*DT

      IF (GTYPE.EQ.2) THEN
        ERRTIME=(4*DIFF*k)**.5
        DO B=-YSPC,YSPC
          ARG=DG*B/(ERRTIME)
          CSTEP=(HALFCDIM)*(1+ERF(ARG))
          DO A=-XSPC,XSPC
            CG(A,B)=CSTEP
          END DO
        END DO
      END IF

      DO I=1,NP
        CALL GRADIENT(CG,I,XVEL,YVEL,ZVEL,X,Y,Z,PT,INCR,NXBND,NYBND)
        ranseed=ranl(iseed)
        IF (ranseed.LT.(PTN*DT)) THEN
          CALL TUMBLE(I,XVEL,YVEL,ZVEL,ISEED)
          IF (TUMBDL.EQ.0) THEN
            CALL NOTUMBLE(I,XVEL,YVEL,ZVEL,X,Y,Z,NXBND,NYBND,YTOTAL)
          ENDIF
        ELSE
          CALL NOTUMBLE(I,XVEL,YVEL,ZVEL,X,Y,Z,NXBND,NYBND,YTOTAL)
        END IF
      END DO

```

```

END DO

IF(DIFFTYPE.EQ.1) THEN
  CALL CONCDIF(CG,DIFF,DIFFTYPE,NXBND,NYBND,DIFFPORE)
END IF

if (k.eq.1 .and. nsim.eq.1)then
  DO I6=YSPC,-YSPC, -1
    write (UNIT=9998,FMT=121) (CG(i,I6),i=-XSPC,XSPC)
  END DO
  DO I6=YSPC,-YSPC, -1
    write (UNIT=9998,FMT=121) (CG(i,I6),i=-XSPC,XSPC)
  END DO
endif

  IF (MOD(K,INCR).EQ.0) THEN

CALL ROOTMEAN(CG,X,Y,Z,XO,YO,ZO,INCR,RAD,ISEED,NFILES)

  IF (NGRAPH1.EQ.1) THEN

IF (K.EQ.NTIME) THEN

  KEY=GETCHARQQ()
endif

  ENDIF

  END IF
50  CONTINUE
  ! Uncomment the following line to have the ArrayViewer
  ! closed automatically.
  ! call faglClose(CG, status)

*   KEY=GETCHARQQ()
    call faglClose(CG, status)
    call DeAllocateArray(status)

    print *, "Done!"

    CLOSE (UNIT=1)
    CLOSE (UNIT=2)
    CLOSE (UNIT=3)
    CLOSE (UNIT=4)
    CLOSE (UNIT=9998)
    CLOSE (UNIT=9999)

    DO 110 I=5,(NFILES/INCR+4)+3+2*(NFILES/INCR)
      CLOSE (UNIT=I)
110  CONTINUE

  END
*-----*
```



```

Module CHANGE
use AVDef
public :: AllocateArray, GetShareName, Recalc, DeAllocateArray
PUBLIC :: GRID, GRADIENT
    DOUBLE PRECISION, allocatable :: CG(:, :)
!DEC$ATTRIBUTES array_visualizer :: CG
INTEGER STATUS
    integer :: lbnd(2) = 0
    integer(4) :: m_nrows=0, m_ncols=0
    character(40) :: title
    character(1) :: key

CONTAINS

Subroutine AllocateArray (XSPC,YSPC,YCAP, status)
    INTEGER XSPC,YSPC,YCAP
    integer(4), intent(OUT) :: status
    allocate(CG(-XSPC:XSPC,-YSPC:YCAP))
    call faglStartWatch(CG, status)
end Subroutine AllocateArray

Function GetShareName()
    character *(35) GetShareName
    character *(35) :: string
    integer(4) :: status
    if (Allocated(CG)) then
        call faglGetShareName(CG, string, status)
    endif
    GetShareName = string
end Function GetShareName

Subroutine DeAllocateArray(status)
    integer(4), intent(OUT) :: status
    if (.not. Allocated(CG)) then
        status = -1
        return
    endif
    call faglEndWatch(CG, status)
    deallocate(CG)
end Subroutine DeAllocateArray

SUBROUTINE GRADIENT(CG,I,XVEL,YVEL,ZVEL,X,Y,Z,PT,INCR,NXBND,NYBND)

PARAMETER (NXMAX=1000, NYMAX=1000)
PARAMETER (NPMAX=20000)
PARAMETER (NKMAX=1)
PARAMETER (NPREVMAX=2)
PARAMETER (NTMAX=100)
PARAMETER (NDENMAX=1000)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INTEGER K,N,NMOTDIM,NUMCAP
INTEGER GTYPE,NGRAPH1,XSPC,YSPC,XCAP,YCAP
INTEGER I,NFILES,DIFFTYPE,NXBND,NYBND,I4
    integer xleftbound, xrightbound
    INTEGER IBIN,BI, J,nspacial,NDET,NPRINT,NDIRECT

```

```

INTEGER XL,YB,XR,YT
  INTEGER NTUMB(npmax,NKMAX)
integer porecode(-NXMAX:NXMAX,-NYMAX:NYMAX)
integer boxcorner(-NXMAX:NXMAX,-NYMAX:NYMAX)
DOUBLE PRECISION X(NPMAX),Y(NPMAX),Z(NPMAX)
DOUBLE PRECISION XVEL(NPMAX),YVEL(NPMAX),ZVEL(NPMAX)
DOUBLE PRECISION XPREV(NPREVMAX,0:NTMAX)
DOUBLE PRECISION YPREV(NPREVMAX,0:NTMAX)
DOUBLE PRECISION CPREV(NPREVMAX,0:NTMAX)
DOUBLE PRECISION DENCELL(-NDENMAX:NDENMAX)
DOUBLE PRECISION DENCELL1(-NDENMAX:NDENMAX)
DOUBLE PRECISION RHO(-NDENMAX:NDENMAX)
* DOUBLE PRECISION PRN(0:NXMAX)
* DOUBLE PRECISION NEGORN(0:NXMAX)
* DOUBLE PRECISION IBIN
DOUBLE PRECISION, dimension(-XSPC:XSPC, -YSPC:YCAP), intent(inout)
& :: CG
DOUBLE PRECISION XLENGTH,YLENGTH,SLIT,DG,DGSQ,CAPLENGTH,YTOTAL,COLCNT
DOUBLE PRECISION KD,CONSUMPCOF,DOTP,PTN,CHI,DIFF,ZLENGTH
DOUBLE PRECISION HALFCDIM,CDIM,ERRTIME,TIME
DOUBLE PRECISION DCX,DCY,XPOINT,YPOINT,DCDY,DCDX,EPS,EPS2
DOUBLE PRECISION PT,CONSUMP,ONE,ZERO,ARG,V
DOUBLE PRECISION XGBL,XGTL,XGBR,XGTR,XDIF,XCENT
DOUBLE PRECISION YGBL,YGTL,YGBR,YGTR,YDIF,YCENT
DOUBLE PRECISION DCLEFT,DCRIGHT,DCTOP,DCBOTTOM
DOUBLE PRECISION TERM1,TERM2,TERM3,TERM4,CCELL
DOUBLE PRECISION DNCGBL,DNCGTL,DNCGBL1,DNCGTL1
DOUBLE PRECISION ORIENT,THETAORN,R,DR,AREA
DOUBLE PRECISION XTEMP,YTEMP,ZTEMP,XFRAC,YFRAC,fraction,fraction2
DOUBLE PRECISION TERM1TMP,TERM2TMP,TERM3YMP,TERM4TMP
DOUBLE PRECISION DC,DX,DY,DDIST,DIST,detecttime
COMMON/CONST/PI,PHI,THETA,NP,NTIME,V,DT,NMOTDIM,K
COMMON/CONST2/XLENGTH,YLENGTH,SLIT,DG,DGSQ,COLCNT
COMMON/INTEG/GTYPE,NGRAPH1,XSPC,YSPC,XCAP,YCAP,N,NUMCAP
COMMON/CGRAD/KD,CONSUMPCOF,DOTP,PTN,CHI,ZLENGTH
COMMON/GRAPH/HALFCDIM,CDIM,ERRTIME,TIME
COMMON/NUMB/ONE,ZERO
COMMON/VELS/VX,VY,VZ
COMMON/DENSE/DENCELL,DENCELL1,RHO,NPRINT,NDIRECT
COMMON/PORES/PORECODE,BOXCORNER
COMMON/POROUS/FRACTION,XLEFTBOUND,XRIGHTBOUND,fraction2
COMMON/INVARENT/TUMBDEL,NTUMB
COMMON/ARRS/BIN
COMMON/temp1/xtemp,ytemp,ZTEMP,TEMP2,TEMP3
* COMMON/MAPPING/PRN,NEGORN

XCENT=X(I)
YCENT=Y(I)
ZCENT=Z(I)

* periodic BOUNDARY

IF (NXBND.EQ.1) THEN
  NUMBOXX=INT(.5+.5*ABS(X(I)/XLENGTH))
  XCENT=X(I)-2*XLENGTH*NUMBOXX*DSIGN(ONE,X(I))
END IF

```

```

IF (NYBND.EQ.1) THEN
NUMBOXY=INT(.5+.5*ABS(Y(I)/YLENGTH))
  YCENT=Y(I)-2*YLENGTH*NUMBOXY*DSIGN(ONE,Y(I))
ENDIF
  NUMBOXZ=INT(.5+.5*ABS(Z(I)/ZLENGTH))
  ZCENT=Z(I)-2*ZLENGTH*NUMBOXZ*DSIGN(ONE,Z(I))

*   REFLECTIVE BOUNDARY
IF (NXBND.EQ.2) THEN
IF (X(I).GT.XLENGTH) THEN
  XDIF=X(I)-XLENGTH
  XCENT=+XLENGTH-XDIF
  END IF
  IF (X(I).LT.(-XLENGTH)) THEN
  XDIF=-X(I)-XLENGTH
  XCENT=-XLENGTH+XDIF
  END IF
END IF
IF (NYBND.EQ.2) THEN
IF (Y(I).GT.YLENGTH) THEN
  YDIF=Y(I)-YLENGTH
  YCENT=+YLENGTH-YDIF
  END IF
  IF (Y(I).LT.(-YLENGTH)) THEN
  YDIF=-Y(I)-YLENGTH
  YCENT=-YLENGTH+YDIF
  END IF
END IF

  XPOINT=(XCENT/DG)+XSPC
  YPOINT=(YCENT/DG)+YSPC
  XL=INT(XPOINT)-XSPC
  YB=INT(YPOINT)-YSPC

  XR=XL+1
  YT=YB+1

  XGBL=XL*DG
  XGTL=XL*DG
  YGBL=YB*DG
  YGTL=YT*DG
  XGBR=XR*DG
  XGTR=XR*DG
  YGBR=YB*DG
  YGTR=YT*DG

*   reflective pore walls
  IF (boxcorner(XL,YB).EQ.3) THEN
  IF (NYBND.EQ.2) THEN
  IF (Y(I).GT.(YGBL+DG/2)) THEN
    YDIF=YGTL-Y(I)
    YCENT=+YGTL+YDIF
    Y(I)=YCENT
  ELSE
    YDIF=Y(I)-YGBL

```

```

        YCENT=YGBL-YDIF
        Y(I)=YCENT
    END IF
ELSE
        XCENT=(XCENT-XVEL(I))
        YCENT=(YCENT-YVEL(I))
        ZCENT=(ZCENT-ZVEL(I))
END IF
        XPOINT=(XCENT/DG)+XSPC
        YPOINT=(YCENT/DG)+YSPC
        XL=INT(XPOINT)-XSPC
        YB=INT(YPOINT)-YSPC
END IF

IF (BOXCORNER(XL,YB).GE.1) THEN
    IF (BOXCORNER(XL,YB).EQ.1) THEN
        DIST=((X(I)-(DG*(XL+.5)))**2+(Y(I)-(DG*(YB+.5)))**2)**.5
        IF (DIST.LT.(DG/2)) THEN
            XCENT=(XCENT-XVEL(I))
            YCENT=(YCENT-YVEL(I))
            ZCENT=(ZCENT-ZVEL(I))
            XPOINT=(XCENT/DG)+XSPC
            YPOINT=(YCENT/DG)+YSPC
            XL=INT(XPOINT)-XSPC
            YB=INT(YPOINT)-YSPC
        ENDIF
    ENDIF

    zcent1=zcent*1
    IF (BOXCORNER(XL,YB).GT.4) THEN
        IF (BOXCORNER(XL,YB).EQ.41) THEN
            DIST=((X(I)-(DG*(XL+1)))**2+(Y(I)-(DG*(YB+1)))**2+ZCENT1**2)**.5
        ELSEIF (BOXCORNER(XL,YB).EQ.42) THEN
            DIST=((X(I)-(DG*(XL)))**2+(Y(I)-(DG*(YB+1)))**2+ZCENT1**2)**.5
        ELSEIF (BOXCORNER(XL,YB).EQ.43) THEN
            DIST=((X(I)-(DG*(XL+1)))**2+(Y(I)-(DG*(YB)))**2+ZCENT1**2)**.5
        ELSEIF (BOXCORNER(XL,YB).EQ.44) THEN
            DIST=((X(I)-(DG*(XL)))**2+(Y(I)-(DG*(YB)))**2+ZCENT1**2)**.5
        ENDIF
        IF (DIST.LT.(DG)) THEN
            XCENT=(XCENT-XVEL(I))
            YCENT=(YCENT-YVEL(I))
            ZCENT=(ZCENT-ZVEL(I))
            XPOINT=(XCENT/DG)+XSPC
            YPOINT=(YCENT/DG)+YSPC
            XL=INT(XPOINT)-XSPC
            YB=INT(YPOINT)-YSPC
        ENDIF
    END IF
END IF

    IF (GTYPE.EQ.4) THEN
        XPOINT=(XCENT/DG)
        YPOINT=(YCENT/DG)
        IF (YPOINT.GE.0) THEN
            YB=INT(YPOINT)

```

```

ELSE
    YB=INT(YPOINT-1)
END IF
IF (XPOINT.GE.0) THEN
    XL=INT(XPOINT)
ELSE
    XL=INT(XPOINT-1)
END IF
END IF

XR=XL+1
YT=YB+1

XGBL=XL*DG
XGTL=XL*DG
YGBL=YB*DG
YGTL=YT*DG
XGBR=XR*DG
XGTR=XR*DG
YGBR=YB*DG
YGTR=YT*DG

X(I)=XCENT+2*XLENGTH*NUMBOXX*DSIGN(ONE,X(I))
Y(I)=YCENT+2*YLENGTH*NUMBOXY*DSIGN(ONE,Y(I))
Z(I)=ZCENT+2*ZLENGTH*NUMBOXZ*DSIGN(ONE,Z(I))

IF (NXBND.EQ.1) THEN
    IF (XR.gt.XSPC) THEN
        XR=-XSPC
    END IF
END IF

IF (NYBND.EQ.1) THEN
    IF (YT.gt.YSPC) THEN
        YT=-YSPC
    END IF
END IF

TERM1= (XGTR-XCENT)*(YGTR-YCENT)/(DGSQ)
TERM2= (XGBR-XCENT)*(YCENT-YGBR)/(DGSQ)
TERM3= (XCENT-XGBL)*(YCENT-YGBL)/(DGSQ)
TERM4= (XCENT-XGTL)*(YGTL-YCENT)/(DGSQ)

* Calculate concentration Gradient

DCLEFT=CG(XL,YT)-CG(XL,YB)
DCRIGHT=CG(XR,YT)-CG(XR,YB)
DCTOP=CG(XR,YT)-CG(XL,YT)
DCBOTTOM=CG(XR,YB)-CG(XL,YB)
DCY=DCLEFT+(XCENT-XGBL)*(DCRIGHT-DCLEFT)/(XGBR-XGBL)
DCX=DCBOTTOM+(YCENT-YGBL)*(DCTOP-DCBOTTOM)/(YGTL-YGBL)
DCDY=DCY/DG
DCDX=DCX/DG

* DCDX=0
* DCDY=1e-4

```

```

IF (BOXCORNER(XL,YB).EQ.11) THEN
  XFRAC=XPOINT-INT(XPOINT)
  YFRAC=YPOINT-INT(YPOINT)
  TERM1tmp= TERM1/3
  TERM2tmp= TERM2/3
  TERM3tmp= TERM3/3
  TERM4tmp= TERM4/3
  IF (XFRAC.GE.0.5.AND.YFRAC.GE.0.5) THEN
    TERM1= 0
    TERM2= TERM2+TERM1TMP
    TERM3= TERM3+TERM1TMP
    TERM4= TERM4+TERM1TMP
    DCDX=DCTOP/DG
    DCDY=DCRIGHT/DG
  ELSEIF (XFRAC.GE.0.5.AND.YFRAC.LT.0.5) THEN
    TERM2= 0
    TERM1= TERM1+TERM2TMP
    TERM3= TERM3+TERM2TMP
    TERM4= TERM4+TERM2TMP
    DCDX=DCBOTTOM/DG
    DCDY=DCRIGHT/DG
  ELSEIF (XFRAC.LT.0.5.AND.YFRAC.GE.0.5) THEN
    TERM4= 0
    TERM1= TERM1+TERM4TMP
    TERM2= TERM2+TERM4TMP
    TERM3= TERM3+TERM4TMP
    DCDX=DCTOP/DG
    DCDY=DCLEFT/DG
  ELSEIF (XFRAC.LT.0.5.AND.YFRAC.LT.0.5) THEN
    TERM3= 0
    TERM1= TERM1+TERM3TMP
    TERM2= TERM2+TERM3TMP
    TERM4= TERM4+TERM3TMP
    DCDX=DCBOTTOM/DG
    DCDY=DCLEFT/DG
  ENDIF
ENDIF

IF (BOXCORNER(XL,YB).GT.4) THEN
  IF (BOXCORNER(XL,YB).EQ.41) THEN
* BOTTOM LEFT TRIANGLE

  X1=XGBL
  X2=XGBR
  X3=XGTL
  Y1=YGBL
  Y2=YGBR
  Y3=YGTL

  A=(X2-X1)*(Y3-Y1)-(Y2-Y1)*(X3-X1)
  A2=(X3-XCENT)*(Y1-YCENT)-(Y3-YCENT)*(X1-XCENT)
  A3=(X1-XCENT)*(Y2-YCENT)-(Y1-YCENT)*(X2-XCENT)
  TERM2=A3/A
  TERM4=A2/A
  TERM1=1-TERM2-TERM4
  TERM3=0

```

```

DCX=TERM1*DCBOTTOM+TERM4*DCBOTTOM
DCY=TERM1*DCLEFT+TERM2*DCLEFT

DCDX=DCBOTTOM/DG
DCDY=DCLEFT/DG
ELSEIF (BOXCORNER(XL,YB).EQ.42) THEN
** BOTTOM RIGHT TRIANGLE
X1=XGBL
X2=XGBR
X3=XGTR
Y1=YGBL
Y2=YGBR
Y3=YGTR

A=(X2-X1)*(Y3-Y1)-(Y2-Y1)*(X3-X1)
A2=(X3-XCENT)*(Y1-YCENT)-(Y3-YCENT)*(X1-XCENT)
A3=(X1-XCENT)*(Y2-YCENT)-(Y1-YCENT)*(X2-XCENT)
TERM3=A3/A
TERM4=A2/A
TERM1=1-TERM4-TERM3
TERM2=0
DCX=TERM1*DCBOTTOM+TERM4*DCBOTTOM
DCY=TERM3*DCRIGHT+TERM4*DCRIGHT

DCDX=DCBOTTOM/DG
DCDY=DCRIGHT/DG
ELSEIF (BOXCORNER(XL,YB).EQ.43) THEN
*TOP LEFT TRIANGLE
X1=XGBL
X2=XGTR
X3=XGTL
Y1=YGBL
Y2=YGTR
Y3=YGTL

A=(X2-X1)*(Y3-Y1)-(Y2-Y1)*(X3-X1)
A2=(X3-XCENT)*(Y1-YCENT)-(Y3-YCENT)*(X1-XCENT)
A3=(X1-XCENT)*(Y2-YCENT)-(Y1-YCENT)*(X2-XCENT)
TERM2=A3/A
TERM3=A2/A
TERM1=1-TERM2-TERM3
TERM4=0
DCX=TERM2*DCTOP+TERM3*DCTOP
DCY=TERM1*DCLEFT+TERM2*DCLEFT

DCDX=DCTOP/DG
DCDY=DCLEFT/DG
ELSEIF (BOXCORNER(XL,YB).EQ.44) THEN
* TOP RIGHT TRIANGLE
X1=XGBR
X2=XGTR
X3=XGTL
Y1=YGBR
Y2=YGTR
Y3=YGTL

```

```

A=(X2-X1)*(Y3-Y1)-(Y2-Y1)*(X3-X1)
A2=(X3-XCENT)*(Y1-YCENT)-(Y3-YCENT)*(X1-XCENT)
A3=(X1-XCENT)*(Y2-YCENT)-(Y1-YCENT)*(X2-XCENT)
TERM2=A3/A
TERM3=A2/A
TERM4=1-TERM2-TERM3
TERM1=0
DCX=TERM2*DCTOP+TERM3*DCTOP
DCY=TERM3*DCRIGHT+TERM4*DCRIGHT
DCDX=DCTOP/DG
DCDY=DCRIGHT/DG
ENDIF
DCDX=DCX/DG
DCDY=DCY/DG
END IF
DOTP=(DCDX)*(XVEL(I)/V)+(DCDY)*(YVEL(I)/V)
DOTP2=(DCDX)*(XVEL(I)/V)+(DCDY)*(YVEL(I)/V)

* Do dot product

CCELL= CG(XL,YB)*TERM1+CG(XL,YT)*TERM2+CG(XR,YT)*TERM3
& +CG(XR,YB)*TERM4

* ngradient=0 for regular spacial
* ngradient=1 for just temporial converted to spacial
* ngradient=2 for time derivative and regular spacial gradient
* ngradient=3 for time derivative and converted spacial gradient
* ngradient=4 for time derivative only

ngradient=0
detecttime=2

if (NGRADIENT.ge.1) then
  NDET=detecttime/DT

DO J=NDET,1,-1
  CPREV(I,J)=CPREV(I,J-1)
  XPREV(I,J)=XPREV(I,J-1)
  YPREV(I,J)=YPREV(I,J-1)
END DO
CPREV(I,0)=CCELL
XPREV(I,0)=X(I)
YPREV(I,0)=Y(I)

IF (K.GT.NDET) THEN
  DC=CCELL-CPREV(I,NDET)
  DX=X(I)-XPREV(I,NDET)
  DY=Y(I)-YPREV(I,NDET)
  DDIST1=V*NDET
  DDIST=(DX**2+DY**2)**.5
  DCDX=DC*DX/(DDIST*DDIST1)
  DCDY=DC*DY/(DDIST*DDIST1)
  DCDT=DC/NDET

IF(DY.ne .ZERO)THEN
  DCY=(DC**2/((1+(DX/DY)**2)))**.5

```



```

    DCX=(DCY*DX/DY)
    DCDX=DCX/DX
    DCDY=DCY/DY
    ELSEif(dx.ne. zero) then
    DCX=(DC**2/((1+(DY/DX)**2)))**.5
    DCY=(DCX*DY/DX)
    DCDX=DCX/DX
    DCDY=DCY/DY
    END IF
**    alternate gradient detection
    if (ngradients.eq.1.or.ngradients.eq.3) then
    DOTP=(DCDX)*(XVEL(I)/V)+(DCDY)*(YVEL(I)/V)
    Endif
    ELSE
    ENDIF
    Else
    ENDIF

    DOTP3=(DCDX)*(XVEL(I)/V)+(DCDY)*(YVEL(I)/V)

    IF (NGRADIEN.EQ.4) THEN
    DOTP=0
    ENDIF
    timedir=DCDT/V
    TIMESUM=timedir+DOTP

    IF (TIMESUM.GT.0) THEN
    IF (NGRADIEN.GE.2)THEN
    IF (NGRADIEN.EQ.4) THEN
    EPS2=(timedir)*(CHI/(V*(1+ccell)**2))
    ELSE
    EPS2=(timedir+DOTP)*(CHI/(V*(1+ccell)**2))
    END IF
    ELSE
    EPS2=(DOTP*CHI/(V*(1+ccell)**2))
    END IF
    eps=(exp(-EPS2))
    PTN=PT*eps
    ELSE
*    eps3=(-DOTP*CHI/(V*(1+CCELL)*(1+ccell)))
*    five=5
*    eps4=min(eps3,five)
*    PTN=PT*exp(eps4)
    PTN=PT
    END IF

    IF (K.EQ.1)THEN
    nsim=0
    if (nsim.eq.1) then
    WRITE(UNIT=9999,FMT=68) X(I),Y(I)
    WRITE(UNIT=9999,FMT=68) X(I),Y(I)
    endif
    ENDIF

*    cell density calcs

```

```

IF (NPRINT.EQ.1.AND.K.EQ.1) THEN
  IF ((XCENT).LT.(SLIT).and.(XCENT).GE.-(SLIT)) THEN
    IF (NDIRECT.EQ.0) THEN
      DNCGTL1=(YCENT-YGBL)/DG
      DNCGBL1=1-DNCGTL1
      DENCELL1(YB)=DENCELL1(YB)+DNCGBL1
      DENCELL1(YT)=DENCELL1(YT)+DNCGTL1
    ELSE
      DNCGTL1=(XCENT-XGBL)/DG
      DNCGBL1=1-DNCGTL1
      DENCELL1(XL)=DENCELL1(XL)+DNCGBL1
      DENCELL1(XR)=DENCELL1(XR)+DNCGTL1
    ENDIF
  END IF
END IF

```

```

IF (MOD(K,INCR).EQ.0) THEN
  IF (NPRINT.EQ.1) THEN
    IF ((XCENT).LT.(SLIT).and.(XCENT).GE.-(SLIT)) THEN
      IF (NDIRECT.EQ.0) THEN
        DNCGTL=(YCENT-YGBL)/DG
        DNCGBL=1-DNCGTL
        DENCELL(YB)=DENCELL(YB)+(DNCGBL)
        DENCELL(YT)=DENCELL(YT)+(DNCGTL)
      ELSE
        DNCGTL=(XCENT-XGBL)/DG
        DNCGBL=1-DNCGTL
        DENCELL(XL)=DENCELL(XL)+(DNCGBL)
        DENCELL(XR)=DENCELL(XR)+(DNCGTL)
      ENDIF
    END IF
  END IF

```

```

*   change to turn on orientation distribution
      if (NMOTDIM.eq.4) then
        IF (YVEL(I).GE.V)THEN
          THETAORN=0
        ELSE IF (YVEL(I).LE.-V)THEN
          THETAORN=PI
        ELSE
          THETAORN=ACOS(YVEL(I)/V)
        END IF
        IBIN=INT(THETAORN*180/PI)
        ORIENT=SIGN(thetaorn,XVEL(I))
        IF (ORIENT.LT.0) THEN
          NEGORN(IBIN)=NEGORN(IBIN)+1
        ELSE
          PRN(IBIN)=PRN(IBIN)+1
        END IF
      END IF

```

```

*   calculate radial distribution

R=((X(I))**2+(Y(I))**2)**.5

DR=DG

```

```

        BI=INT(R/DR)
*       AREA=PI*(((BI+1)*DR)**2-(BI*DR)**2)/(10000**2) !converted to cm^2
        IF (NDIRECT.EQ.0) THEN
            If (Y(I).GT.ZERO) THEN
                AREA1=PI*(((BI+1)*DR)**2-(BI*DR)**2)/(10000**2) !converted to cm^2
            ELSE
                AREA2=PI*(((BI+1)*DR)**2-(BI*DR)**2)/(10000**2) !converted to cm^2
            END IF
        ELSE
            If (X(I).GT.ZERO) THEN
                AREA3=PI*(((BI+1)*DR)**2-(BI*DR)**2)/(10000**2) !converted to cm^2
            ELSE
                AREA4=PI*(((BI+1)*DR)**2-(BI*DR)**2)/(10000**2) !converted to cm^2
            END IF
        ENDIF
        IF (NDIRECT.EQ.0) THEN
            If (Y(I).GT.ZERO) THEN
                RHO(BI)=RHO(BI)+(1/(2*AREA1))
            ELSE
                RHO(-BI)=RHO(-BI)+(1/(2*AREA2))
            END IF
        ELSE
            If (X(I).GT.ZERO) THEN
                RHO(BI)=RHO(BI)+(1/(2*AREA3))
            ELSE
                RHO(-BI)=RHO(-BI)+(1/(2*AREA4))
            END IF
        ENDIF
    ENDIF

*       calculate number of cells in capillary

    IF (YCENT.GE.YLENGTH) THEN
        NUMCAP=NUMCAP+1
    END IF

*       print out cell coordinates

68    FORMAT (F11.4,T20,F11.4,T35,F11.4,T50,F10.6,T62,I6)

    IF(N.EQ.1) THEN
        if (NMOTDIM.eq.2) then
            WRITE (UNIT=4+K/INCR,FMT=68) X(I),Y(I) !,dotp,timedir!XCENT,YCENT
        else
            WRITE (UNIT=4+K/INCR,FMT=68) X(I), Y(I) ,Z(I)
        endif
    ENDIF
    IF(N.EQ.3) THEN
        WRITE(UNIT=9999,FMT=68) X(I),Y(I) !,Z(I) !,orient
*       WRITE (UNIT=4+K/INCR,FMT=68) X(I), Y(I) !XCENT,YCENT !,Z(I)
        IF (K.EQ.NTIME) THEN
            WRITE(UNIT=5,FMT=68) X(I),Y(I)
        endif
    END IF
END IF

*       Depleting concentration

```

```

      IF (CONSUMPCOF.GT.0) THEN

      CG(XL,YB)=DMAX1(ZERO,CG(XL,YB)-TERM1*CONSUMPCOF)
      CG(XL,YT)=DMAX1(ZERO,CG(XL,YT)-TERM2*CONSUMPCOF)
      CG(XR,YT)=DMAX1(ZERO,CG(XR,YT)-TERM3*CONSUMPCOF)
      CG(XR,YB)=DMAX1(ZERO,CG(XR,YB)-TERM4*CONSUMPCOF)
      ENDIF

      RETURN
      END SUBROUTINE GRADIENT

* UPDATE CONCENTRATION SUBROUTINE-----*

      SUBROUTINE CONCDIF(CG,DIFF,DIFFTYPE,NXBND,NYBND,DIFFPORE)

      PARAMETER (NXMAX=1000, NYMAX=1000)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER K,N,NMOTDIM,NUMCAP
      INTEGER GTYPE,NGRAPH1,XSPC,YSPC,XCAP,YCAP
      INTEGER JDIF,NDIF
      INTEGER I,NFILES,DIFFTYPE,NXBND,NYBND,I4
      integer xleftbound, xrightbound
      integer porecode(-NXMAX:NXMAX,-NYMAX:NYMAX)
      integer boxcorner(-NXMAX:NXMAX,-NYMAX:NYMAX)
      DOUBLE PRECISION
      XLENGTH,YLENGTH,SLIT,DG,DGSQ,CAPLENGTH,YTOTAL,COLCNT
      DOUBLE PRECISION KD,CONSUMPCOF,DOTP,PTN,CHI,DIFF
      DOUBLE PRECISION HALFCDIM,CDIM,ERRTIME,TIME
      DOUBLE PRECISION ONE, ZERO,fraction, DIFFPORE,fraction2
      DOUBLE PRECISION D2CGDX2(-NXMAX:NXMAX,-NYMAX:NYMAX)
      DOUBLE PRECISION D2CGDY2(-NXMAX:NXMAX,-NYMAX:NYMAX)
      DOUBLE PRECISION, dimension(-XSPC:XSPC, -YSPC:YCAP), intent(inout)
      &    :: CG
      COMMON/CONST2/XLENGTH,YLENGTH,SLIT,DG,DGSQ,COLCNT
      COMMON/INTEG/GTYPE,NGRAPH1,XSPC,YSPC,XCAP,YCAP,N,NUMCAP
      COMMON/NUMB/ONE,ZERO
      COMMON/PORES/PORECODE, BOXCORNER
      COMMON/POROUS/FRACTION,XLEFTBOUND,XRIGHTBOUND,fraction2

      IF(DIFFTYPE.EQ.1) THEN

*center *****
      DO JDIF=-YSPC,YSPC
        DO NDIF=1-XSPC,XSPC-1
          IF (PORECODE(NDIF,JDIF).EQ.6) THEN
            D2CGDX2(NDIF,JDIF)=(2*CG(NDIF+1,JDIF)-2*(CG(NDIF,JDIF)))
          &    /DGSQ
          ELSE
            IF (PORECODE(NDIF,JDIF).EQ.4) THEN
              D2CGDX2(NDIF,JDIF)=(2*CG(NDIF-1,JDIF)-2*(CG(NDIF,JDIF)))
            &    /DGSQ
          ELSE
            IF (PORECODE(NDIF,JDIF).EQ.5) THEN
              D2CGDX2(NDIF,JDIF)=0

```

```

        ELSE
        IF (PORECODE(NDIF,JDIF).EQ.40) THEN
            D2CGDX2(NDIF,JDIF)=0
        ELSE
            D2CGDX2(NDIF,JDIF)=(CG(NDIF+1,JDIF)-2*(CG(NDIF,JDIF))
&      +CG(NDIF-1,JDIF))/DGSQ
        END IF
        END IF
        ENDIF
        END IF
        END DO
        END DO

* center *****

        DO JDIF=1-YSPC,YSPC-1
            DO NDIF=-XSPC,XSPC
                IF (PORECODE(NDIF,JDIF).EQ.2) THEN
                    D2CGDY2(NDIF,JDIF)=(2*CG(NDIF,JDIF-1)-2*(CG(NDIF,JDIF)))
&      /DGSQ
                ELSE
                    IF (PORECODE(NDIF,JDIF).EQ.8) THEN
                        D2CGDY2(NDIF,JDIF)=(2*CG(NDIF,JDIF+1)-2*(CG(NDIF,JDIF)))
&      /DGSQ
                    ELSE
                        IF (PORECODE(NDIF,JDIF).EQ.5) THEN
                            D2CGDY2(NDIF,JDIF)=0
                        ELSE
                            IF (PORECODE(NDIF,JDIF).EQ.20) THEN
                                D2CGDY2(NDIF,JDIF)=0
                            ELSE
                                D2CGDY2(NDIF,JDIF)=(CG(NDIF,JDIF+1)-2*(CG(NDIF,JDIF))
&      +CG(NDIF,JDIF-1))/DGSQ
                            END IF
                        ENDIF
                        END IF
                        END IF
                        END DO
                    END DO
                ***** no flux edges *****
                IF (NYBND.EQ.0) THEN
                    DO JDIF=-YSPC,YSPC
                        D2CGDX2(-xspc,JDIF)=(2*CG(-xspc+1,JDIF)-2*(CG(-xspc,JDIF)))
&      /DGSQ
                        D2CGDX2(xspc,JDIF)=(2*CG(xspc-1,JDIF)-2*(CG(xspc,JDIF)))
&      /DGSQ
                    enddo
                ENDIF
                IF (NXBND.EQ.0) THEN
                    DO NDIF=-XSPC,XSPC
                        D2CGDY2(NDIF,-yspc)=(2*CG(NDIF,-yspc+1)-2*(CG(NDIF,-yspc)))
&      /DGSQ
                        D2CGDY2(NDIF,yspc)=(2*CG(NDIF,yspc-1)-2*(CG(NDIF,yspc)))
&      /DGSQ
                    enddo
                ENDIF

```

```

***** periodic edges
IF (NYBND.EQ.1) THEN
  DO JDIF=-YSPC,YSPC-1
    D2CGDX2(-XSPC,JDIF)=(CG(1-XSPC,JDIF)-2*(CG(-XSPC,JDIF))
    &   +CG(XSPC-1,JDIF))/DGSQ
    D2CGDX2(XSPC-1,JDIF)=(CG(XSPC-2,JDIF)-2*(CG(XSPC-1,JDIF))
    &   +CG(-XSPC,JDIF))/DGSQ
  ENDDO
ENDIF

  IF (NXBND.EQ.1) THEN
    DO NDIF=-XSPC,XSPC-1
      D2CGDY2(NDIF,-YSPC)=(CG(NDIF,1-YSPC)-2*(CG(NDIF,-YSPC))
      &   +CG(NDIF,YSPC-1))/DGSQ
      D2CGDY2(NDIF,YSPC-1)=(CG(NDIF,YSPC-2)-2*(CG(NDIF,YSPC-1))
      &   +CG(NDIF,-YSPC))/DGSQ
    END DO
  ENDIF

  IF (GTYPE.EQ.4) THEN
    *****CAP CENTER
    DO JDIF=YSPC,YCAP
      DO NDIF=1-XCAP,XCAP-1
        D2CGDX2(NDIF,JDIF)=(CG(NDIF+1,JDIF)-2*(CG(NDIF,JDIF))
        &   +CG(NDIF-1,JDIF))/DGSQ
      END DO
    END DO
    DO JDIF=0,YCAP-1
      DO NDIF=-XCAP,XCAP
        D2CGDY2(NDIF,JDIF)=(CG(NDIF,JDIF+1)-2*(CG(NDIF,JDIF))
        &   +CG(NDIF,JDIF-1))/DGSQ
      END DO
    END DO
    ***** CAP edges *****
    DO JDIF=YSPC,YCAP
      D2CGDX2(-xCAP,JDIF)=(2*CG(-xCAP+1,JDIF)-2*(CG(-xCAP,JDIF)))
      &   /DGSQ
      D2CGDX2(xCAP,JDIF)=(2*CG(xCAP-1,JDIF)-2*(CG(xCAP,JDIF)))
      &   /DGSQ
    enddo
    DO NDIF=-XCAP,XCAP
      D2CGDY2(NDIF,YCAP)=(2*CG(NDIF,YCAP-1)-2*(CG(NDIF,YCAP)))
      &   /DGSQ
      D2CGDY2(NDIF,YSPC)=(CG(NDIF,YSPC+1)-2*(CG(NDIF,YSPC))
      &   +CG(NDIF,YSPC-1))/DGSQ
      D2CGDY2(NDIF,YSPC-1)=(CG(NDIF,YSPC)-2*(CG(NDIF,YSPC-1))
      &   +CG(NDIF,YSPC-2))/DGSQ
    enddo
  ENDIF

  ***CALC NEW CONC

  DO JDIF=-YSPC,YSPC
    DO NDIF=-XSPC,XSPC
      IF (ndif.LT.XRIGHTBOUND.AND.ndif.GT. XLEFTBOUND) THEN
        CG(NDIF,JDIF)=DMAX1(ZERO,CG(NDIF,JDIF)

```

```

&      +DIFFPORE*(D2CGDX2(NDIF,JDIF)+D2CGDY2(NDIF,JDIF)))
      ELSE
        CG(NDIF,JDIF)=DMAX1(ZERO,CG(NDIF,JDIF)
&      +DIFF*(D2CGDX2(NDIF,JDIF)+D2CGDY2(NDIF,JDIF)))
      ENDIF
    END DO
  END DO

*** PERIODIC Conc *****
  IF (NYBND.EQ.1) THEN
    DO JDIF=1-YSPC,YSPC-1
      CG(XSPC,JDIF)=CG(-XSPC,JDIF)
    ENDDO
  ENDIF
  IF (NXBND.EQ.1) THEN
    DO NDIF=1-XSPC,XSPC-1
      CG(NDIF,YSPC)=CG(NDIF,-YSPC)
    ENDDO
    CG(XSPC,YSPC)=CG(-XSPC,-YSPC)
    CG(-XSPC,YSPC)=(CG(-XSPC,YSPC-1)+CG(1-XSPC,-YSPC))/2
    CG(XSPC,-YSPC)=CG(-XSPC,YSPC)
  ENDIF

*****
*****CAPILLARY
  IF (GTYPE.EQ.4) THEN
    DO JDIF=YSPC-1,YCAP
      DO NDIF=-XCAP,XCAP
        CG(NDIF,JDIF)=DMAX1(ZERO,CG(NDIF,JDIF)
&      +DIFF*(D2CGDX2(NDIF,JDIF)+D2CGDY2(NDIF,JDIF)))
      END DO
    END DO
  END IF

  END IF

  RETURN
END SUBROUTINE CONCDIF

```

```

* GRID SETUP SUBROUTINE-----*
*   The GRID subroutine sets up the concentration gradient at
*   fixed points in a grid. The concentration gradient is a
*   linear gradient from 0 to CONC/KD in the Y direction and the origin
*   is in the center of the grid.
*   ! AVDef is the AViz module file

```

```

SUBROUTINE GRID(CG,DG)

```

```

PARAMETER (NXMAX=1000, NYMAX=1000)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
integer status, arrayData
integer :: lbnd(2) = 0
INTEGER GTYPE,NGRAPH1,XSPC,YSPC,XCAP,YCAP,A,B
integer xleftbound, xrightbound
integer porecode(-NXMAX:NXMAX,-NYMAX:NYMAX)

```

```

integer boxcorner(-NXMAX:NXMAX,-NYMAX:NYMAX)
DOUBLE PRECISION, dimension(-XSPC:XSPC, -YSPC:YCAP), intent(inout)
&    :: CG
    DOUBLE PRECISION HALFCDIM,CDIM,ERRTIME,TIME
DOUBLE PRECISION ARG,DG, YTOTAL ,fraction2,fraction
COMMON/INTEG/GTYPE,NGRAPH1,XSPC,YSPC,XCAP,YCAP,N,NUMCAP
COMMON/GRAPH/HALFCDIM,CDIM,ERRTIME,TIME
    COMMON/PORES/PORECODE, BOXCORNER
    COMMON/POROUS/FRACTION,XLEFTBOUND,XRIGHTBOUND,fraction2

IF (GTYPE.eq.1) THEN
    DO B=-YSPC,YSPC
        DO A=-XSPC,XSPC
            CG(A,B)=CDIM*(YSPC+B)/(2*YSPC)
        END DO
    END DO
ENDIF

IF (GTYPE.eq.2) THEN
    DO B=-YSPC,YSPC
        ARG=DG*B/(ERRTIME)
        CSTEP=(HALFCDIM)*(1+ERF(ARG))
        DO A=-XSPC,XSPC
            CG(A,B)=CSTEP
        END DO
    END DO
ENDIF

IF (GTYPE.eq.3) THEN
    DO B=-YSPC,YSPC
        DO A=-XSPC,XSPC
            CG(A,B)=cdim
        END DO
    END DO
DO B=-YSPC,YSPC
    DO A=-XSPC,XSPC
        IF (BOXCORNER(A,B).EQ.41) THEN
            CG(A+1,B+1)=22
        ENDIF
        IF (A.LT.XRIGHTBOUND.AND.A.GT. XLEFTBOUND) THEN
            CG(A,B)=cdim
            IF (PORECODE(A,B).EQ.5) THEN
                CG(A,B)=0
            ENDIF
        ENDIF
    ENDDO
ENDDO
ENDIF

    IF (GTYPE.eq.4) THEN
        DO B=YSPC+1,YCAP
            DO A=-XCAP,XCAP
                CG(A,B)=CDIM
            END DO
        END DO
    DO A=-XSPC,XSPC

```



```

        CG(A,-YSPC)=0
        CG(A,1-YSPC)=0
    END DO
    ENDIF

    IF (GTYPE.eq.5) THEN
        DO B=-YSPC,YSPC
            DO A=-XSPC,XSPC
                CG(A,B)=CDIM*(B)/(YSPC)
            END DO
        END DO
    END DO
    ENDIF

    IF (GTYPE.eq.6) THEN
        DO B=-YSPC,YSPC
            DO A=-XSPC,XSPC
                CG(A,B)=CDIM*(2*YSPC+B+A)/(2*(YSPC+XSPC))
            END DO
        END DO
    ENDIF

    IF (GTYPE.eq.7) THEN
        DO A=-XSPC,XSPC
            DO B=-YSPC,1
                CG(A,B)=0
            END DO
            CG(A,0)=HALFCDIM
            DO B=1,YSPC
                CG(A,B)=CDIM
            END DO
        END DO
    ENDIF
endif

    IF (NGRAPH1.EQ.1) THEN
! Call StartWatch to let the AView lib know we're interested in viewing CG
        print *, "Initializing array data"
        call fag!StartWatch(CG, status)
        lbnd(1:size(shape(CG))) = lbound(CG)
        call fag!LBound(CG, lbnd, status)
        print *, "Starting Array Viewer"
        call fag!Show(CG, status)
        ! Set the title bar on ArrayViewer
        call fag!Name(CG, "concentration plot", status)
        write(title, 202)
202    format("Time step: ", I5)
    ENDIF

    RETURN
    END SUBROUTINE GRID

* TUMBLE SUBROUTINE-----*
*
* The TUMBLE subroutine determines the new velocity vector following
* a tumble. The routine gets the turn angles theta and phi from a
* random number formula and a turn angle function (TUNANG), respectively.

```

* The subroutine then calls VELN to determine the new vector components.

*

```
SUBROUTINE TUMBLE(I,XVEL,YVEL,ZVEL,ISEED)
PARAMETER (NPMAX=20000)
  PARAMETER (NKMAX=10000)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  INTEGER TUMBDEL
  INTEGER NTUMB(npmax,NKMAX)
  DOUBLE PRECISION XVEL(NPMAX),YVEL(NPMAX),ZVEL(NPMAX)
REAL BIN(0:100)
COMMON/CONST/PI,PHI,THETA,NP,NTIME,V,DT,NMOTDIM,K
COMMON/VELS/VX,VY,VZ
COMMON/ARRS/BIN
COMMON/INVARENT/TUMBDEL,NTUMB
```

```
IF (NMOTDIM.EQ.3) THEN
  PHI=TUNANG(ISEED)
  THETA=2*PI*RAN1(ISEED)
  VX=XVEL(I)
  VY=YVEL(I)
  VZ=ZVEL(I)
  CALL VELN(VXN,VYN,VZN)
  XVEL(I)=VXN
  YVEL(I)=VYN
  ZVEL(I)=VZN
END IF
```

```
IF (NMOTDIM.EQ.2) THEN
  THETA1=TUNANG(ISEED)
  DES=(1-2*RAN1(ISEED))
  THETA=SIGN(THETA1,DES)
  VX=XVEL(I)
  VY=YVEL(I)
  CALL VELN(VXN,VYN,VZN)
  XVEL(I)=VXN
  YVEL(I)=VYN
END IF
```

```
IF (TUMBDEL.EQ.1) THEN
  ntumb(I,k)=1
ENDIF
```

```
RETURN
END SUBROUTINE TUMBLE
```

*

* NOTUMBLE SUBROUTINE-----*

*

* The NOTUMBLE subroutine determines the new position of a cell if
* no tumble occurs.

*

```
SUBROUTINE NOTUMBLE(I,XVEL,YVEL,ZVEL,X,Y,Z,NXBND,NYBND,YTOTAL)
PARAMETER (NPMAX=20000)
  PARAMETER (NKMAX=1)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  INTEGER K,N,NMOTDIM,NUMCAP
```

```

    INTEGER I,NXBND,NYBND,GTYPE
    INTEGER TUMBDEL
    INTEGER NTUMB(npmax,NKMAX)
    DOUBLE PRECISION X(NPMAX),Y(NPMAX),Z(NPMAX)
    DOUBLE PRECISION XVEL(NPMAX),YVEL(NPMAX),ZVEL(NPMAX)
    DOUBLE PRECISION XLENGTH,YLENGTH,SLIT,DG,DGSQ,CAPLENGTH,COLCNT
    DOUBLE PRECISION abx,XTEMP,ZTEMP,YTEMP,ABZ,YTOTAL
    DOUBLE PRECISION TEMP2,TEMP3
    REAL BIN(0:100)
    COMMON/CONST/PI,PHI,THETA,NP,NTIME,V,DT,NMOTDIM,K
    COMMON/CONST2/XLENGTH,YLENGTH,SLIT,DG,DGSQ,COLCNT
    COMMON/INTEG/GTYPE,NGRAPH1,XSPC,YSPC,XCAP,YCAP,N,NUMCAP
    COMMON/VELS/VX,VY,VZ
    COMMON/ARRS/BIN
    COMMON/temp1/xtemp,ytemp,ZTEMP,TEMP2,TEMP3
    COMMON/INVARENT/TUMBDEL,NTUMB

    XTEMP=X(I)
    YTEMP=Y(I)
    ZTEMP=Z(I)

    IF (NMOTDIM.EQ.3) THEN
        TEMP2=z(I)
        X(I)=X(I)+XVEL(I)
        Y(I)=Y(I)+YVEL(I)
        Z(I)=Z(I)+ZVEL(I)
        TEMP3=z(I)
        abX=abs(X(I))
        abZ=abs(Z(I))
        *   Z=COLCNTZ+1

        IF (SLIT.LT.abx) THEN
            IF (NXBND.EQ.0) THEN
                X(I)=XTEMP
                Y(I)=YTEMP
                Z(I)=ZTEMP
            END IF
            COLCNT=COLCNT+1
            END IF

            IF (NYBND.EQ.0) THEN
                abY=abs(Y(I))
                IF (YLENGTH.LT.abY) THEN
                    Y(I)=YTEMP
                    X(I)=XTEMP
                    Z(I)=ZTEMP
                END IF
            END IF
        END IF

    IF (NMOTDIM.EQ.2) THEN
        TEMP2=Y(I)
        X(I)=X(I)+XVEL(I)
        Y(I)=Y(I)+YVEL(I)
        TEMP3=Y(I)
        abx=abs(X(I))

```

```

IF (GTYPE.EQ.4) THEN
  IF (Y(I).GE.YLENGTH.AND.SLIT.LT.ABX) THEN
    X(I)=XTEMP
    Y(I)=YTEMP
  end if
  IF (Y(I).GE.YTOTAL) THEN
    X(I)=XTEMP
    Y(I)=YTEMP
  ENDIF
  IF (Y(I).LE.-YLENGTH) THEN
    X(I)=XTEMP
    Y(I)=YTEMP
  ENDIF
  IF (X(I).GE.XLENGTH) THEN
    X(I)=XTEMP
    Y(I)=YTEMP
  ENDIF
  IF (X(I).LE.-XLENGTH) THEN
    X(I)=XTEMP
    Y(I)=YTEMP
  ENDIF
ELSE
  IF (SLIT.LT.abx) THEN
    IF (NXBND.EQ.0) THEN
      X(I)=XTEMP
      Y(I)=YTEMP
    END IF
    COLCNT=COLCNT+1
  END IF

  IF (NYBND.EQ.0) THEN
    abY=abs(Y(I))
    IF (YLENGTH.LT.abY) THEN
      Y(I)=YTEMP
      X(I)=XTEMP
    END IF
  END IF
END IF
END IF

IF (TUMBDL.EQ.1) THEN
  L=.1/DT
  IF (K.ge.L) THEN
    DO J=1,(L-1)
      m=k-J
      ntumb1=ntumb(I,m)
      if (ntumb(I,m).eq.1) then
        Y(I)=YTEMP
        X(I)=XTEMP
        Z(I)=ZTEMP
      endif
    end do
  ENDIF
ENDIF

```

```

      RETURN
      END SUBROUTINE NOTUMBLE
*-----*
*
* ROOTMEAN SUBROUTINE-----*
*
* The ROOTMEAN subroutine determines the motility of the cells based
* on a root mean square calculation. This routine also outputs the
* motility coefficient. The final purpose of the subroutine is to
* output the position of the cells to the sequential output files.
*
      SUBROUTINE ROOTMEAN(CG,X,Y,Z,XO,YO,ZO,INCR,RAD,ISEED,NFILES)
      PARAMETER (NPMAX=20000)
      PARAMETER (NXMAX=1000, NYMAX=1000)
      PARAMETER (NDENMAX=1000)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER FILENM*10,CGRAPH*12
      INTEGER K,N,NMOTDIM,NUMCAP
      INTEGER GTYPE,NGRAPH1,XSPC,YSPC,XCAP,YCAP
      integer xleftbound, xrightbound
      INTEGER IBIN,BI,stat,BI2,BI3
      INTEGER I,NFILES, SEED1,NPRINT,NDIRECT
      INTEGER I3,FIRST,LAST,LOW,LARGE,C,F,E
      DOUBLE PRECISION X(NPMAX),Y(NPMAX),Z(NPMAX)
      DOUBLE PRECISION XO(NPMAX),YO(NPMAX), ZO(NPMAX)
      DOUBLE PRECISION, dimension(-XSPC:XSPC, -YSPC:YCAP), intent(inout)
&      :: CG
      DOUBLE PRECISION DENCELL(-NDENMAX:NDENMAX)
      DOUBLE PRECISION DENCELL1(-NDENMAX:NDENMAX)
      DOUBLE PRECISION DENCELLAVG(-NDENMAX:NDENMAX)
      DOUBLE PRECISION DENCELL1AVG(-NDENMAX:NDENMAX)
*      DOUBLE PRECISION PRN(0:NXMAX)
*      DOUBLE PRECISION NEGORN(0:NXMAX)
      DOUBLE PRECISION RHO(-NDENMAX:NDENMAX)
      DOUBLE PRECISION KD,CONSUMPCOF,DOTP,PTN,CHI,DIFF
      DOUBLE PRECISION
      XLENGTH,YLENGTH,SLIT,DG,DGSQ,CAPLENGTH,YTOTAL,COLCNT
      DOUBLE PRECISION HALFCDIM,CDIM,ERRTIME,TIME
      DOUBLE PRECISION ARG,CMSQ,fraction,fraction2
      DOUBLE PRECISION OMEGA,COEFF,AVGTA,RANMOT
      REAL BIN(0:100)
      COMMON/CONST/PI,PHI,THETA,NP,NTIME,V,DT,NMOTDIM,K
      COMMON/CONST2/XLENGTH,YLENGTH,SLIT,DG,DGSQ,COLCNT
      COMMON/INTEG/GTYPE,NGRAPH1,XSPC,YSPC,XCAP,YCAP,N,NUMCAP
      COMMON/GRAPH/HALFCDIM,CDIM,ERRTIME,TIME
      COMMON/DENSE/DENCELL,DENCELL1,RHO,NPRINT,NDIRECT
      COMMON/PORES/PORECODE,BOXCORNER
      COMMON/VELS/VX,VY,VZ
      COMMON/ARRS/BIN
      COMMON/temp1/xtemp,ytemp,TEMP2,TEMP3
*      COMMON/MAPPING/PRN,NEGORN
      COMMON/POROUS/FRACTION,XLEFTBOUND,XRIGHTBOUND,fraction2
      common/info/avgta,AVGCOS,PT,SEED1

      RSQ=0

```

- * Pick if want relative motility or absolute
- * motttype=1 for absolute motttype=2 for relative

```

motttype=1

IF (NMOTDIM.EQ.3) THEN
  DO J=1,NP
    DX=X(J)-XO(J)
    DY=Y(J)-YO(J)
    DZ=Z(J)-ZO(J)
    RSQ=RSQ+(DX*DX+DY*DY+DZ*DZ)
    RSQ2=RSQ2+(DX*DX+DY*DY)
    IF (MOTTTYPE.EQ.2) THEN
      XO(J)=X(J)
      YO(J)=Y(J)
      ZO(J)=Z(J)
    ENDIF
  END DO
  OMEGA=RSQ/NP
  OMEGA1=OMEGA/100000000
  IF (MOTTTYPE.EQ.2) THEN
    COEFF=OMEGA/(6*incr*dt)
  ELSE
    COEFF=OMEGA/(6*TIME)
  ENDIF
  RANMOT=(V/DT)**2/(3*PT*10000**2*(1-AVGCOS))
  *****2D MOTILITY
  coeff2=(rsq2/np)/(4*time)*(1.0/10000.0)**2
  RANMOT=COEFF2
  END IF

IF (NMOTDIM.EQ.2) THEN

  DO J=1,NP
    if (X(J).GT.0) THEN
      DX=X(J)-XO(J)
      DY=Y(J)-YO(J)
      RSQ=RSQ+(DX*DX+DY*DY)
      RSQX=RSQX+(DX*DX)
      RSQY=RSQY+(DY*DY)
      IF (MOTTTYPE.EQ.2) THEN
        XO(J)=X(J)
        YO(J)=Y(J)
      ENDIF
    ENDIF

  NP1=NP1+1
  ELSE
    DX=X(J)-XO(J)
    DY=Y(J)-YO(J)
    RSQ2=RSQ2+(DX*DX+DY*DY)
    NP2=NP2+1
  ENDIF
  END DO

*      OMEGA=RSQ/NP

```

```

OMEGA=RSQ/NP1
  OMEGA1=OMEGA/100000000
OMEGAX=RSQX/NP
  OMEGA1X=OMEGAX/100000000
  OMEGAY=RSQY/NP
  OMEGA1Y=OMEGAY/100000000
  IF (MOTTYPE.EQ.2) THEN
    COEFF=OMEGA/(4*incr*dt)
  ELSE
    COEFF=OMEGA/(4*TIME)
    COEFFX=OMEGAX/(4*TIME)
    COEFFY=OMEGAY/(4*TIME)
  ENDIF

coeff2=(rsq2/np2)/(4*time)*(1.0/10000.0)**2

*   CONVERT TO CM^2/S

  RANMOT=(V/DT)**2/(2*PT*10000**2*(1-AVGCOS))

*   coeff2=(rsq2/np)/(4*time)*(1.0/10000.0)**2
  RANMOT=COEFF2

ENDIF

*Conversion um^2/s ---> cm^2/s
  COEFF=COEFF*(1.0/10000.0)**2
  WRITE(UNIT=3,FMT=61),K,COEFF,TIME,fraction,OMEGA1,avgt,RANMOT
  & ,fraction2
61  FORMAT (I8,T10,E12.5,T25,F10.2,T43,F10.6,T57,F12.8,T70,F12.3,
  & T85,E12.4,T100,F8.4)
55  FORMAT (I5,T10,E12.4,T25,F10.2,T43,I10,T57,F12.10,T70,F12.10,
  & T85,F12.10)
58  FORMAT (I5,T10,F16.6,T30,F11.3,T46,F6.3,T59,F10.6)
56  FORMAT (F14.6,T20,F14.6,T40,F14.6,T60,F6.2)
57  FORMAT (F10.3,T15,F10.3,T30,F10.3,T46,F6.3,T59,F10.6)
59  FORMAT (I5,T10,F16.4,T30,F16.4,T45,F16.4,T62,F8.6)
69  FORMAT (I7,T10,E16.9,T30,F16.10,T45,F16.4,T62,F8.6)
32  FORMAT (F14.2,T20,F14.2)
121 Format (600(1x,f6.2))
  M1=6+NFILES/INCR
  M2=M1+NFILES/INCR+1
  M3=M1-1
  M4=M2+NFILES/INCR+1
  NUMCAP=0

IF (N.EQ.2) THEN
  DO I2=1,NP
    WRITE(UNIT=M1+NP,FMT=56) X(I2),Y(I2),Z(I2),RAD
  END DO
ENDIF

IF (N.EQ.3) THEN
  DO I6=YSPC,-YSPC, -1

```

```

      write (UNIT=9998,FMT=121) (CG(i,I6),i=-XSPC,XSPC)
    END DO
  END IF

```

```

      IF (NPRINT.EQ.1) THEN

```

```

***** colcnt

```

```

      WRITE (UNIT=M1,FMT=32) COLCNT !,COLCNTZ

```

```

***** LINEAR DENSITY

```

```

      IF (NDIRECT.EQ.0) THEN

```

```

        LARGE=YCAP

```

```

        LOW =-YSPC

```

```

      ELSE

```

```

        LARGE=XSPC

```

```

        LOW=-XSPC

```

```

      ENDIF

```

```

      FIRST=LOW+5

```

```

      LAST=LARGE-5

```

```

      DO C=FIRST,LAST,10

```

```

        DO F=-5,5

```

```

          E=C+F

```

```

          DENCELL1AVG(C)=DENCELL1AVG(C)+DENCELL1(E)

```

```

        DENCELLAVG(C)=DENCELLAVG(C)+DENCELL(E)

```

```

      ENDDO

```

```

      DENCELLAVG(C)=DENCELLAVG(C)/DENCELL1AVG(C)

```

```

      CMSQ=C*DG/10000

```

```

      IF (NDIRECT.EQ.0) THEN

```

```

        WRITE (UNIT=M1+K/INCR,FMT=58) C,CMSQ,DENCELL(C),DENCELLAVG(C)

```

```

      &      ,CG(0,C)

```

```

      ELSE

```

```

        WRITE (UNIT=M1+K/INCR,FMT=58) C,CMSQ,DENCELL(C),DENCELLAVG(C)

```

```

      &      ,CG(C,0)

```

```

      ENDIF

```

```

      ENDDO

```

```

      DO I3=LOW,LARGE

```

```

        DENCELLAVG(I3)=DENCELL(I3)/DENCELL1(I3)

```

```

      CMSQ=I3*DG

```

```

      IF (NDIRECT.EQ.0) THEN

```

```

        WRITE (UNIT=M1+K/INCR,FMT=57) CMSQ,DENCELL(I3),DENCELL1(I3),

```

```

      &      DENCELLAVG(I3),CG(0,I3)

```

```

      ELSE

```

```

        WRITE (UNIT=M1+K/INCR,FMT=57) CMSQ,DENCELL(I3),DENCELL1(I3),

```

```

      &      DENCELLAVG(I3),CG(I3,0)

```

```

      ENDIF

```

```

      ENDDO

```

```

      DO I4=LOW,LARGE

```

```

        DENCELLAVG(I4)=0

```

```

        DENCELL(I4)=0

```

```

        DENCELL1AVG(I4)=0

```

```

      ENDDO

```

```

***** radial cell density

```

```

      IF (NDIRECT.EQ.0) THEN

```



```

DO BI=(-YSPC),(YSPC)
  BI2=BI
  BI3=BI2*DG
  WRITE (UNIT=M2+K/INCR,FMT=69) BI3,RHO(BI),CG(0,BI2)
  RHO(BI)=0
END DO
ELSE
  DO BI=(-XSPC),(XSPC)
    BI2=BI
    BI3=BI2*DG
    WRITE (UNIT=M2+K/INCR,FMT=69) BI3,RHO(BI),CG(BI2,0)
    RHO(BI)=0
  END DO
ENDIF
***** CONCENTRATION AT LAST TIME STEP

IF (K.EQ.NTIME) THEN
  DO I6=(YSPC),-yspc, -1
    write (UNIT=M3,FMT=121) (CG(i,I6),i=(-Xspc),(Xspc))
  END DO
END IF

***** ORIENTATION

*      DO IBIN=0,179
*      WRITE (UNIT=M4+K/INCR,FMT=59) IBIN,PRN(IBIN) !,NEGORN(IBIN)
*      &      ,YO(IBIN+1)
*      PRN(IBIN)=0
*      NEGORN(IBIN)=0
*      END DO
ENDIF

IF (NGRAPH1.EQ.1) THEN
*   IF (K.EQ.NTIME) THEN
*   ! Change the title to reflect the changes in the data set
*     call faglName(CG, "conc plot", status)
*     print *, "Informing the viewer array data has been changed."
*     call faglUpdate(CG, status)
*   END IF
ENDIF

RETURN
END SUBROUTINE ROOTMEAN

end module CHANGE
*-----*
*
* VELN SUBROUTINE-----*
*
* This subroutine calculates the new cellular velocity components
* after a tumble.
*
SUBROUTINE VELN(VXN,VYN,VZN)

  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  REAL BIN(0:100)

```

```

COMMON/CONST/PI,PHI,THETA,NP,NTIME,V,DT,NMOTDIM,K
COMMON/ARRS/BIN
COMMON/VELS/VX,VY,VZ

```

```

      IF (NMOTDIM.EQ.2) THEN
        VXN=VX*COS(THETA)-VY*SIN(THETA)
        VYN=VX*SIN(THETA)+VY*COS(THETA)
      END IF

```

```

      IF (NMOTDIM.EQ.3) THEN
        SQR=(VX**2.0+VY**2.0)**(0.5)
        IF (SQR.EQ.REAL(0)) THEN
          VXN=V*COS(THETA)*SIN(PHI)
          VYN=V*SIN(THETA)*SIN(PHI)
          VZN=V*COS(PHI)
        ELSE
          VXN=VX*VZ*SIN(PHI)*COS(THETA)/SQR-V*VY*SIN(THETA)*SIN(PHI)/SQR
+   +VX*COS(phi)
          VYN=VY*VZ*SIN(PHI)*COS(THETA)/SQR+V*VX*SIN(THETA)*SIN(PHI)/SQR
+   +VY*COS(phi)
          VZN=-SQR*COS(THETA)*SIN(PHI)+VZ*COS(PHI)
        END IF
      END IF

```

```

      RETURN
      END

```

```

*-----*

```

```

* INITVEL SUBROUTINE-----*

```

```

*
* Initialize Velocity by generating a random phi and theta and calling
* VELN to generate velocity components.
*

```

```

      SUBROUTINE INITVEL(XVEL,YVEL,ZVEL,ISEED)
      PARAMETER (NPMAX=20000)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER I
      DOUBLE PRECISION XVEL(NPMAX),YVEL(NPMAX),ZVEL(NPMAX)
      REAL BIN(0:100)
      COMMON/CONST/PI,PHI,THETA,NP,NTIME,V,DT,NMOTDIM,K
      COMMON/ARRS/BIN
      COMMON/VELS/VX,VY,VZ

```

```

      DO I=1,NP
      IF (NMOTDIM.EQ.3) THEN
        PHI=ACOS(1-2*RAN1(ISEED))
        THETA=2*PI*RAN1(ISEED)
        VX=0
        VY=0
        VZ=V
        CALL VELN(VXN,VYN,VZN)
        XVEL(I)=VXN
        YVEL(I)=VYN
        ZVEL(I)=VZN
      END IF

```

```

        IF (NMOTDIM.EQ.2) THEN
            THETA1=2*PI*RAN1(ISEED)
            DES=(1-2*RAN1(ISEED))
            THETA=SIGN(THETA1,DES)
            VX=V*COS(THETA)
            VY=V*SIN(THETA)
            XVEL(I)=VX
            YVEL(I)=VY
        END IF
    END DO

    RETURN
    END

*-----*
*
* RAN1 FUNCTION-----*
*
* This function generates a random number between 0 and 1.
*
    FUNCTION RAN1(IDUM)
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    DIMENSION R(97)
    PARAMETER (M1=31104,IA1=625,IC1=6571)
    PARAMETER (M2=12960,IA2=1741,IC2=2731)
    PARAMETER (M3=14000,IA3=1541,IC3=2957)
    DATA IFF /0/

    RM1=1.0/M1
    RM2=1.0/M2

    IF (IDUM.LT.0.OR.IFF.EQ.0) THEN
        IFF=1
        IX1=MOD(IC1-IDUM,M1)
        IX1=MOD(IA1*IX1+IC1,M1)
        IX2=MOD(IX1,M2)
        IX1=MOD(IA1*IX1+IC1,M1)
        IX3=MOD(IX1,M3)
        DO 11 J=1,97
            IX1=MOD(IA1*IX1+IC1,M1)
            IX2=MOD(IA2*IX2+IC2,M2)
            R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1
11    CONTINUE
        IDUM=1
    ENDIF
    IX1=MOD(IA1*IX1+IC1,M1)
    IX2=MOD(IA2*IX2+IC2,M2)
    IX3=MOD(IA3*IX3+IC3,M3)
    J=1+(97*IX3)/M3
    IF(J.GT.97.OR.J.LT.1)PAUSE
    RAN1=R(J)
    R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1
    RETURN
    END
*-----*
*

```

```

* TURNANG FUNCTION-----*
*
*   This function calculates the phi turn angle when a cell tumbles. *

FUNCTION TUNANG(ISEED)
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  DOUBLE PRECISION AVGTA, SUMTA
  REAL BIN(0:100)

  COMMON/CONST/PI,PHI,THETA,NP,NTIME,V,DT,NMOTDIM,K
  COMMON/ARRS/BIN
  COMMON/VELS/VX,VY,VZ
  common/info/avgta,AVGCOS,PT,SEED1
  WW=RAN1(ISEED)*100
  YY=INT(WW)
  ZZ=WW-YY
  TUNANG=(ZZ*BIN(YY)+(1-ZZ)*BIN(YY+1))*PI/180
  sumta=sumta+tunang*(180/PI)
  numb=numb+1
  avgta=sumta/numb
  SUMCOS=SUMCOS+COS(TUNANG)
  AVGCOS=SUMCOS/NUMB
  RETURN
END
*-----*

```

function erf(X)

```

c-----
c Copyright by C. Pozrikidis, 1999
c All rights reserved.
c This program is to be used only under the
c stipulations of the licencing agreement.
c-----
c-----
c Error function
c-----
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  erf = 1.0-erfc(X)
  Return
  End

```

```

=====
function erfc(X) ! complementary error function
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)

```

```

c-----
c Complementary Error function
c-----
  T = 1.0/(1.0 + 0.3275911*ABS(X))
  erfc = T*EXP(-(X)**2)*(0.254829592 +T*(-0.284496736 +
+      T*( 1.421413741 +T*(-1.453152027 +T*1.061405429))))
  If(X.lt.0.) erfc = 2.0-erfc
  Return
  End

```

SUBROUTINE BOXSUB(xspc,yspc,DG)

```

PARAMETER (NXMAX=1000, NYMAX=1000)
integer seed, input, cord, xnew, ynew, x, y, i, sumnew
integer xspc, yspc, length, count
integer xleftbound, xrightbound
integer porecode(-NXMAX:NXMAX,-NYMAX:NYMAX)
integer boxcorner(-NXMAX:NXMAX,-NYMAX:NYMAX)
double precision prob, fraction, DG, fraction2
REAL PI
COMMON/PORES/PORECODE, BOXCORNER
COMMON/POROUS/FRACTION,XLEFTBOUND,XRIGHTBOUND,fraction2
*****

open (unit = 100, file = 'cord.txt')
  open (unit = 200, file = 'grid.txt')
  open (unit = 300, file = 'particle.txt')
open (unit = 400, file = 'particle1.txt')

input=xspc
PI=4.0*ATAN(1.0)
length=2*input+1
seed = 99999
seed = -235865
seed = -843297
seed = -287681
*****

*****

count = 0
1 format (I4, I4, 1X, F10.8, I2, I2, I2)
2 format (A)
3 format (f10.4,4x,f10.4)
*****

xleftbound= -input + 2
* xleftbound= 0
* xrightbound= xspc /4
xrightbound= xspc - 4

do ynew = -yspc + 2, yspc - 4
  y = -ynew + (yspc+1)
*****

*
* boxcorner - overlap
* porecode - diffusion
*
* porecode numbers:
* #2 - bottom center
* #4 - left center
* #5 - center
* #6 - right center
* #8 - top center
*
*****

do xnew = xleftbound, xrightbound
  x = xnew + (input+1)
*****
pick how many rows for solid wall on multislit simulations
nblock =0

```

```

if (nblock.eq.1) then
  IF (MOD(ynew,10).EQ.0) THEN
    boxcorner(xnew, ynew) = 3
    if (xnew.ne.xleftbound) then
      porecode(xnew, ynew) = 2
      porecode(xnew, ynew+1) = 8
    endif
    write (400, 3) (real(xnew)+.5,real(ynew)+.5)
    count3 = count3 + 1
  end if
elseif (nblock.eq.2) then
  IF (MOD(ynew,3).EQ.0) THEN
    boxcorner(xnew, ynew) = 3
    boxcorner(xnew+1, ynew+1) = 3
    if (xnew.eq.xleftbound) then
      porecode(xnew, ynew + 1) = 4
    else if (xnew.eq.Xrightbound) then
      porecode(xnew, ynew + 1) = 6
    else
      porecode(xnew, ynew) = 2
      porecode(xnew, ynew+2) = 8
      porecode(xnew, ynew + 1) = 5
    endif
    write (400, 3) (real(xnew),real(ynew))
    write (400, 3) (real(xnew+1),real(ynew+1))
    count3 = count3 + 2
  end if
elseif (nblock.eq.3) then
  IF (MOD(ynew,4).EQ.0) THEN
    boxcorner(xnew, ynew) = 3
    boxcorner(xnew+1, ynew+1) = 3
    boxcorner(xnew+2, ynew+2) = 3
    if (xnew.eq.xleftbound) then
      porecode(xnew, ynew + 1) = 4
    else if (xnew.eq.Xrightbound) then
      porecode(xnew, ynew + 1) = 6
    else
      porecode(xnew, ynew) = 2
      porecode(xnew, ynew+3) = 8
      porecode(xnew, ynew + 1) = 5
      porecode(xnew, ynew + 2) = 5
    endif
    write (400, 3) (real(xnew),real(ynew))
    write (400, 3) (real(xnew+1),real(ynew+1))
    write (400, 3) (real(xnew+2),real(ynew+2))
    count3 = count3 + 3
  end if
else
  sumnew=sqrt(real(xnew+1)**2 + real(ynew+1)**2)
  if ((sumnew*DG).lt.501.and.boxcorner(xnew, ynew).lt.1)
&      then
    else if (boxcorner(xnew, ynew) .gt. 1) then
      else
        prob = ran(seed)

```

```

**PROB OF ZERO
*       If (prob .le. 0.93) then

*****
        prob2=0.42
        prob3=0.42
*****

        If (prob .le. PROB2) then
        boxcorner(xnew, ynew) = 0

**PROB OF ONE
        else if (prob .gt. prob2 .and. prob
&               .le. prob3) then

***** 1 for circle  3 for block
        boxcorner(xnew, ynew) = 3
        count1 = count1 + 1

*       write (300, 3) (real(xnew)+.5,real(ynew)+.5)
        write (400, 3) (real(xnew)+.5,real(ynew)+.5)
**PROB OF TWO
        else if (prob .gt. 1.8 .and. prob
&               .le. .90) then
        IF (boxcorner(XNEW-2,YNEW).EQ.41.OR.boxcorner(XNEW+1,YNEW-1)
&          .EQ.41.OR.boxcorner(XNEW,YNEW-1).EQ.2.OR.boxcorner
&          (xnew, ynew) .ge. 1.OR.boxcorner(XNEW,YNEW-2).EQ.41) THEN

            ELSE
            boxcorner(xnew, ynew) = 21
            boxcorner(xnew + 1, ynew) = 22
            porecode(xnew + 1, ynew) = 2
            porecode(xnew + 1, ynew + 1) = 8
            count2 = count2 + 1
            write (300, 3) (real(xnew)+.5,real(ynew)+.5)
            write (300, 3) (real(xnew)+1.5,real(ynew)+.5)
        END IF
**PROB OF FOUR
*       else if (prob .gt. 0.93) then
        else if (prob .gt. PROB3) then

            IF (boxcorner(XNEW+1,YNEW-1).EQ.41.OR.boxcorner(XNEW,YNEW-1)
&             .EQ.2.OR.boxcorner(xnew, ynew) .ge. 1) THEN
            ELSE
***** use for ordered particles
*       IF ((MOD(ynew+YSPC+1,2).EQ.0.AND.MOD(Xnew+XSPC,2).EQ.0).OR.
* & (MOD(ynew+YSPC+0,2).EQ.0.AND.MOD(Xnew+XSPC+0,2).EQ.0))THEN
*       IF ((MOD(ynew+YSPC+1,4).EQ.0.AND.MOD(Xnew+XSPC+1,4).EQ.0).OR.
* & (MOD(ynew+YSPC+3,4).EQ.0.AND.MOD(Xnew+XSPC+3,4).EQ.0))THEN
*       porebox(xnew, ynew) = 4
            boxcorner(xnew, ynew) = 41
            boxcorner(xnew + 1, ynew) = 42
            boxcorner(xnew, ynew + 1) = 43
            boxcorner(xnew + 1, ynew + 1) = 44
            if (porecode(xnew + 1, ynew).eq.8) then

```

```

        porecode(xnew + 1, ynew) = 20
        else
            porecode(xnew + 1, ynew) = 2
        endif
        if (porecode(xnew, ynew + 1).eq.6) then
            porecode(xnew, ynew + 1) = 40
        else
            porecode(xnew, ynew + 1) = 4
        endif
        porecode(xnew + 1, ynew + 1) = 5
        porecode(xnew + 2, ynew + 1) = 6
        porecode(xnew + 1, ynew + 2) = 8
        count4 = count4 + 1
        write (300, 3) (real(xnew)+1,real(ynew)+1)
    ENDIF
*   END IF
        else
            end if
        end if
    End If
    end do
end do
*****
*
*   cord printout
    do ynew = -yspc + 2, yspc - 4
    do xnew = -xspc + 2, xspc - 4
        write (100, 1) xnew, ynew, prob, ! porebox(xnew,ynew),
&        boxcorner(xnew, ynew), porecode(xnew, ynew)
    end do
end do
*****
**
*****
***
*   fraction = 100.00 * (count3+count1*PI*(.5)**2+COUNT2*2*(1)**2
    fraction = 100.00 * (count3+count1+COUNT2*2*(1)**2
& +COUNT4*PI*(1)**2)/((xrightbound-xleftbound+2)*(2*yspc - 4))
    fraction2=100*(2*count1+COUNT4*4/3*PI*(1)**3)/
& ((xrightbound-xleftbound+2)*(2*yspc - 4)*2)
    print 400, fraction
    print 400, fraction2
400    format ('% solids = ', F10.6, '%')
    close(unit=100)
    close(unit=200)
    close(unit=300)
    close(unit=400)
    return
end

```


APPENDIX B

Instructions on how to use the chemotaxis simulation:

The simulation code consists of the three files `main.for`, `chemovidgraphpore2.f`, and `boxsub3.f`. Single cell parameters and system initial conditions are read in from an input file called `cellvidpore.prn`. The names for the output files are in the `names.prn` file. The normalized accumulative probabilities at a hundred equally spaced angles for the turn angle distribution are in the input file `points.prn`. Other turn angle distributions can be used for polar bacteria or gaussian distributions by changing the name in the `read` statement in the `main.for` program.

The boundary conditions, dimensionality, choice of cell initial location, initial concentration distribution, diffusion, and output options are chosen by defining switches at the beginning of the `main.for` program. The addition of an alternate temporal sensing method can be specified in the `gradient` subroutine.

Uncommenting the call to the `boxsub` subroutine will include particles in the simulation for cells to swim around. The amount of time for a tumble can be specified independent of the time step.

APPENDIX C

Code for matlab:

```
clear all
echo off
clc
[a, b] = textread('C:\chemotaxis\allcells.prm', '%f %f');
[d, e] = textread('C:\chemotaxis\particle.txt', '%f %f');
Conc = load('C:\chemotaxis\allconc.prm');

xlength=7000
dg=200;
xspc=xlength/dg;
numframes=40;
numcells=10000;
al = xspc*2;
alp1=al+1;

axis image;
set(gca,'XGrid','on')
set(gca,'YGrid','on')
contourf(Conc(alp1:-1:1, 1:alp1), 20);
axis image;
colormap(.5*cool + .5*jet);
shading flat;
q = xlabel('x position');
w = ylabel('y position');
title('cell position');
    plot(d+al/2+1,e+al/2+1, 'go', 'Markersize', 3,'Markerfacecolor', 'g');
axis image;
colormap(.5*cool + .5*jet);
shading flat;
fig1 = figure(1);
xlim([1 alp1]);
ylim([1 alp1]);
winsize = get(fig1, 'Position');
winsize(1:2) = [1 1];

A = moviein(numframes, fig1, winsize);
axis image
k = 1;
for t = 1:numframes
    for i = 1:numcells
        x(i, t) = a(k)/dg + xspc+1;
        y(i, t) = b(k)/dg + xspc+1;
```

```

        k = k + 1;
    end
end
z = 1;
for t = 1:numframes
    time=t
    contourf(Conc(z+al:-1:z, 1:alp1),20);
    axis image
    grid on;
    shading flat
    hold on
    set(gca,'XGrid','on')
    set(gca,'YGrid','on')
    plot(d+al/2+1,e+al/2+1, 'go', 'Markersize', 3,'Markerfacecolor', 'g');
    p = plot(x(1:numcells, t), y(1:numcells, t), 'ko', 'Markersize', 1.5,'Markerfacecolor',
    'k');
    set(gca,'XGrid','on')
    set(gca,'YGrid','on')
    grid on;
    A(:, t)=getframe(fig1, winsize);
    grid on;
    hold off
    grid on
    xlim([1 alp1]);
    ylim([1 alp1]);
    drawnow
    z = z+alp1;
end
movie(fig1, A, 1, 10, winsize)
save cellsim.mat A;
mpgwrite(A, jet,'c:\simulation\hetpores\4002d.mpg');

```

REFERENCES

<http://water.usgs.gov/wid/html/bioremed.html>, February, 2000.

Adler, J., and B. Tempelton. 1967. The Effect of Environmental Conditions on the Motility of *Escherichia coli*. *Journal of General Microbiology*. 46:175-184.

Adler, J. 1973. 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.

Alt, W. 1980. Biased Random Walk Models for Chemotaxis and Related Diffusion Approximations. *J. Math. Biology*. 9:147-177.

Barton, J.W., R.M. Ford. 1995. Determination of Effective Transport Coefficients for Bacterial Migration in Sand Columns. *American Society for Microbiology*. 91(9):3329-3335.

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

Berg, H.C. 1983. Random Walks in Biology. Princeton University Press, New Jersey.

Berg, H.C., and L. Turner. 1990. Chemotaxis of bacteria in glass capillary arrays. *Biophysical Journal*. 58:919-930.

Biondi, S. A., J. A. Quinn, and H. Goldfine. 1998. Random Motility of Swimming Bacteria in Restricted Geometries. *AIChE Journal*. 44:1923-1929.

Bornbusch, A., and W. Conner. 1986. Effects of Self-steered Turn Size and Turn Bias Upon Simulated Chemoklinotactic Behavior. *J. Theor. Biol.* 122:7-18.

Brosilow, B. J., R. M. Ford, S. Sarman, and P. T. Cummings. 1996. Numerical Solution of Transport Equations for Bacterial Chemotaxis: Effect of Discretization of Directional Motion. *SIAM J. Appl. Math.* 56:6:1639-1663.

Brown, D. A., and H. C. Berg. 1974. Temporal Stimulation of Chemotaxis in *Escherichia coli*. *Proc. Nat. Acad. Sci. USA*. 71:1388-1392.

Chang CC, K. Merritt. 1994. Infection at the site of Implanted Materials with and without Pre adhered Bacteria. *Journal of Orthopaedic Research*. 12, 526-531.

Criddle, C, M Dybas, D Hyndman, D Wiggert, et al. 1997. The Schoolcraft field bioaugmentation experiment: Evaluation of in-situ bioaugmentation to remediated an aquifer contaminated with carbon tetrachloride. *State of Michigan Department of Environmental Quality Technical report no. 4* (Contract # Y40386).

Dillon, R., L. Fauci and D. Gaver III. 1995. A Microscale Model of Bacterial Swimming, Chemotaxis and Substrate Transport. *J. theor. Biol.* 177:325-340.

Dillon, R., L. Fauci. 2000. A Microscale Model of Bacterial and Biofilm Dynamics in Porous Media. *Biotechnol Bioeng.* 68:536-547.

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.

Duffy, K. J., and R. M. Ford. 1997. Turn Angle and Run Time Distributions Characterize Swimming Behavior for *Pseudomonas putida*. *Journal of Bacteriology.* 179:1428-1430.

Emerson D., M. Worden, J. Breznak. 1994. A Diffusion Gradient Chamber for Studying Microbial Behavior and Separating Microorganisms. *Applied and Environmental Microbiology.* 60:1269-1278.

EPA bulletin. Lammars Barrel Factory Superfund Site, Bevercreek OH. August 1997.

Ford, R. M., and D. A. Lauffenburger. 1991. Measurement of Bacterial Random Motility and Chemotaxis Coefficients: II. Application of Single-Cell-Based Mathematical Model. *Biotechnol Bioeng.* 37:661-672.

Ford, R. 1992. Mathematical Modeling and Quantitative Characterization of Bacterial Motility and Chemotaxis. *In* Modeling the Metabolic and Physiologic Activities of Microorganisms. John Wiley & Sons, Inc, New York.

Frymier, P., R. Ford and P. Cummings. 1994. Analysis of Bacterial Migration: I. Numerical Solution of the Balance Equation. *AIChE Journal.* 40:704-715.

Frymier, P., R. Ford and P. Cummings. 1993. Cellular Dynamics Simulations of Bacterial Chemotaxis. *Chemical Engineering Science.* 48:687-699

Gestwicki, J.E., L.L. Kiessling. 2002. Inter-receptor communication through arrays of bacterial chemoreceptors. *Nature.* 415:81-84.

Gristina AG, W. Costerton. 1985. Bacterial Adherence to Biomaterials and Tissue: The Significance of Its Role in Clinical Sepsis. *Journal of Bone and Joint Surgery.* 67-A,2, 264-273.

Gristina AG, RA Jennings, PT Naylor, ON Nyvik, and LX Webb. 1989. Comparative in vitro antibiotic resistance of surface colonizing coagulase negative staphylococci. *Antimicrob Agents Chemother.* 33, 813-816.

Hamilton A. 1987. Biofilms: Microbial Interactions and Metabolic Activities. *Symp. Soc. Gen. Microbiol.*,41 361-385.

- Jacoby M. 2001. Custom-made biomaterials. *Science & Technology*. 79(6):30-35
- Liu, Z. 1996. A Method for Measuring Bacterial Chemotaxis Parameters in a Microcapillary. *Biotechnol Bioeng*. 51:120-125.
- Macnab, R. M., and E. Koshland, Jr. 1972. The Gradient-Sensing Mechanism in Bacterial Chemotaxis. *Proc. Nat. Acad. Sci. USA*. 69:2509-2512.
- Madigan, M. T., J. M. Martinko, j. Parker. 2000. Brock Biology of Microorganisms. Prentice Hall, New Jersey.
- Marx, R. B., and M. D. Aitken. 2000. A Material-Balance Approach for Modeling Bacterial Chemotaxis to a Consumable Substrate in the Capillary Assay. *Biotechnol Bioeng*. 68:308-315.
- Mesibov, R., G. Ordal, and J. Adler. 1973. The Range of Attractant Concentrations for Bacterial Chemotaxis and the Threshold and Size of Response over this Range. *Journal of General Physiology*. 62:203-223.
- Newton, B.J, W.M Jerrell. 1999. A Procedure to estimate the Response of Aquatic Systems to Changes in Phosphorus and Nitrogen Inputs. *USDA National Water and Climate Center*.
- Nikata T., K. Sumida, J. Kato, and H. Ohtake. 1992. Rapid Method for Analyzing Bacterial Behavioral Responses to Chemical Stimuli. *Applied and Environmental Microbiology*. 58:2250-2254.
- Penny, A. S, and D. Haldeman. 1997. *The Microbiology of the Terrestrial Deep Subsurface*. Lewis Publishers: New York.
- Ratner B, A. Soffman, F. Schoen, J. Lemons. 1996. Biomaterials Science, An Introduction to Materials in Medicine. Academic Press, San Diego, CA, p. 206.
- Rivero-Hudec, M. and D. A. Lauffenburger. 1986. Quantification of Bacterial Chemotaxis by Measurement of Model Parameters Using the Capillary Assay. *Biotechnol Bioeng*. 28:1178-1190.
- 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.
- Serban P. F., M. T. Widman, M.R. Worden. 1998. In situ mapping of community-level cellular response with catalytic microbiosensors. *Biosensors & Bioelectronics*. 13:1197-1203.

- Slonczewski J, Macnab R, Alger J, Castle A. 1982. Effects of pH and Repellent Tactic Stimuli on Protein Methylation Levels in *Escherichia coli*. *Journal of Bacteriology*. 152, 384-399, 1982.
- Strauss, I., P. Frymier, C. Hahn and R. Ford. 1995. Analysis of Bacterial Migration: II. Studies with Multiple Attractant Gradients. *AIChE Journal*. 41:402-414.
- Strickler D, R. McLean. 1995. Biomaterials associated infections: The Scale of the Problem. *Cells and Materials*, 5, 2, 167-182.
- Tatara, G; M. Dybas, C. Criddle. 1993. Effects of Medium and Trace Metals on Kinetics of Carbon Tetrachloride Transformation by *Pseudomonas* sp. Strain KC. *Applied and Environmental Microbiology*. 2126-2131.
- Tso, W., and J. Adler. 1974. Negative Chemotaxis in *Escherichia coli*. *Journal of Bacteriology*. 118:560-576.
- Van Der Drift C, J. Duiverman, H. Bexkens, A. Krijnen. 1975. Chemotaxis of a Motile *Streptococcus* toward Sugars and Amino Acids. *Journal of Bacteriology*. 124, 3, 1142-1147.
- Vickers, M. G. 1981. Ideal and Non-ideal Concentration Gradient Propagation in Chemotaxis Studies. *Exp Cell Res*. 136:91-100.
- Weinberg ED. Iron and Susceptibility to infectious disease. 1974. *Science*. 184, 952-956, 1974.
- Widman, M., D. Emerson, C. Chiu, M. Worden. 1997. Modeling Microbial Chemotaxis in a Diffusion Gradient Chamber. *Biotechnol. Bioeng*. 55:191-205.
- Widman, M. 1997. Engineering Applications of Microbial Chemotaxis. PhD Dissertation Michigan State University.
- Witt, M. E., M. J. Dybas, M. R. Worden, C. S. Criddle. 1999. Motility-Enhanced Bioremediation of Carbon Tetrachloride-Contaminated Aquifer Sediments. *Environ. Sci. Technol*. 33:2958-2964.

MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 02334 8729