CHARACTERIZATION OF A PASSIVE DIFFUSION MICRODEVICE FOR ASSAYS OF CHEMOTAXIS AND MORPHOGENESIS.

By

Ipshita Chakraborty

Thesis

Submitted to the Faculty of the Graduate
School of Vanderbilt University in partial
fulfillment of the requirements for the
degree of

MASTER OF SCIENCE

in

Biomedical Engineering

August, 2007

Nashville, Tennessee

Approved:

John P. Wikswo

Robert J. Roselli

ACKNOWLEDGEMENTS

I would like to thank Dr. John Wikswo, my supervisor, for his able guidance and supervision of my research and his encouragement to exceed the achievable. I am extremely thankful to Philip Samson who originally fabricated the diffusion gradient microdevice and provided me with constructive suggestions in the initial period of training. Without his constant inputs and guidance, this thesis would not have been possible.

A great deal of gratitude goes to Dr. Narcisse A. N'Dri without whose constant inputs and troubleshooting expertise this project would not have reached completion. I also take this opportunity to thank Dr. Momchil Velkovsky who gave several helpful inputs regarding the mathematical portions of this thesis.

A big thank you goes to my parents for giving me the independence to choose my own path in life and from whom I have learnt several important lessons. I would also like to thank my friends in 1A and 1B who made my time at Nashville and Vanderbilt highly enjoyable. A special thank you goes to Saikat Sengupta for providing me with infrastructure to perform my computational experiments.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

I.1 Chemotaxis and Morphogenesis

Cell migration is a process that plays a central role in a wide variety of biological and physiological phenomena. It is an essential component of three pivotal biological processes—embryogenesis, wound healing and metastasis. The advent of bioengineering has also made cell migration crucial for technological applications such as tissue engineering where it assists in the colonization of biomaterials scaffolding. Cell migration has both physical and chemical dimensions underlying its occurrence. Therefore, in order to understand it as an integrated process, it is essential to have a thorough understanding of the physical and chemical properties of the multicomponent assemblies that are involved. Cell migration has been described as a 'physically integrated chemical process' [1] and the rationale of this statement is evident by the fact that it is physically coordinated both spatially and temporally by a number of chemical gradients. The studies on cell migration since the past decade have all aimed towards gaining a detailed perspective on the physicochemical nature of cell migration.

Cell migration can be influenced by a range of factors from sunlight to complex gradients of biologically significant molecules. Chemotaxis, one of the types of cell migration, is characterized by the directed motion of cells in response to external chemical gradients of soluble molecules called chemoattractants. Cells can also be repelled by biological substances in which case they move away from the direction of the concentration gradient. Chemotaxis is a pivotal part of processes such as the movement of macrophages and neutrophils during wound healing, the migration of neural crest cells, the movement of endothelial cells in the early stages of cancer and metastasis.

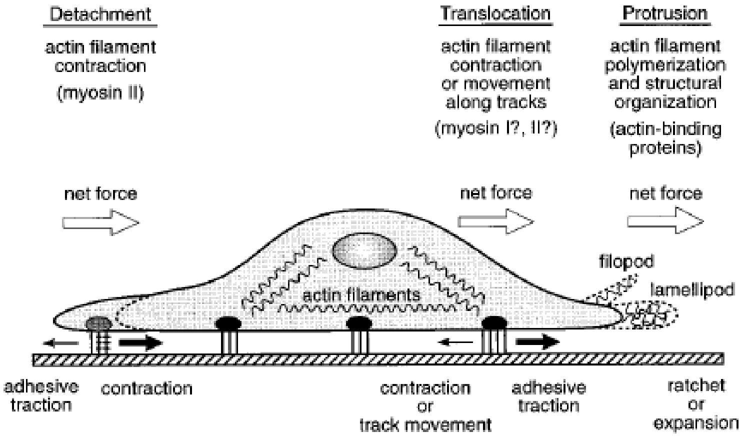


Figure 1: Illustration of the basic process of cell migration [2]

The orientation of the direction of motion is achieved by the preference of the pseudopod extension towards or away from the higher chemoattractant concentration. The pseudopod is the result of the morphological polarization and membrane extension of a migrating cell [3].

Wound healing is a fundamental process that is directed and managed through chemotaxis. The formation of a wound initiates a cascade of events such as inflammation, new tissue formation and tissue remodeling which lead to partial reconstruction of the wounded area. The first step in wound healing is the formation of a clot which is initiated by the vasoconstriction of the injured blood vessel and the activation of the intrinsic coagulation cascade by the endothelial cells and platelets [4]. The clot that forms is made of collagen, platelets, thrombin and fibronectin and these factors release cytokines and growth factors that initiate the inflammatory response. This stage marks the advent of chemotactic processes in wound healing. Immediately as the clot is formed cellular signals are generated to bring about the migration of neutrophils to the wound site. This migratory movement occurs due to the responses of neutrophils to gradients of Interleukin (IL)-1, tumor necrosis factor (TNF)-α, transforming growth factor (TGF)-β, PF4 and bacterial products [5]. Macrophages are also attracted to the wound site due to the release of the same chemoattractants. Endothelial cells also help to bring macrophages and fibroblasts to the wound by the synthesis of IL-1 which, in turn, stimulates macrophages and neutrophils to express more of their own chemoattractant cytokines and growth factors [6]. Macrophages release the chemoattractants platelet-derived growth factor (PDGF), TNF-α, IL-6, granulocyte colony stimulating factor (G-CSF) and granulocyte macrophage colony stimulating factor (GM-CSF) to recruit more fibroblasts and macrophages [7]. Fibroblasts then release IFN-γ which induces monocytes to turn into macrophages. The process of wound healing can be summarized in this way as a cascade of chemotactic events where the lymphocytes are the cells involved in the response and the molecular gradients are formed by the cytokines and growth factors.

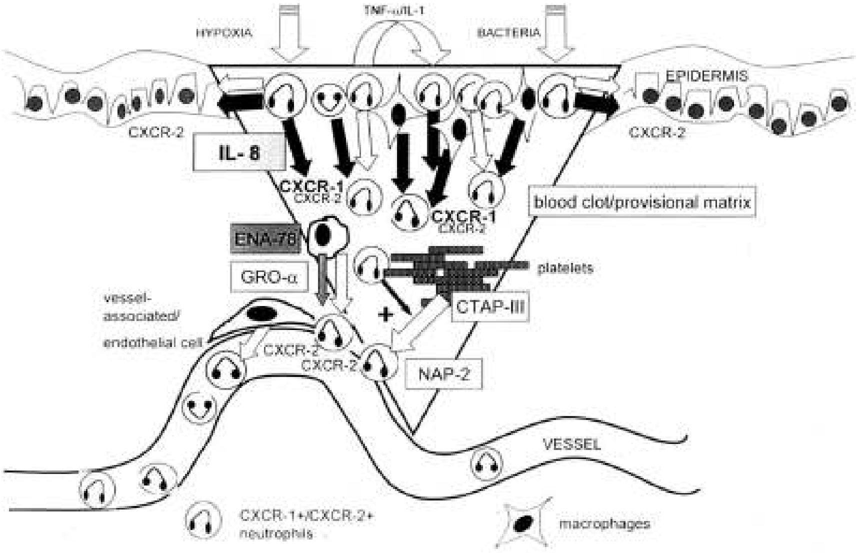


Figure 2: An illustration of neutrophils during human epithelial wound healing [2]

Chemotactic cell migration is thought to play an integral part at several stages during cancer. The motility of invasive cancer cells is determined by the surrounding microenvironment. Subtle changes in the cancer cell's interaction with the extracellular matrix as well as growth factors and cytokines define whether a cell becomes invasive or remains stationary in the tumor mass. The motility of cancer cells is being studied through the intravital imaging of cells, at single cell resolution, inside tumors within living animals. Protein expression analysis of macrophages and tumor cells invading jointly has revealed how these cells are attracted to each other and how they play synergistic roles during cancer invasion [8]. Tumor cells express CSF-1, which stimulates macrophage chemotaxis, whereas macrophages express epidermal growth factor (EGF), which stimulates tumor cell chemotaxis [9]. Several motility factors, both tumor and host-derived, as well as growth factors such as basic fibroblast growth factor (bFGF), TGF-α, epidermal growth factor (EGF), PDGF and hepatocyte growth factor (HGF) can induce chemotactic responses in tumor cells and endothelial cells [10]. Angiogenic factors such as VEGF are also potent chemoattractors for endothelial cells and for certain tumor cells that express VEGF receptors [11]. The second groups of key players in cancer cell invasion are the chemokines. Chemokines are a family of over 40 small (8 kDa) related proteins with the function of moving cells along a chemotactic gradient, either to organize cells within an organ or to facilitate the movement of leucocytes around the body. In the case of metastasis, a strong association is observed between the soluble chemokines and the cell surface, especially to sugar residues [12]. Experiments using transwells have suggested that target tumor cells move towards increasing concentrations of the chemokine but it is still difficult to provide an explanation as to how an effective gradient can be maintained within a fast-flowing blood vessel or at a significant distance from the chemokine source in conditions of no flow [13]. The current state of cancer research suggests that several of these signaling pathways such as those involving PDGF and EGF are being uncovered. However, the complexity of interaction of gradients and the behavior of cells in the vicinity of these complex gradients indicate that additional research will be required to fully understand biological responses.

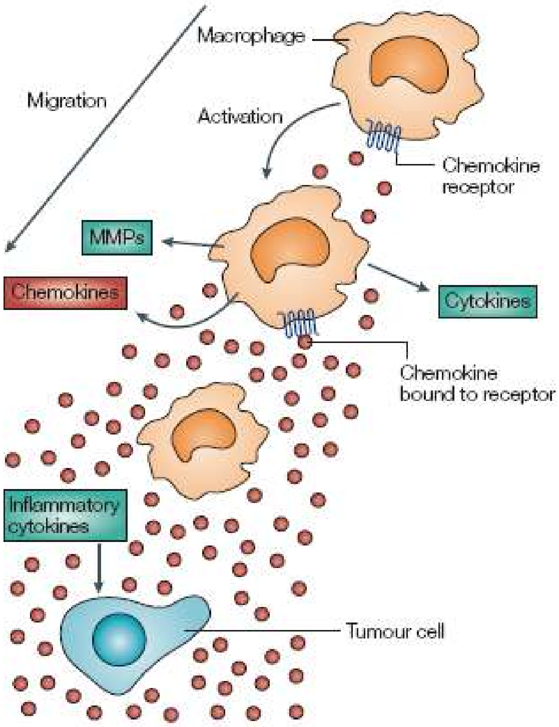


Figure 3: Illustration of a chemokine gradient in cancer accompanied by macrophage migration [14]

The development of the vertebrate embryo also results from a series of complex and highly interconnected cell migration processes, giving rise to the difference between the fate map of the blastoderm and the definitive location of different tissues. This extensive reorganization of the basic regions of the early embryo is a result of morphogenetic movements, including individual cell migration in response to molecular gradients and cell sheet spreading during gastrulation [15]. The formation of the neural crest in the embryo and its subsequent development into the craniofacial and enteric nervous systems is governed by several chemotactic processes involving a complex array of molecular cues which form gradients at different levels [16]. The fibroblast growth factor FGF and bone morphogenetic protein (BMP) signaling systems [17], for instance, play numerous roles in the development of the cranial neural crest, being involved in the migration, survival, patterning and specification of several neural crest domains. FGF-8 and 2 have been demonstrated to produce a chemotactic response in cranial neural crest cells *in vitro* and FGF-2 can produce a chemotactic response *in vivo* [18]. The migratory and proliferative behavior of primordial germ cells has also been studied extensively and factors such as TGF-β and the Steel/c-kit signaling pathway have been implicated in the chemotactic responses and motility of these cells [19]. However, a clear and definitive picture of the factors controlling the movement of primordial germ cells in the living embryo is still lacking.

In all processes where chemotaxis and cell migration are involved, the molecular components are being identified at a very rapid rate compared to the rate at which their mechanism of action in migration is being uncovered. The study of cell migration and chemotaxis as processes in which all these components participate as an integrated, dynamic system is the goal of all current research on these topics and given this goal, it is important to have devices that allow quantitative measurements of cellular responses to chemotactic gradients.

I.2 Chemotaxis devices

A great deal of information on chemotaxis has been accumulated in the past 45 years about the molecular components involved but there is not yet sufficient information of the effect of these molecular players on cells. The major limitation of chemotaxis studies has been in terms of generating gradients in a controlled manner and maintaining them since cells need to sense changes in concentration gradients to direct their motion. The earliest methods of creating gradients of chemoattractants utilized collagen, fibrin or agarose gels [19]. These methods relied on the diffusion of the chemokine through the gel. Recent methods are not based on gels and include the Zigmond Chamber [20], the Dunn Chamber [21] and an optical chemotaxis assay system [22]. The Zigmond and Dunn chambers work on the principle of diffusion over a short distance between two fluid-filled chambers. However, these methods do not allow the formation of gradients of different shapes and are also limited in their ability to maintain the gradients over long periods of time. The generation of both complex and linear biologically active gradients is required to study the role of such gradients and the correlation between their shapes and the behavior of cells in them. It is also essential to be able to maintain these concentration gradients at conditions close to steady-state.

Microfluidic devices are ideal for the creation of such controlled chemotactic gradients. Microfabrication methods allow the creation of arbitrary microchannel designs through which cells can crawl. A unique feature of fluids in microfluidic devices is laminar flow which can be attributed to the microscale dimensions of such devices. In laminar flows, mixing takes place only through diffusion and there is no convection component to consider, unlike in macroscale flows. This makes the process of gradient formation simple and provides ease of calculation while mathematically characterizing the gradients in the device. The class of laminar flow micromixers that utilize this property of laminar flow to pattern surface and solution properties at the microscale was developed by Jeon et al. [23]. These devices have been fabricated for the study of chemotaxis and can create simple, controlled and linear gradients—their goal is to not mix the streams fully and maintain heterogeneous gradients of the desired shape and magnitude. The device consists of two main parts, one of which is the gradient-generating portion and the other is the observation portion. Pyramidal branched arrays of microchannels that split, combine and mix fluid streams form the gradient-generating portion. Each microchannel contains a different portion of the chemoattractant and these are recombined in the main channel such that the concentration gradient is perpendicular to the direction of flow and the gradient is maintained throughout the length. The observation portion of the device consists of the main channel where the cells are placed. This device can be used to manipulate the microenvironment required by cells for motility. The underlying principle for this device is the use of low Reynolds numbers to eliminate the possibility of any inertial movements such as advection that can arise in diffusive flow. The gradient can be maintained at steady-state and cell migration can be observed and recorded in real time which enables the user to determine the chemotaxis coefficient without prior knowledge of parameters such as the variation of motility with chemoattractant concentration. The potential problem with this device is gradient smearing due to the diffusion of chemoattractants at the device surface when streams of different compositions merge at the end. Dertinger et al. [24] extended Jeon's pyramidal scheme by combining multiple pyramidal networks in parallel, generating a broad range of concentration profiles using relatively fewer inlets. They demonstrated that increasing the number of inlets on each pyramidal network from two to three provides a greater range of concentration profiles. Lin et al. [25] modified the pyramidal model by creating both linear and non-linear gradients and controlling them using different flow rates for the inlets.

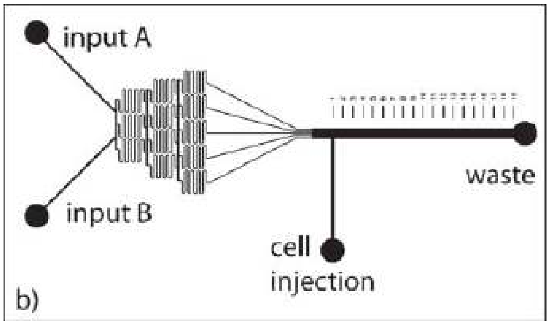


Figure 4: Passive diffusion device fabricated by Jeon et al. [23]

They achieved this by the integration of a separate mixing module with the gradient-generating and observation network.

Wikswo et al. have addressed the above problem in their experiments conducted with the device. In the experiments performed by Walker et al. at the Vanderbilt Institute for Integrative Biosystems Research and Education (VIIBRE) directed by Dr. John Wikswo, the flow used to maintain the gradient creates a small drag force on the cells and the diffusion between the streams causes the chemokine gradient to vary down the length of the device [23]. The concentration gradient within the device was visualized and quantified using the fluorescent molecule FITC-dextran. The two input streams to the device are divided and mixed until five streams are created, all of which flow into the main microchannel. The experiments performed by Wikswo et. al. have characterized two aspects of the device. Firstly, the cross-sectional linearity of the chemokine gradient depends on the position down the length of the microchannel. Cells attached near the beginning of the microchannel experience a step-like cross sectional profile while cells attached further down the microchannel experience a smoother cross-section. Secondly, although higher flow rates are desirable from a gradient viewpoint they can affect cell movement through the effects of shear forces on cells.

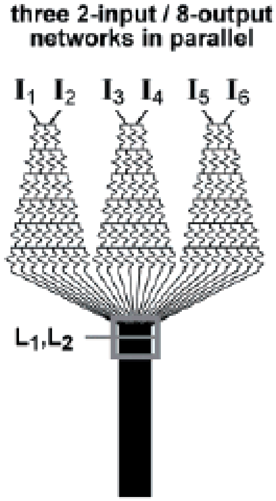

three 2-input / 8-output networks in parallel
I_1 I_2 I_3 I_4 I_5 I_6
L_1,L_2

Figure 5: Extension of the Jeon device to generate gradients of more complex shapes [24]

The device class developed by Jeon et al., is a valuable tool for chemotaxis research but the investigator has to consider the dynamic effects of shear force intrinsic to this method of developing gradients and investigators must also be extremely careful to provide steady regulated flow rates and pay particular attention to bubbles or other obstructions in the flow dividing network which can disturb or modulate the intended gradient.

I.3 A microfabricated multi-chambered diffusion device

An effort initiated in 2005 from VIIBRE has resulted in the development of a class of multi-chambered implantable cell trap devices that can be used to examine the combinatorial effects of the gradients of diffusive substances. Figure 6 shows a variant of such a device which can be used for *in vitro* studies of crossed chemokine gradients.

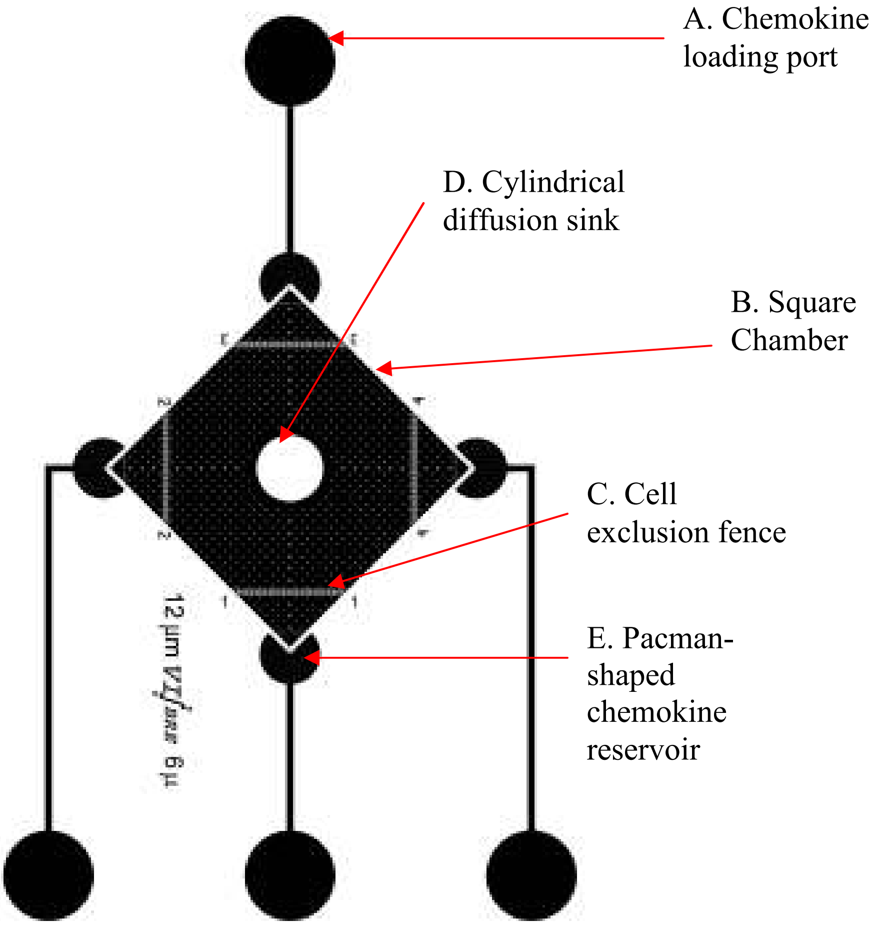

A. Chemokine loading port
D. Cylindrical diffusion sink
B. Square Chamber
C. Cell exclusion fence
E. Pacman-shaped chemokine reservoir
12 µm

Figure 6: VIIBRE microfabricated crossed gradient diffusion device

This device is designed to allow the formation of gradients by substances diffusing freely in the chamber [B]. It is a square chamber with an upraised cylinder in the centre. It has four inlets through which four different substances can be introduced and a central, large volume cylinder which acts as the central diffusion sink [D]. The device features cell exclusion fences which prevent cells from entering and clogging the narrow flux limiting channels which separate the pacman reservoirs from the square cell assay chamber. During experiments, the device is placed in a Petri dish that is filled with water to the top such that the device is completely submerged in water. The uniqueness of the structure of the device is evident in the entry of substances into the device and their diffusion through it. It has four ‘pacman’-shaped structures [E] at each inlet at a short distance from the actual opening. The pacman structures are connected to the main device through short and narrow flux limiting channels. The pacman-shaped structures act as finite reservoirs which provide us with a mechanism for allowing the free diffusion of substances into the device without active pumping. The channels connecting these reservoirs to the main chamber are extremely narrow compared to the rest of the device (Figure 2-G). This relatively high difference in dimensions provides us with a controlled delivery of biologically relevant substances from the reservoirs into the device. In experiments, cells are initially introduced into rectangular cell assay chamber and then observed as the crossed diffusion gradients engendered by the four chemokine reservoirs are created over time as substances diffuse through the narrow flux limiting channels [Fig. 7].

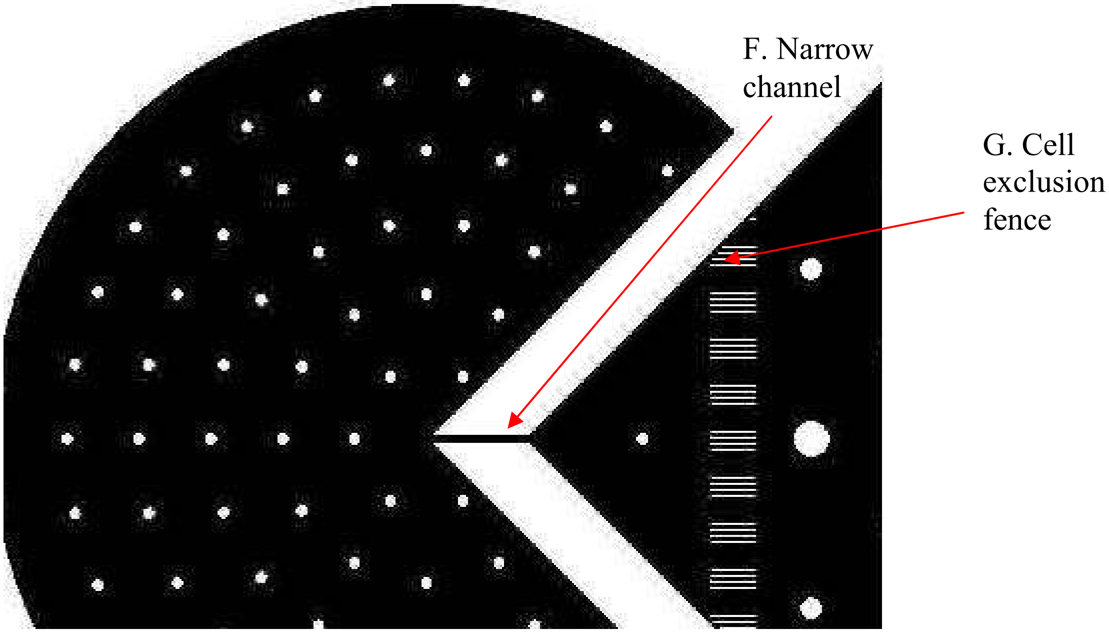


Figure 7: Magnified view of the channel and cell exclusion fence

A characteristic of this device that differentiates it from Jeon type laminar flow mixer devices is the gradient experienced by cells in the device from the beginning of the chamber to the central sink. The cells experience a smooth time dependent gradient derived by device geometry with no step-like fluctuation in any cell assay.

We require a detailed visualization of these gradients to analyze the behavior of cells in combinatorial gradients of different substances. As the device is still in its first generation, the subsequent goal for it is to achieve maximum possible functionality which necessitates optimization of the design of this device. To determine the modifications that can be made to the design, the current performance of the device should be thoroughly studied.

I.4 Objectives

The traditional empirical method for the development and optimization of a microfluidic device is a cyclical process. The initial prototype is fabricated and experiments are performed. The parameters are changed after interpretation of the results and the device is re-fabricated to optimize its performance. The performance of the device is characterized by a number of parameters which can depend on the device itself such as length of the chamber and the diameter of the sink or can also depend upon the properties of the substance introduced in the device such as diffusivity. Moreover, in order to control the formation of different combinatorial gradients we need to know the correlation between the properties of the device and the properties of the substances introduced in the device. The device, therefore, needs to be analyzed from a fluid mechanics and diffusion perspective. The analysis of device performance including all the necessary parameters becomes a multivariable problem where the effect of each parameter is determined by varying it and keeping the other variables constant. The experimental approach to solving such a multivariable problem involves both manufacturing resources and time as the device has to be fabricated each time a structural property is changed and a new experiment has to be performed each time a different molecule is made to enter the device. A computational fluid dynamics (CFD) model is an efficient solution to such a problem as it allows us to analyze the device by varying a single or a combination of parameters in much less time and using only a simulation. This enables us to select those cases where modifications in the device give us maximum benefit in terms of optimizing the function of the device. Advances in computational capacities and availability of software specific to fluidic analysis have made it possible to develop such models. Diffusion is not a standard problem in CFD and the device modeled in this thesis functions exclusively through diffusion, without involving any convection or even cross-advection. However, the principles for solving diffusion problems numerically are similar to those used for solving fluid dynamics problems and the geometry of the device is complex. The goal of developing a model for the diffusion device was to have an initial version which analyzes diffusion in the device and then extend it in the future to include those features where convection also needs to be analyzed. The choice of CFD provided us with this capability.

In the context of this thesis, the computational fluid dynamics (CFD) model is also an endeavor to understand the feasibility of modeling microfluidic devices utilizing the Fluent [26] and GAMBIT [27] computational fluidic software packages. A detailed discussion on modeling in GAMBIT and Fluent will be provided in Section *II.2.3*. Fluent is typically used for laminar flow or turbulent flow in macroscale problems. A flow or even a diffusion problem in the microscale involves values with extremely small magnitudes, in the order of femtometres. This thesis is an effort to understand and interpret the way Fluent handles small numbers and the extent to which a microfluidic device can be approximated with a CFD model. We have also shed light on the advantages of using CFD software, the issues encountered in the process and methods to circumvent these issues, if they exist.

It is essential to define the variables and cases for the model before it is set up. Parameters related to the device that are to be studied in this project using the model include the length of each side of the square chamber, the diameter and height of the central sink cylinder and ratio of these two properties. Our approach in modeling the passive diffusion device in Fluent has been one of sequential approximation. We initially developed a model of the device which consists of only a chamber and a sink with an infinite source outside the device. This simple geometry allowed us to assess the validity of the results obtained from Fluent in the absence of complex structures that affect the diffusion process in the actual device such as the narrow channel and a finite reservoir. The initial results were strictly to assess Fluent as a useful tool for simulating microfluidic devices. However, the cases studying the effects of particular architectural parameters on the function of the device can be taken as good approximations to estimating the functional trend of the device when each of these parameters is altered.

CHAPTER II

CHARACTERIZATION OF A PASSIVE DIFFUSION MICRODEVICE FOR ASSAYS OF CHEMOTAXIS AND MORPHOGENESIS

II.2 Materials and Methods

We have used a wide range of tools and techniques for the development of the computational model. Ranging from the fundamental principles of diffusion to visualization and contouring algorithms, they have been incorporated into the analysis at different stages. The most significant tool used in this process has been the commercial CFD software package named Fluent. We have used a separate section in this chapter to discuss the capabilities of Fluent, the interrelation between the preprocessing and the processing parts of the software and the sequence of processes leading to the final solution.

II.2.1 Diffusion—theories and solution methods

Diffusion is the process by which matter is transported from a region of high concentration to a region of lower concentration as a result of random molecular motions [28]. Transfer of heat by conduction is also due to random molecular motions and there is an obvious analogy between the two processes. This fact was first recognized by Fick who put diffusion on a quantitative basis by adopting the mathematical equation of heat conduction derived originally by Fourier [29]. In this approach, diffusion is first considered as a macroscopic phenomenon and an approximate differential equation is derived to characterize it. A second form of the diffusion equation which is the form derived by Fick, is obtained by combining the approximate differential equation with the continuity equation [30]. We define fluence or flux density, an important characteristic of fluids, as the amount of matter transported across an imaginary surface per unit area per unit time [31]. It can be represented by a vector pointing in the direction the matter moves and is denoted by **j**. Movement of molecules from one region to another takes place only due to a difference of concentrations in the regions. The change in concentration is denoted by $\frac{\partial C}{\partial x}$ when the change is concentration is small enough to be approaching zero. We can assume the diffusion problem to be one-dimensional for simplification and consequently, make the statement that there is no net convection if $\frac{\partial C}{\partial x} = 0$ and there is flow when $\frac{\partial C}{\partial x} \neq 0$. Considering that the difference in concentration is small and following from the definition of flux density, we can also conclude that the flux density is linearly proportional to the concentration gradient. Therefore,

$$j_x \propto \frac{\partial C}{\partial x}.$$

The corresponding equation for this relation is

$$j_x = -D\frac{\partial C}{\partial x} \quad (1.1)$$

where the constant D is called the diffusion coefficient and has units m^2s^{-1} in the S.I. system. This relationship is called Fick's first law of diffusion and the negative sign shows that the flow is in the direction from higher concentration to lower concentration. If the equation for the first law is differentiated we get:

$$\frac{\partial j_x}{\partial x} = -D\frac{\partial^2 C}{\partial x^2} \quad (1.2)$$

Fick's second law can easily be derived by considering the following situation [32]. An element of volume in the form of a rectangular parallelepiped is considered whose sides are parallel to the axes of coordinates and are of length 2dx, 2dy and 2dz. Let the center of the element be at P(x, y, z) where the concentration of the diffusing substance is C. We also assume that ABCD and A`B`C D` are the faces perpendicular to the axis of x as illustrated in Figure 3.

Therefore, the rate at which the diffusing substance enters the element through the face ABCD in the x-plane is given by

$$4dydz(j_x - \frac{\partial j_x}{\partial x} dx)$$

where j_x is the rate of transfer through unit area of the corresponding plane through P.

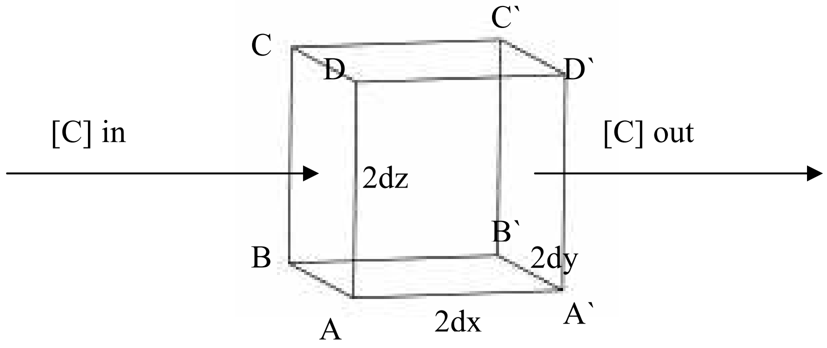


Figure 8: Rectangular parallelepiped for demonstrating Fick's Second Law of Diffusion

Similarly, the rate of loss of diffusing substance through the face A`B`C`D` is given by

$$4dydz(j_x + \frac{\partial j_x}{\partial x} dx)$$

The net increase in the rate of diffusing substance can then be obtained by summing up the transfer and loss to get

$$-8dxdydz\frac{\partial j_x}{\partial x}$$

In a similar, from the other faces we get

$$-8dxdydz\frac{\partial j_y}{\partial y}\, and - 8dxdydz\frac{\partial j_z}{\partial z}.$$

Hence, the total rate for the increase of diffusing substance in the element is given by

$$-8dxdydz\frac{\partial j_x}{\partial x}\ -8dxdydz\frac{\partial j_y}{\partial y}\, and - 8dxdydz\frac{\partial j_z}{\partial z}$$

The rate at which the amount of diffusing substance in the element increases is also given by

$$8dxdydz\frac{\partial C}{\partial t}$$

Equating the two expressions given above we get

$$\frac{\partial C}{\partial t}+\frac{\partial j_x}{\partial x}+\frac{\partial j_y}{\partial y}+\frac{\partial j_z}{\partial z}=0$$

as the equation of continuity [33]. The continuum hypothesis states that the number of particles or mass or in the case of an incompressible fluid, the volume, remains constant.

Replacing the flux terms in the last three expressions from equation 1.1 we get

$$\frac{\partial C}{\partial t}=D(\frac{\partial^2 C}{\partial x^2}+\frac{\partial^2 C}{\partial y^2}+\frac{\partial^2 C}{\partial z^2})\ \ (1.3)$$

$$\text{or}\frac{\partial C}{\partial t}=D\nabla^2 C\ \ (1.4)$$

as Fick’s Second Law of Diffusion. It combines Fick’s first law with the continuum hypothesis and tells us how the concentration at a particular point changes with time. Fick’s second law has been used for modeling the diffusion device in this thesis.

The diffusion equation has several general solutions when the diffusion coefficient is constant depending on the initial and boundary conditions. The conditions for which the diffusion equation is solved can be broadly divided into two categories—time-dependent and steady-state. We use steady-state conditions to solve the diffusion equation when the system is approaching a constant concentration or, in other words, the rate of change in concentration approaches zero. This can be written as

$$\frac{\partial C}{\partial t} = 0 .$$

In all other cases where the nature of the diffusion system is not completely known and the time required to reach steady-state has to be determined, the time-dependent method is used in which equation 1.4 is solved.

The diffusion equation can be solved analytically or computationally, depending on the type of geometry being considered. Analytical solutions for the diffusion equation consist mostly of simple geometries with defined initial and boundary conditions. However, all applicable geometries cannot be accurately modeled as ideal shapes and hence, the diffusion equation is solved computationally in such cases. The computational method requires a set of initial and boundary conditions as inputs along with the equation. It discretizes the problem space into small parts for each of which it obtains an approximate solution and then integrates over the whole space to obtain the final solution. The method of calculating diffusive flux computationally has been shown below [34] to demonstrate the transition from continuous to discrete domains. The diffusion equation, in the continuous form, is written as:

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2} \qquad (2.1)$$

In order to derive the discretized form of (1) we define each of the derivatives in their discrete forms and equate them as follows. It should be noted that although we have taken the concentration variable to be in terms of position on the x-axis and time, the same process that we outline below is applicable to systems for over 2 variables. Hence, the time derivative in (1) is expressed as:

$$\frac{\partial C(x,t)}{\partial t} = \frac{C(x,t+\Delta t) - C(x,t)}{\Delta t}$$

The second derivative with respect to position can be expressed in terms of a centered formula. This states that for any function $f(x)$ the line joining the points $f(x+h)$ and $f(x-h)$ is also an approximation to the slope at the point $f(x)$ where h is the quantity in terms of which the function $f(x)$ can be expanded as a Taylor series. Therefore, the first derivative from first principles and using this centered formula is written as

$$f'(x) = \lim_{h->0} \frac{f(x+h) - f(x-h)}{2h}$$

The second derivative, according to the same method and using Taylor series expansions can be written as

$$f''(x) = \frac{f(x+h) + f(x-h) - 2f(x)}{h^2}$$

The second derivative of concentration with respect to position is, therefore, expressed as

$$\frac{\partial^2 C(x,t)}{\partial x^2} = D\frac{C(x+h,t) + C(x-h,t) - 2C(x,t)}{h^2}$$

Therefore, the discretized diffusion equation in the finite time central difference (FTCS) form is

$$\frac{C(x,t+1) - C(x,t)}{\Delta t} = D\frac{C(x+h,t) + C(x-h,t) - 2C(x,t)}{h^2}$$

II.2.2 Computational Fluid Dynamics

The division of a complete problem space into discrete elements which serve as the solution nodes forms the fundamental principle of computational fluid dynamics (CFD). In classical fluid dynamics, the fluid is treated as a continuum and the description of a body of fluid can be given using either its position or its velocity as the reference value. In the Lagrangian method of describing a fluid, a fluid particle is characterized by a position vector $x(a,t)$ where the position is dependent on a property of the fluid, a and its transition in time [35]. In most cases, a is assumed to be the position of the fluid particle at time *t=0***.** The Lagrangian method is the standard mode of characterization and is used as the foundation for deriving the major equations that characterize the field of flow. All fluids where the stress-strain ratio curve is linear are called Newtonian fluids and their flow is characterized by three equations—the continuity equation, conservation of momentum and conservation of energy. The equations for the conservation of momentum are known as the Navier-Stokes equations. The solution of these three fundamental equations forms the core of understanding fluid flow.

The approximate solution of the steady and unsteady Navier-Stokes equations for incompressible fluids as well as any other equations associated with fluid flow continue to be active areas of research. Several numerical schemes have been developed for the calculation of these equations and the most significant factor in the process has been the proper specification of boundary conditions, especially when solid walls are present. The steady improvement in the speed and memory size in computers since the 1950s has led to the progression of solution methods from numerical analysis to computational fluid dynamics. According to Chapman, Green, Rubbert and Jameson, CFD offers four major advantages when compared to experimental fluid dynamics [36]. Firstly, the lead time in design and development is significantly reduced. Secondly, CFD can simulate flow conditions not reproducible in experimental model tests. Thirdly, CFD provides highly detailed and comprehensive information and is cost-effective. Lastly, CFD involves lower energy consumption than actual construction of fabrication of the object being modeled.

The process of obtaining solutions to flow, diffusion or other similar equations computationally consists of two main stages. The first stage converts the continuous partial differential equations and auxiliary (boundary and initial) conditions into a discrete set of algebraic equations. This first stage is called discretization. The replacement of individual differentiated terms in the partial differentiated equations by algebraic expressions connecting nodal values on a finite grid introduces an error. Therefore, it is important to choose algebraic expressions such that this error is minimized. The second stage of the process consists of the actual solution of the algebraic equations in an iterative fashion until satisfactory values are obtained for the approximate solutions. Three major methods are available for obtaining the solution to these equations—finite difference, finite element and finite volume.

The finite difference in the discrete space is analogous to the derivative in continuous space. It was developed in the 1920s by A.Thom [37] for the solution of nonlinear hydrodynamic equations and has since continued to be used for initial solutions of the Navier-Stokes equations. The method involves three basic steps—dividing the solution into grids of nodes, approximating the partial differential equation to finite differences that relate the solution to grid points and solving the finite difference equations subject to the prescribed initial and boundary conditions. The finite differences employed can be of the forward, backward or central kind as described below and the schemes incorporating these different finite difference techniques are the explicit, implicit and Crank-Nicholson Methods respectively. Table 1.1 illustrates the mathematical definitions of each of these finite differences.

Table 1: Different finite difference schemes and their application in the diffusion equation

Method Name	**Finite Difference Formula**	**Method Description**
Explicit Method	$\frac{u_j^{n+1}-u_j^n}{k}=\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{h^2}$	Forward difference at time t_n and second order central difference for space, x_j.
Implicit Method	$\frac{u_j^{n+1}-u_j^n}{k}=\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{n+1}^{j-1}}{h^2}$	Backward difference at time t_{n+1} and central difference for the space derivative at position x_j.
Crank-Nicholson Method	$2(\frac{u_j^{n+1}-u_j^n}{k})=\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{n+1}^{j-1}}{h^2}+\frac{u_{j+1}^n-2u_j^n+u_{j+1}^n}{h^2}$	Central difference at time $t_{n+1/2}$ and second order central difference for the space derivative x_j

[38]

A finite element method discretization of a continuous partial differential equation is based on a piecewise representation of the solution in terms of specified basis functions. These basis functions can be defined using a number of methods of which the most common method is Galerkin's method [39]. The boundary value problem which consists of the partial differential equations expressed as functions along with the appropriate boundary conditions is first converted to its weak or variational form. This form expresses the problem as a function of integrals and their derivatives as opposed to functions of numbers. The variational form which is essentially an infinite dimensional linear problem is replaced with a finite dimensional form using piecewise linear functions. The given polygonal space is triangulated and the finite dimensional form actually consists of functions that are linear, quadratic or even polynomial on each triangle of the chosen triangulation. The advantage of the finite element method is that it can handle more complex geometries compared to the finite difference method which can handle only rectangular blocks or its simple variations. Moreover, the finite difference method approximates the operator (the derivative) and solves the problem on the nodes or intersection of edges on the grid. The finite element method, on the other hand, uses exact operators but approximates the solution basis functions and can solve the problem at the interior of elements or cells as well as at the nodes, depending on our needs.

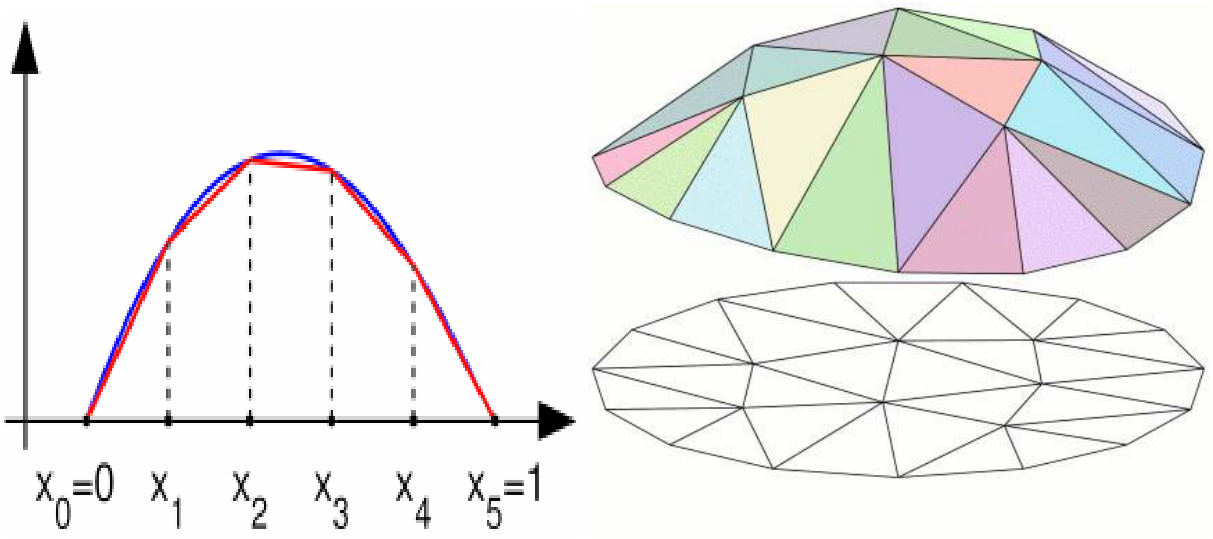


Figure 9: Piecewise linear approximation of a continuous function and triangulation of a surface using the finite element method [40]

The finite element method is mostly used in structural mechanics rather than in fluid mechanics, despite its advantages over the finite difference method. The most commonly used calculation scheme in CFD software is the finite volume method [41]. This method is similar to FEM as it also uses an integral form of the original partial differential equation. However, in the finite volume method, the region of integration is taken to be a control volume Ω_i, associated with the point coordinate x_i. The finite volume method can be considered as a hybrid of the finite difference and finite element methods. It converges much faster than the finite element method with the run time of the two methods differing by a ratio of 10^5, on an average. The finite volume method also uses a reduced system of weights, compared to the finite element. The finite element method uses a basis function to denote the weight function while the finite volume method uses a piecewise function to denote the weights. The finite volume method is therefore, a reduced version of the finite element method and due to its faster convergence time is used more extensively in computational fluid dynamics.

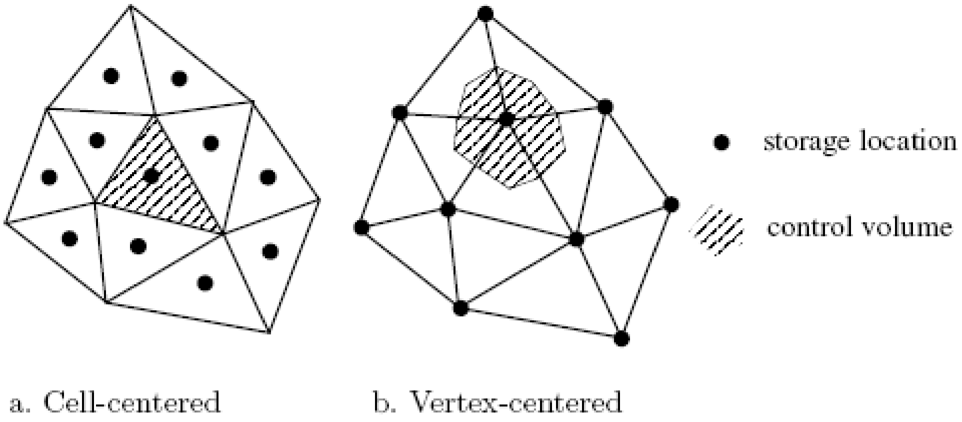


Figure 10: Illustration of control volume variants used in the finite volume method [42]

II.2.3 Fluent and Gambit—fundamentals of fluidic modeling

Fluent is a widely used software package for CFD analysis and is currently packaged and marketed by Ansys, USA. It is designed to handle both turbulent and laminar flow analysis and supports a large number of flow processes such as combustion, convection-diffusion, species reaction and turbo propulsion. The basic Fluent package consists of three main components:

- Gambit for geometry development
- Fluent, the solver
- FlexLM, the licensing software

A given flow problem is initially analyzed to find the simplest form of the original geometry in which the flow takes place. The geometry might have structural or other architectural components which either do not contribute to the flow or do not contribute significantly to it. The geometry that is set up in GAMBIT is exported as a mesh which is then set up as a case in Fluent. The materials for the simulation, boundary and initial conditions, relaxation factors and other parameters related to the simulation are set up in Fluent. The data obtained from Fluent is postprocessed using a number of tools that will be discussed in the next section.

GAMBIT is a preprocessing section of Fluent exclusively used for setting up the geometry for the CFD problem, assigning boundary conditions and fluid zones. Developing the geometry in GAMBIT begins with the creation of vertices, edges, faces and volumes. The final shape is then meshed using an appropriate meshing scheme. This is a significant step during geometry development as it is the time-limiting and calculation-limiting factor in the simulation. A finer mesh leads to refinement in data but results in a tradeoff in run-time. The objective of the recurring process of computational fluid dynamics is to achieve the optimal balance between mesh refinement and the running time for the calculation. The procedure for setting up geometry, the options available and the theories of meshing algorithms are discussed in greater detail in Appendix A.

The geometry developed in GAMBIT is exported as a mesh file (.msh) and is read into Fluent, the solver environment. The materials required for the simulation—solid, liquid or gaseous are assigned here. The boundary conditions for the calculation are set up in GAMBIT but can be modified in Fluent. All other parameters required for evaluating the solution of the discretized diffusion equation such as relaxation factors, initial conditions, discretization schemes and calculation schemes are all set up in Fluent. The assignment of the time discretization scheme and the timestep size is the final crucial step while setting up a simulation. Larger timesteps decrease the stability of unstable solutions. Partial differential equations that have multiple time scales and are discretized through the finite element or finite volume methods are challenging for the cache-based architecture that is prevalent in most PCs at present. The optimization of run-time and mesh quality simultaneously is, therefore, an integral part of the modeling process. The procedure for setting up a simulation case in Fluent and issues to be aware of in the process are discussed in further detail in Appendix B and Appendix C.

The final stage of CFD modeling is the postprocessing and visualization of data from the solver is discussed in the next section. In a study done by Smith et al. [43] of the relationship between the running time of a CFD program and the number of nodes in an unstructured mesh they found that the execution time decreased exponentially as the number of vertices assigned to each node on a parallel computational system decreased (Figure 12).

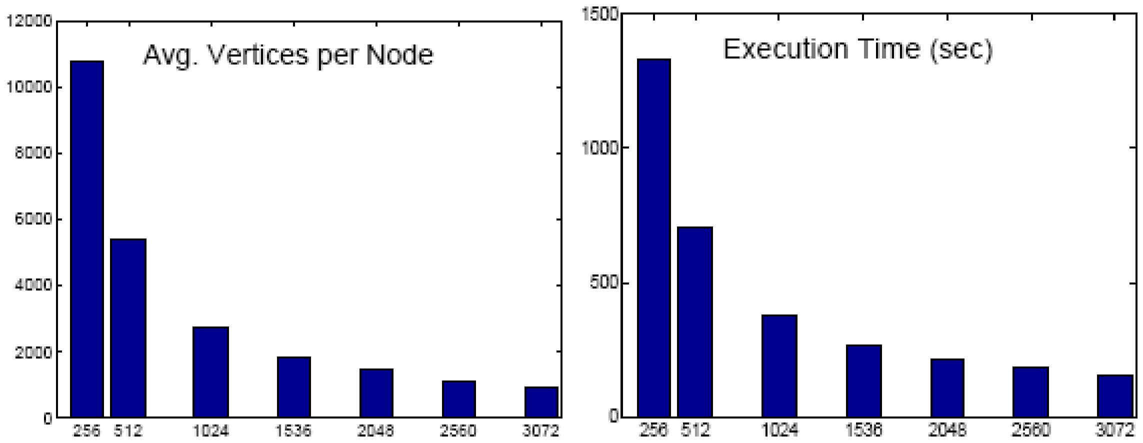


Figure 11: Experiment showing the decrease in time with a decrease in the number of vertices per node [43]

II.2.4 Visualization and analysis of CFD data

The data obtained from any CFD solver environment can be in one of the two main formats—binary or text. The preferred choice for storage is generally binary as it occupies twelve to fourteen times less space compared to the text files. Binary files present limitations in terms of processing the data due to their compact size. The information is highly encoded and can be read bysome visualization programs but the process of reading the files takes a great deal of time (~15mins/file). Moreover, the binary case and data files are written out by Fluent in a nonspecific manner, i.e., all the output parameters applicable to the simulation in every part of the grid are written out in the data files. This makes it difficult for the user to sort out the required information and select the parts of the grid on which the reported data is required. It is also not possible to report certain important features of interest such as volume and surface integrals. Binary files can be read directly in programs such as Tecplot [44] that can regenerate the mesh and the solution properties from the files. This feature gives a useful complete view of the geometry and the appearance of the solution as contours or streamtraces.

In order to extract surfaces and report the data selectively on those surfaces we can use these binary case and data files in Fluent to extract any desired surface, beginning from a point to a sub-zone. These surfaces and the data associated with them can be reported in eight different file formats, among which we have used ASCII and Tecplot. Reporting data using a different format not only allows us to select the zone of interest but also the variables of interest. Another advantage of these text-based formats is immediate visibility of data which enables the user to see the raw data and check for any fundamental errors within it.

In this theisis my first focus was on verifying that Fluent could be used to generate credible results using simplified device geometry and realistic diffusion coefficients. Accordingly, before using a new software environment for large scale calculations, we did a ‘pilot’ application using the device geometry in its simplest form and all the basic functions in Fluent that would be used in later simulations. The postprocessed data was not graphed using Matlab during these initial stages. Fluent files were directly read in Tecplot and points were extracted from Tecplot manually. These have been graphed using Microsoft Excel which has enough features to allow us to assess the concentration values and trends on a macroscale basis.

We have used the ASCII and Tecplot data file formats extensively to report data from selected surfaces. The surfaces that we have created using data and case files in Fluent are lines, rakes and planes. The detailed procedure for extracting these surfaces and reporting data from them is in Appendix B.

II.2.5 Geometries and boundary conditions used in Fluent

The simulations that we have modeled in Fluent, from preliminary geometrical representations to the complete device, have been run after setting up the geometries in GAMBIT with the required dimensions and specifications. In order to test the validity of Fluent before progressing to large-scale calculations, as mentioned in section *I.4*, we have used a simple version of the actual geometry of the device. This model consists of the square chamber and the cylindrical sink with an infinite source outside the device. The square diffusion chamber has each side equal to 7500μm and is 20μm tall. At the corners of the square are 12μm wide openings- this is assumed to be the initial width of the narrow channel. The sink is cylindrical with a diameter and height equal to 2000μm. This is the first model where we use a sizing function to create the mesh to adjust for mesh sizes at the corners of the chamber.

We simulated the simple version of the 3D device initially and then approximated it to a 2D diffusion problem in later simulations. This simplification was possible due to two major reasons. Firstly, the ratio of the height of the square chamber to the diameter and height of the cylinder is in the order of 10^2—high enough to allow us to neglect the third dimension. Secondly, we show later [section II.3.3] numerically that the 3D and its 2D approximation can be treated as equivalent problems and therefore, the 2D model can easily be used in all situations to get feasible results. The square chamber is approximated to a square whose vertices are truncated by 12μm wide openings. The cylinder becomes a circle in the middle of the square with the same diameter.

We used this 2D representation of the simple geometry to perform simulations in which we tested the effect of changing different parameters on the device. The parameters that we tested are the length of the square chamber, the diameter of the cylindrical sink, the ratio of the length to the diameter and the diffusion coefficient of the diffusing substance. We ran 3-4 cases for each parameter and then summarized the results with an approximate polynomial derived by using the method of least squares [45].

In the next stage of simulations, we advanced to simulating the device geometry incorporating all parts of the actual device including the finite reservoir and the narrow restriction channel. The dimensions of the device were also changed in this model to equate them to the dimensions of the actual device. The length of each side of the square chamber is still 3750μm. The width of the channel was made 100μm and its length was 200μm. The radius of the cylindrical sink was 2000μm and its height was 4000μm. The reservoir at the end of the channel was made a half-cylinder with the same radius and height. This model had significant changes with respect to meshing control, boundary conditions and initialization of the solution, summarized in Table 2.

Our final set of models emulated the original device exactly in dimensions and a great deal in terms of shape. We knew the area of the reservoir (80mm^2) and using this value we designed a semicircular reservoir at the end of the channel with a radius equal to 1414.21μm. The remaining dimensions of the device remained the same as in previous models. The following table (Table 2) lists each model created in GAMBIT as well as the boundary and initial conditions specific to that particular model. The diffusion coefficient for the material used for the simulation is denoted by D.

Table 2: List of the models and simulations run in Fluent, from preliminary to final

#	Case Description	Geometry	Boundary and Initial Conditions
1.	Simplified 3D model D = 1.2E-05m^2/s	Square chamber and central cylindrical sink	No-flux condition at side and bottom *faces* Concentration at cylinder-top: 1E-26M Concentration of infinite source: 1M
2.	2D approximation of simplified 3D model D = 1.2E-05m^2/s	Square ≅ chamber Circle ≅ Cylinder	No-flux boundary conditions at side *edges* Concentration at the circumference of the circle: 1E-26M Concentration of the substance entering at the inlet: 1M.
4.	2D model with restriction fence D$_{[chamber]}$ = 1.2E-05m^2/s D$_{[fence] =}$ 1.2E-06m^2/s	Diameter of circle: 4000µm Fence ≅ Trapezium at a distance of 375µm from the top right corner, width 100µm.	Same as 2D model
5.	2D device with different diameters for the circle D = 1.2E-05m^2/s	3 cases—constant square dimensions and variable circle dimensions--2000µm, 3000µm and 4000µm	Same as for the 2D device
6.	3D slice with and without the barricade layer D = 1.2E-05m^2/s	Slice of 3D device Width of the opening between the chamber and the cylinder: With barricade = 5µm Without barricade = 20µm	Concentration of the substance entering the device is 1M.

Table 2, continued from previous page

7.	2D device with different lengths for the square side D = 1.2E-05m^2/s	3 cases—same dimensions for the circle diameter and different lengths for the sides of the square--7500µm, 10000µm, 30000µm and 50000µm	Same as for the 2D device
8.	2D device with different square-side to circle diameter ratios D = 1.2E-05m^2/s	3 cases—length of square side is 30000µm in every case Diameter of the circle is 2000µm, 3000µm and 4000µm	Same as for the 2D device
9.	2D device with different diffusivities for the entering substance D = 1.2E-05m^2/s	4 cases—dimensions are same as those of the 2D device. Diffusivities are varied by 10^{-1} in each case	Same as for the 2D device
10.	Original device with half-cylindrical reservoir D = 5E-10m^2/s	Similar to the simplified 3D model. Diameter of sink: 4000µm Height of sink: 4000µm **Additions** Narrow channel: Width = 100µm Length = 200µm Half-cylindrical reservoir: Radius = 2000µm Height = 4000µm	No flux conditions on all outer faces, including the top of the sink Initial concentration of fluid in the reservoir: 1mole/m^3 Initial concentration of fluid in the rest of the device: zero
11.	Exact 3D model of original device D = 3E-11m^2/s	**Alterations** Semicircular reservoir with area 80mm^2. Size functions used in both the chamber and the reservoir. 5 cases with channel widths 20µm, 40µm, 80µm, 150µm and 200µm	Same as in the previous model.

In our final set of simulations using Fluent, we also obtain information about the gradient in every cell of the device grid. We need this information to observe the formation of gradients on a planar portion of the device. We have the option of exporting the concentration data from the planar surface to another visualization program and then computing the gradients on the plane. Fluent exports either the center or the node of every cell in the grid that falls within the plane. This approach leads to loss in accuracy as we have to form a matrix of points for any visualization program and thus, have to use linear interpolation in order to determine the concentration and consequently, the gradient values. We can use Fluent to calculate the gradient value of each cell by writing a user-defined function (UDF) [46] using the macros supplied by Fluent and integrating it to our simulation. Fluent calculates the gradient in every cell using the theoretical formula for diffusion gradient that is derived from Fick's Laws as

$$J = -D\nabla\phi$$

where J denotes the flux of the diffusive substance, D its diffusion coefficient, ϕ the function denoting its concentration and ∇ the gradient operator.

This provides with data values directly from the simulation that can be visualized on the grid itself using Tecplot. The UDF is a program written in C, using macros inherently built in Fluent and constructs from the programming language itself. The results from the UDF calculations are stored in memory locations that are also defined by the user. The process for setting up the simulation with a UDF has been explained in greater detail in the tutorial in Appendix B.

II.3 Preliminary results for validation of Fluent

II.3.1 The basic 3D computational domain—steady-state and flux

The first set of results from Fluent is from the simplified version of the original device. The simulation was run for 300 timesteps with each timestep being equal to a second. The number of timesteps is derived from the diffusion time estimation formula:

$$T = \sqrt{\frac{l^2}{D}}$$

Figure 12 shows the contour diagram of the system at 150 seconds and by correlating the color map and the contour diagram we can observe the progression of the substance in the device. The concentration decreases as we reach the cylinder and drops to 0.05M over a distance of 3738µm along the x-axis.

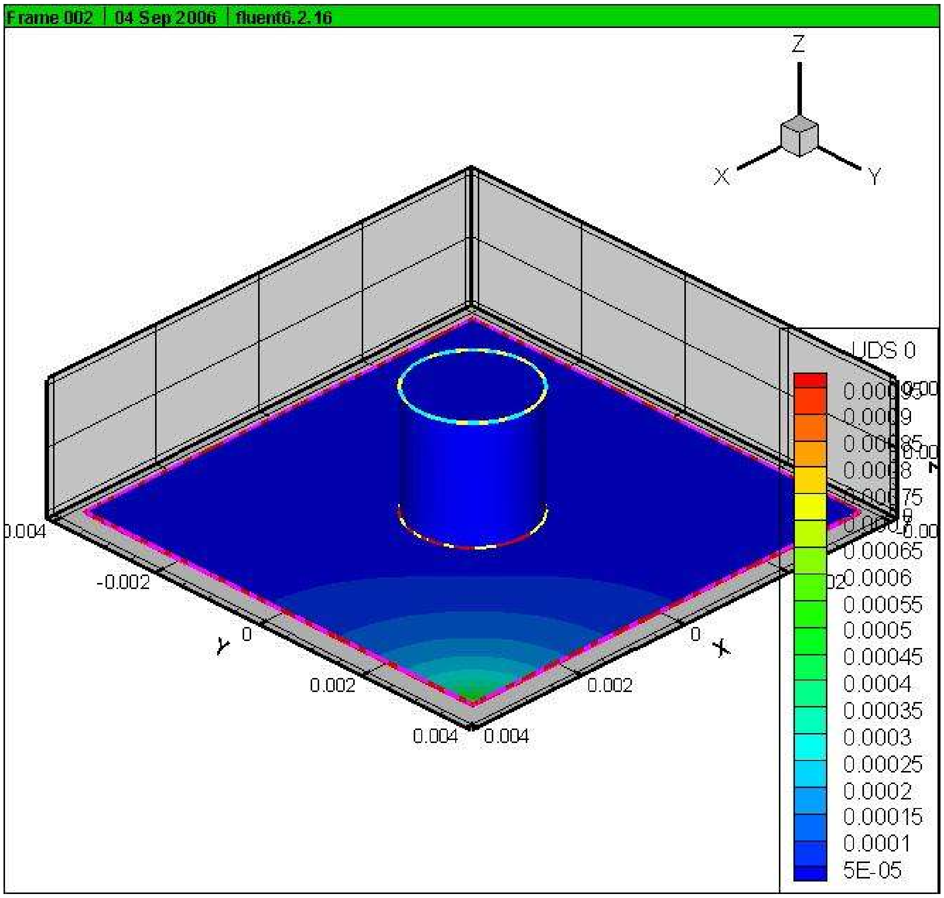


Figure 12: Contour plot of the 3D model showing the concentration gradient at 150 seconds

The system does not reach steady state with respect to the true physical definition of steady state. However, we need to establish a criterion to determine the time when the device reaches an optimum concentration distribution. We establish the condition for optimum concentration distribution according to the following mathematical relationship:

$$\frac{1}{N}\sum_{i=1}^{N}(u_i^{\,t+1} - u_i^{\,t}) \leq 1e - 06$$

where u_{t+1} is the concentration at time (t+1) seconds and u_t is the concentration at time *t* seconds. The change in time, Δt is not shown as its value is equal to one. N is the number of coordinates across which concentration values are taken, which in this case is 15. We choose such a criterion as an error margin of this order ensures that the concentration levels do not change in an amount that is biologically significant. In later experiments, such criteria are not used as we know the number of days for which we run an experiment in the actual device and use that information to establish the number of timesteps for the simulation.

The criterion established for this simulation takes into account the average of the differences in concentrations at two consecutive timesteps and is based on the epsilon-delta principle [16]. The value that is set as equivalent to an epsilon value for optimum concentration is 1e-06 as the concentration values for different coordinates are in the order of 1e-04 and the difference in concentration appears in the seventh decimal place and delta (Δt) is equal to one. Figure 13 shows the 3D system approaching the established condition and data has been collected in intervals of 20 seconds except after 140 seconds. After 140 seconds, data is also collected at 145 seconds and the values for concentration are examined more closely for every 5 seconds till 180 seconds. The raw data has been read directly in Tecplot and a slice is extracted from the 3D device by inserting a plane at the points (3750, 3750, 2E-05), (0, 0, 0) and (-3750, -3750, 2E-05), shown in Figure 13. Data points have then been extracted along a straight line running diagonally through the chamber till the beginning of the cylinder. The number of points extracted at each time point is 15 and the data from these points has been plotted using Microsoft Excel. Each point extracted is expressed in terms of x, y and z coordinates and the graph in Figure 14 displays the concentration at different time-points with distance decreasing from the entrance of the chamber to the edge of the cylinder. The concentration undergoes a steeper decrease after the first 10μm and gradually approaches values in the order of 1E-20 near the cylinder. It is observed that after 145 seconds the concentration changes by 1e-06M only after

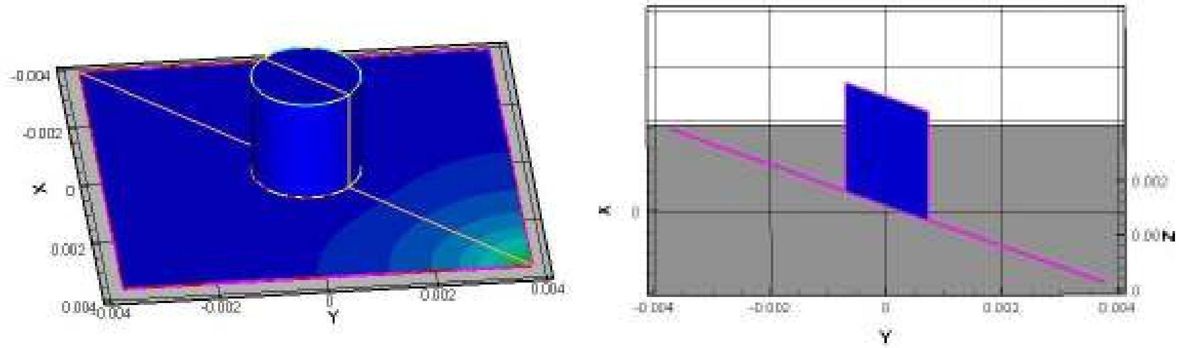


Figure 13: Simple 3D device viewed in Tecplot and the extracted 2D slice used for obtaining concentration values

every 30 seconds, implying that the rate of change of concentration decreases. The time-points between which the rate of change of concentration reaches an average less than the decided epsilon are 145 and 150 seconds.

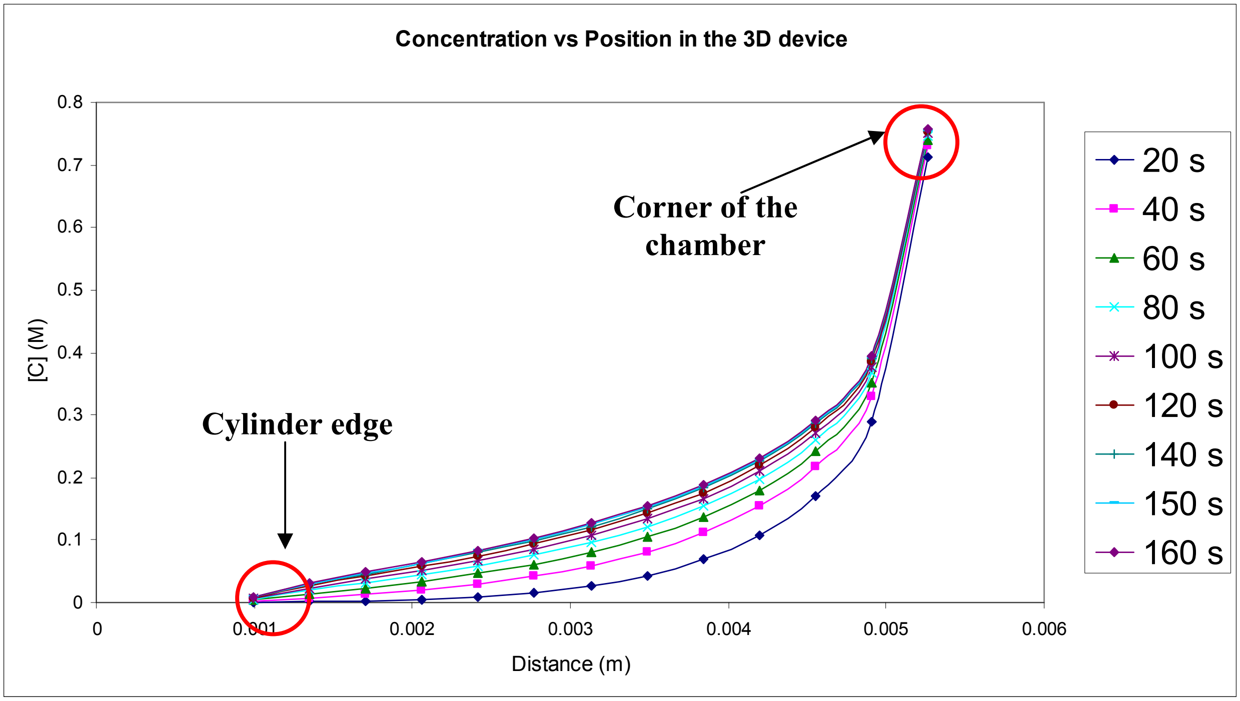


Figure 14: Concentration versus position on the X-axis plotted at different time-points for the 3D model

The flux at the channel where the diffusing substance enters the device is also calculated. As the data obtained on concentration is with respect to x, y and z coordinates, the discretized version of the diffusion equation is used, derived in section *II.1.1* as

$$\frac{C(x,t+1)-C(x,t)}{\Delta t}=D\frac{C(x+h,t)+C(x-h,t)-2C(x,t)}{h^2}$$

The left side of the above equation is used in the calculation of flux which has been done using a Matlab code. The value of h is 4E-06m^2 and that of D has been mentioned previously as 1.2E-05m^2/s. Figure 15 displays the trend of flux values from 10 seconds to 300 seconds and it is observed that the flux starts approaching a constant value at time values beyond 140 seconds.

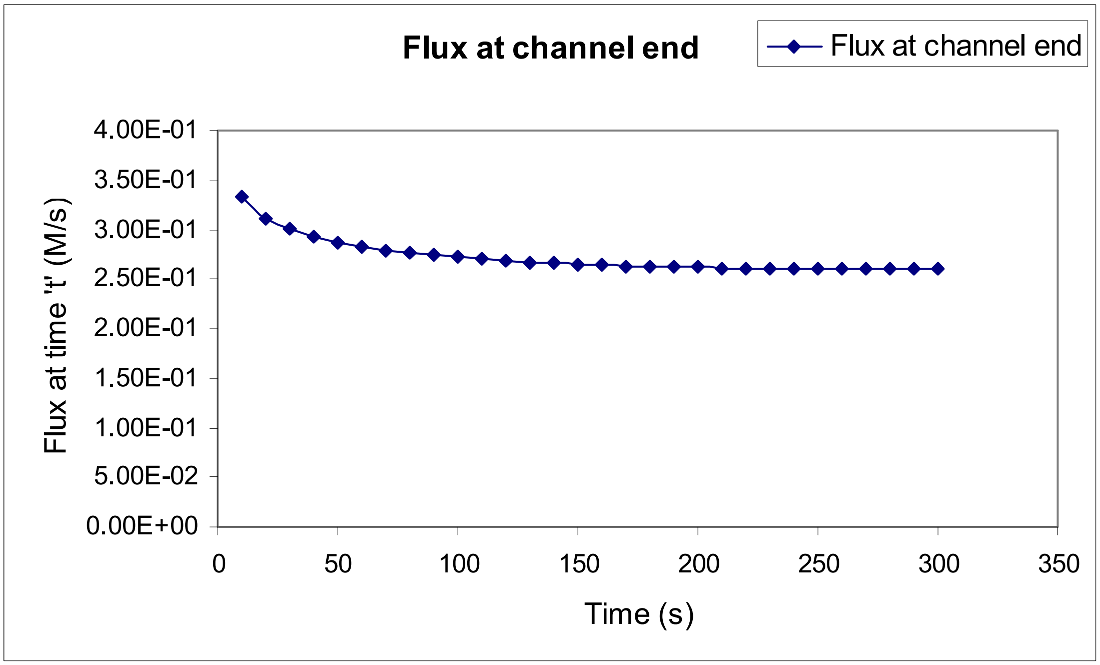


Figure 15: Flux entering the chamber, at time-points spaced 10 seconds apart

We obtain the concentration in the device in discrete forms at regular intervals, corresponding to the x, y and z coordinates.

II.3.2 The 2D approximation of the basic 3D domain

The simplified 3D model consists of a complex grid and takes approximately 6 hours to run on a Windows NT 4.0 Workstation. The running time for such models is related exponentially to the number of nodes in the mesh and the 3D model for this diffusion device has 68,500 nodes. We created the 2D approximation model to reduce the complexity of the grid and to provide an approximation model which can run faster. Using this model, we get a system that takes 135 seconds to approach steady-state using the same diffusive materials as well as the boundary conditions and materials as used in the 3D model. Figure 16 shows the results obtained from this simulation.

The system is shown approaching our established condition and data is plotted from 20 seconds to 140 seconds at intervals of 20 seconds till 120 seconds after which data is plotted at 130, 135 and 140 seconds. We observe from the results that the trend of concentration levels is similar to that in the 3D device but scaled down in terms of diffusion time and scaled up with respect to the concentration values. This happens as the substance is now moving over a 2D grid instead of a 3D grid and does not have to cover the z dimension which is mostly contained in the cylinder in the 3D model.

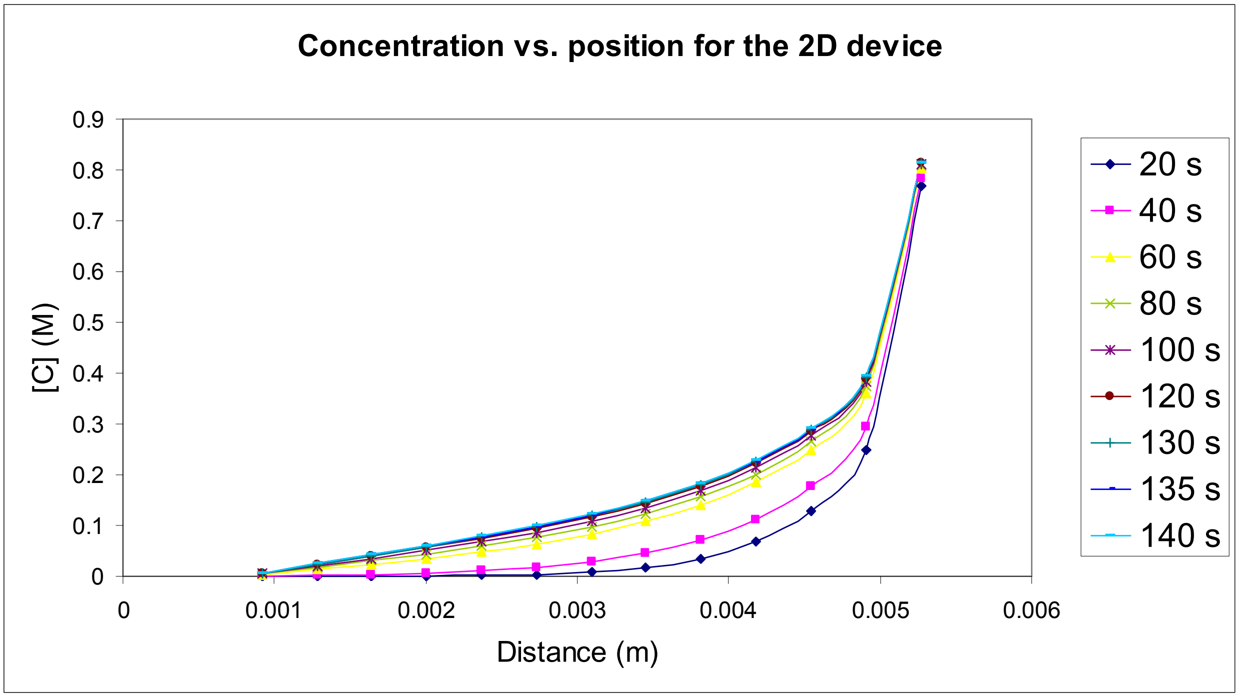


Figure 16: Concentration versus position on the X-axis for the 2D model

II.3.3 Equivalence of the 2D and 3D grids

We have used the 2D model to obtain a number of preliminary results examining the effects of various parameters, both architectural and physical, on the function of the device. We compared the 2D and 3D models prior to exclusively using the 2D model in order to determine the magnitude of error between the two models. We plotted the concentration over space from the 2D and 3D models at 140seconds. This plot indicated that the concentration values in the 2D and 3D models can almost be superimposed from the entrance to the chamber till the edge of the cylinder. We also plotted the difference in concentration values between the 3D and 2D models to obtain a magnified view of the error margin. The graph in Figure 17 indicates that the similarity in concentration values is maximum near the inlet where the substance enters and minimum near the hole where the cylinder begins (in the 3D model).

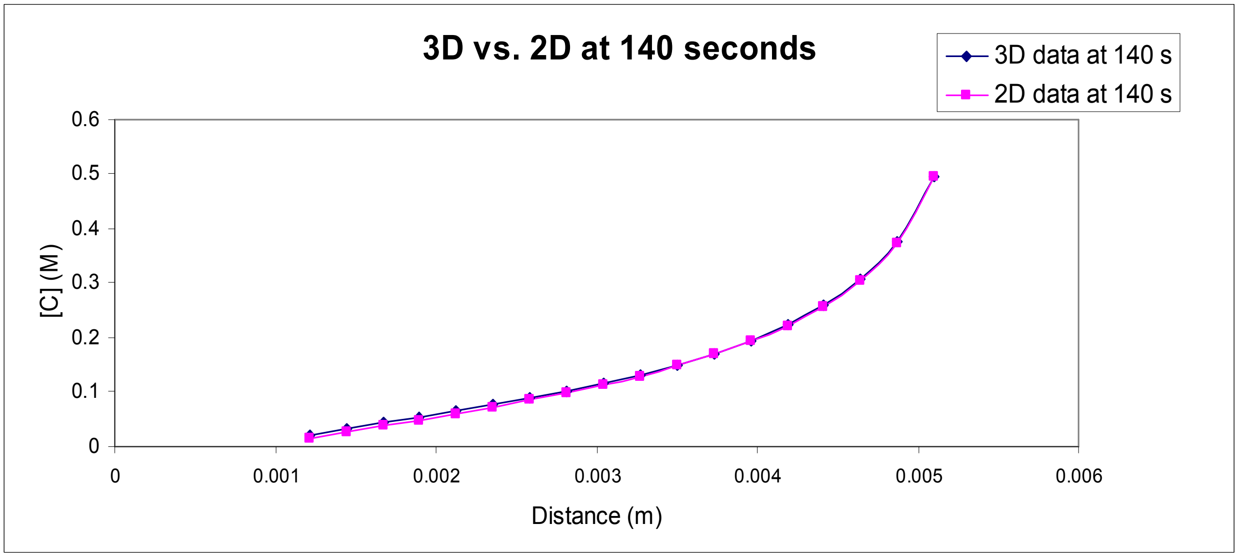


Figure 17: Concentration profiles over space for the 2D and 3D models at 140 seconds

To clarify this fact, we have also plotted the difference in concentration at the same coordinates in space for the 2D and 3D models, as shown in Figure 18.

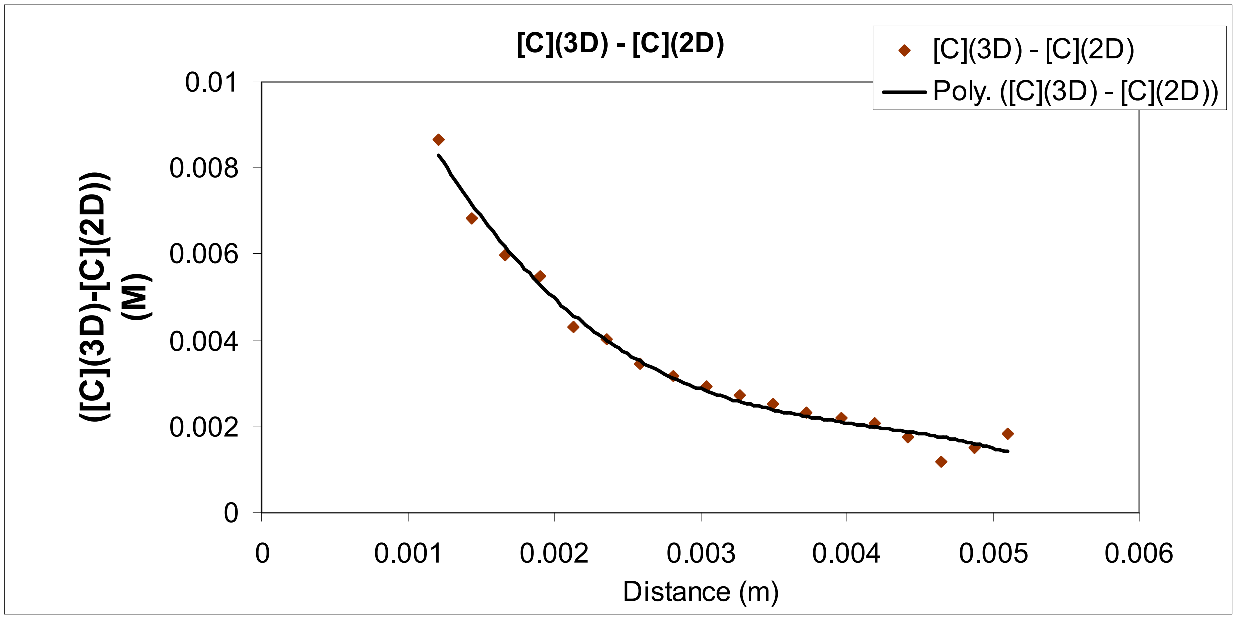


Figure 18: The difference in concentration over space, between the 2D and 3D models

We can see that further along the x-axis the magnitude of error decreases. This is an expected result as the regions of higher concentration are similar in both the 2D and 3D models—the height of the chamber is not significant compared to the width and length. It has been calculated from the numerical data that the standard deviation for the data is 1.38e-04kmoles.

II.3.4 Parametric studies with the basic 2D grid

The first parameter that we have studied using the 2D model is the addition of a fence in the device which is 100µm wide and 375µm from the corner, as described in section *II.2.2*. The fence, being a restriction structure, is expected to distort the radially diffusing gradient in the device. The fence is designed such that the diffusivity of the substance entering the fence is reduced by half on encountering it, which, in turn, accounts for the restriction in movement of substances in the device. Our model contains a simple version of this fence to examine the feasibility and method of adding this feature in Fluent and establishing proof of concept for the effect of the fence. The points have been extracted along the same diagonal line as in all other cases for the 2D model and the diffusivity of the substance after entering the fence is assumed to be 1% of the original diffusivity, as opposed to 50%, in order to see a more visible effect.

The results from the simulation show a significant slowing down of diffusive flow in the model with the fence which can be seen in Figure 19. At 180 seconds the difference between the concentrations obtained in the model with the fence and the simple 2D model, at the corner where diffusion starts, is more than 0.1M.

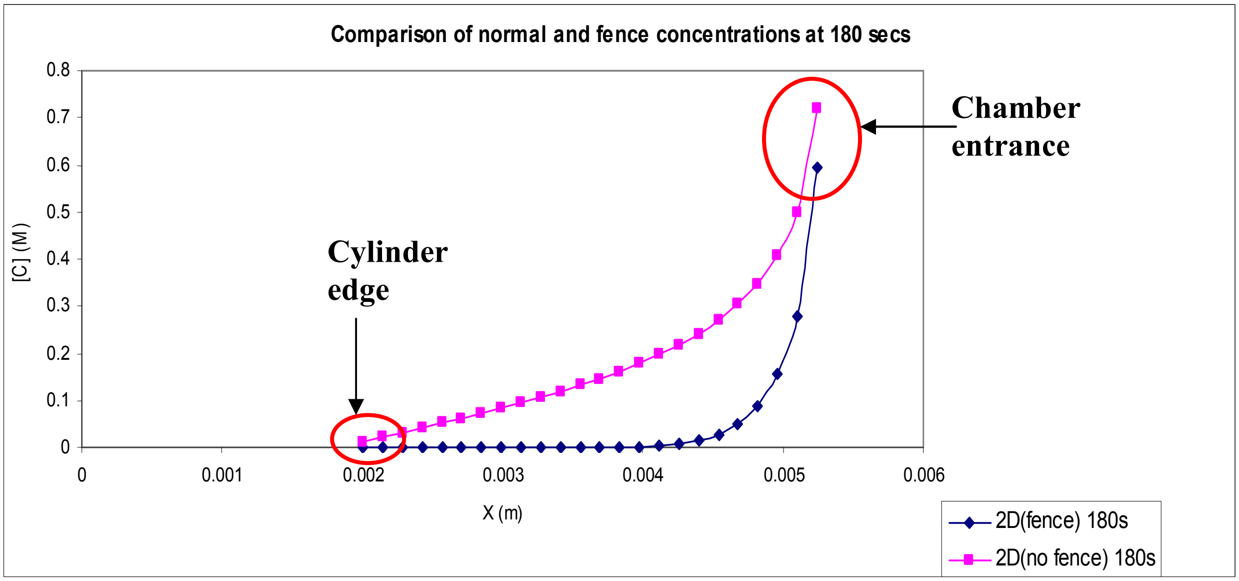


Figure 19: Comparison of concentration levels in the 2D device with the addition of a fence

The difference in flow can also be observed from the contour diagrams which are obtained for 140 seconds (Figures 20 and 21).

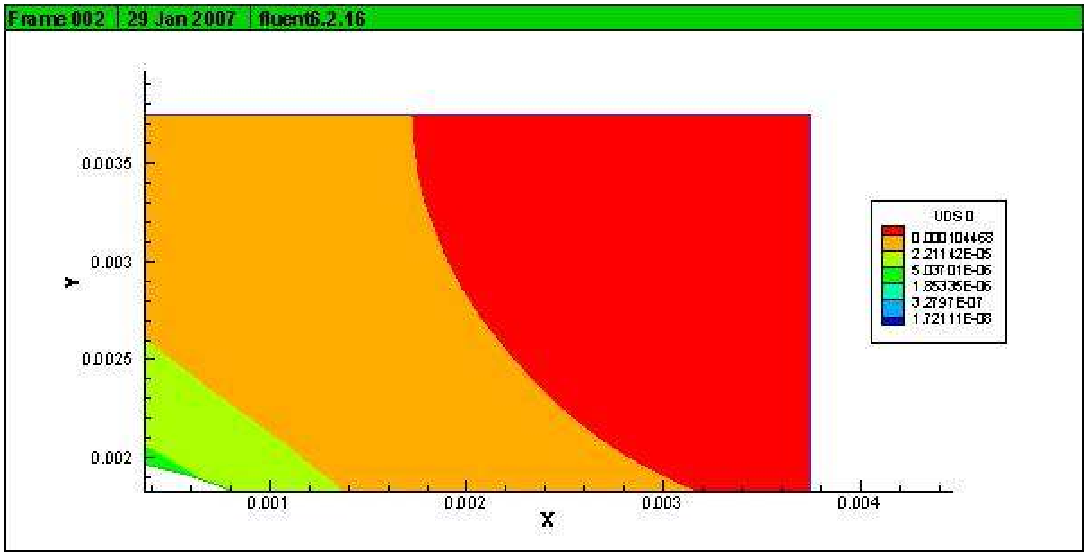


Figure 20: Contour plot of the device with no fence showing the concentration over space

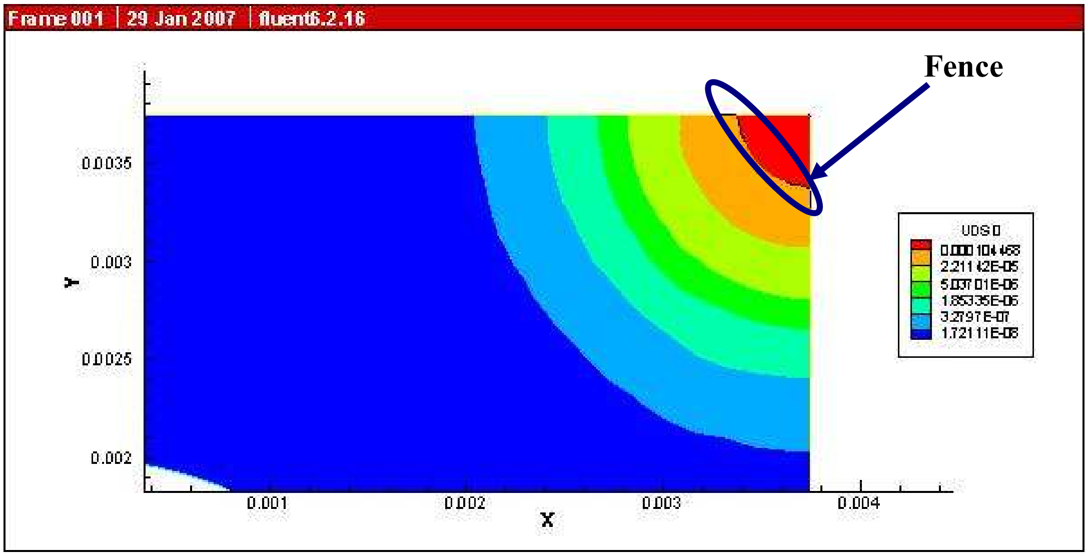


Figure 21: Concentration over space in the device with the fence (circled in blue)

In the contour diagrams, we can observe from the color maps that the concentration levels in the model without the fence exceed those of the model with the fence by at least two orders of magnitude.

The second property that we studied through simulations was the diameter of the cylinder. We used the 2D model for this study and made three versions of this model with circles of diameter 2mm, 3mm and 4mm. We ran each of these models to satisfy the criteria established in section *II.3.1* and observed the effects of the increasing diameter on the time each model took to reach the same concentration level. Firstly, the distance traversed by the diffusing substance from the corner to the hole is less and therefore, it diffuses faster. Secondly, the distortion in the gradient around the hole is greater due to the higher degree of curvature of the hole. Figure 22 illustrates these aspects of the diffusive flow. The y-axis has the time-points at which each of the models satisfies our established delta-epsilon criterion. We obtain an empirical relation between the cylinder diameter and the time required to approach similar concentration levels by fitting an appropriate equation to the data points using the method of least squares. The final equation that we get is

$$T_{oc} = 3E - 6D^2 - 0.0225D = 170$$

where T_{oc} denotes the time taken to reach the optimum concentration level according the delta-epsilon criterion established in section *II.3.1.*

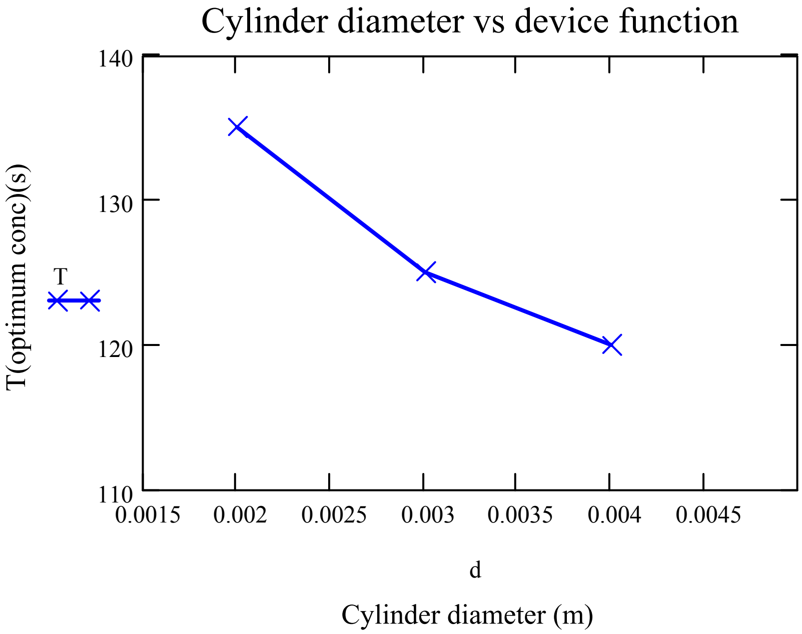


Figure 22: Variation in the time required by the 2D model to approach optimum concentration levels with variation in the diameter of the central hole

We also tested the 2D model for variation in concentration profiles with changes in diffusion coefficients. We vary the diffusion coefficient such that each new diffusion coefficient is less than the previous one by a factor of 0.1. This mode of variation is chosen for simplifying calculations as well as because the biological molecules that we ultimately want to use in our experiments have diffusion coefficients that can be expressed in terms of exponents of ten. We plot the logarithms of the diffusion coefficients instead of the actual diffusion coefficient values which are very small. The plot in Figure 23 shows the logarithm of the diffusion coefficients plotted against the time taken by each model to satisfy our established delta-epsilon criterion. We observe from this plot that the time taken to reach the same concentration level rises steeply with a decrease in the diffusion coefficient.

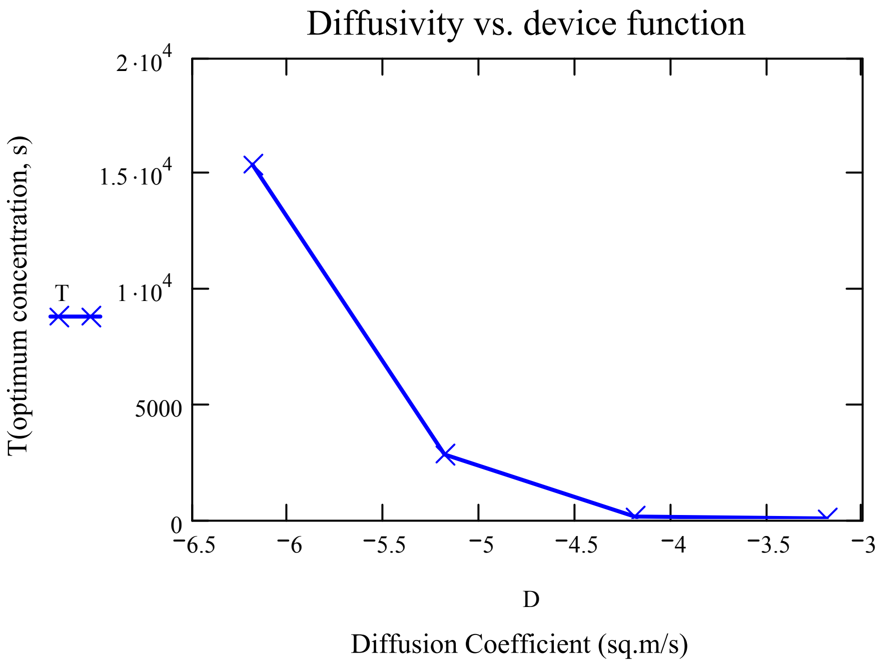


Figure 23: Variation in the time required by the 2D model to approach optimum concentration levels with varying diffusion coefficients

Using the method of least squares again, we derive the equation relating this time to the diffusion coefficient of the diffusing substance. The equation obtained is

$$T_{oc} = -(1197D^3 + 13741D^2 + 52262D + 65682)\,.$$

We ran a set of simulations with the 2D model varying the length of the square chamber. The range of lengths taken varies from 7500μm to 50000μm. The length of the side of the chamber is an important determinant of the time taken by the substance to travel to the central hole and then up the cylinder. Figure 28 shows the variation in the time required by the 2D model to satisfy our established criterion with increases in the length of the side of the base. Four cases have been taken with base-lengths of 7500μm, 10000μm, 30000μm and 50000μm. The increase in time required to reach steady-state increases sharply after the first two points as the difference between the lengths of the sides also increases from 2500μm to 20000μm.

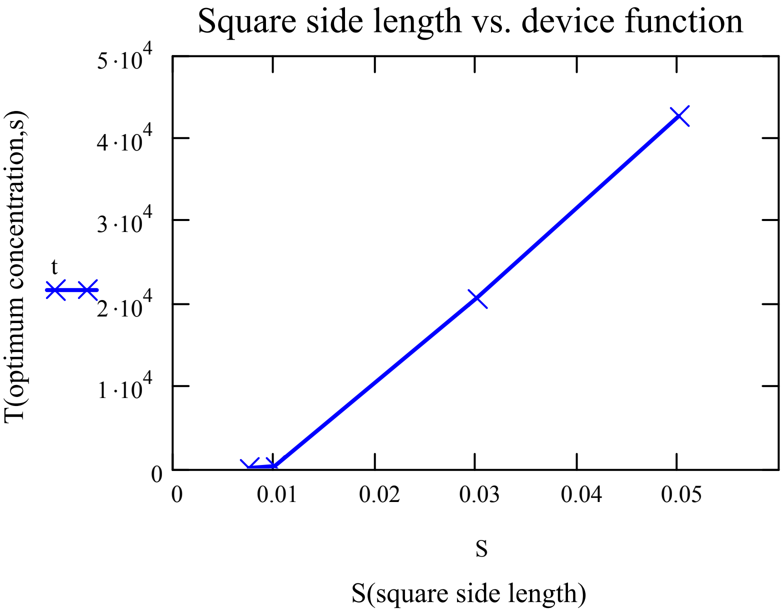


Figure 24: Variation in the time required by the 2D model to approach steady-state with different lengths for the square side

We make estimates to find the equation which gives the minimum regression value for this curve and use the method of least squares to obtain the coefficients. We obtain the following equation

$$T_{oc} = -9\times10^{8} S^{3} + 8\times10^{7} S^{2} - 1\times10^{6} S + 4605$$

Our last set of parametric simulations has the ratio of the cylinder's diameter to the chamber's length as the independent variable. The two major factors which determine the path of diffusion in the device are the diameter of the central hole and the length of the side of the base. The longer the length of the side of the base, the longer a substance takes to reach the central hole. The diameter of the hole also impacts the concentration gradient as the distortion of the gradient is higher with a larger diameter while the time taken by the substance to reach the central hole is shorter.

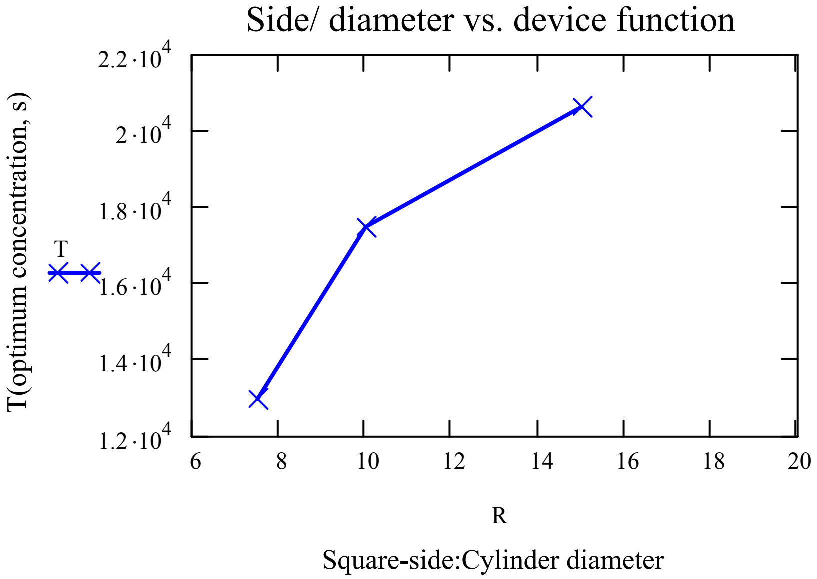


Figure 25: Variation in the time required by the 2D approximation system to approach steady-state with changes in the ratio of length of the base to diameter of the hole

However, the combined effect of these two parameters is not intuitive and so we ran the 2D model of the device with a base length of 3cm with three different ratios. The curve takes a form that is almost parabolic with a steady increase in the time required to approach steady-state with an increase in the ratio of side to diameter.

We use the method of least squares once more to find the relationship between the ratio of side to diameter and the time taken by each model to reach the optimum concentration. Fitting a polynomial of degree 2 to the data points we find that it is an almost perfect fit and we get the equation

$$T_{oc} = -158R^2 + 4566.5R - 12403.$$

II.4 Results from experiments in Fluent

II.4.1 3D grid with a semi-cylindrical reservoir

The validation of preliminary results in Fluent was the cue to proceed to the next step—simulation of actual experiments in Fluent with a model that emulates the physical device in somewhat more detail.

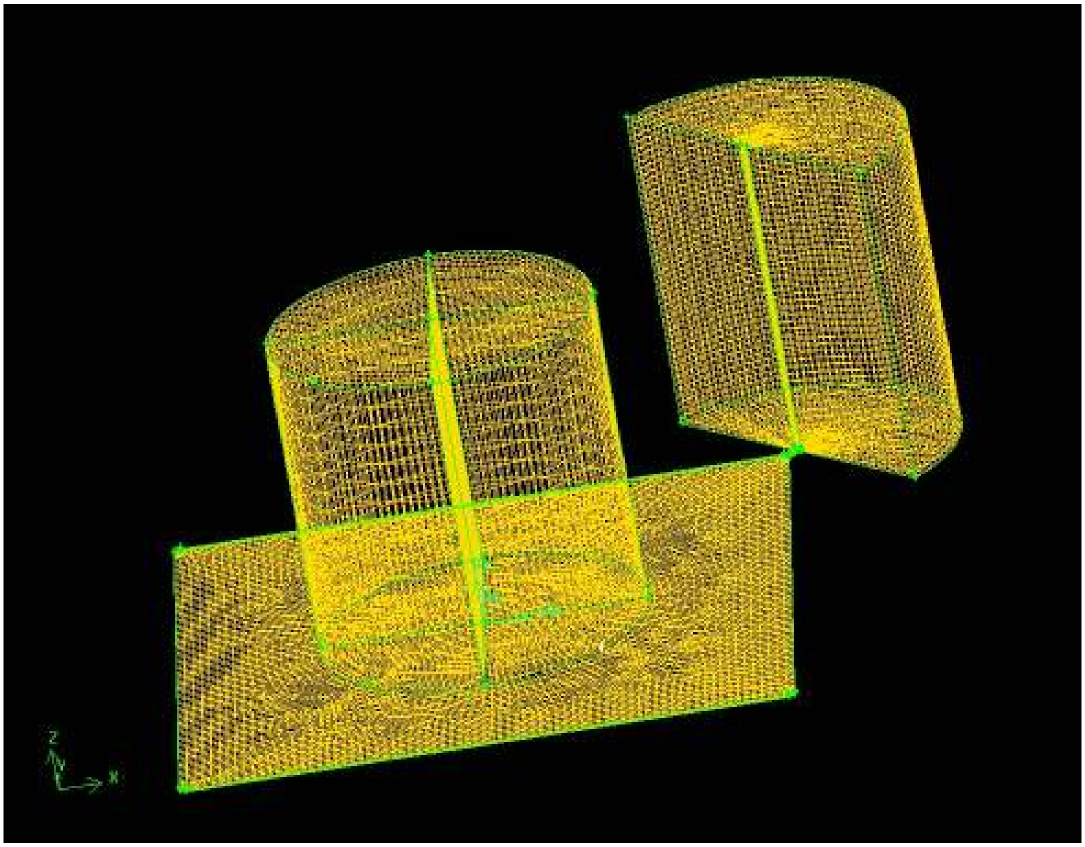


Figure 26: Mesh of the 3D model with a narrow channel and a semi-cylindrical reservoir

We developed the model with the semi-cylindrical reservoir as described in section II.2.5 and ran 2 sets of simulations for 950400 seconds of computational time which equals eleven days. Figure 26 shows the meshed grid of the 3D model with the channel and the reservoir. We initially plotted the profiles of concentration over space in the device at time intervals of 24 hours in Microsoft Excel. The points were extracted from the beginning of the channel to the middle of the base of the cylinder in Tecplot. The plot in Figure 27 shows the concentration profile of the diffusing substance at these points at every 24 hours of simulation time. There are steep drops in the concentration at two critical points of the device:

- The beginning of the channel to the beginning of the chamber
- The beginning of the chamber to the beginning of the cylinder

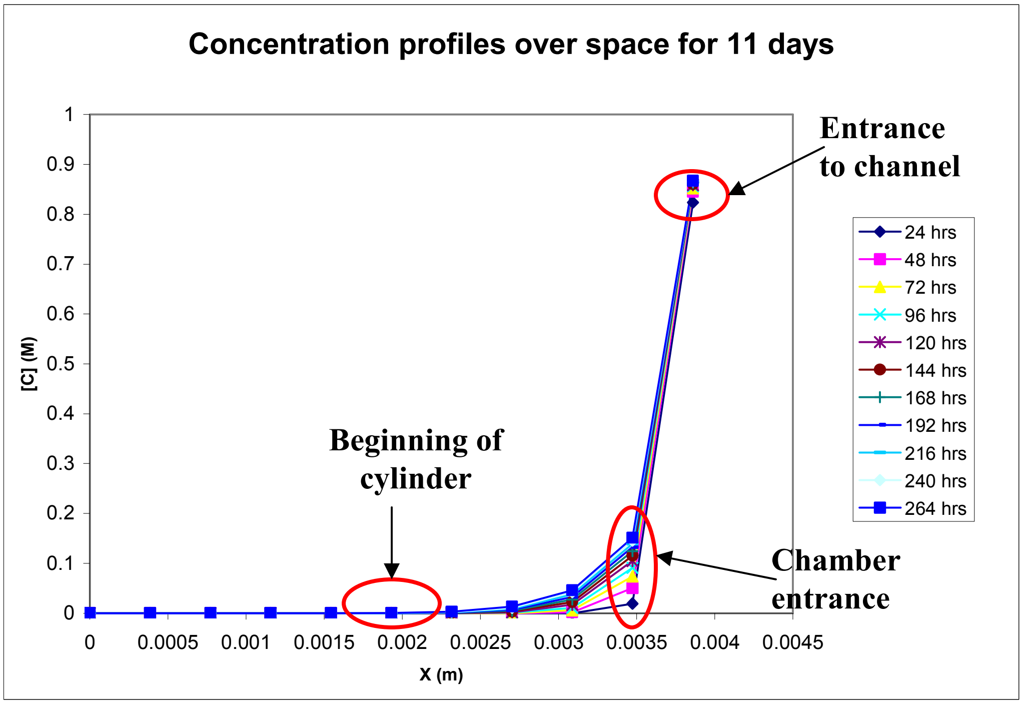


Figure 27: Concentration profile over space at every 24 hours in the device with the semi-cylindrical reservoir

We also plotted the concentration over space in the device beginning from the chamber instead of the channel to observe how the concentration rises in time at the entrance to the chamber. The plot in Figure 28 illustrates the same and we observe that there is steady increase in the concentration at the entrance to the chamber with each interval of one day. We also plotted the concentration profiles over space at intervals of three days in the device. In this case, we plotted the logarithms of the concentration values instead of the actual concentration values which are very small, in the order of 10^{-22}. The trend of concentration values decreases in steepness, as expected, with increase in time. The plot in Figure 29 shows the graph with the logarithms of concentration values plotted over space at three different time-points, spaced three days apart.

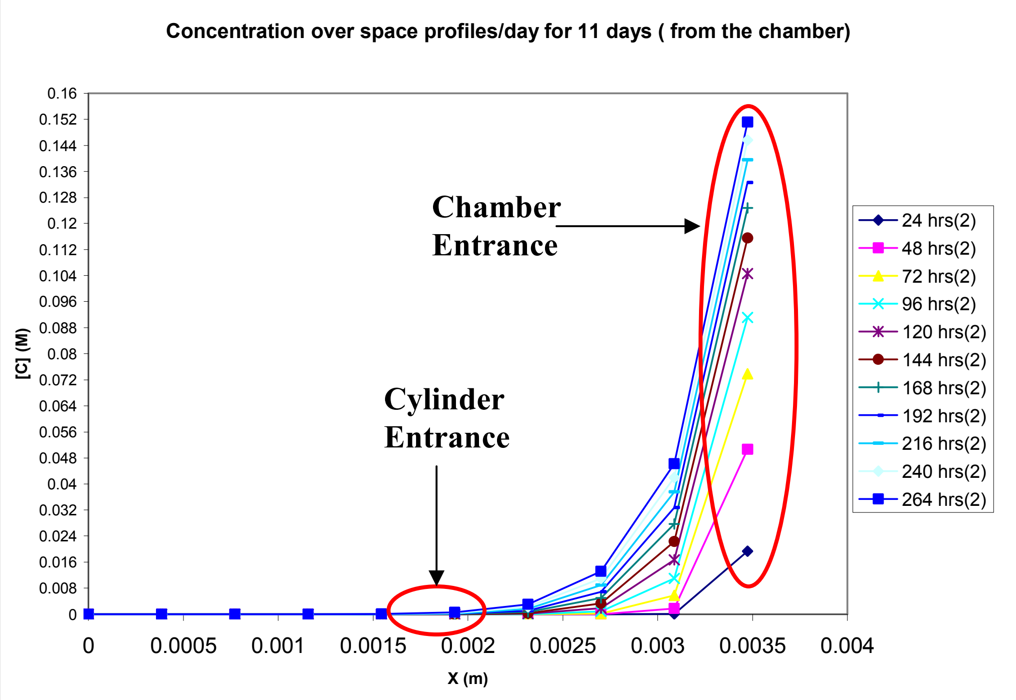


Figure 28: Concentration profile over space in the device with the semi-cylindrical reservoir beginning with the entrance to the chamber

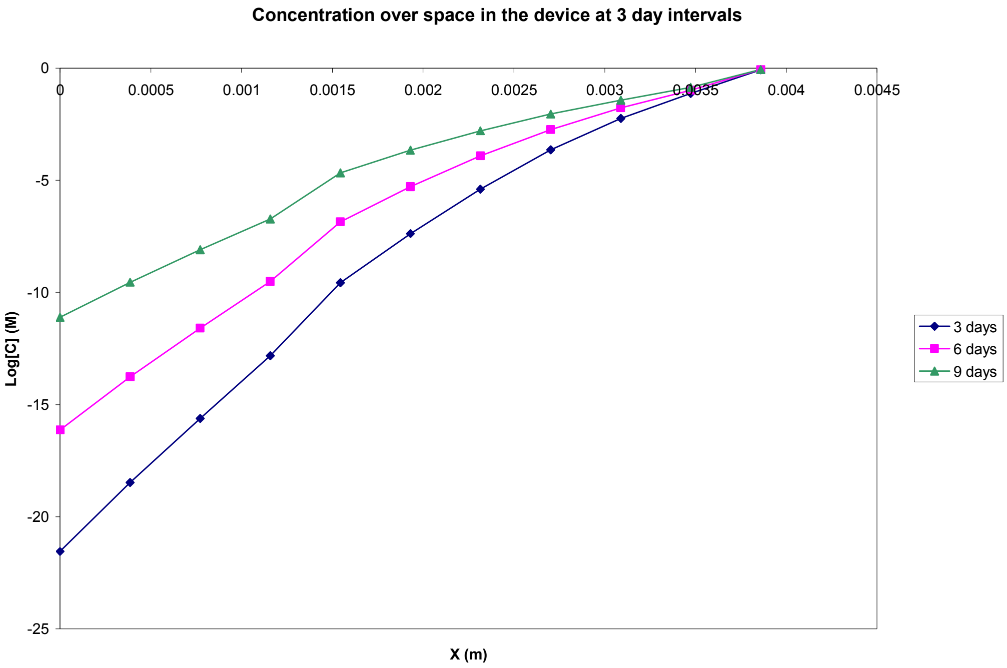


Figure 29: Logarithms of concentration values plotted against space at three time-points, spaced three days apart

II.4.2 Concentration profiles in devices with varying channel widths

Our most significant objective in performing computational simulations in Fluent is to examine the effect of the restriction channel on the concentration levels and gradient formation in the device. We modified the previous 3D model, as described in section *II.2.5*, and ran each case for 950,400 seconds or 11 days and observed the concentration profiles in each model with a different channel width. In order to minimize numerical diffusion in each case we have used different initial concentrations which have been summarized in Table 3. The range of concentration values required for maintaining cells in the device for the duration of an experiment (11 days) is 15-19ng/ml which can be expressed as 6.67E-10-7.13E-10M/m^3. The maximum concentration of the substance in the reservoir cannot exceed the order of millimoles. Fluent takes concentration input in the form of kilomoles which implies that the value used for initial concentration would have been 6.67E-06 moles/l. However, due to numerical diffusion, which we shall explain in greater detail in Appendix D, we have scaled up the values of initial concentration used in the simulations by 3-5 orders of magnitude.

Table 3: Initial concentration values corresponding to each device with different channel widths

Channel Width (µm)	Initial Concentration (M)
20	6.67E2
40	6.67E3
80	6.67E1
150	6.67E1
200	6.67E1

In these experiments we extracted data along different geometrical entities:

- Lines, extracted along the diagonal from the chamber entrance to the beginning of the cylinder
- Rakes, which are similar to lines, with the exception that we can specify the number of points on them
- Planes along the surface of the chamber, at a height halfway along the z -axis

Figure 30 displays the results from the device with the 20 micron channel. This channel is extremely narrow compared to the rest of the device and offers a high degree of restriction to the entering fluid. We have displayed the concentration level of fluid in the device at one, three, five, nine and eleven days on a line running diagonally through the device.

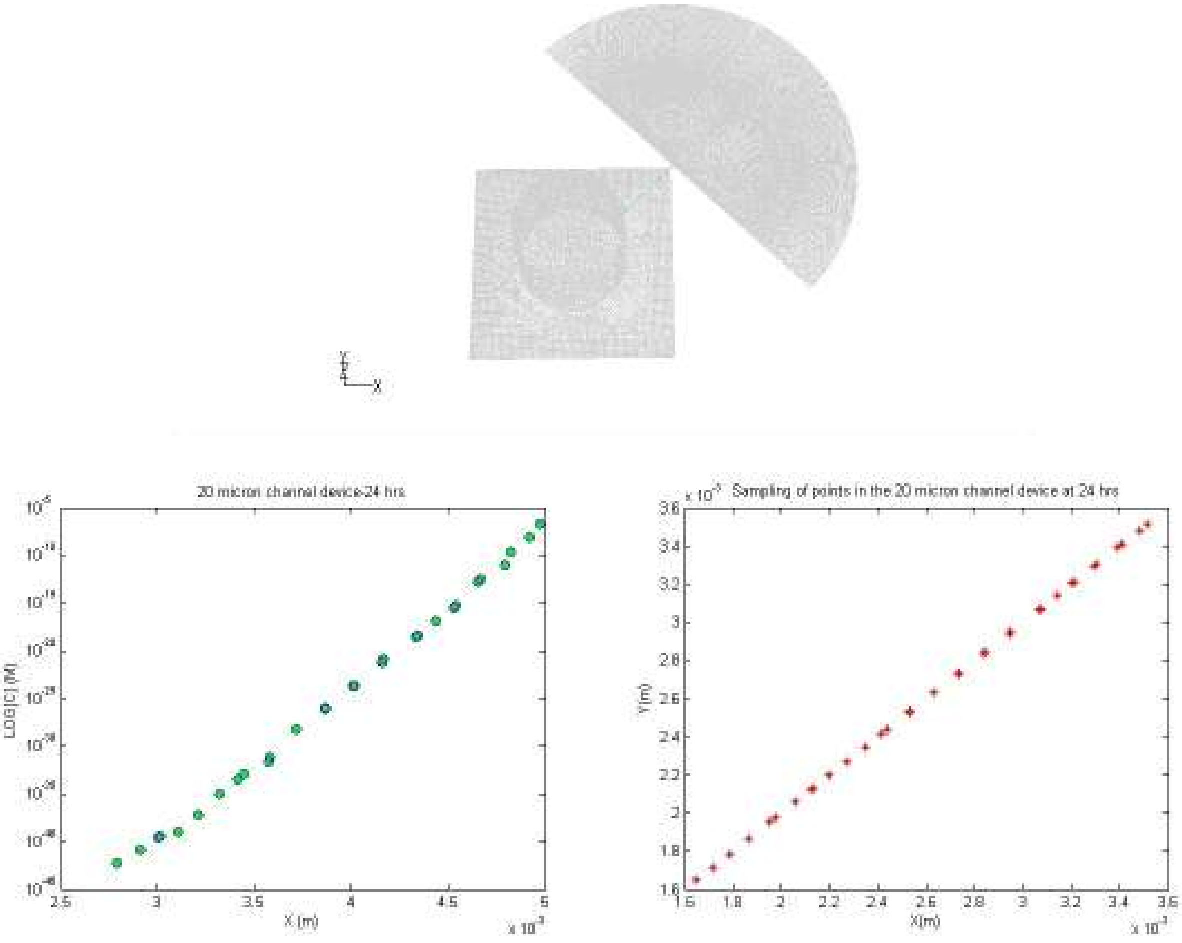


Figure 30: Logarithm of concentration values plotted over space in the model with the 20 micron channel at 1, 3, 5, 7, 9 and 11 days

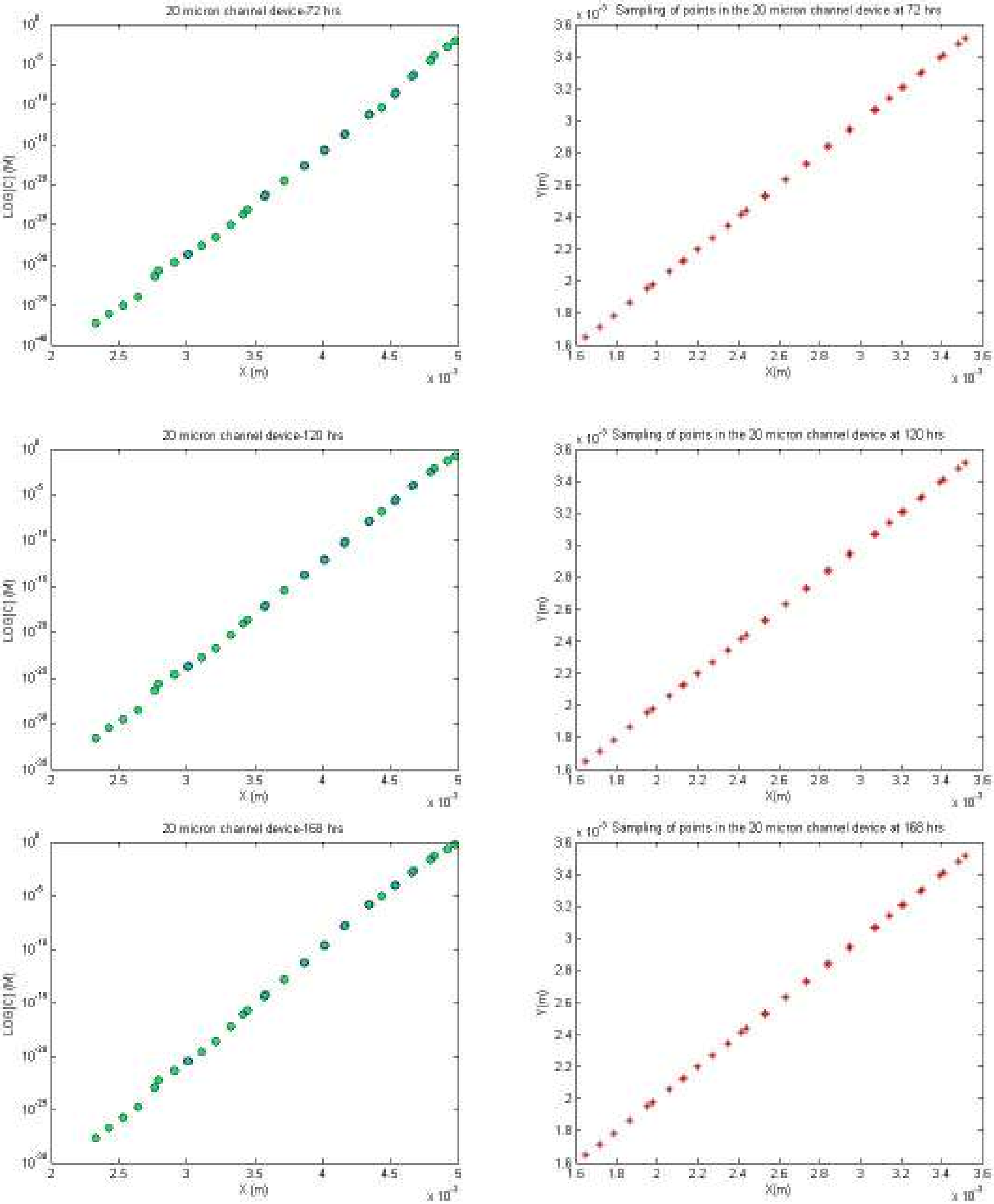


Figure 30-contd.

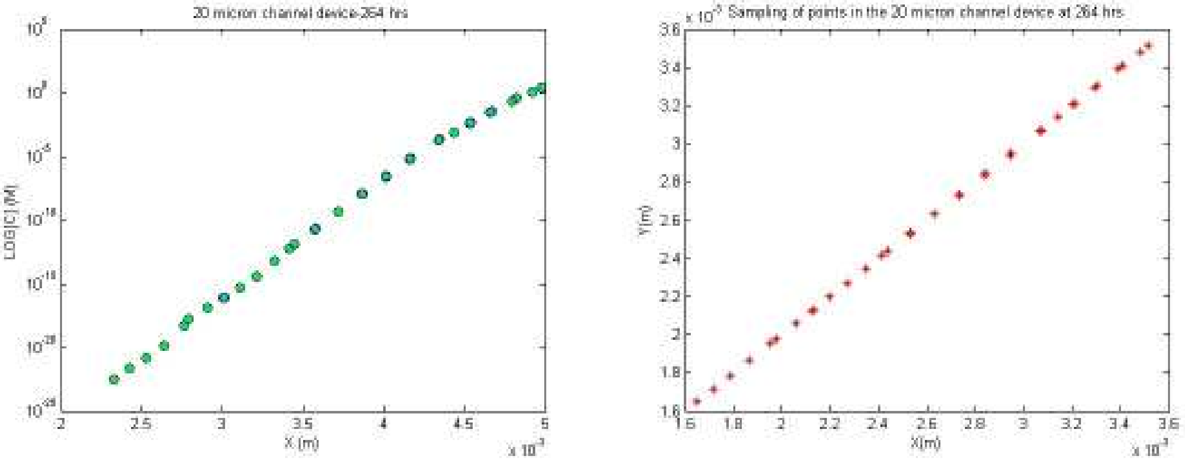
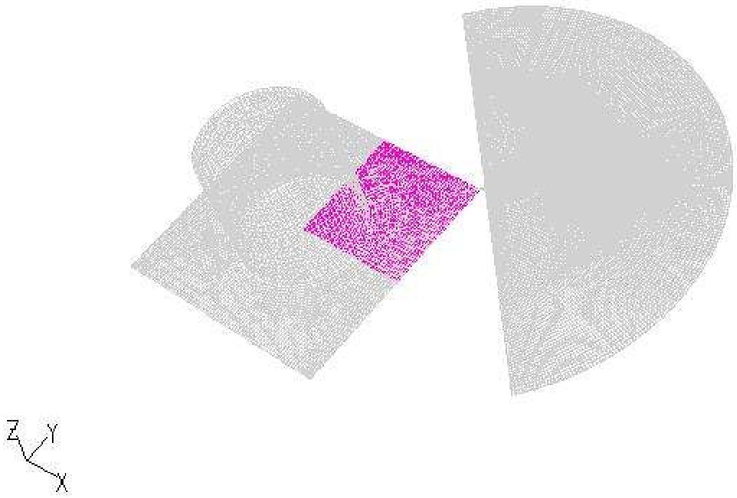


Figure 30-contd.

The points in Figure 30 have been sampled along a line which runs diagonally along the length of the chamber, beginning at the entrance and stopping just before the edge of the cylinder. A plot of the points extracted from the lines has been given beside each plot illustrating a concentration profile. The concentration plots start showing greater distortions after values greater than 1E-25 as this is the smallest value processed by Matlab. The logarithms have been plotted against the radial coordinates of the extracted points while the sampled points have been shown in terms of x and y coordinates. In the figures, we see that the points sampled are not uniformly spaced as they are extracted at all cells in the grid through which our specified line passes. Consequently, the logarithms of the concentration values do not follow a smooth linear trend at the initial time-points (24 hrs). The phenomenon of numerical diffusion has been explained in greater detail in Appendix D. The profiles and sampling of points gets smoother with time as numerical diffusion decreases. Figure 31 shows contour plots of the same model where we have extracted all the data on a plane extending from the center of the cylinder till the end of the chamber. It is halfway across the height of the chamber, as shown in the mesh plot in

Figure 31.The contour plots show a steady increase in the concentration levels within the plane over time.



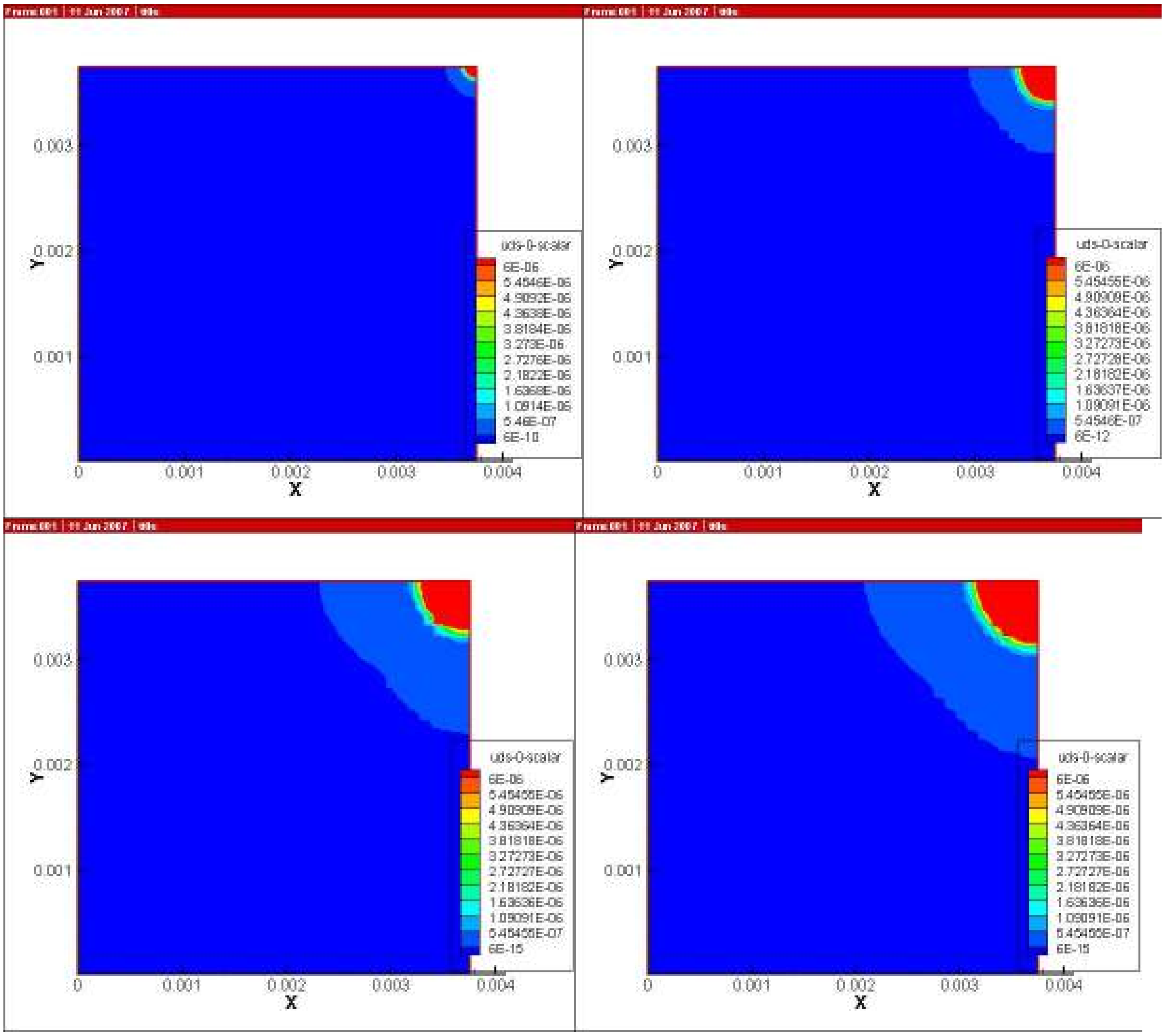


Figure 31: Concentration levels in the model with the 20 micron channel on a plane at a height of 1.3µm across the length of the chamber and extending radially from the center of the cylinder

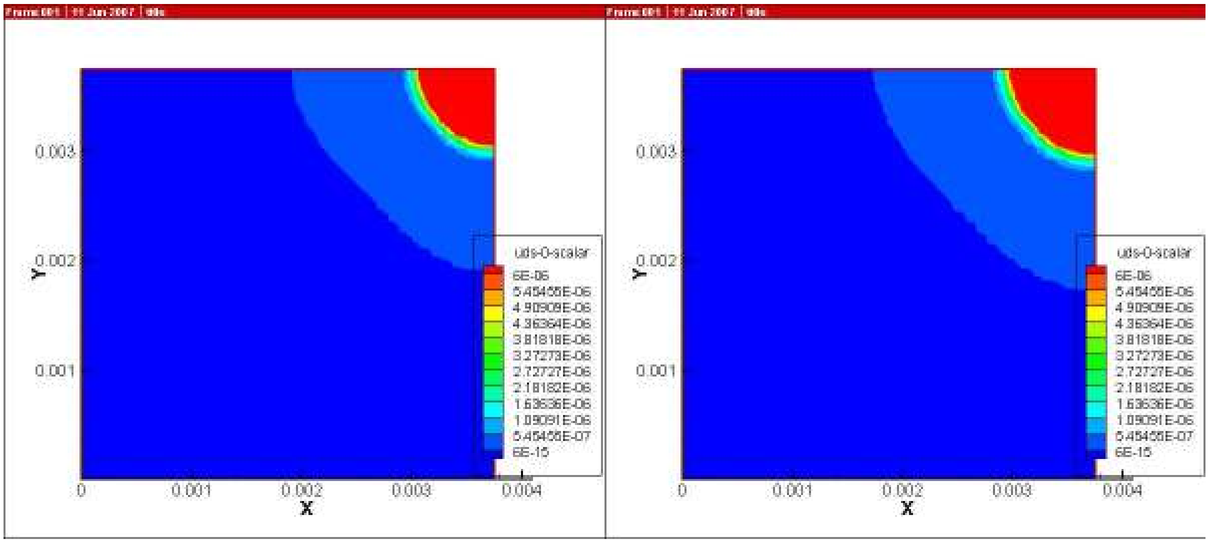


Figure 31-contd.

We have plotted contour levels ranging in concentration values from 6.67mM to 6.67pM as this is the relevant range of concentration values that will be used in actual experiments. Distortion in contour levels is the result of numerical diffusion as well as the contour algorithms used in all standard programs. These plots have been done using Tecplot 10.0. Figures 32 and 33 show similar results from a model with a channel of width 40 microns. We first plotted the logarithms of concentration values at the diagonal over space at different time-points. We also plotted the contours of concentration levels within the plane as in the previous model. The concentration levels in the model with the 40µm wide channel are only around ten times higher than those in the model with the 20µm wide channel. This change is most noticeable towards the corner of the chamber and approximately till halfway across the distance from the corner to the edge of the cylinder. This effect can be observed more clearly in the contour plots.

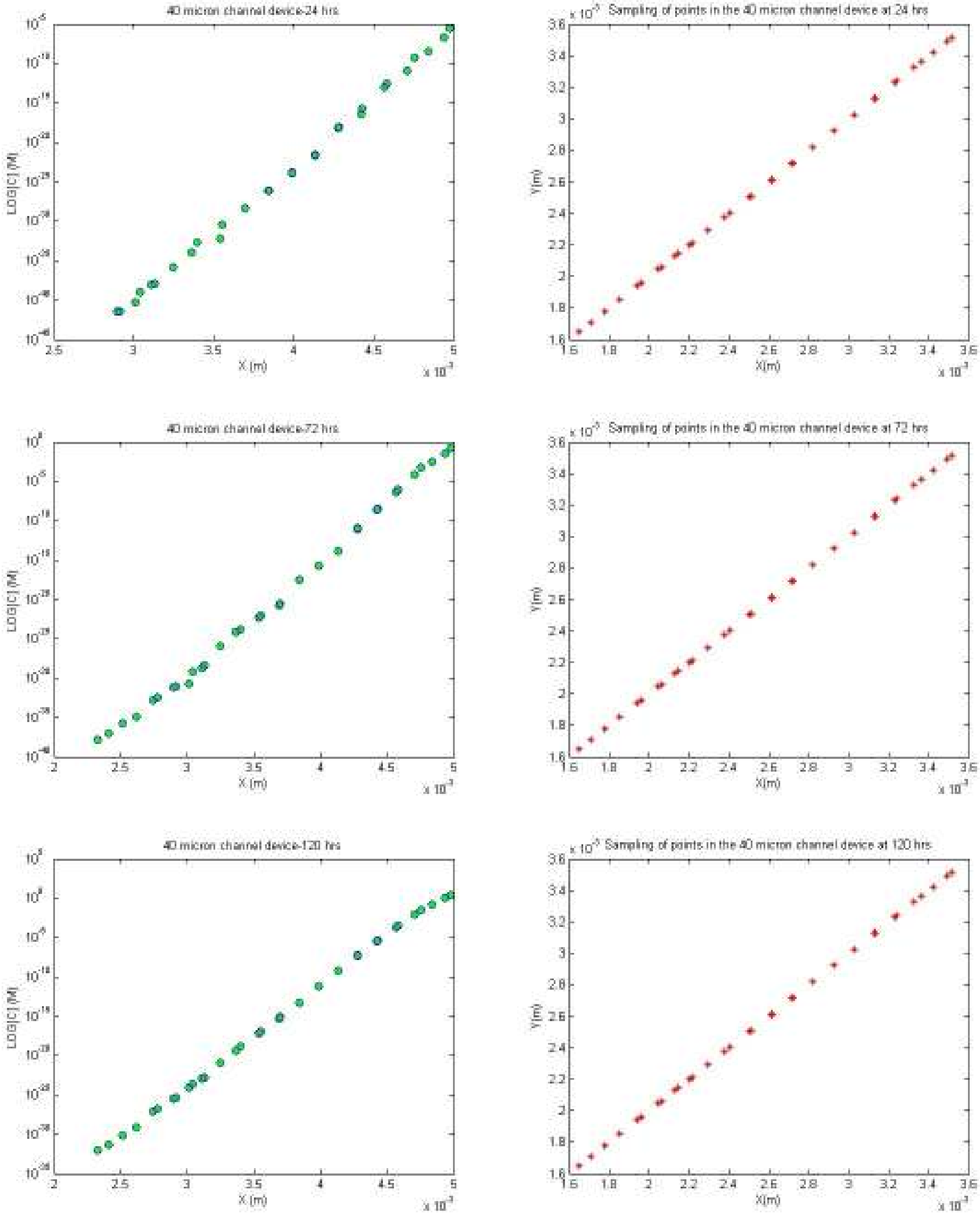


Figure 32: Logarithm of concentration values plotted over space in the model with the 40 micron channel at 1, 3, 5, 7, 9 and 11 days

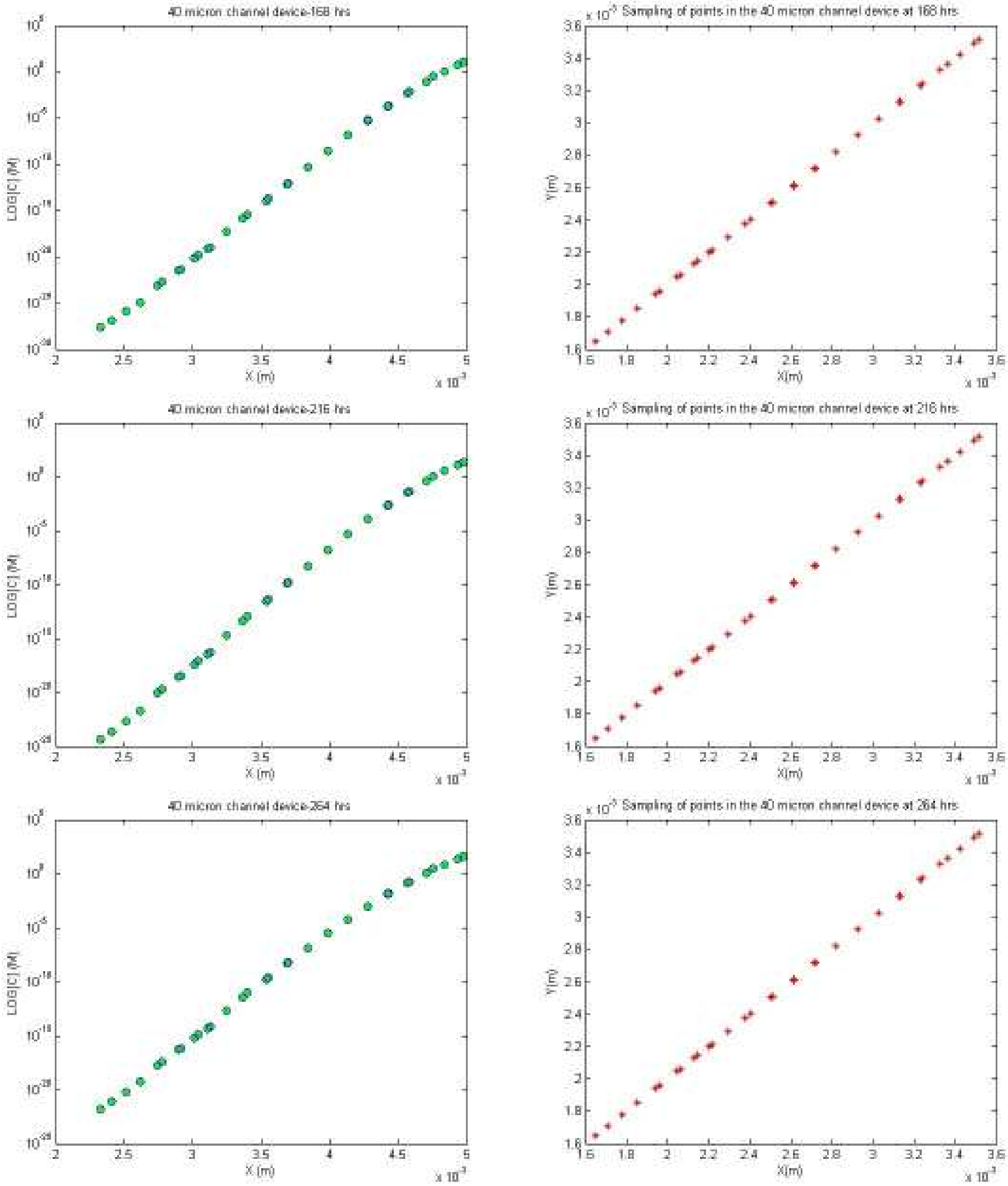


Figure 32-contd.

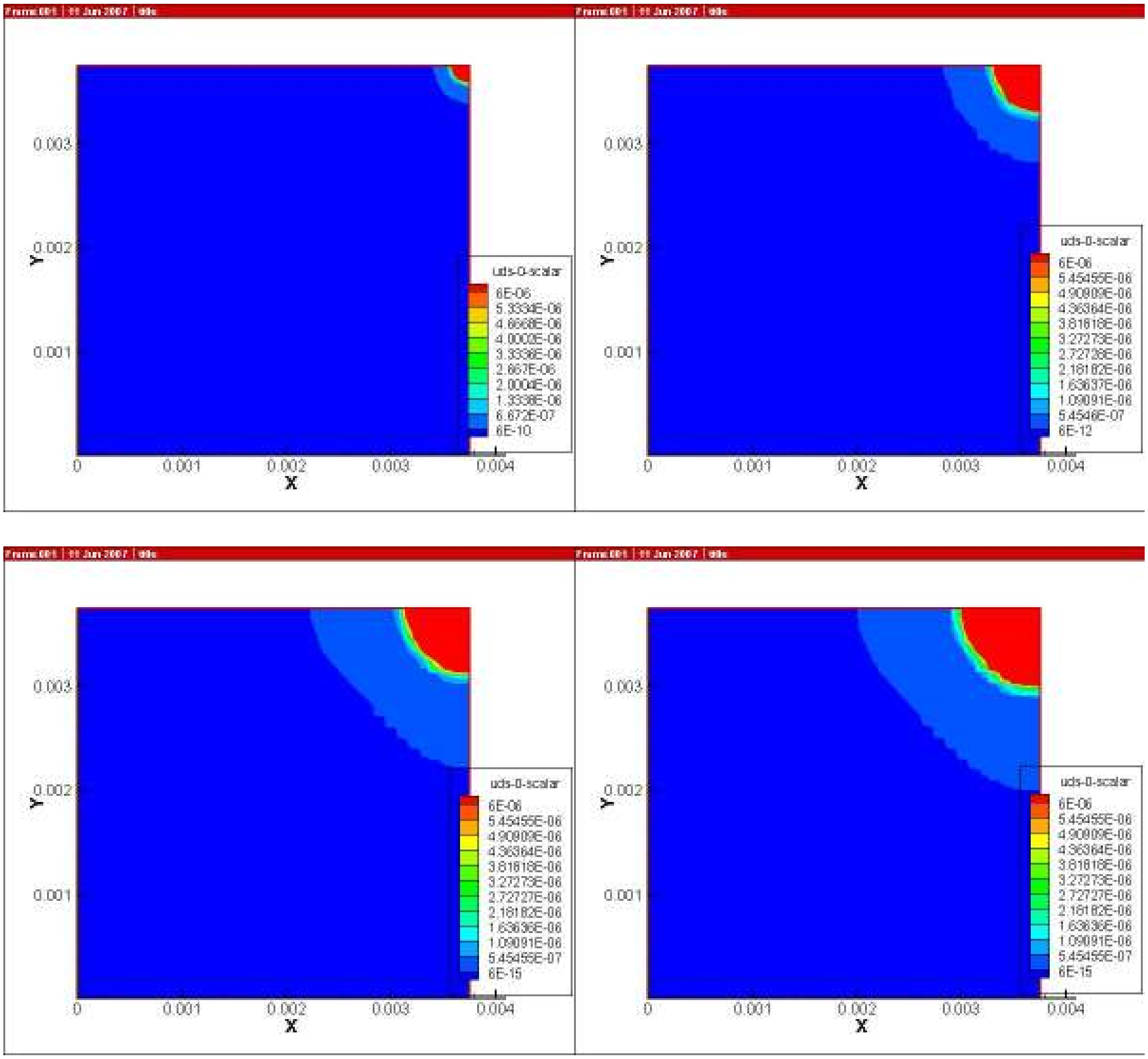


Figure 33: Concentration levels in the model with the 40 micron channel on a plane at a height of 1.3µm across the length of the chamber and extending radially from the center of the cylinder

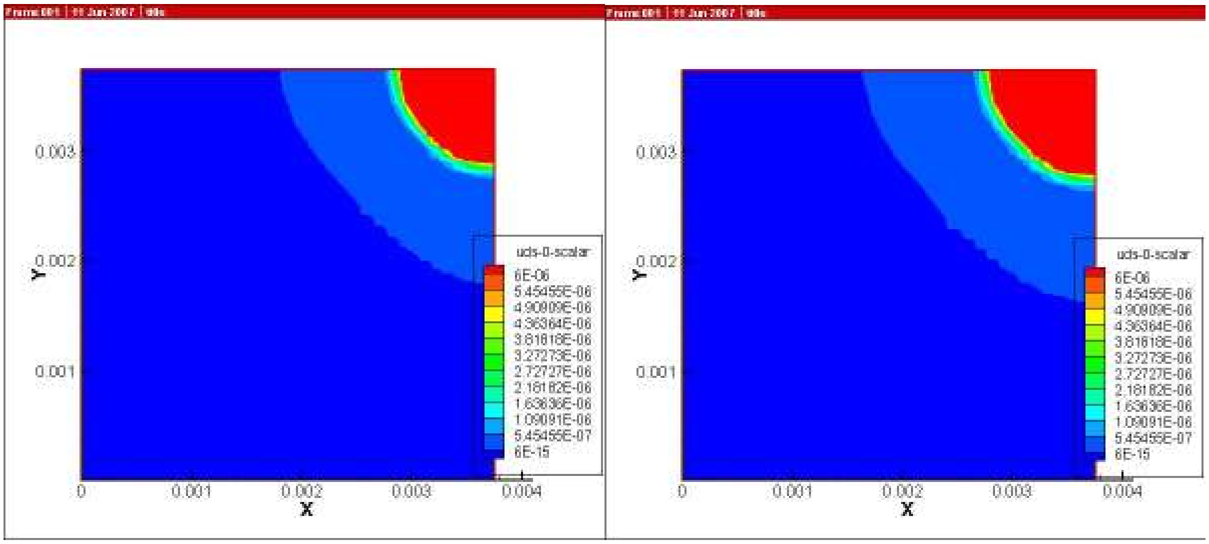


Figure 33-contd.

We can observe in the above contour plots that the distortion in contour levels has decreased compared to the levels in the device with the 20µm wide channel. It continues to be seen in contour levels corresponding to lower values of concentration. The values for concentration decrease exponentially over space. Hence, we do not specify a single delta-value by which Tecplot can calculate the contour levels; instead we specify the number of contour levels which allows Tecplot to directly calculate the delta-values for each contour level. We plotted the results of the model with a channel of width 80 microns in a similar way. Figures 35 and 36 display the results from this simulation. We observe that with an increase in the channel width to 80µm there is a considerable increase in concentration levels within the device. The higher width of the channel allows a greater amount of material to flow into the chamber at any given time-point. This is evident from both the semilog as well as the contour plots where we observe that the concentration levels increase at a much faster rate and the distortion in the logarithm values has also decreased. The device reaches a maximum concentration 6 mM in a day unlike in the 20µm device where it takes at least 30 hours.

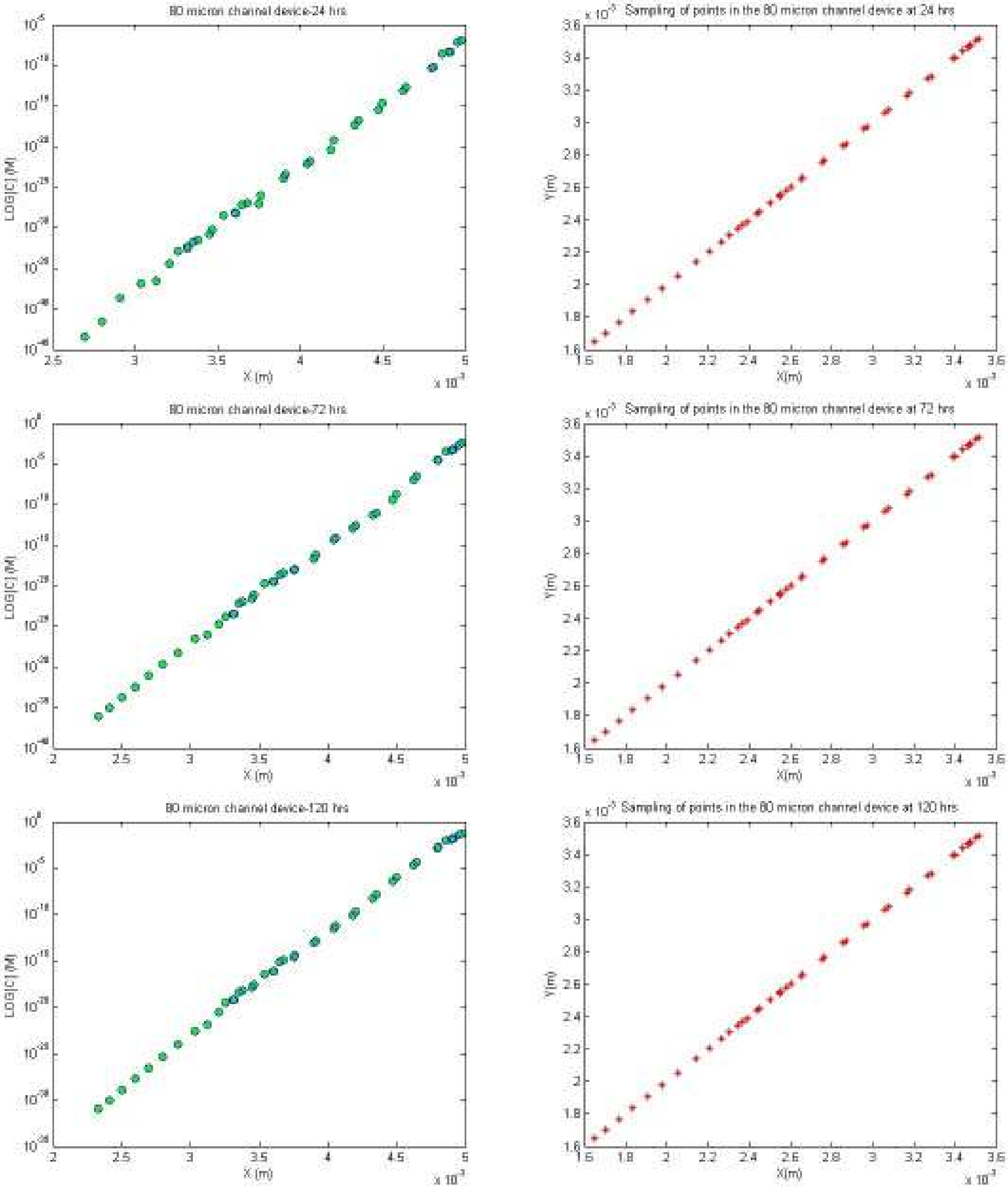


Figure 34: Logarithm of concentration values plotted over space in the model with the 80 micron channel at 1, 3, 5, 7, 9 and 11 days

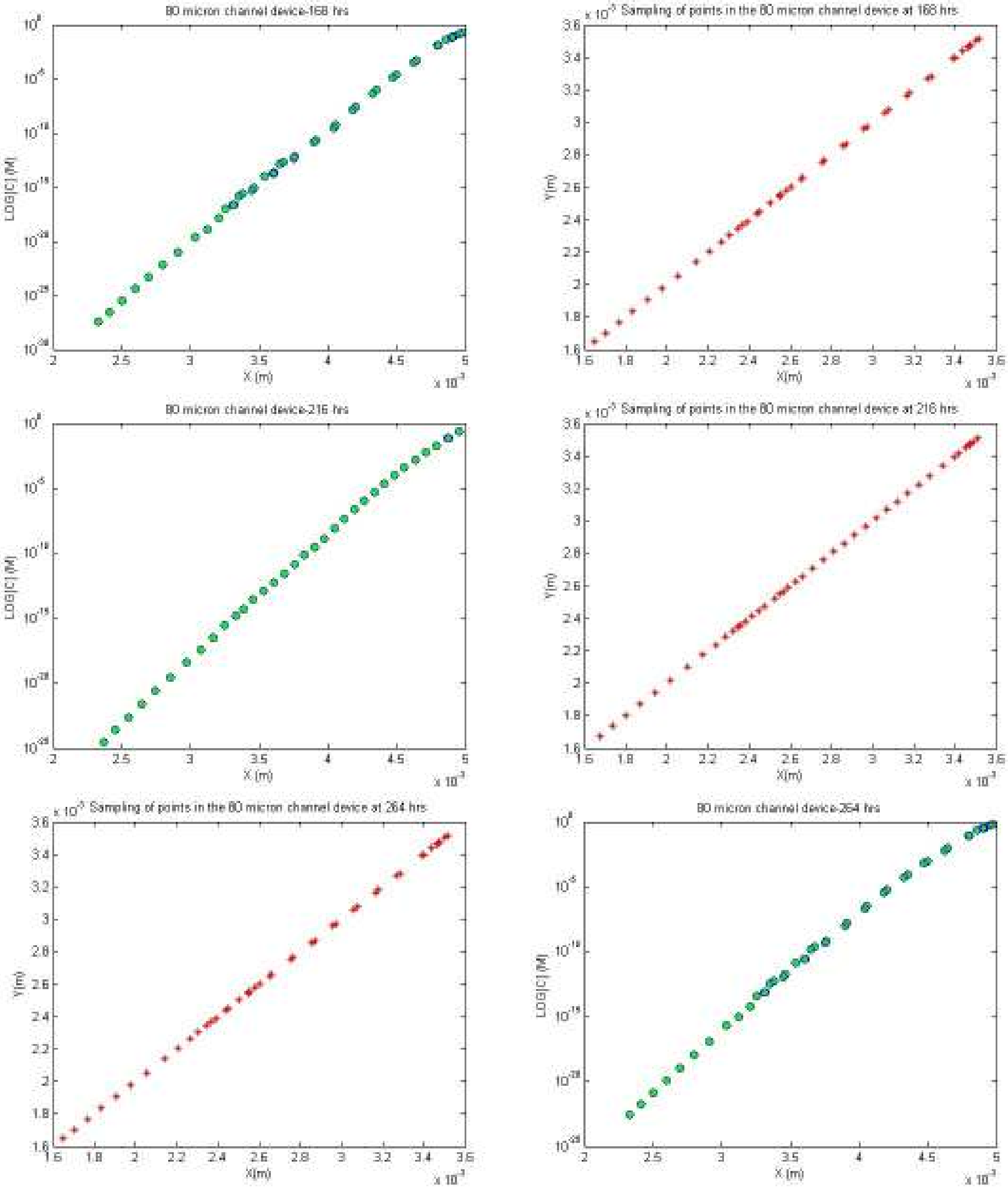


Figure 34-contd.

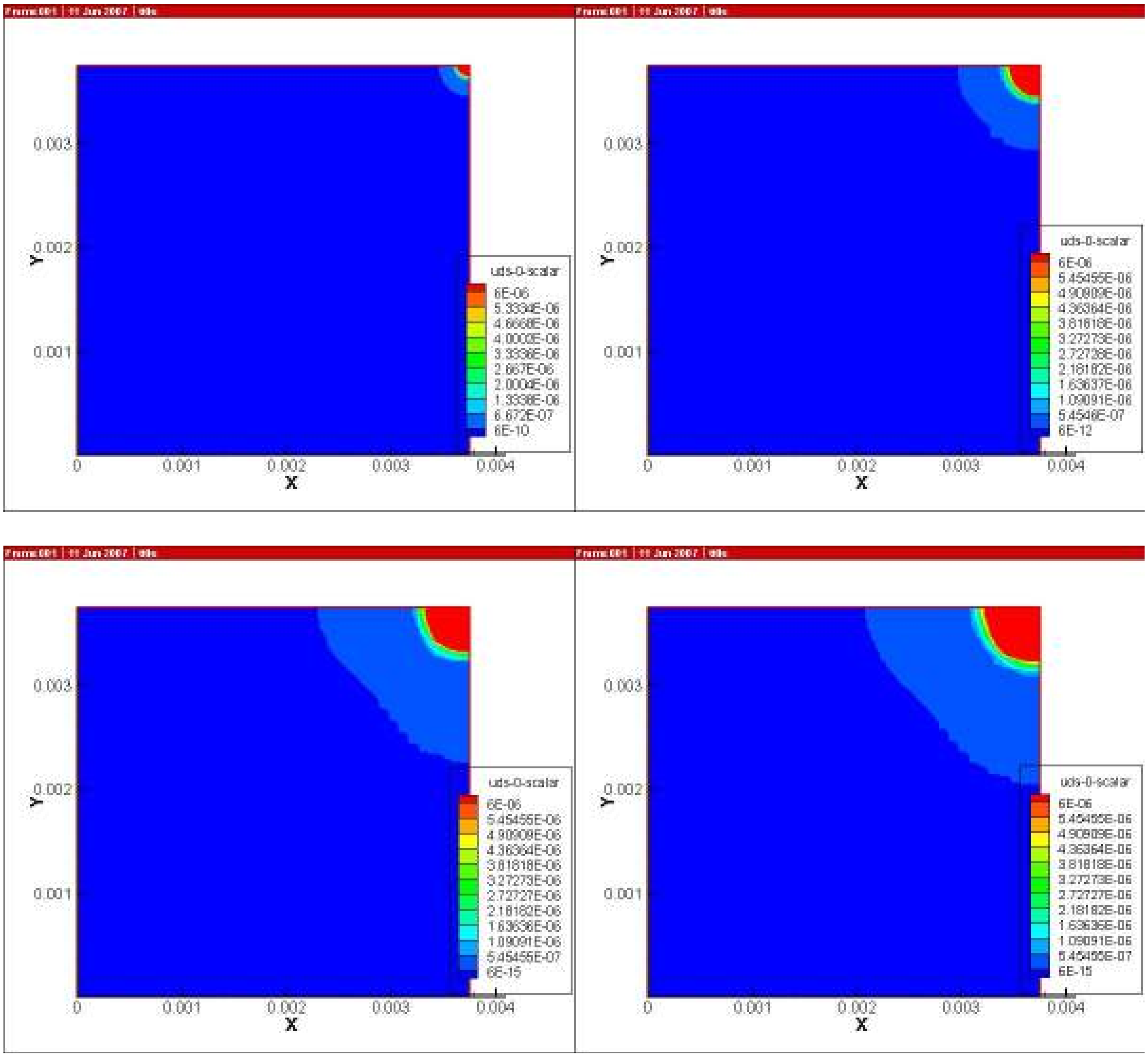


Figure 35: Concentration levels in the model with the 80 micron channel on a plane at a height of 1.3μm across the length of the chamber and extending radially from the center of the cylinder

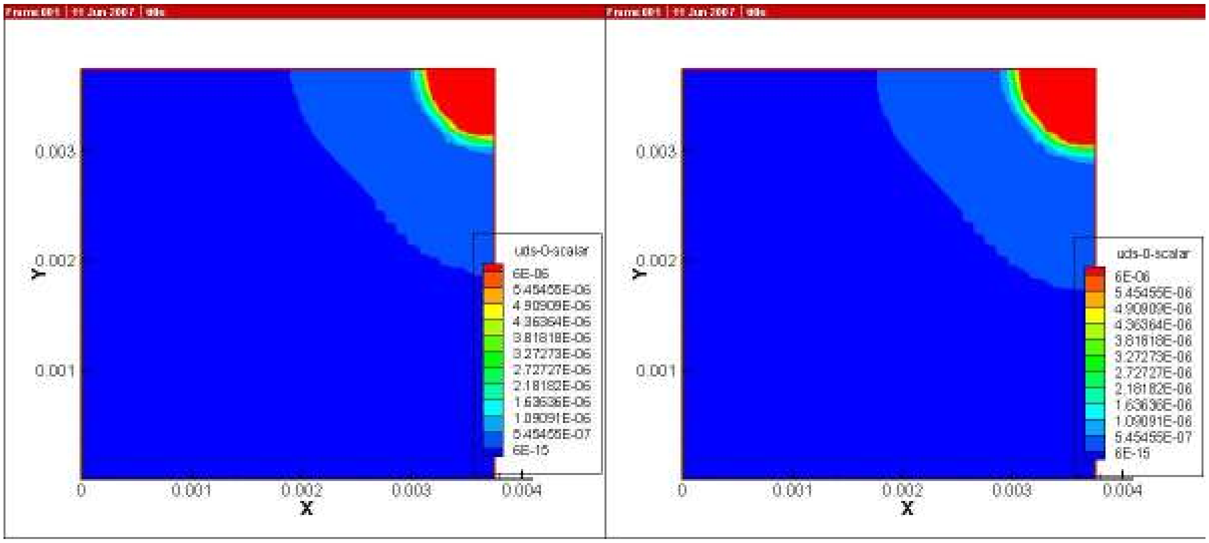


Figure 35-contd.

Our next simulation used a model with a channel of width 150μm. We can observe from the results in figures 37 and 38 that an increase in the channel width by 70μm increases the rate at which the concentration increases within the chamber.

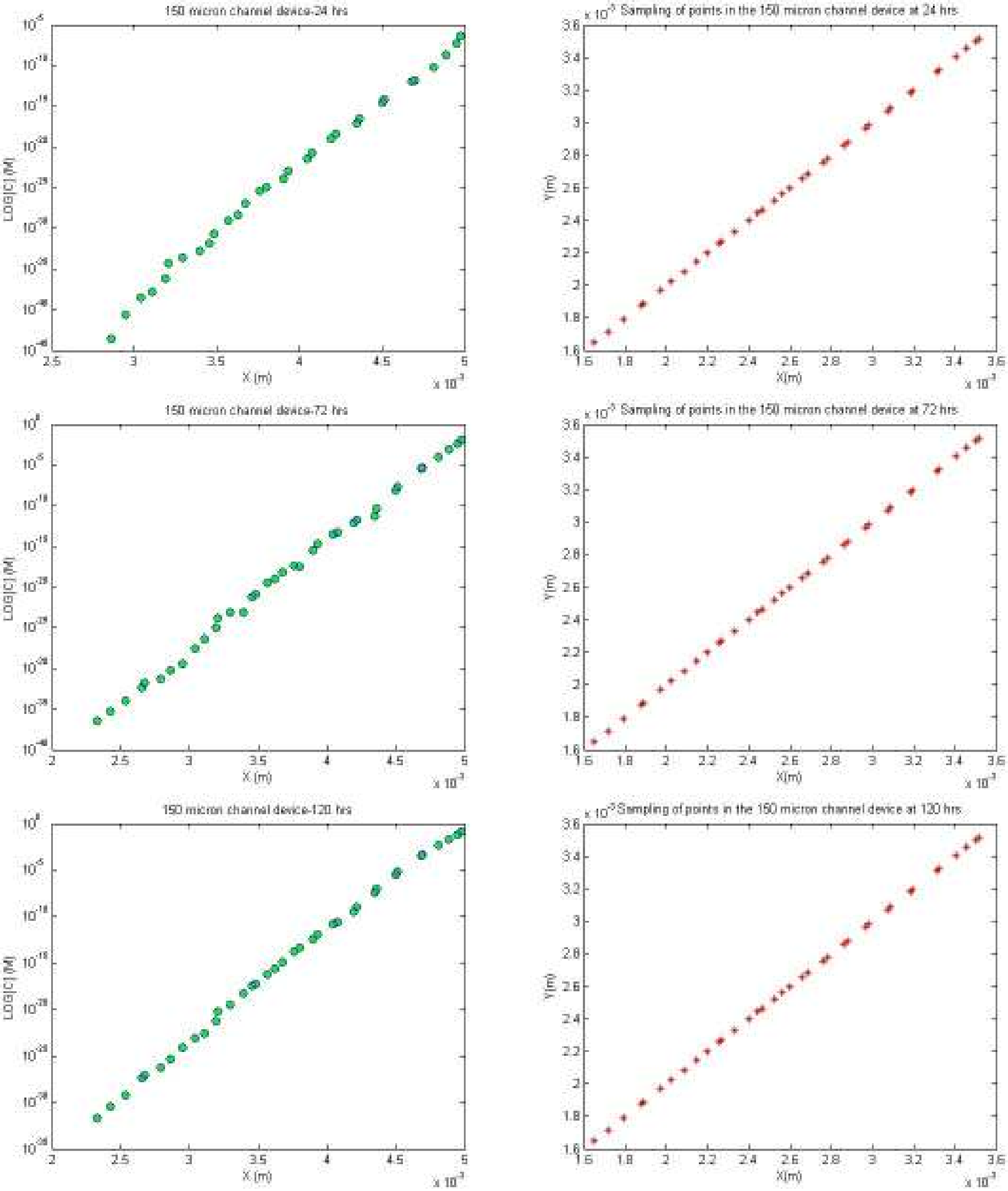


Figure 36: Logarithm of concentration values plotted over space in the model with the 150 micron channel at 1, 3, 5, 7, 9 and 11 days

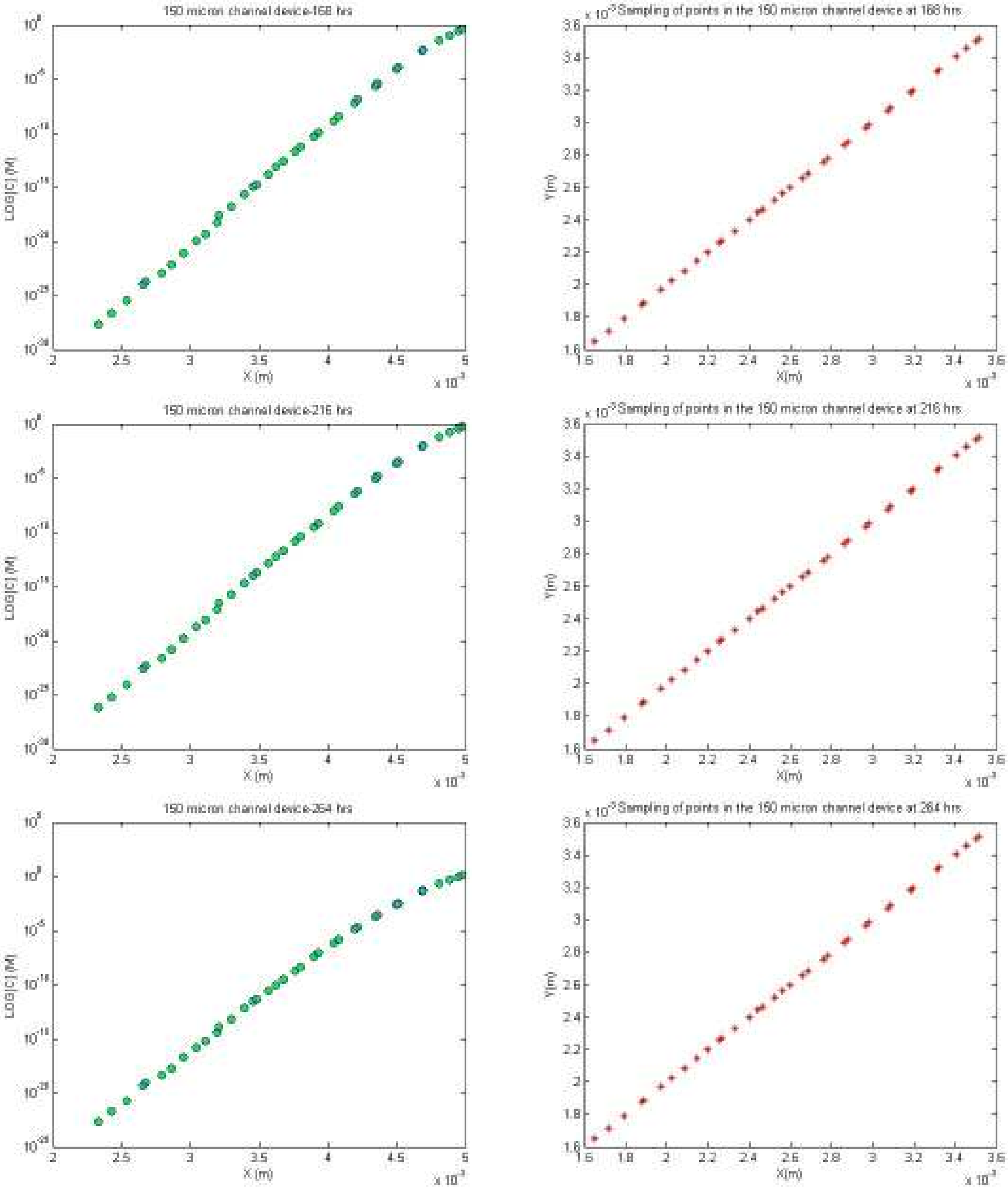


Figure 36-contd.

The concentration at the entrance to the chamber rises to millimolar level in less than 18 hours. The distortion in values due to numerical diffusion is also considerably decreased during the initial time-points. Comparing the plot at 24 hours in this device to the plot for the device with the 20µm wide channel at 24 hours, we can see that numerical diffusion is considerably decreased in the former.

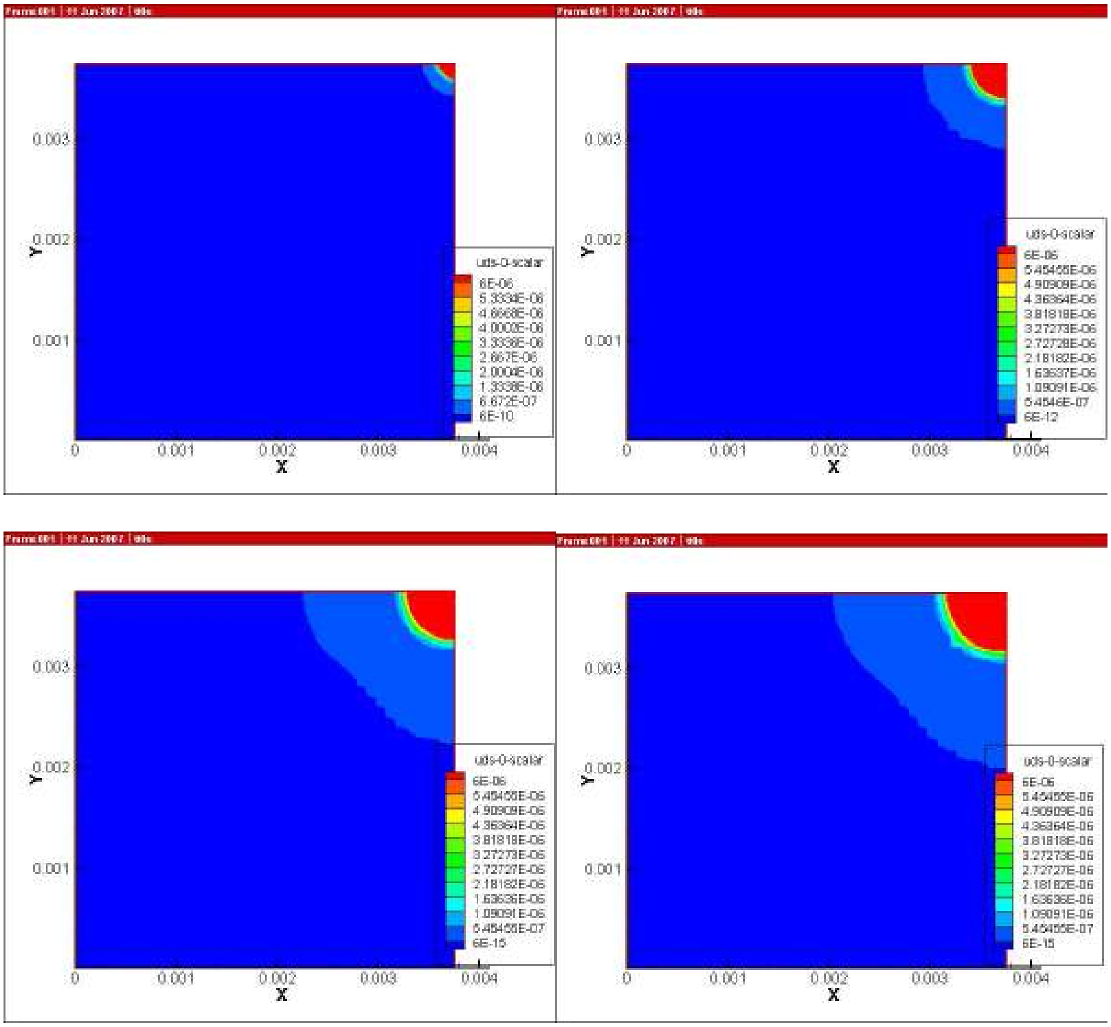


Figure 37: Concentration levels in the model with the 150 micron channel on a plane at a height of 1.3µm across the length of the chamber and extending radially from the center of the cylinder

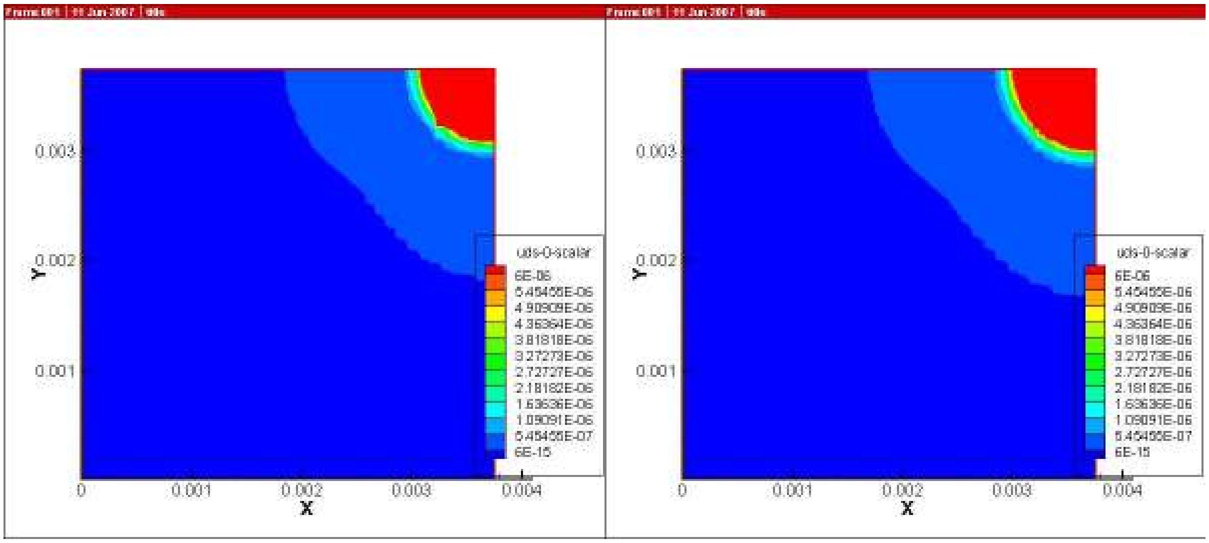


Figure 37-contd.

Our last simulation was with a model with a channel of width 200μm and this is the highest channel width that we have used among all our simulations. The results in Figures 38 and 39 show that there is not a significant difference in terms of concentration levels between the devices with 150μm and 200μm wide channels. However, there is a marked difference in the smoothness of the semilog plots. We observe that from 72 hours the logarithms of the concentration values form a smoother plot than in the previous models.

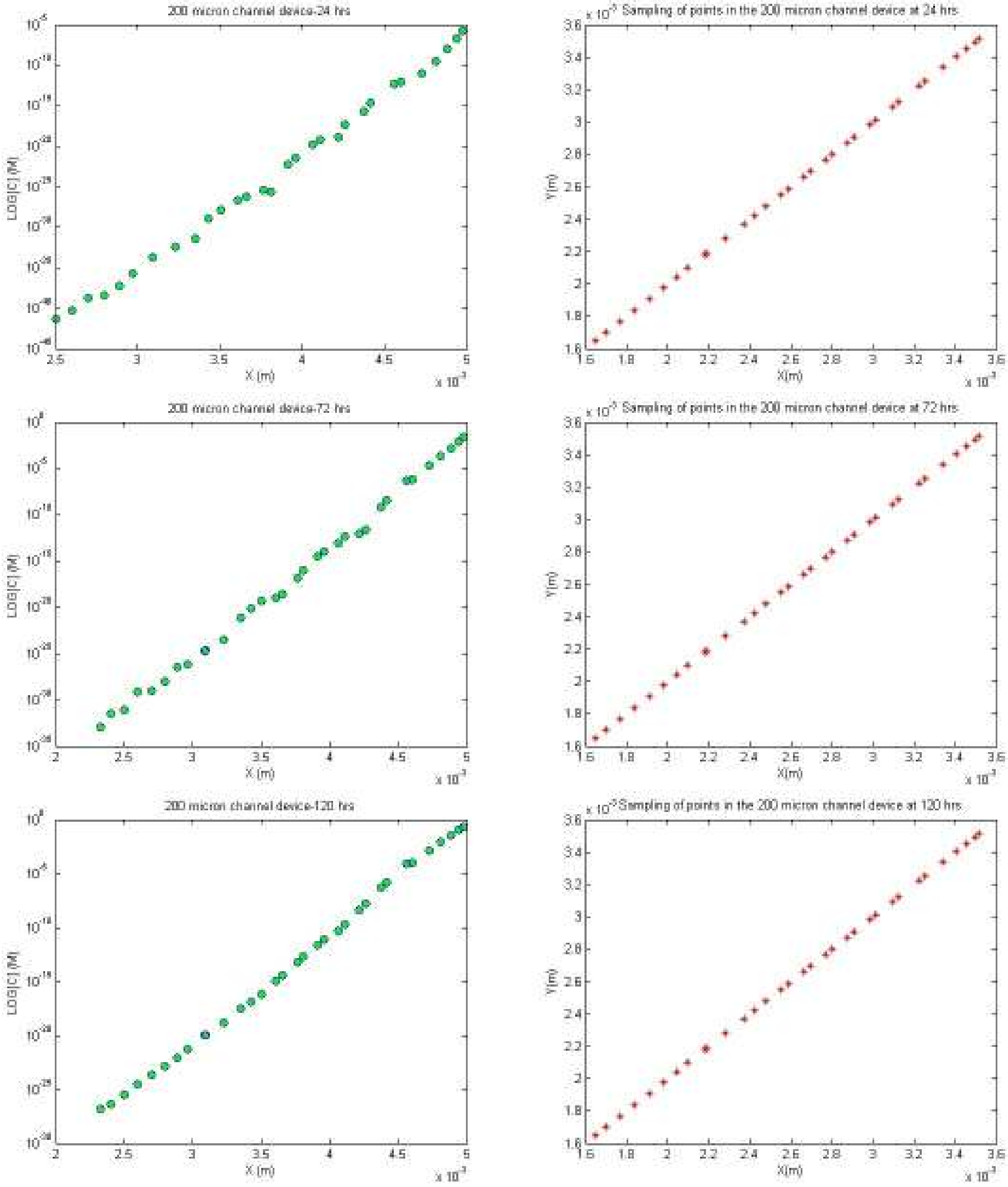


Figure 38: Logarithm of concentration values plotted over space in the model with the 200 micron channel at 1, 3, 5, 7, 9 and 11 days

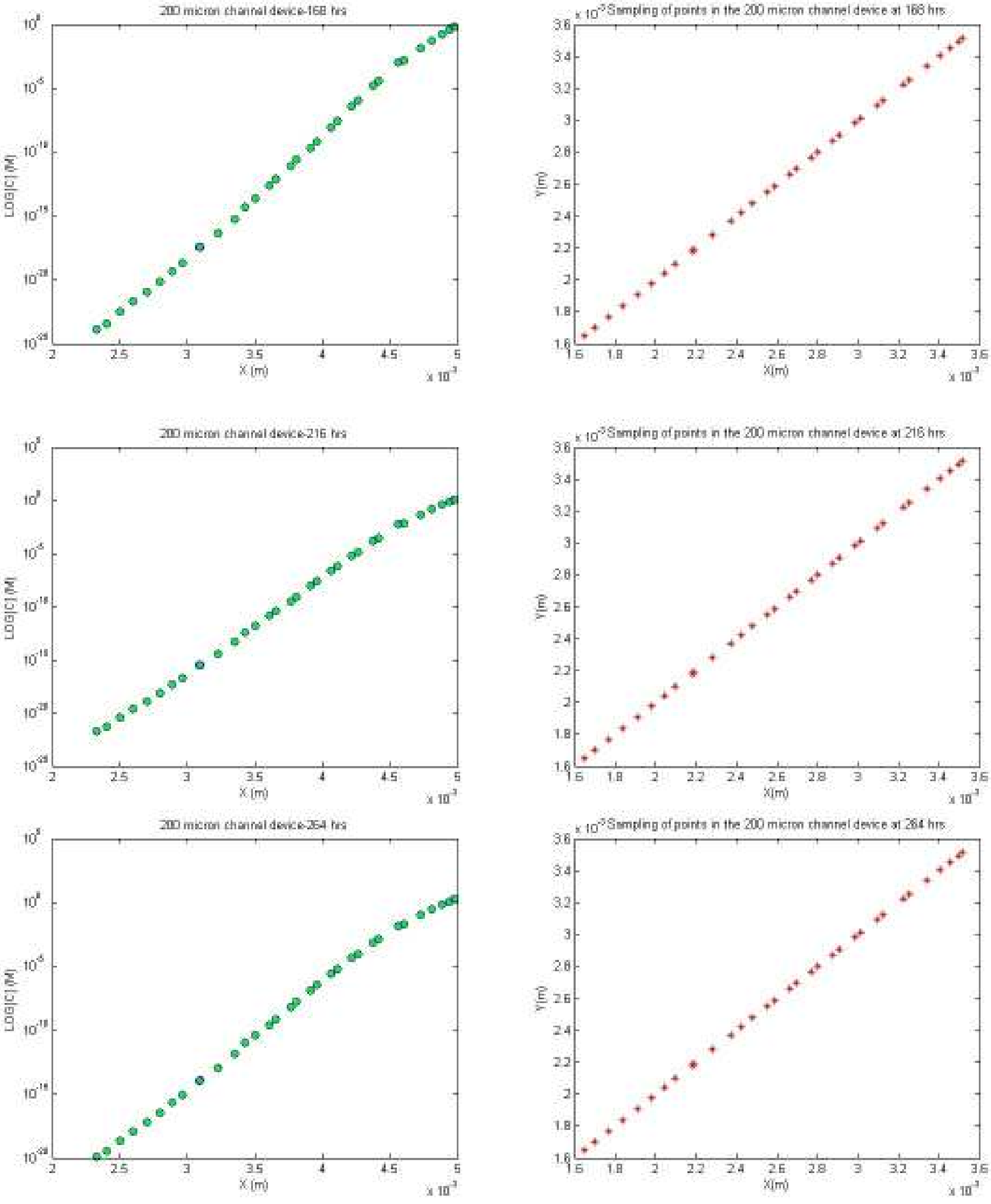


Figure 38-contd.

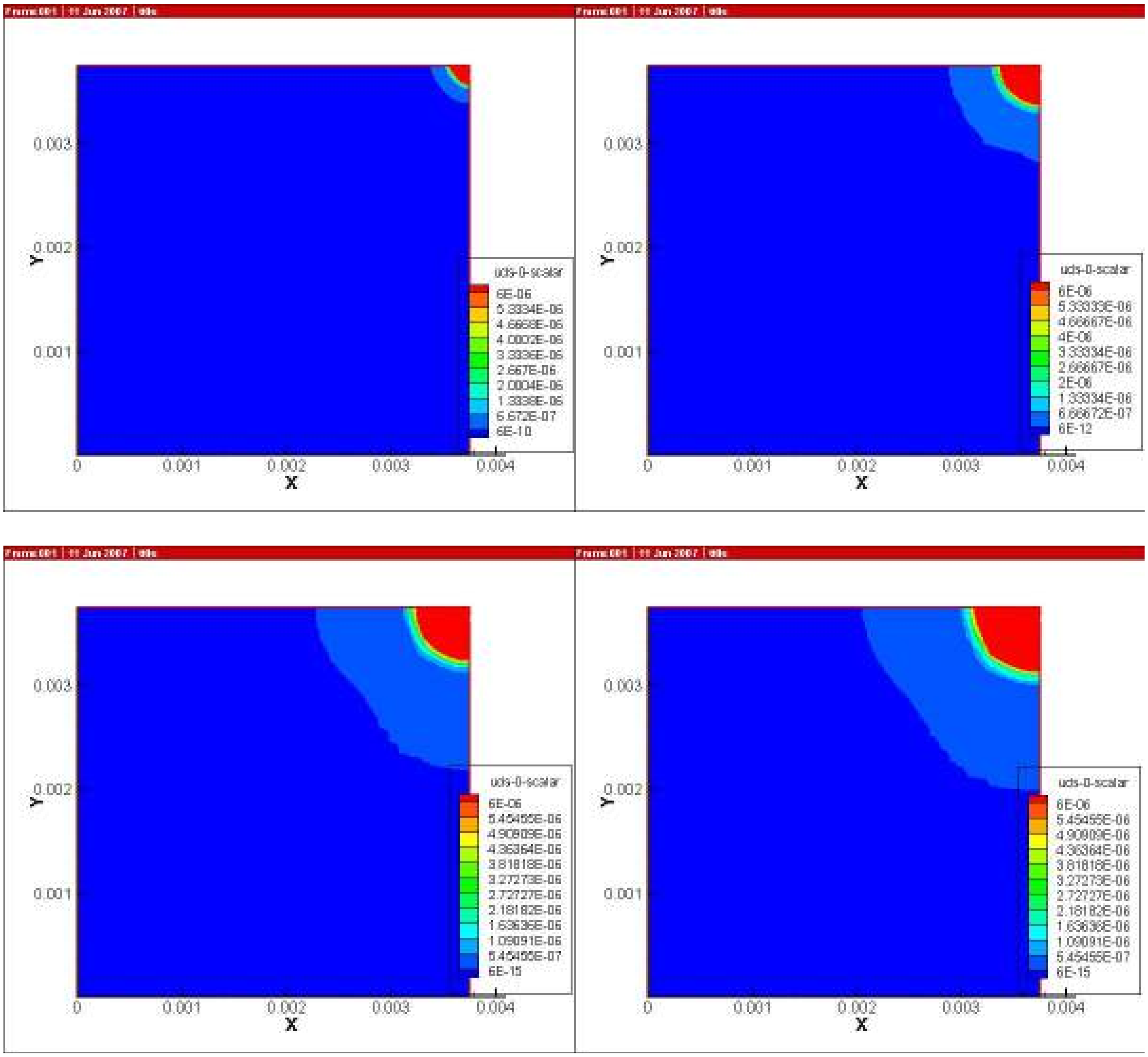


Figure 39: Concentration levels in the model with the 200 micron channel on a plane at a height of 1.3µm across the length of the chamber and extending radially from the center of the cylinder

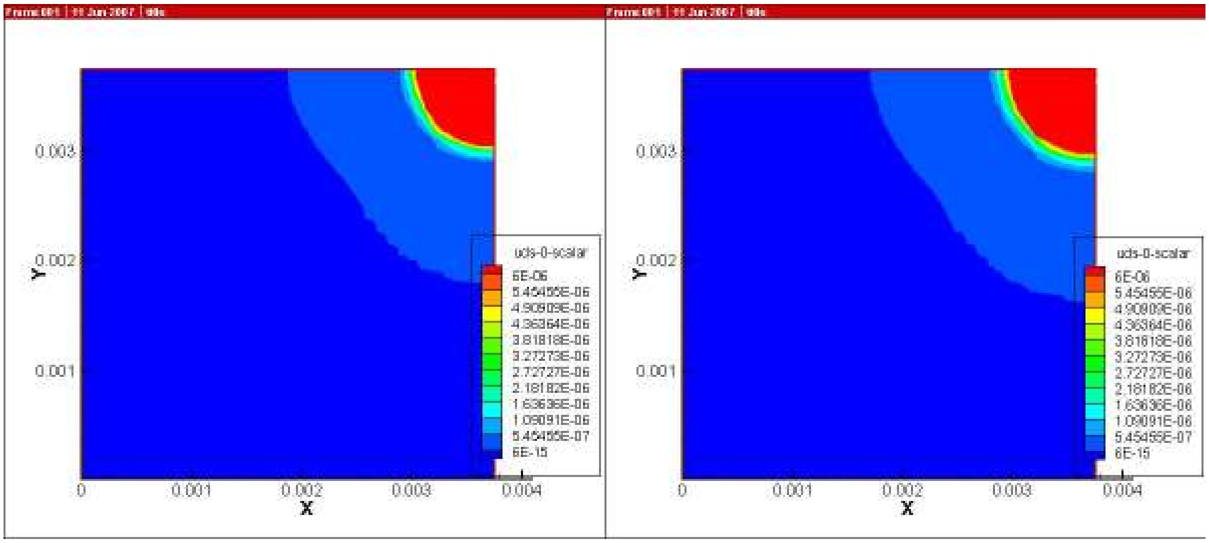
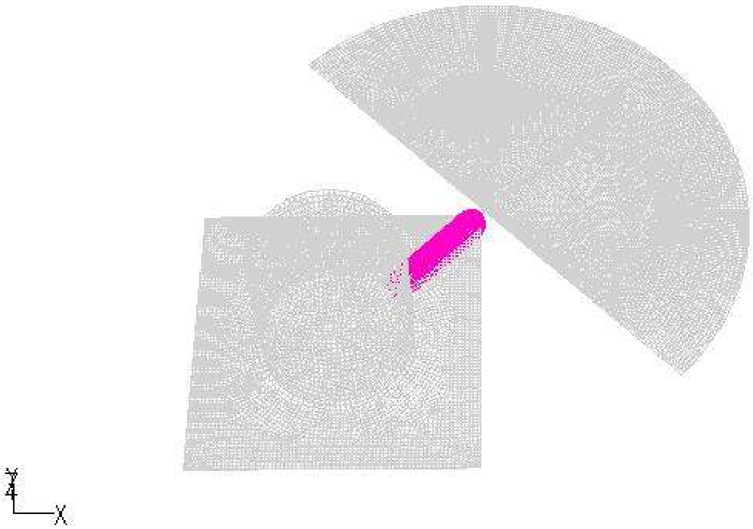


Figure 39-contd.

II.4.3 Gradient profiles in devices with varying channel widths

We have also computed and studied the gradient of the diffusive material over space at different time points in each device with a different channel width. In extracting information about the gradient, we used a diagonally drawn rake in Fluent. The difference between a line and rake is that we can specify the number of points to be extracted on a rake. These points may not always be actual grid intersection points and are uniformly spaced. Therefore, the values for diffusion parameters on rakes are not directly from the simulation results but are interpolated by Fluent on the basis of the simulation results. We also obtained gradient values on a plane extracted in the same manner as we did for the contour plots of concentration values. The concentration values and gradient values for each device at each time-point have been graphed on the same plot. The concentration values, as described before, decrease exponentially and on closer inspection, we can infer that the values extracted from Fluent are based on a linear interpolation scheme that Fluent uses to calculate the concentration values.



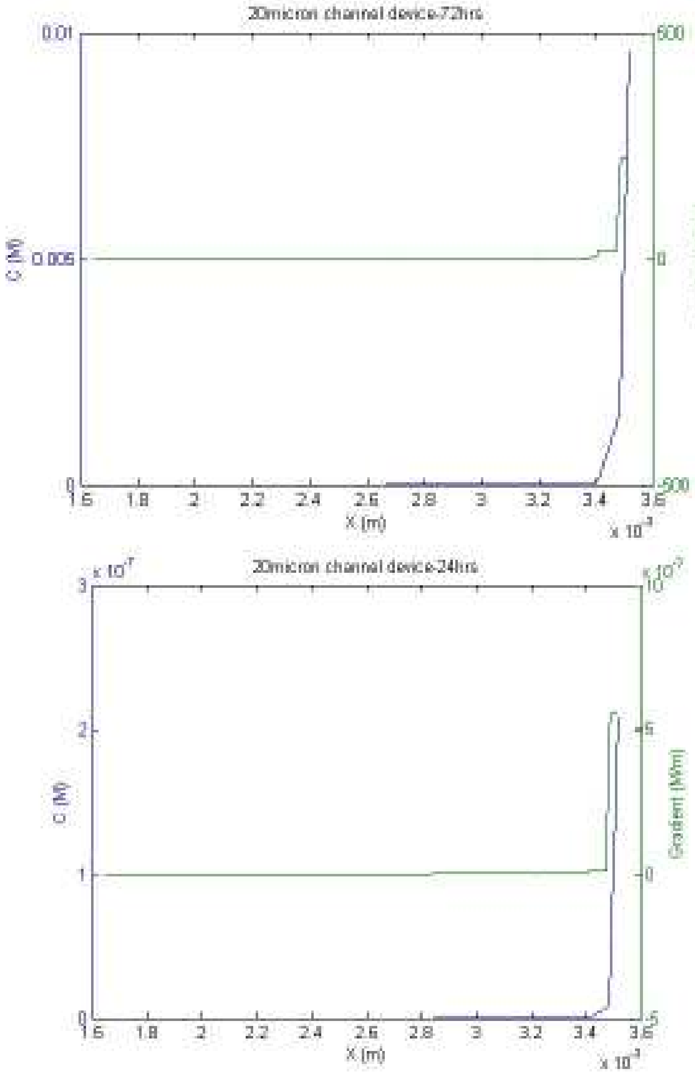


Figure 40: Plots showing the concentration and gradient profiles for the device with the 20μm wide channel at 1, 3, 5, 7, 9 and 11 days. The concentration and gradient plots are in blue and green respectively

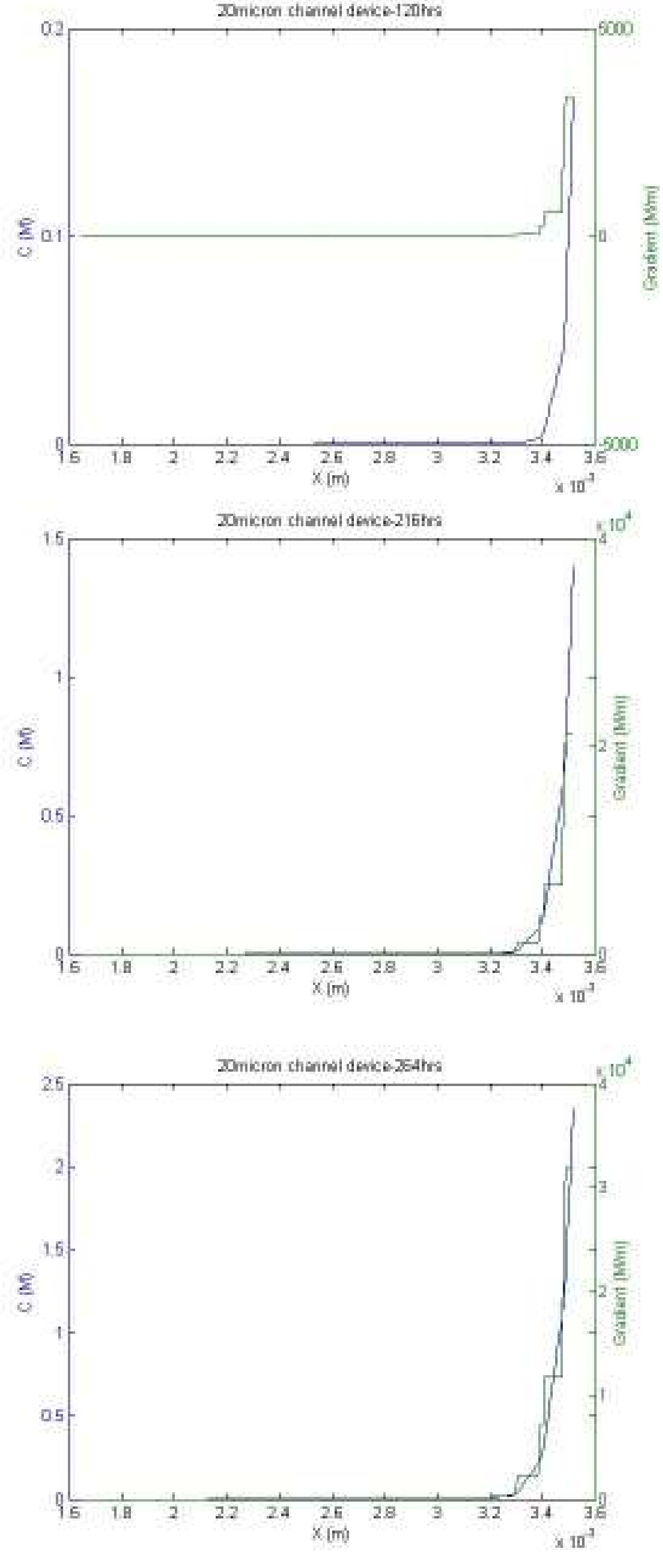


Figure 40-contd.

The gradient values are calculated using the discretized expression for gradient calculation described in section II.2.1. The gradient plots, as expected, are plots that resemble step-functions which again confirms the linearity of the trend of concentration values calculated by Fluent. We observe that once the channel width reaches 80μm the concentration and gradient plots decrease in steepness and this trend continues in the last simulation with a channel width of 200μm. The scales of the Y-axis on the left and right sides are different to account for the gradient and concentration value ranges. The plots have been generated using Matlab and the code can be obtained from Appendix D.

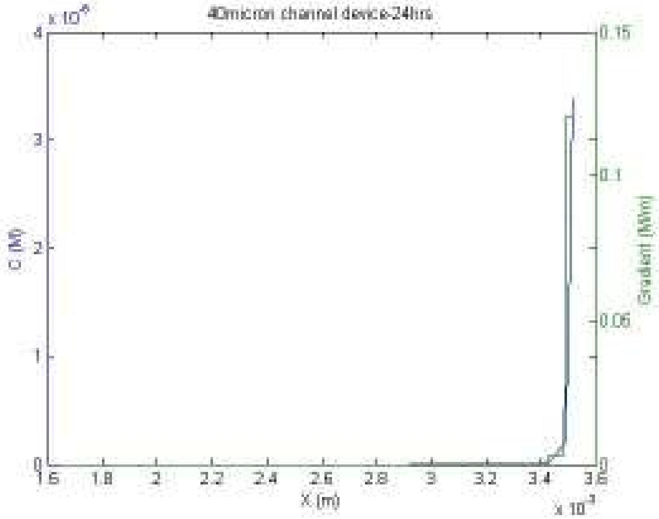


Figure 41: Plots showing the concentration and gradient profiles for the device with the 40μm wide channel at 1, 3, 5, 7, 9 and 11 days

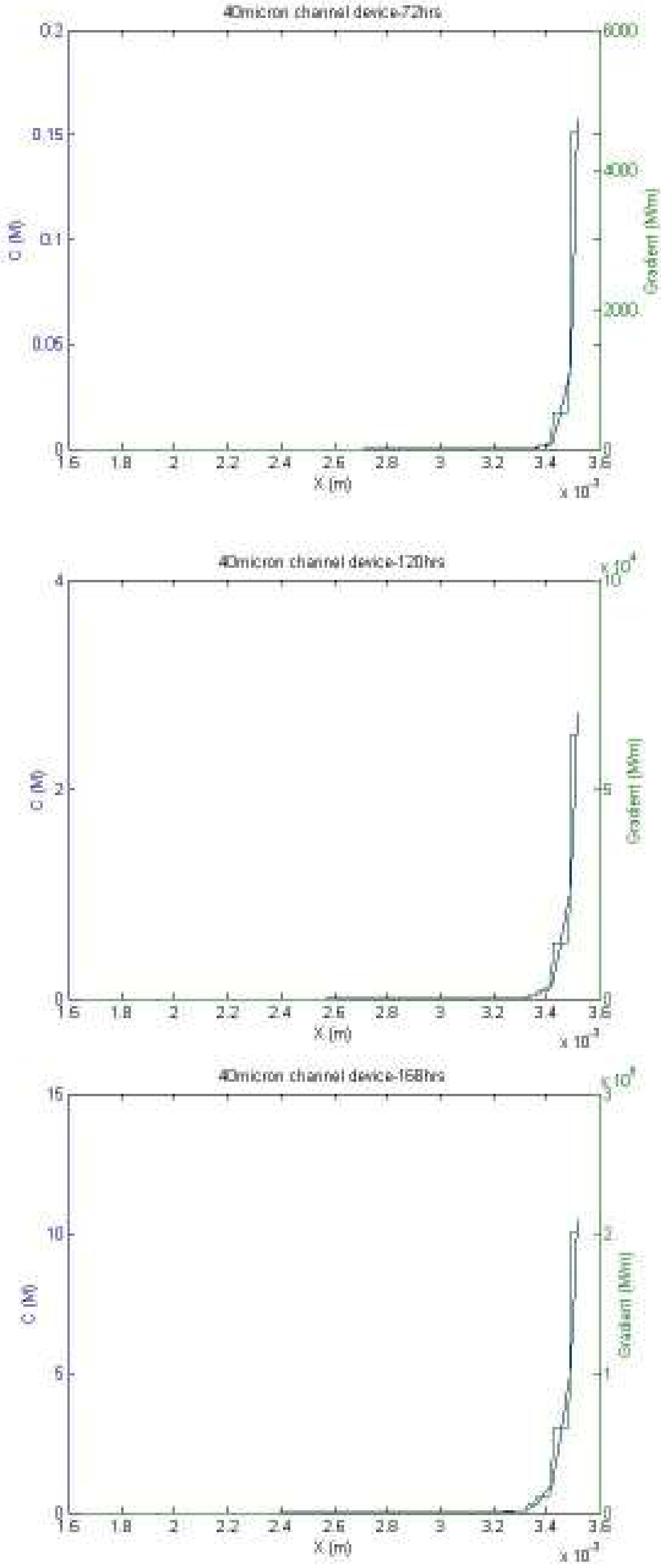


Figure 41-contd.

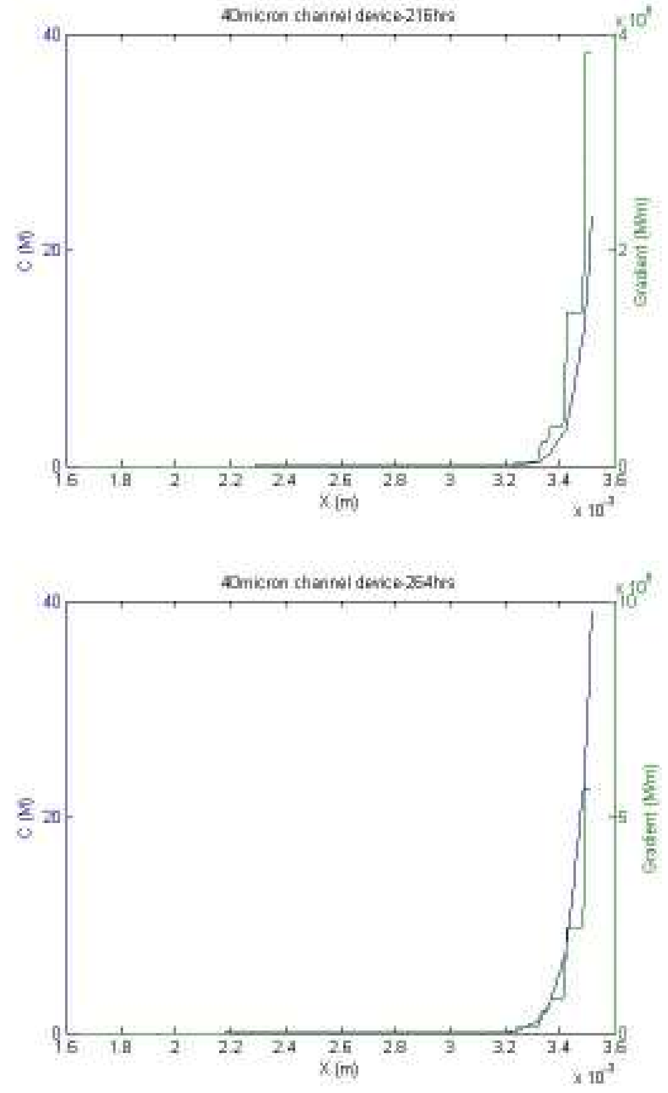


Figure 41-contd.

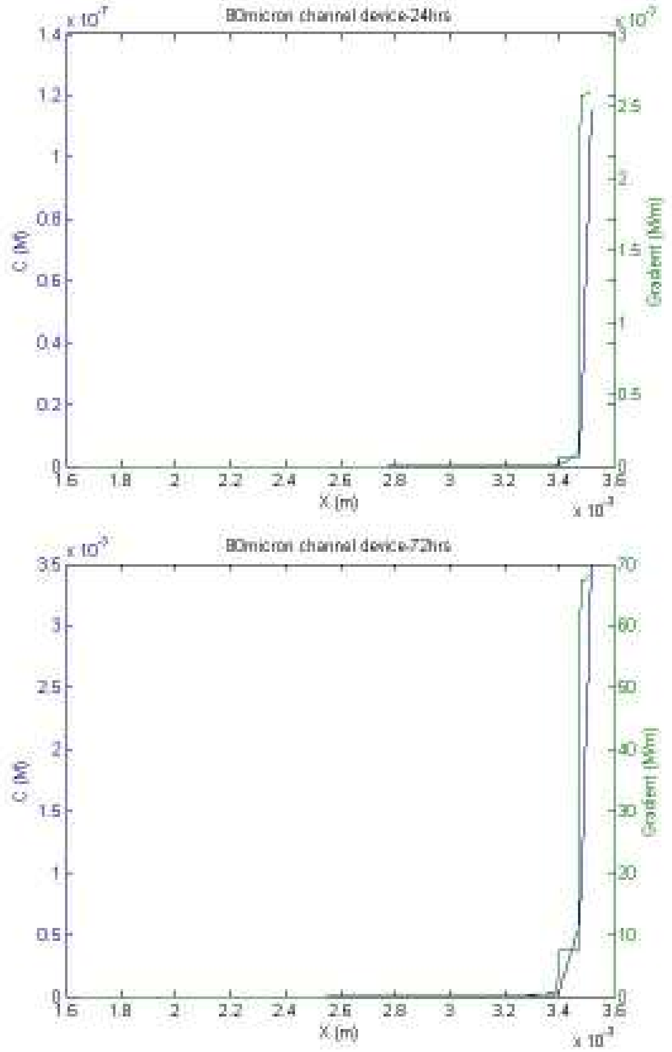


Figure 42: Plots showing the concentration and gradient profiles for the device with the 80µm wide channel at 1, 3, 5, 7, 9 and 11 days

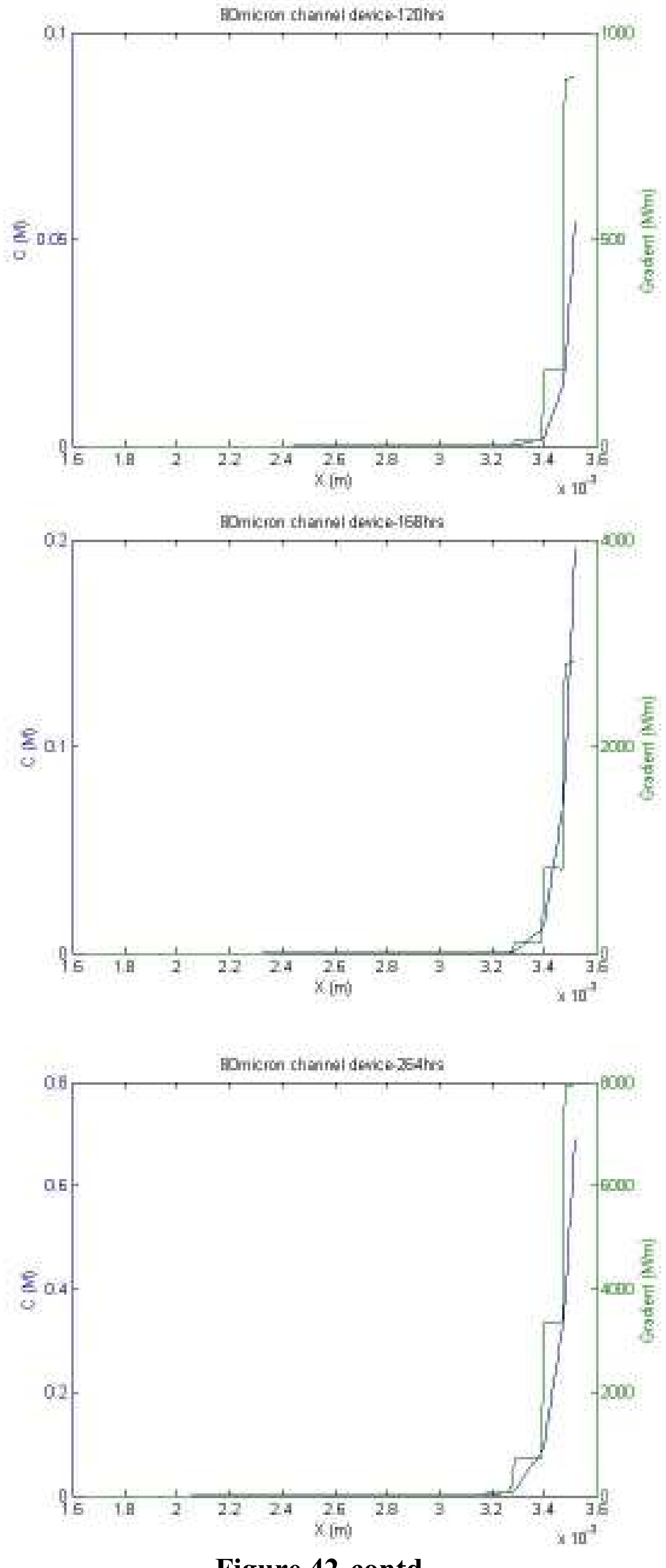


Figure 42-contd.

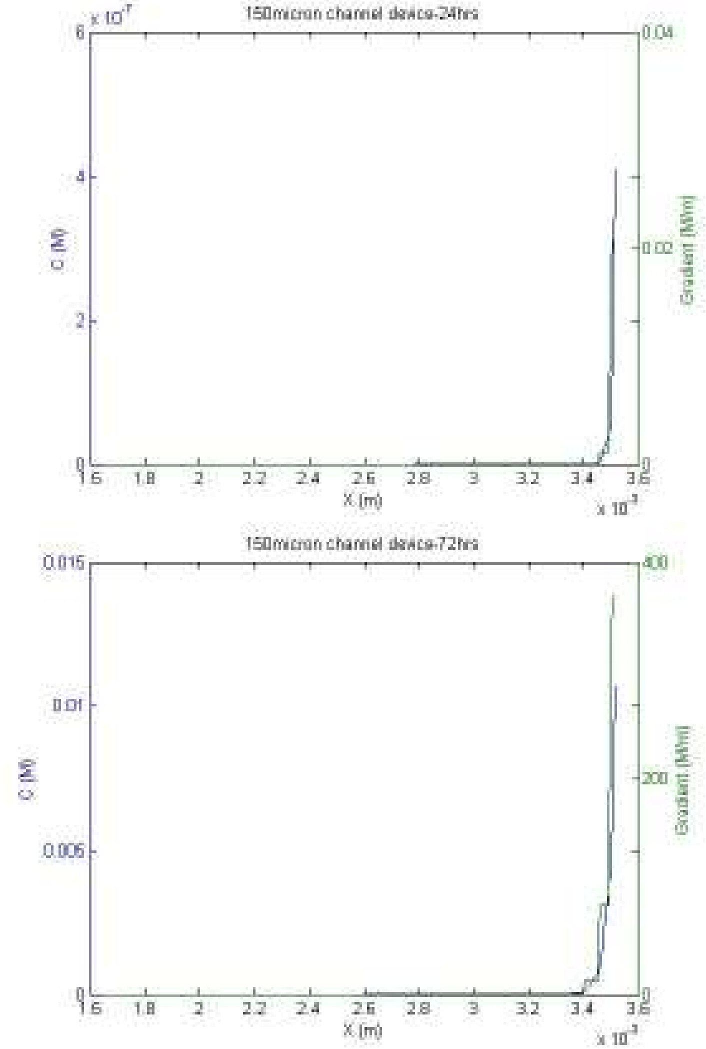


Figure 43: Plots showing the concentration and gradient profiles for the device with the 150µm wide channel at 1, 3, 5, 7, 9 and 11 days

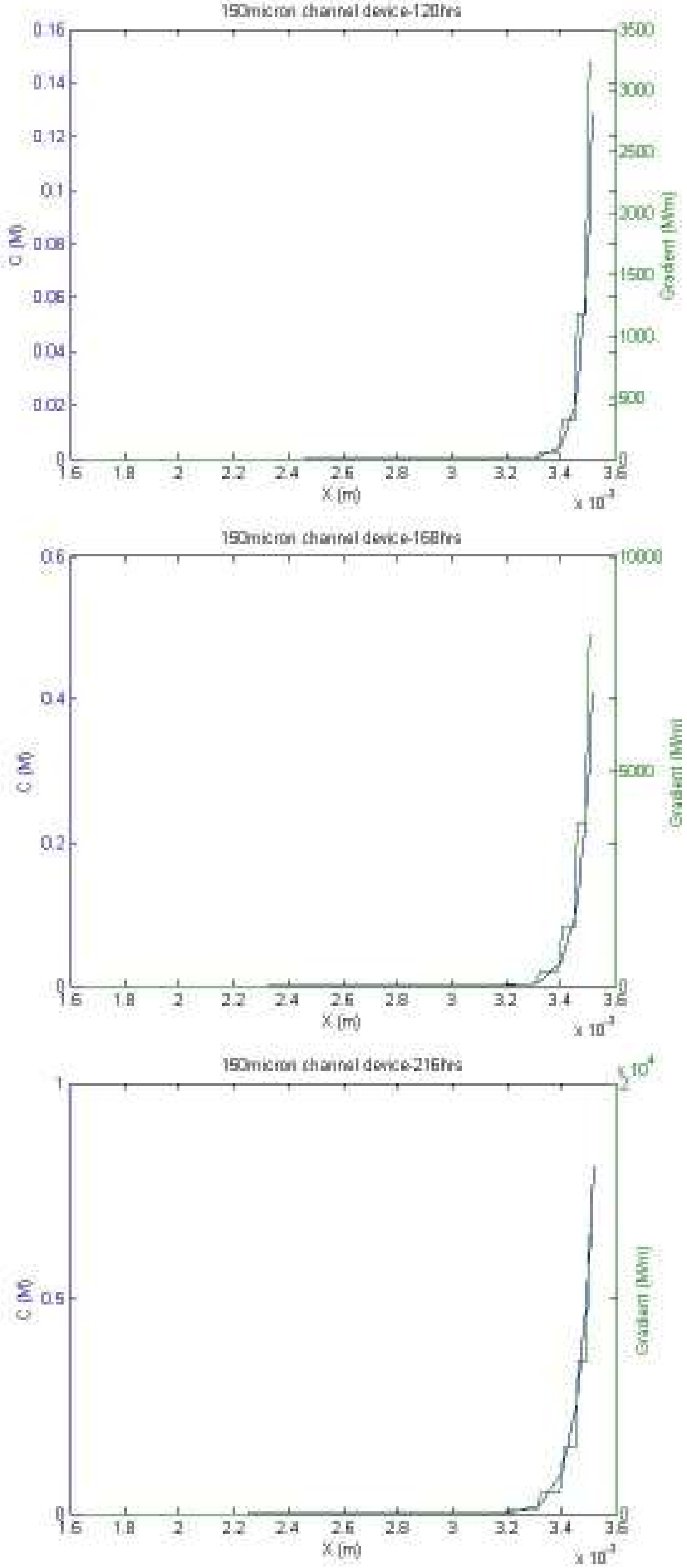


Figure 43-contd.

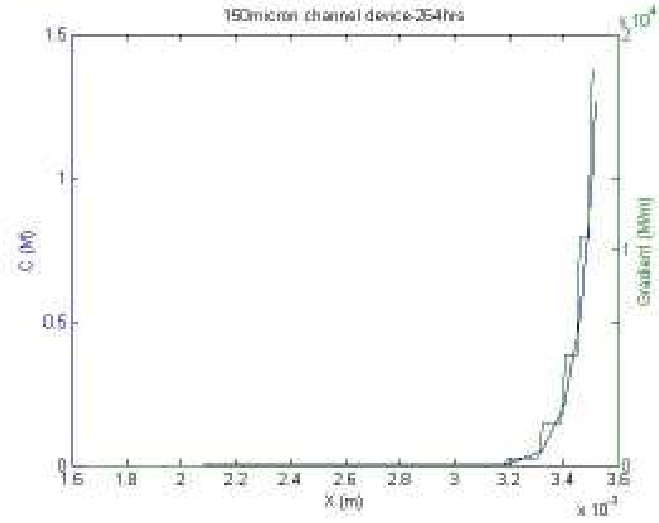


Figure 43-contd.

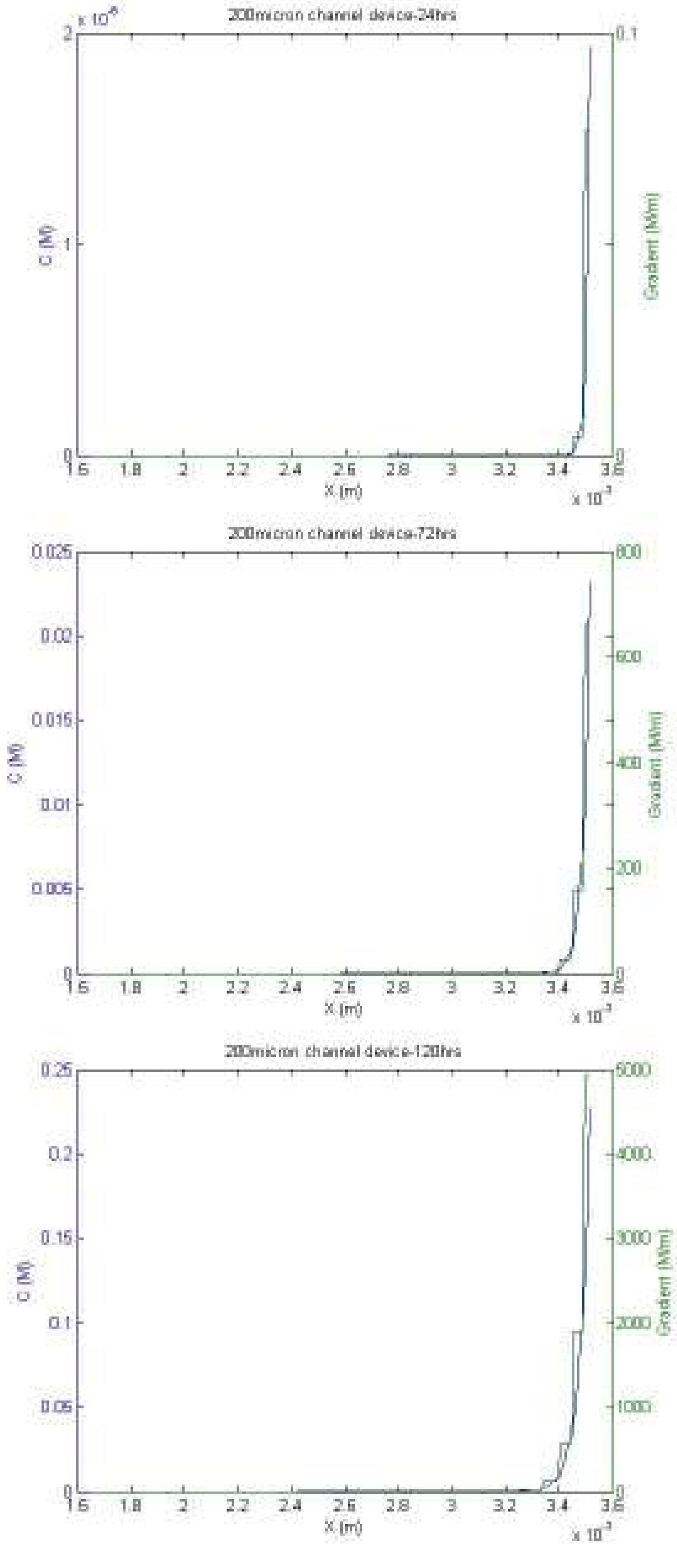


Figure 44: Plots showing the concentration and gradient profiles for the device with the 200μm wide channel at 1, 3, 5, 7, 9 and 11 days

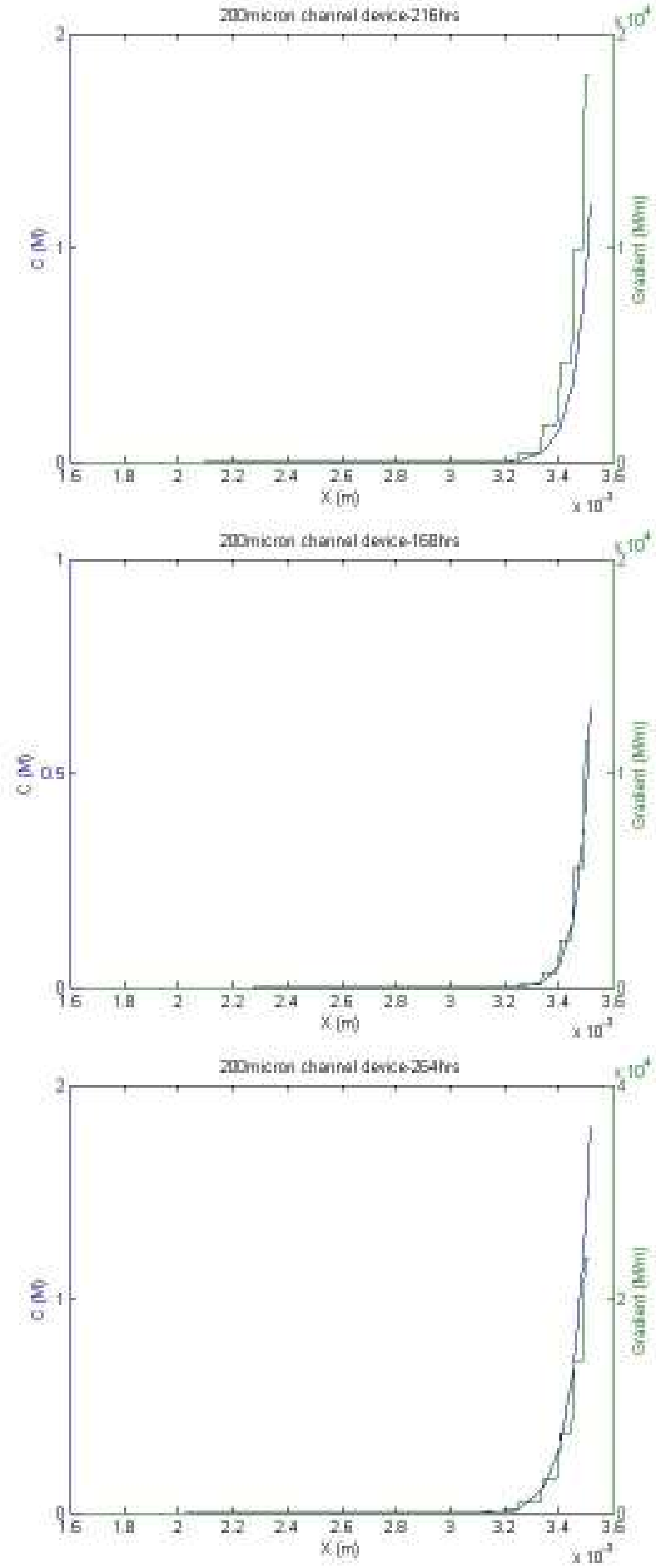
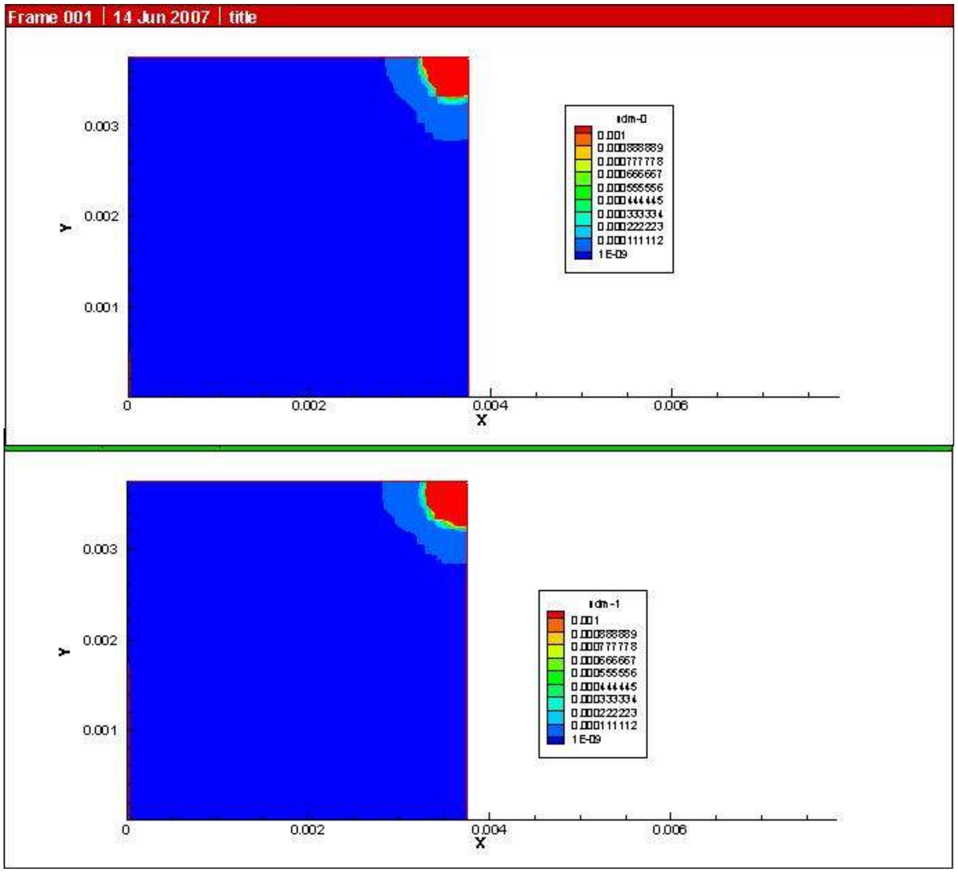


Figure 44-contd.

We also examined the gradient of the diffusive substance on the same plane that we defined for viewing contours of concentration levels. Figure 45 shows the contour plots of gradients on the same plane in devices having channels of width 20µm, 40µm, 80µm, 150µm and 200µm. We have shown the data obtained at 3, 7 and 11 days in this section to maintain the uniformity of time-intervals and enable comparison of gradient levels in devices with channels of differing widths.



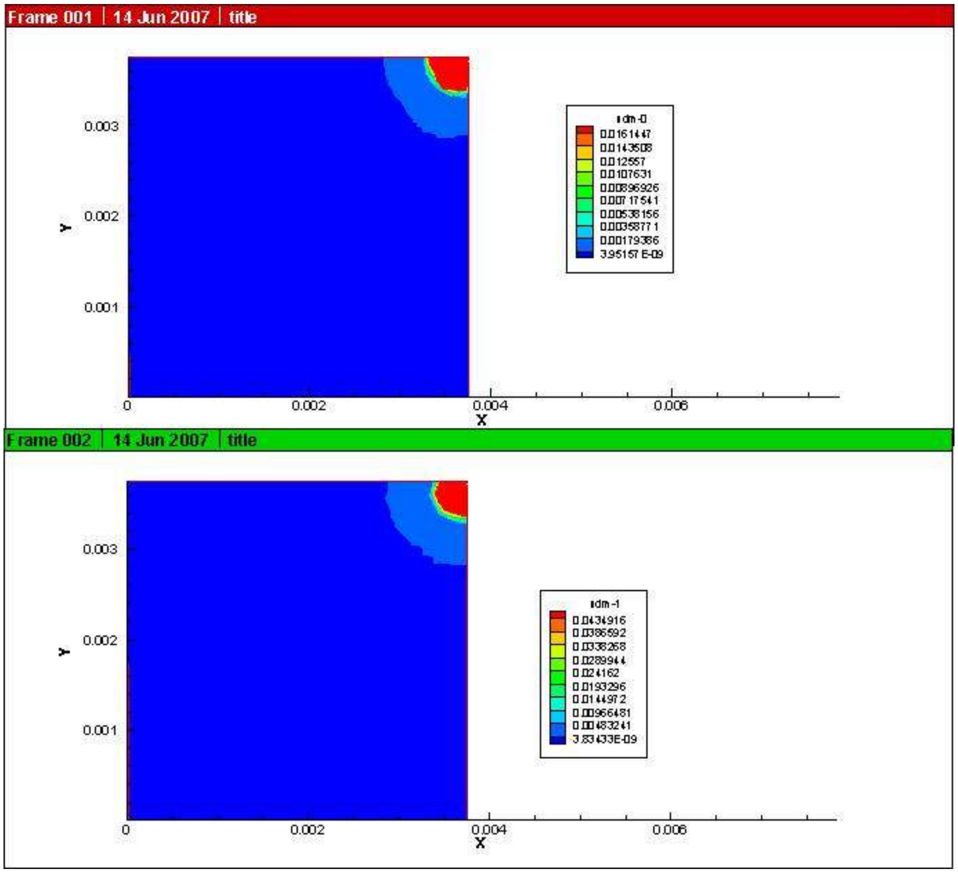


Figure 45: Contour plots showing the gradients in the 20µm, 40µm, 80µm, 150µm and 200µm channel devices at 3 days

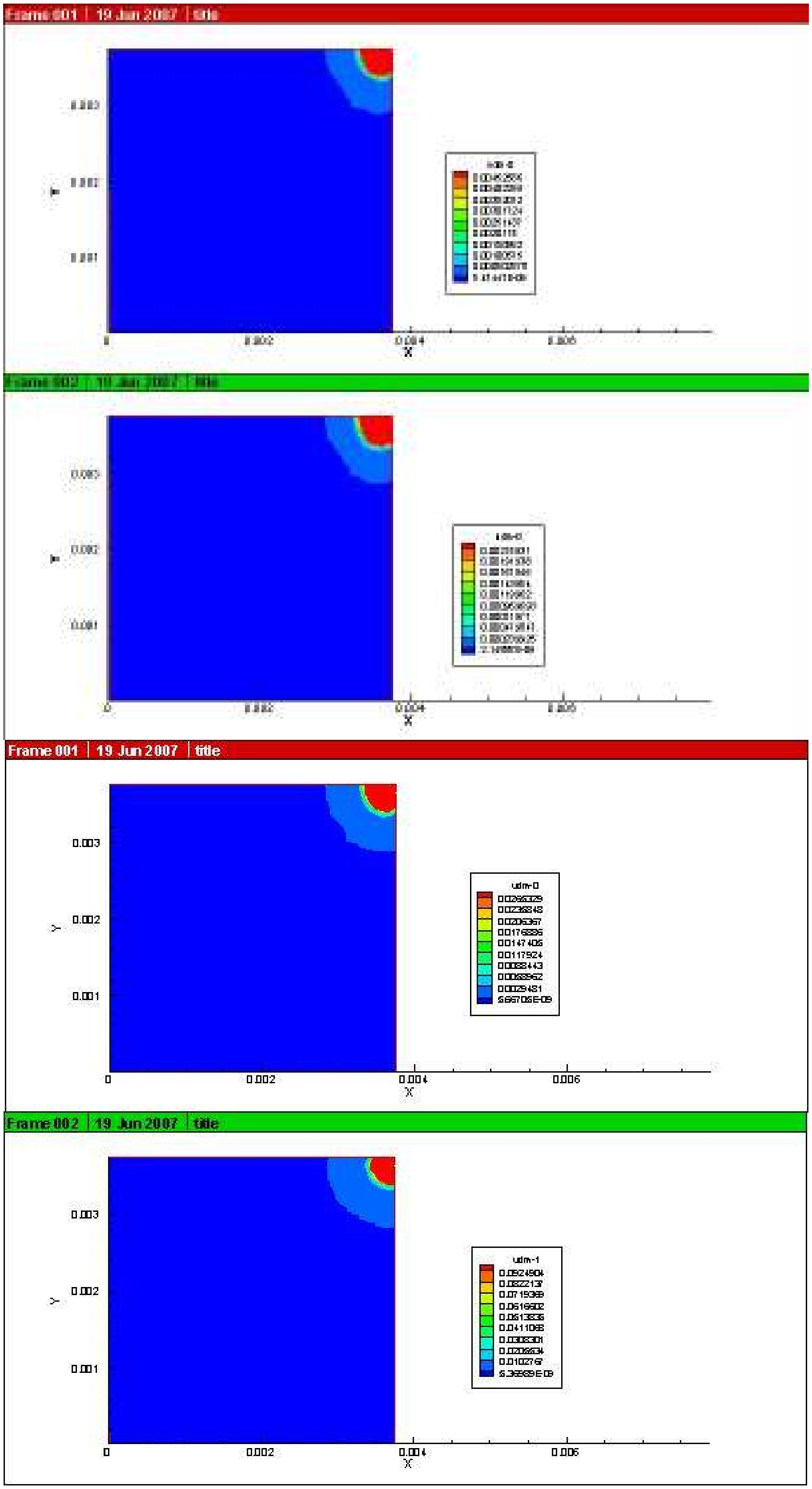


Figure 45-contd.

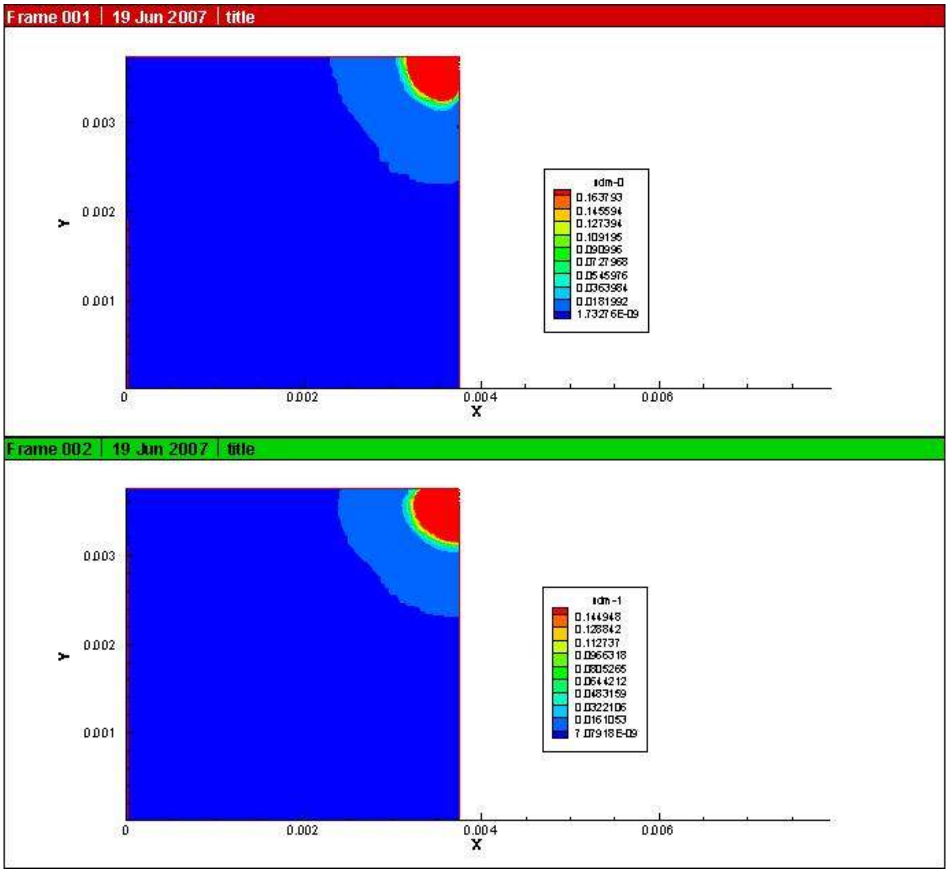


Figure 45-contd.

The contour plots of gradients on the plane at 3, 7 and 11 days have been displayed in Figures 44, 45 and 46 respectively. Each subplot of these plots exhibits data from device with a channel of a particular width. There are five subplots in each plot, indicating that the contours are from devices with channels of widths 20μm, 40μm, 80μm, 150μm and 200μm respectively. The gradient values calculated in Fluent are from each cell in the volume grid are calculated with respect to the x, y and z coordinates of the central node of each cell as these are in-built grid value storage variables provided by Fluent. Each subplot in figures 44, 45 and 46 contains two frames indicating the gradient contours in the x-direction (red frame) and in the y-direction (green frame).

The color scale on these contour plots is generated using Tecplot according to the gradient values provided by Fluent. The maximum, minimum and delta values are decided after several executions of the contour plot and the combination is identified that most visibly displays the gradation in gradient levels in each direction. There is not a marked difference in gradient levels at 3 days among the different devices but the distortion in contour levels decreases as the width of the channels increase. This indicates that the diffusion takes place smoothly and the flow increases with increasing channel width. Figure 46 shows the gradient contours in devices with varying channel widths at 7 days.

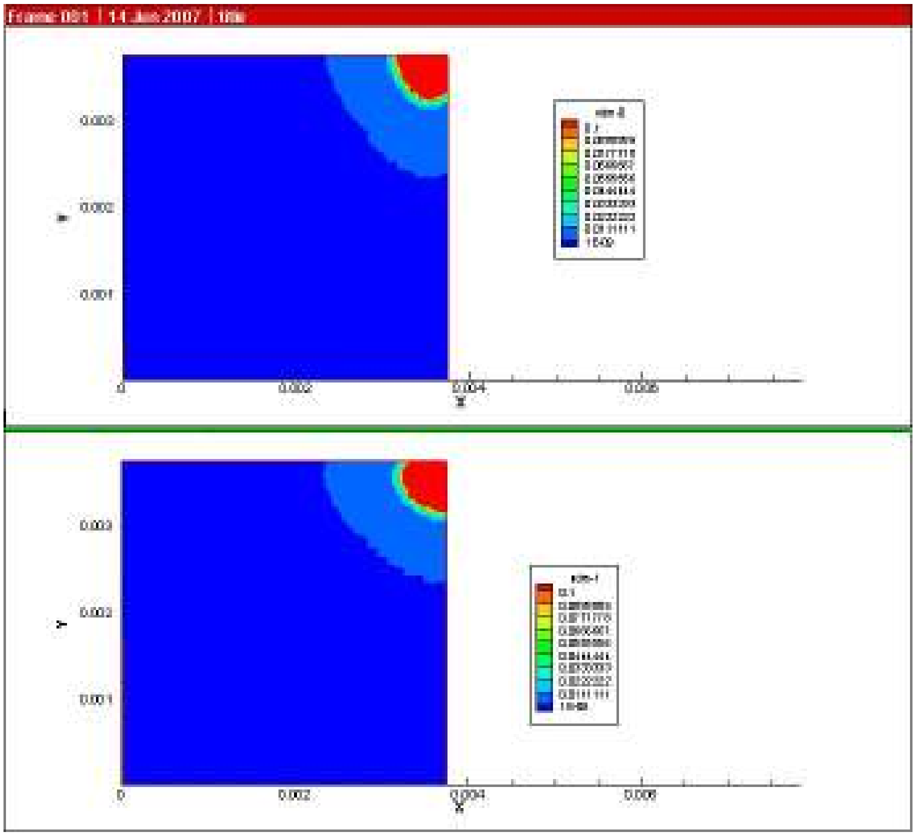


Figure 46: Contour plots showing the gradients in the 20µm, 40µm, 80µm, 150µm and 200µm channel devices at 7 days

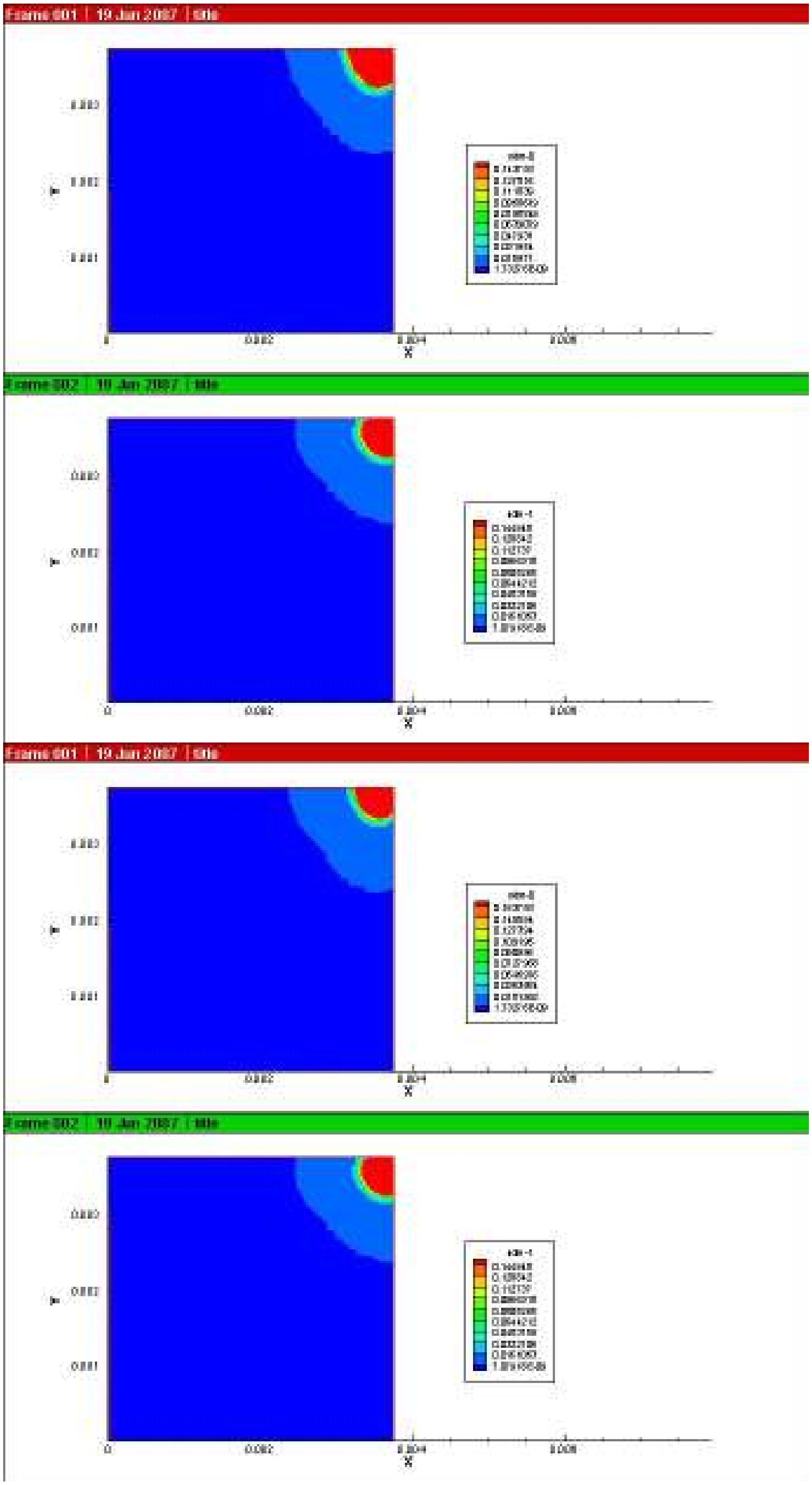


Figure 46 contd.

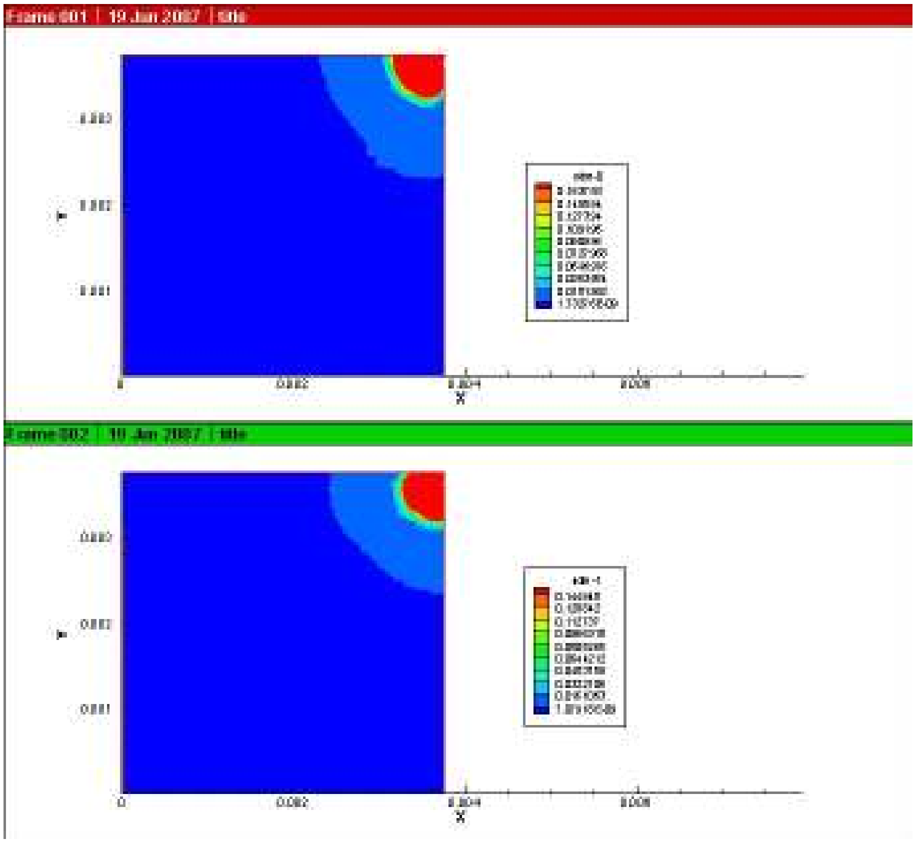


Figure 46-contd.

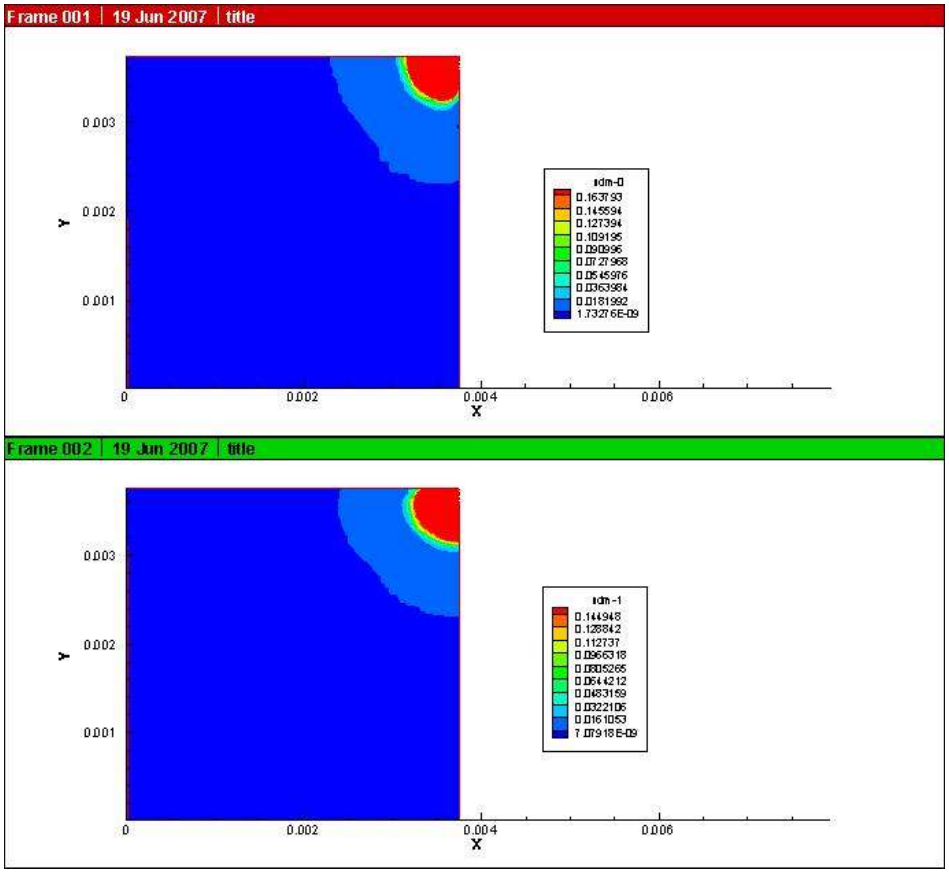


Figure 46-contd.

In these contour plots, we can see a discernible difference between the contour levels in the different devices. The devices with channel widths 80µm, 150µm and 200µm show a distinct increase in gradient over space. The contour levels are smoother overall and in the devices with 20µm and 40µm wide channels the gradient is greater in the x direction compared to the y direction. Figure 47 shows the gradient contours in all the devices at 11 days and we observe the increase in gradient levels among the devices with channels of width 80µm, 150µm and 200µm.

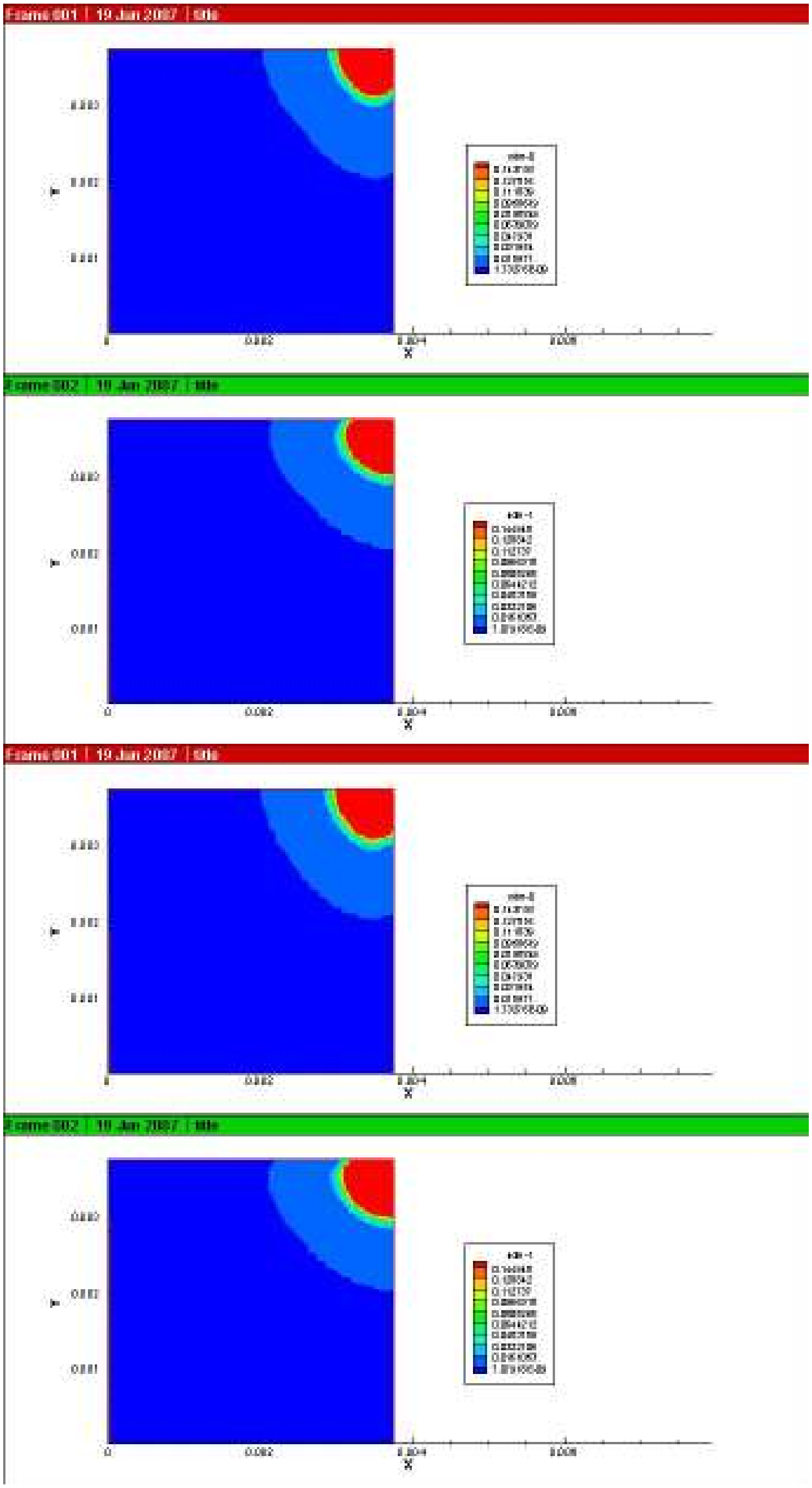


Figure 47: Contour plots showing the gradients in the 20µm, 40µm, 80µm, 150µm and 200µm channel devices at 11 days

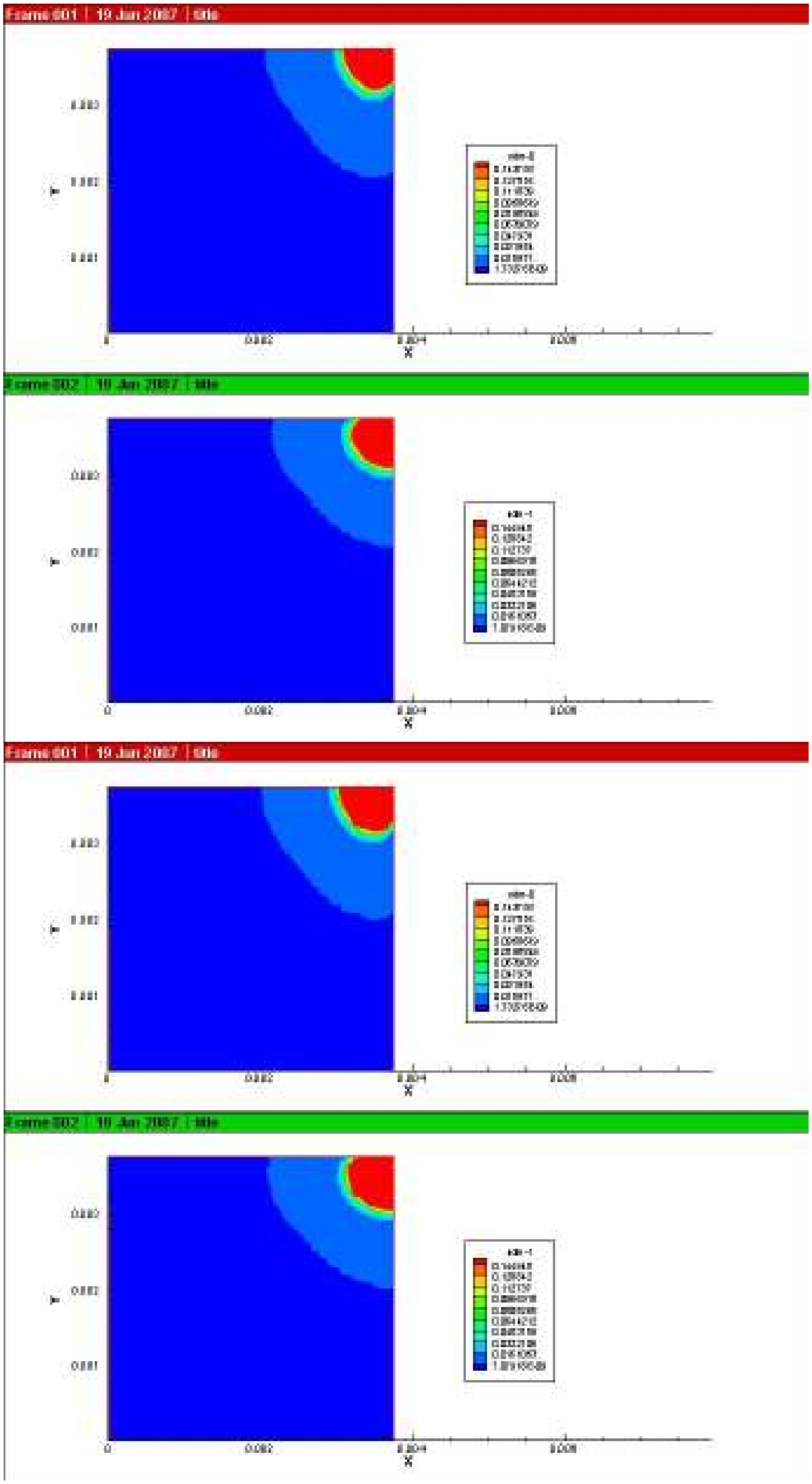


Figure 47-contd.

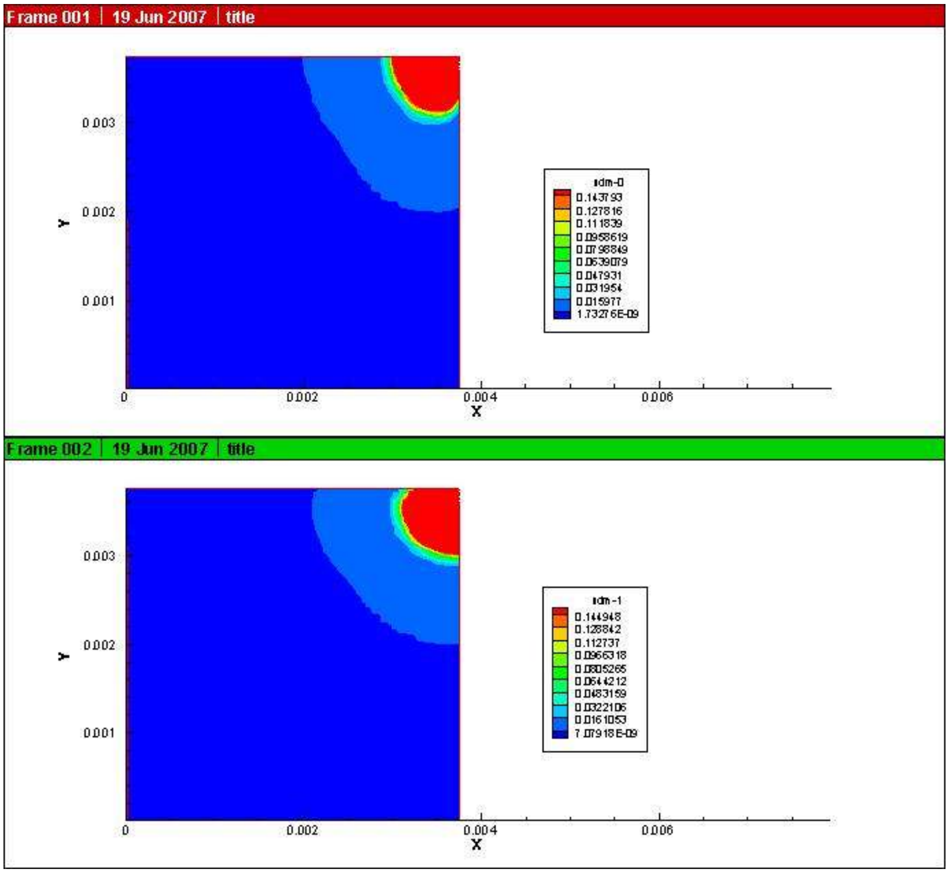


Figure 47-contd.

CHAPTER III

CONCLUSION AND FUTURE WORK

III.1 Discussion and conclusion

We developed a CFD model for our microfabricated passive diffusion device with the following objectives, as stated in section *I.4*:

- The validation of Fluent as a tool for simulating microfluidic devices and providing design paradigms for their fabrication
- The study of how architectural and material parameters affect the simplest form of the device
- The study of the restriction channel as an impedance structure and how it affects the diffusive flow, concentration levels and gradient formation in the device.

We established through our preliminary results that Fluent is a credible tool for developing CFD models of microfluidic devices. The plots in Figures 13 and 14 in section *II.3.1* clearly demonstrate that the preliminary results are in accordance with the theoretical principles of diffusion. The concentration is high at the entrance to the chamber, almost equal to the initial concentration value while at the edge of the cylinder it starts approaching extremely small values in the vicinity of 1E-014. Our boundary condition in this preliminary model states that the concentration at the top of the cylinder is clamped to zero and these results agree with this boundary condition. However, we also observe in the graphs that the values towards the edge of the cylinder are not discernible and the concentration values drop to zero when the actual values are close to zero but not equal to it. The range of the values that we encounter in the concentration levels of the device is in the order of 1E-20 and therefore, we cannot get a complete picture of the concentration trends plotting the actual values. We can plot the logarithm of the concentration values as we have done in later models but in addition it is necessary for us to get a picture of the actual concentration values.

In the later models that emulate the actual diffusion device, we have demonstrated first that we can simulate the device with a finite reservoir and obtain valid results which are again in accordance with the theory of diffusion. We observe here that the range of concentration values increases to V which creates further challenges in viewing the entire range of concentration values on a single plot. In this simulation we also come across numerical diffusion for the first time that has been explained in greater detail in Appendix D. In order to address the challenge of viewing the complete range of concentrations obtained, in our later models we developed three kinds of plots

- Concentration levels and gradient along a user-specified line drawn diagonally from the entrance of the chamber to the edge of the cylinder.
- Concentration and gradient levels on a plane radiating from the center of the cylinder and at a height halfway through the chamber.
- Semilog plots of concentration points along lines drawn diagonally across the grid from the entrance of the chamber to the edge of the cylinder.

This collection of graphs for each case provides us with a clear picture of concentration and gradients along the x and y axes as well as the planar view of these quantities in the form of contour plots. We find that towards the initial periods of time, from 0-24 hours, the concentration along the chamber reaches values ranging from 2.09E-07M to 1.47E-44M. The progression of these values can be clearly seen in the semilog plots where the entire range of values has been plotted. The graphs illustrating the actual concentration and gradient values do not show the entire range of data, especially beyond 3000μm after which the concentration values start dropping below 1E-10M. However, these graphs are necessary for us to understand how the concentration levels correlate to the gradient levels in the chamber. In the contour plots, we observe that beyond values in the range of 1E-15 the contour lines become distorted and the contour levels indistinct. This occurs both due to numerical diffusion and limited control over contour algorithms which has been addresses in greater detail in Appendix C. In the semilog plots we also observe that in cases with smaller channel widths (20μm and 40μm), the graphs for the earlier time-points (24 and 72 hours) are not completely smooth. We have displayed the sampling of points in Fluent alongside each semilog plot and from these graphs, it is evident that the sampled points are not regularly spaced. They are actual points from the grid where the specified line intersects it and as every grid cell is not of the same size, the points are not uniformly placed along the line. We have used a line instead of a rake in these plots as the former provides us with data directly from the simulation while the latter uses Lagrangian interpolation to obtain concentration values for points that are not actually present on the grid. As our final tool for analysis, we also developed semilog plots for concentration levels in different devices

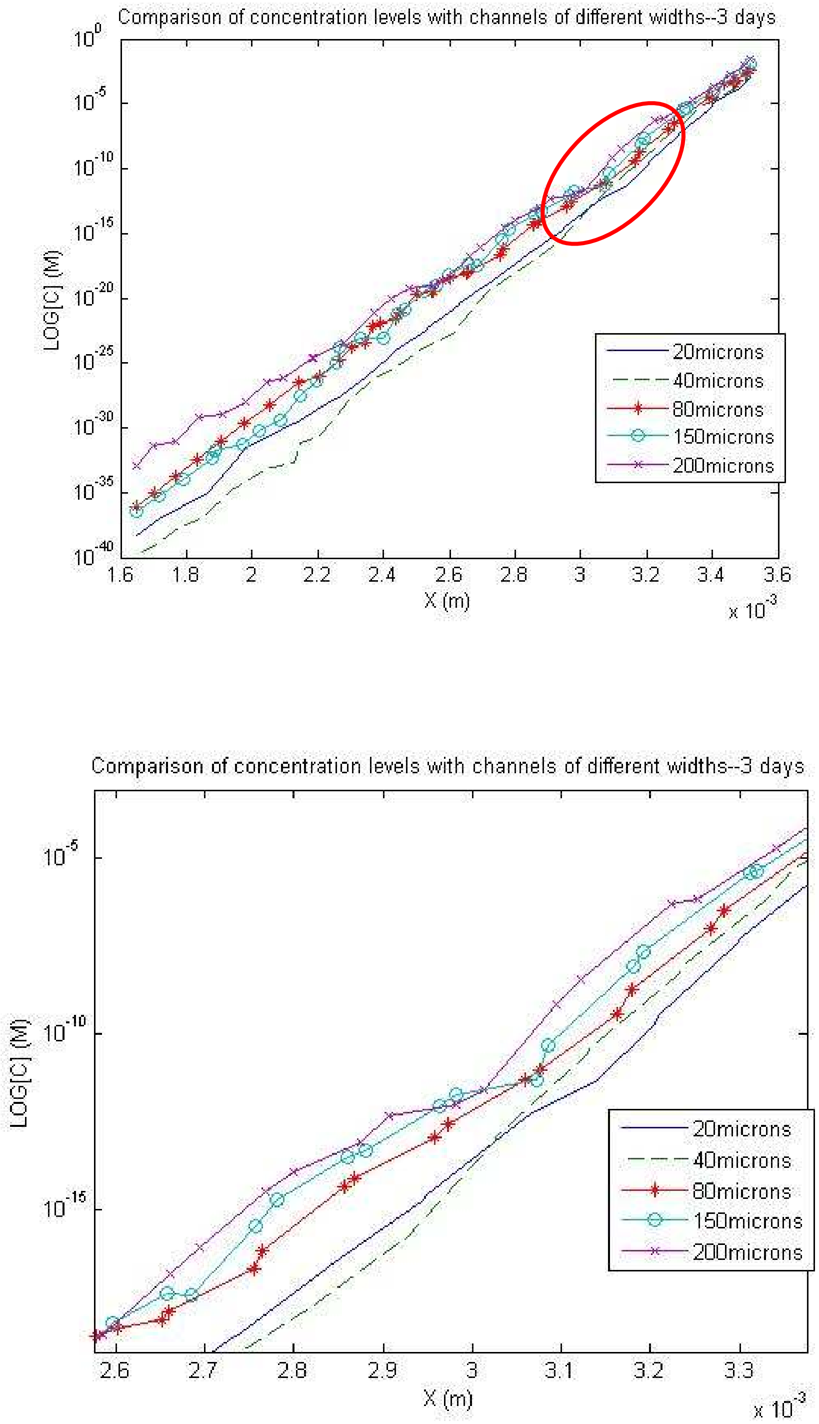


Figure 48: Concentration levels at 3 days in devices with different channel widths

with varying channel widths at particular time-points. The results in Figure 48 show the comparison of the concentration levels in devices with varying channel widths at 3 days. The part encircled in the first plot in red has been magnified in the second plot. We observe in the plots that the difference in concentration levels broadens out only towards the end when we approach the edge of the cylinder. The device with the channel of width 200µm shows relatively higher concentration levels compared to the other models. The magnified view indicates that with the exception of distortions due to non-uniform sampling, all the devices show their expected trends in concentration levels. The distortion makes it challenging to delineate clear trends but we an observe that while the 20µm and 40µm channel devices maintain lower concentrations, the devices with the three higher channel widths show a rise in concentration at the same point in space.

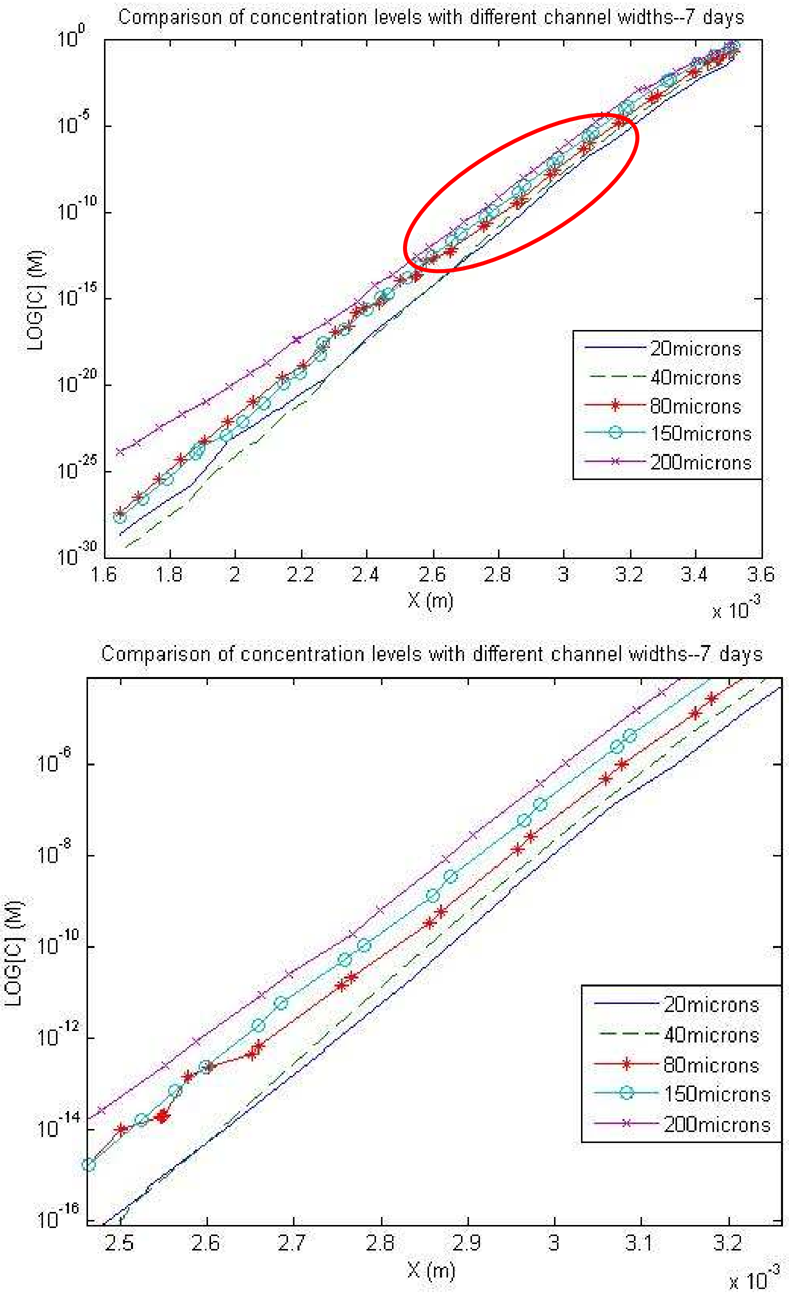
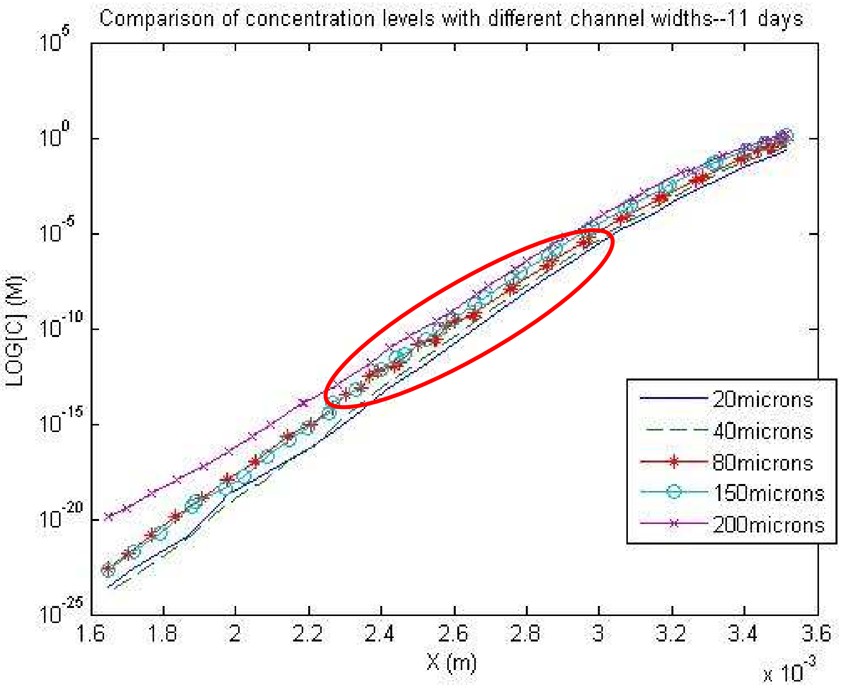


Figure 49: Concentration levels at 7 days in devices with different channel widths

The results in figure 49 show a comparison of concentration levels again but this time at seven days. The distortion is considerably at this time point and it is easier to distinguish the individual concentration trends from each other. In the magnified view, we observe that although there is a distinct difference in concentration levels among the devices with channels of width 20µm, 40µm and 80µm, the difference is not as distinct at 150µm, where it almost overlaps with the 80µm line over space. The 200µm channel device, however, continues to be distinct from all the others, showing high and discernible concentration levels which differ from the 20µm channel device by at least 1E-02 orders of magnitude. Our final plots in Figure 50 show similar data from all the models at 11 days. The distortion decreases to a minimal level at this time-point, allowing us to discern the trends of each model in the complete plot.



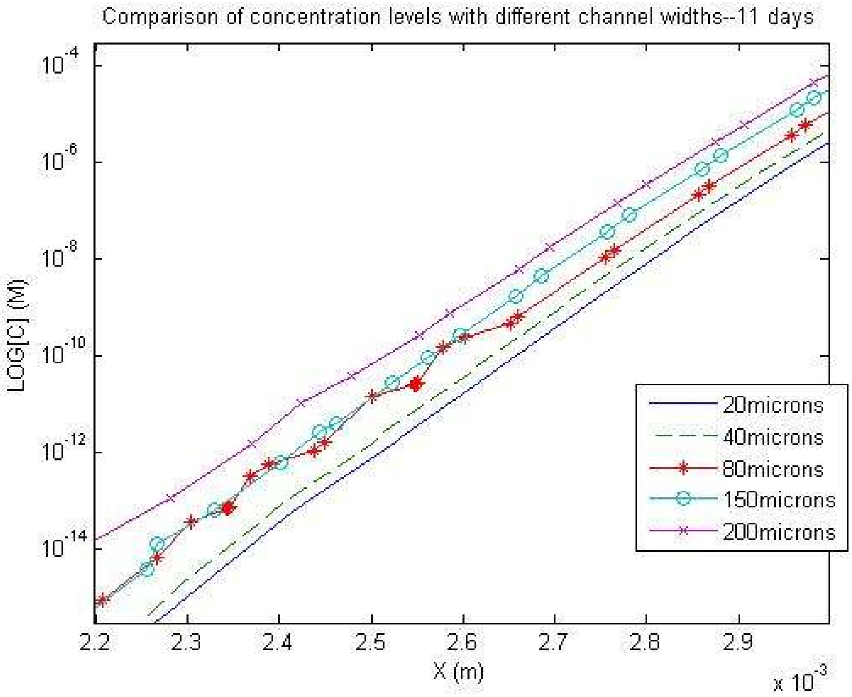


Figure 50: Concentration levels at 11 days in devices with different channel widths

We can infer from the above data that the width of the restriction channel plays a significant part as a parameter that influences diffusive flow and gradient formation. The distinct trend of the device with the 200µm channel at all time-points and the rise in concentration levels between 40µm and 80µm support this fact. In terms of the range of concentrations that we are interested in, it can be achieved faster through devices with higher channel widths, preferably in the 150-200µm range. However, these are estimates from computational simulations and should be validated with experimental data in order to fabricate the device and refine it for use in experiments.

III.2 Future Work

We have developed a robust computational model for our microfabricated passive diffusion device. As mentioned in the previous work, it is imperative to substantiate such work with valid experimental or analytical data prior to the fabrication of the device which will be used in biological experiments. In terms of the computational model, although we have examined numerous basic features, we can still add certain structures to it that are part of the original, fabricated device. The fence whose effects we have examined in section *II.3.4*, is a part of the original device. A set of simulations can therefore be designed to study the restriction channel in the presence of the fence. This will give us a compounded restriction effect due to the effect of two restriction structures. We can also add the rectangular posts further inside the chamber that are present for structural support.

In addition to the structures that we can add to the device, we can also make further attempts to address some of the issues discussed in Appendix D. Numerical diffusion is largely dependent on the timestepping scheme used in the simulation. Our present simulations have already attempted to minimize numerical diffusion by using the inherent timestepping schemes. We can define our own timestepping scheme using an adaptive procedure in which the first few timesteps are run with a step size less than one second and close to the minimum value allowed by Fluent which is 1E-05 second. We can increase the timestep at a uniform rate following these first few timesteps.

In conclusion, the purpose of this study is to serve as a guideline for simulating not only this particular device but any device functioning on similar principles. The issues discussed in Appendix D as well as the guides provided in Appendices A and B are guidelines that are applicable to any passive diffusion device.

APPENDIX A

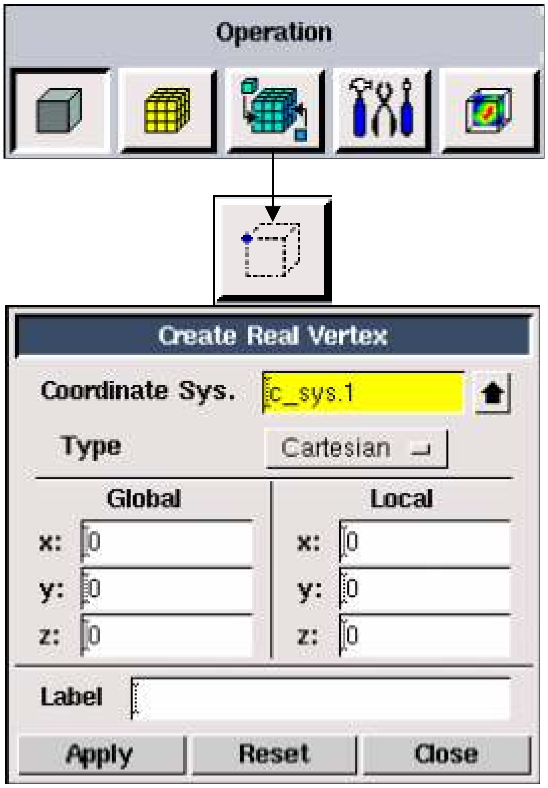
Procedure for model construction in GAMBIT

GAMBIT is a preprocessing component of Fluent where we set up the geometry of a fluidic domain and assign its boundary and fluid zones to prepare it for running a CFD simulation in the Fluent solver environment. GAMBIT is based on a graphical user interface and a hierarchical system of geometry creation. The user can develop the geometry in GAMBIT beginning with coordinates of points and stepping up to volumes. The first step for setting up geometry depends on the basic shape of the original geometry. We can use volumes or faces if the basic shape of the geometry corresponds to a fundamental shape such as a cylinder, sphere or parallelepiped. In the case of irregular geometries the best option is to start with point coordinates, draw the base in 2D and then extrude it to form a 3D entity.

The basic set of commands for creating any geometrical entity (point, edge, face or volume) in GAMBIT is

Operations -> Geometry -> Geometrical entity -> Create (entity) with (option)

Figure 49 shows an example of such a sequence of commands to create a vertex by specifying x, y and z coordinates.



Coordinate Sys. c_sys.1

Type Cartesian

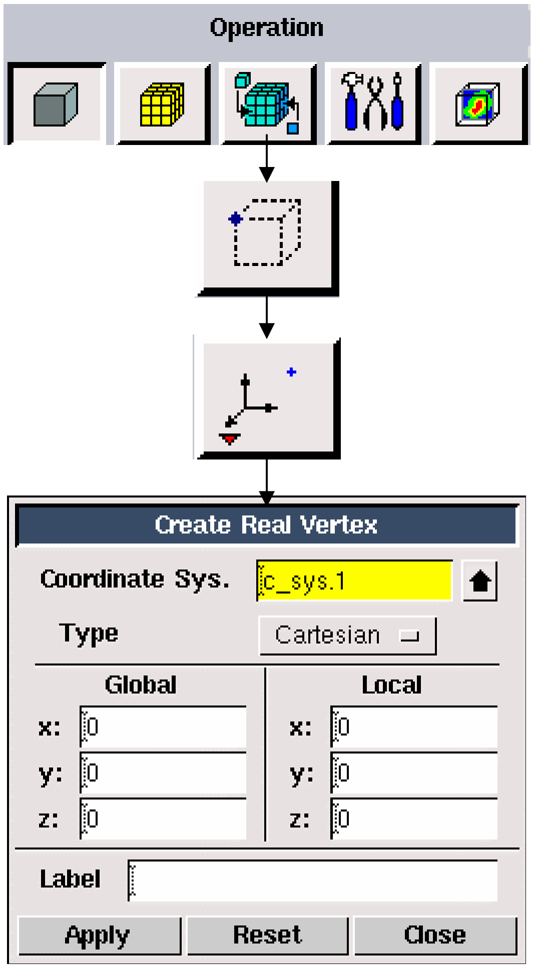
	Global		Local
x:	0	x:	0
y:	0	y:	0
z:	0	z:	0

Label

Figure 51: Vertex creation panel in GAMBIT

The options that can be specified with each geometrical entity allow us to create it in different circumstances. As an example, a volume can be created by joining together sets of existing faces or by applying motion to an existing face along a specified path. The choice of the option, therefore, depends on whether the geometry is regular or irregular, symmetric or asymmetric. The sweep volume option, for instance, works with shapes that are symmetrical about an axis of rotation; it does not work with shapes that are irregular in terms of volume. The following part of this appendix illustrates the exact process for creating the geometry of our passive diffusion microdevice in GAMBIT; it serves as a tutorial for any user beginning to create geometries in GAMBIT.

We set up the initial set of vertices in GAMBIT by using the Geometry icon (the first below), clicking on vertex and finally on ‘Create new vertex’.



Coordinate Sys. c_sys.1

Type Cartesian

	Global		Local
x:	0	x:	0
y:	0	y:	0
z:	0	z:	0

Label

Figure 52: Vertex creation panel in detail

The coordinates for the square base are a function of the width of the channel and are determined using geometry and trigonometry. Let w be the width of the channel and let us assume that we have extended a corner of the chamber to form a square with a vertex at the corner, as shown in Figure 53.

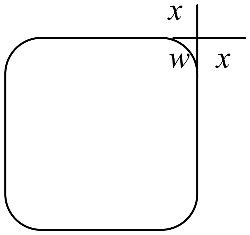


Figure 53: Schematic for determining the coordinates of the chamber vertices

According to the above figure, the two small sides belong to a right angled triangle and are equal in length since they are parts of edges of a square. Therefore, we can conclude

$$2x^2 = w^2$$
$$\Rightarrow x = \sqrt{\frac{w}{2}}$$

and using the value of x we can find the x and y coordinates for the vertices of the chamber. If the length of each side of the chamber is denoted by l and the vertices of the chamber by a set of points $P_1 ....P_8$, the points are specified as

Table 4: Symbolic measures for the x and y coordinates of the chamber vertices

Vertex	X	Y
P_1	$l-\sqrt{\frac{x}{2}}$	l
P_2	l	$(l-\sqrt{\frac{x}{2}})$
P_3	l	$-(l-\sqrt{\frac{x}{2}})$
P_4	$l-\sqrt{\frac{x}{2}}$	$-l$
P_5	$-(l-\sqrt{\frac{x}{2}})$	$-l$
P_6	$-l$	$-(l-\sqrt{\frac{x}{2}})$
P_7	$-l$	$l-\sqrt{\frac{x}{2}}$
P_8	$-(l-\sqrt{\frac{x}{2}})$	l

We use the ‘Create Straight Edge’ option to join these vertices and form the edges.

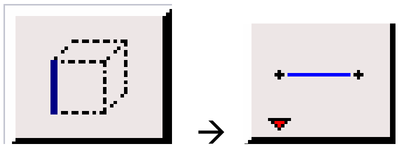


Figure 54: Edge creation panel in GAMBIT

We copy the vertices on one corner using the 'Move/Copy Vertices' tool. The *x* and *y* coordinates for translating the vertices are determined using the following method.

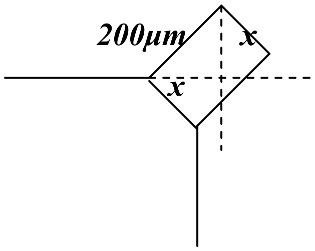
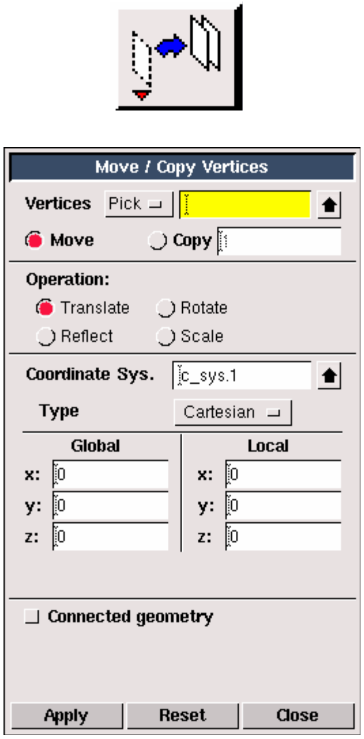


Figure 55: Schematic for determining the coordinates of the channel vertices

According to the above figure, we can write

$$2x^2 = 200^2$$
$$\Rightarrow x = 144.41$$

Figure 56 shows the Move/Copy Vertices panel where the parameters for copying vertices are filled.



Vertices Pick

● Move ○ Copy 1

Operation:

● Translate ○ Rotate

○ Reflect ○ Scale

Coordinate Sys. c_sys.1

Type Cartesian

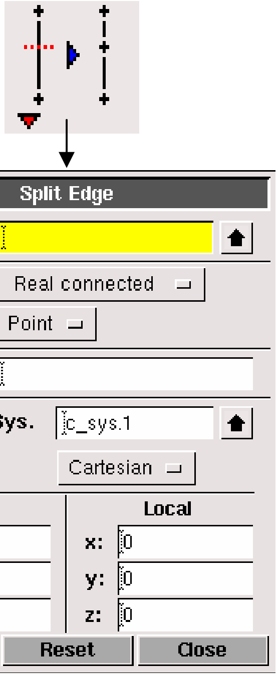
Global		Local	
x:	0	x:	0
y:	0	y:	0
z:	0	z:	0

□ Connected geometry

Figure 56: The Move/Copy vertices panel in GAMBIT

We connect the copied vertices to their parent vertices to create the narrow channel.

We also bisect the outer edge of the narrow channel using the ‘Split Edge’ tool and entering 0.5 in the U-value section as shown in Figure 57.



Split Edge

Edge:

Type: Real connected

Split With: Point

U Value:

Coordinate Sys.: c_sys.1

Type: Cartesian

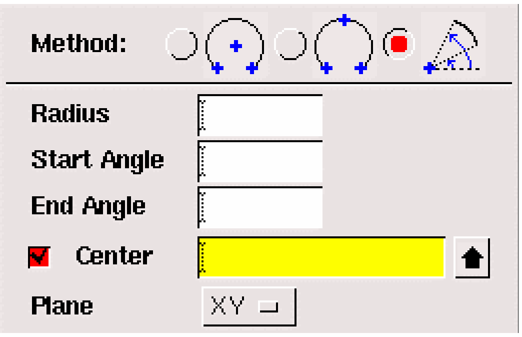
Global		Local	
x:	0	x:	0
y:	0	y:	0
z:	0	z:	0

Figure 57: The Split Edge panel in GAMBIT

We know the area of the reservoir and for simplicity of both geometry creation and calculation we create a reservoir of semicircular shape. By fitting the area of the reservoir against the given value we can find the radius of the semicircle. We use the equation

$$\frac{\pi r^2}{2} = A_r$$

where r is the radius of the semicircle and A_r is the given area of the reservoir. We use the "Create Real Edge" toolbar and select circular edge as an option. In the "Create Circular Edge" panel we again select the option for drawing an arc by specifying the start and end angles and the radius. By trial and error we can ascertain the start and end angles as -45° and 135° respectively.



Method:

Radius

Start Angle

End Angle

☒ Center

Plane XY

Figure 58: The Circular Edge drawing panel in GAMBIT

We join the end points of the arc to the end vertices of the channel with straight edges. We also delete the outer edge of the channel as well as the bisection vertex without including the lower geometry to make it convenient for us to extrude the device to 3D. We use the 'Create real face' option to create a face, including all the remaining edges.

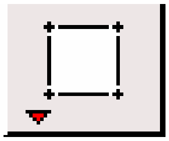
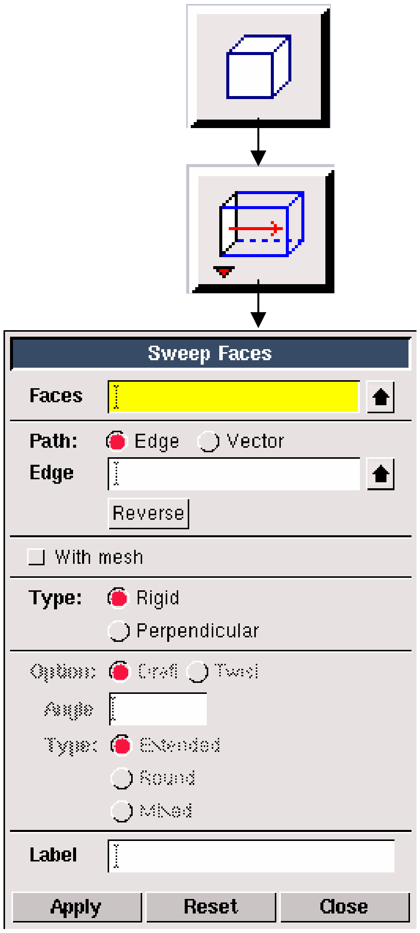


Figure 59: The Create Face button in GAMBIT

We create the initial volume from the face created in the above step. We use the 'Sweep Faces' option from the volume toolbar to do a perpendicular sweep in the positive Z direction with a magnitude of 26, a draft angle of 360 degrees and the 'mixed' option. The result produces the 3D view of the chamber, channel and reservoir.



Sweep Faces

Faces

Path: ● Edge ○ Vector

Edge

Reverse

□ With mesh

Type: ● Rigid

○ Perpendicular

Option: ● Draft ○ Twist

Angle

Type: ● Extended

○ Round

○ Mixed

Label

Apply Reset Close

Figure 60: The Sweep Faces option panel in GAMBIT

We now need to draw the boundaries separating the chamber, channel and reservoir. We draw in the faces that separate the reservoir, channel and chamber by joining the edges of the channel on both sides and using the “Create Face” and “Split Volume with Face” tools.

We have to draw the central sink to complete drawing the basic geometry. We know the radius of the central sink and can use this value to determine two points on the circumference of the base of the sink. These two points along with the center are required by GAMBIT to draw a circle. We can find the coordinates of these points by drawing an isosceles right-angled triangle within the circle and bisecting the right angle. The resulting triangles are also isosceles and right-angled and consequently, we can write

$$2x^2 = R^2$$

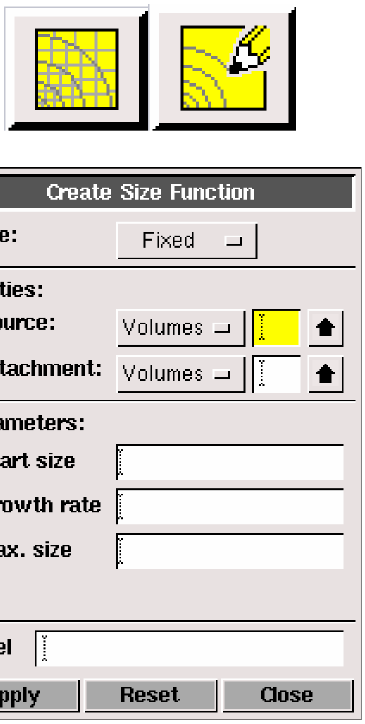
where R is the radius of the sink. We use the “Create Circular Face” tool to create this circle. To extrude the sink to its correct height we sweep this circular face using the perpendicular option by 4000 units along the positive Z-axis with a draft angle of 360 degrees and a ‘mixed’ option.

We need to make some adjustments in geometry for ease of meshing. As we are using structured and regular meshes with rectangular or hexahedral elements we should have volumes that are also regular. We split the reservoir in half using the “Split Edge with Vertex”, “Create Straight Edge”, “Create Face from Edges” and “Split Volume with Face” tools. We also split the cylindrical sink into 4 parts along the XZ and YZ planes for ease of meshing.

We need size functions for meshing all parts of the device with the exception of the narrow channel since that itself is the structure that creates the high aspect ratio in the device. The cylinder is also meshed without a sizing function as it is not connected directly to the channel. The size function created for each of the other parts is the same and the start size of the function always corresponds to the mesh density of the channel. Using the ‘Tools’ menu and the ‘Create Sizing Function’ submenu we enter the parameters for the sizing function as follows

- Start size: 10
- Growth rate: 1.2
- End Size: 100

The source for the function can be specified as the edges, faces or volumes. Using the lowest denomination leads to refined meshing and therefore we specify it as edges. We use all the edges at the corner for the chamber and for the reservoir we use the edges that are common with the channel. The attachment is the volume to which we want to attach this sizing function and we specify it accordingly in each case.



Create Size Function

Type: Fixed

Entities:

Source: Volumes

Attachment: Volumes

Parameters:

Start size

Growth rate

Max. size

Label

Figure 61: The panel for creating size functions in GAMBIT

We can now mesh the reservoir using the “Mesh Volumes” option and unchecking the “Apply Sizing” option as we have already created sizing functions for these parts. The small channel is meshed using a mesh size of 10. We mesh the top face of the chamber before meshing the entire volume in order to maintain uniformity in meshing. The reason has been explained in detail in Appendix C. We mesh the top face of the chamber using the Pave option and Quad elements.

The entire volume of the chamber is then meshed using the Cooper scheme. The cylinder is then meshed uniformly with the Cooper scheme and a mesh size of 250.

The last step in GAMBIT is to specify the boundary and fluid zones for the device. Boundary zones are specified as the no-flux zones or the walls within which the fluid will be contained. The space inside the boundaries is specified as fluid zones. We specify two fluid zones—one for the reservoir and another for the rest of the device. This makes it convenient for us to assign an initial fluid concentration to the reservoir in Fluent. The mesh is exported from GAMBIT using the “Export” option from the “File” menu.

APPENDIX B

Procedure for simulation setup in Fluent

Fluent is the main solver component of the package and mesh files exported from GAMBIT can be read into it as cases. The most fundamental and significant steps carried out in Fluent for simulating any fluidic process are

- Scaling and manipulation of grid which includes conversion to unstructured grids
- Specification of any physical model, if used and the solver model
- Specification of materials used in the simulation
- Setting up of boundary conditions (initial values for parameters like velocity and pressure)
- Defining scalars, function or memory, if used, for the simulation
- Specification of solution properties and under-relaxation factors
- Initialization of the solution and setting p of display windows
- Definition of time schemes (fixed/adaptive) and timestep size(s)

The procedure for setting up the simulation for our microfabricated diffusion device has been given below, as a sample tutorial for customizing Fluent to set up a simulation for a given physical fluidic situation.

We read in the mesh in FLUENT from the GAMBIT files.

File → Write → Case

We scale the grid by a factor of 1E(-06) to convert it to micrometers

Grid → Scale

We check the grid for inverted or skew elements using the ‘Check Grid’ option

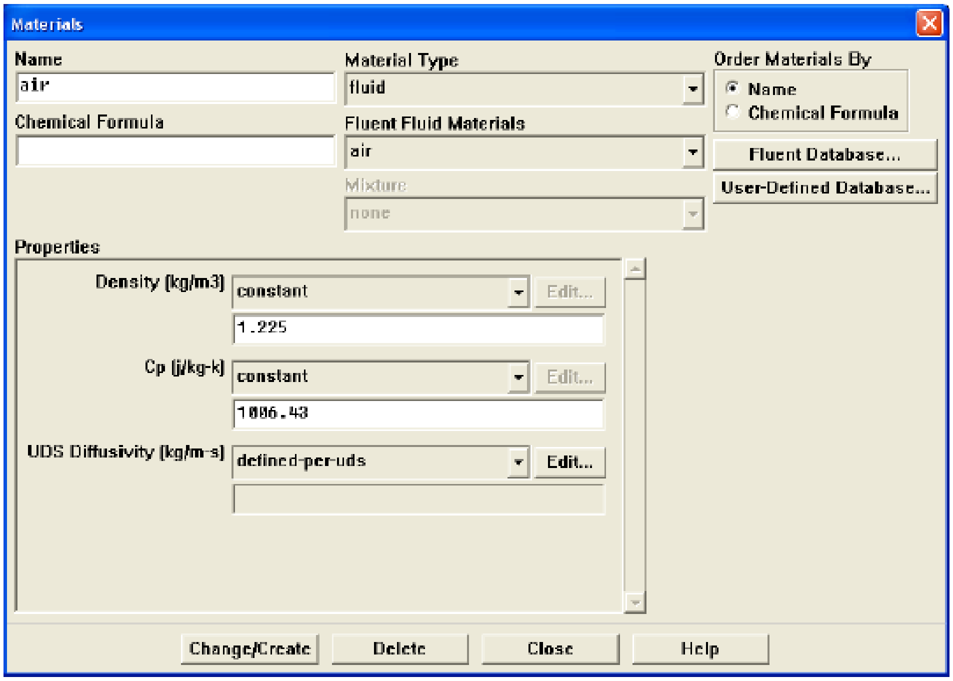
Grid → Check

We now define the solver model by using the 'unsteady' solver option and choose the second-order explicit time discretization scheme.

Define → Model

We also define the materials for the simulation. We select 'Fluent Database' in the materials and copy 'water-liquid' from it on to the solver panel.

Define → Materials → Fluent Database



Name: air

Material Type: fluid

Order Materials By: ● Name ○ Chemical Formula

Chemical Formula:

Fluent Fluid Materials: air

Mixture: none

Properties

Density (kg/m3): constant, 1.225

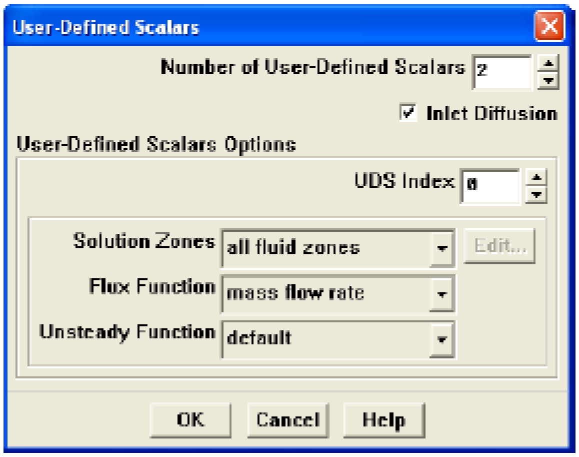
Cp (j/kg-k): constant, 1006.43

UDS Diffusivity (kg/m-s): defined-per-uds

Figure 62: The Materials panel in Fluent

Our diffusive material is to be denoted by a scalar and so we define a single UDS (user-defined scalar) for the experiment.

Define → User-defined → Scalars



Number of User-Defined Scalars 2

☑ Inlet Diffusion

UDS Index 0

Solution Zones all fluid zones

Flux Function mass flow rate

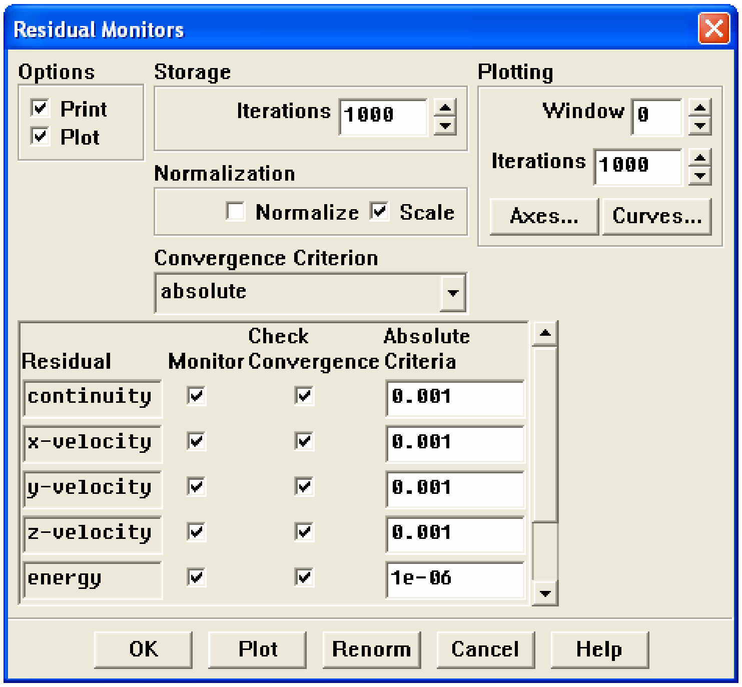
Unsteady Function default

Figure 63: The user-scalar definition panel in Fluent

We now go back to the materials section and fill in the diffusion coefficient value in the 'UDS diffusivity' section.

In the 'Solve' panel we select the 'Monitors' option and set the display of residuals and time at every timestep

Solve → Monitors → Residuals/Statistic



Residual Monitors

Options

☑ Print
☑ Plot

Storage

Iterations 1000

Normalization

□ Normalize ☑ Scale

Convergence Criterion

absolute

Plotting

Window 0

Iterations 1000

Residual	Monitor	Check Convergence	Absolute Criteria
continuity	☑	☑	0.001
x-velocity	☑	☑	0.001
y-velocity	☑	☑	0.001
z-velocity	☑	☑	0.001
energy	☑	☑	1e-06

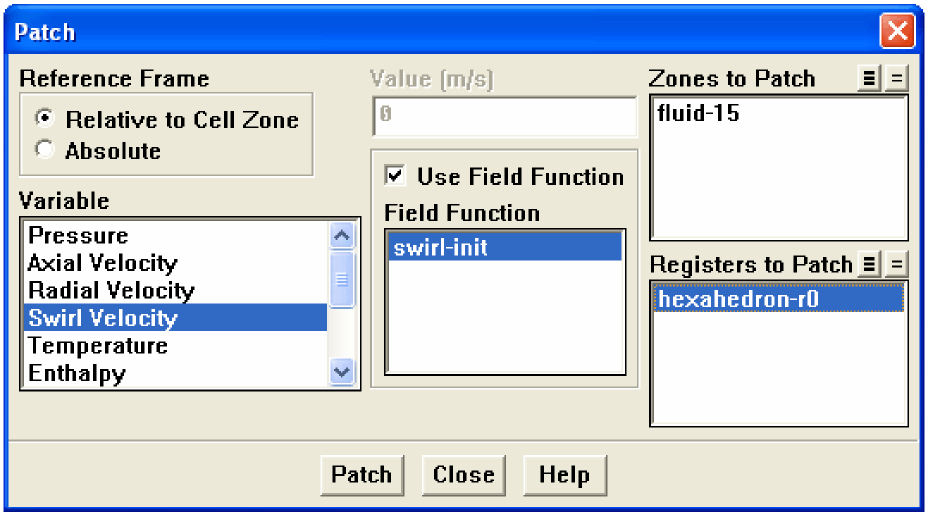
Figure 64: The residuals panel in Fluent

We can now specify the solution method for the experiment from the ‘Solution Controls’ panel. We de-select the ‘Flow’ option as and solve for the UDS only. As we are not solving for flow the default values can be used for the other parameters.

Solve → Controls → Solution

Provide the initial conditions for the experiment. Using the initialize option, first initialize the entire flow field to zero and then using the ‘Patch’ option initialize the fluid zone in the reservoir to the desired initial concentration.

Solve → Initialize → Initialize/Patch



Reference Frame

● Relative to Cell Zone
○ Absolute

Variable

Pressure
Axial Velocity
Radial Velocity
Swirl Velocity
Temperature
Enthalpy

Value (m/s) 0

☑ Use Field Function

Field Function

swirl-init

Zones to Patch

fluid-15

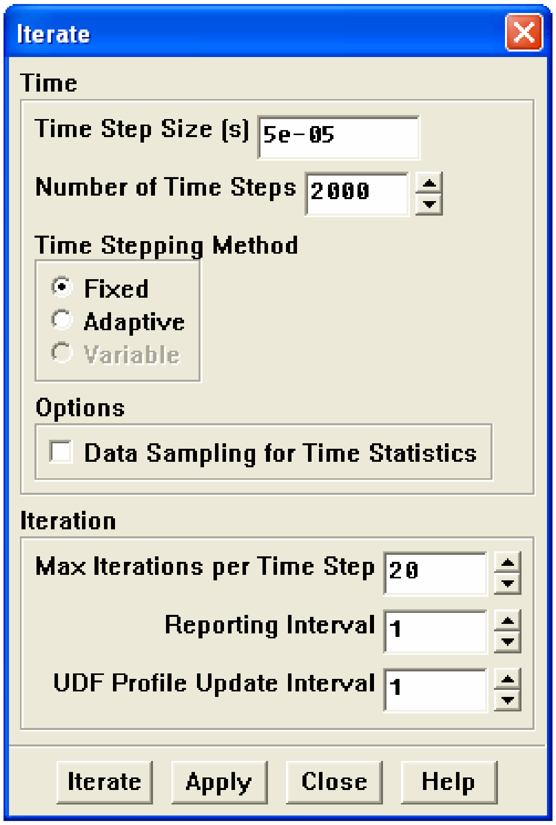
Registers to Patch

hexahedron-r0

Figure 65: The ‘Patch’ initialization panel in Fluent

To start running the experiment we use the ‘Iterate’ option. We define fixed timesteps with each timestep equal to 3600. In the ‘Autosave’ option, enable files to be written out at every timestep. The ‘frequency’ in this section denotes timestep frequency and not actual time.

Solve → Iterate



Iterate

Time

Time Step Size (s) 5e-05

Number of Time Steps 2000

Time Stepping Method

● Fixed
○ Adaptive
○ Variable

Options

□ Data Sampling for Time Statistics

Iteration

Max Iterations per Time Step 20

Reporting Interval 1

UDF Profile Update Interval 1

Figure 66: The iteration and time-step definition panel in Fluent

APPENDIX C

User-defined Functions and their utilization

User-defined functions (UDFs) are an additional feature provided in Fluent that allow the user to add a number of enhancements to the standard features of the Fluent code. The additional capabilities that a user can add to a simulation by hooking a UDF to it are

- Customization of boundary conditions, material property definitions, surface and volume reaction rates, source terms in Fluent transport equations, source terms in user-defined scalar (UDS) transport equations, diffusivity functions, etc.
- Adjustment of computed values on a once-per-iteration basis.
- Initialization of a solution.
- Asynchronous (on demand) execution of a UDF
- Execution at the end of an iteration, upon exit from Fluent, or upon loading of a compiled UDF library.
- Post-processing enhancement.
- Enhancement of existing Fluent models (e.g., discrete phase model, multiphase mixture model, discrete ordinates radiation model).

There are some fundamental rules that are applicable to every UDF that can be summarized as follows:

- UDFs are written in the C programming language.
- UDFs must have an include statement for the `udf.h` file.
- UDFs must be defined using `DEFINE` macros supplied by Fluent Inc.

- UDFs utilize predefined macros and functions supplied by Fluent Inc. to access Fluent solver data and perform other tasks.
- UDFs are executed as interpreted or compiled functions.
- UDFs are hooked to a Fluent solver using a graphical user interface panel.
- UDFs use and return values specified in SI units.

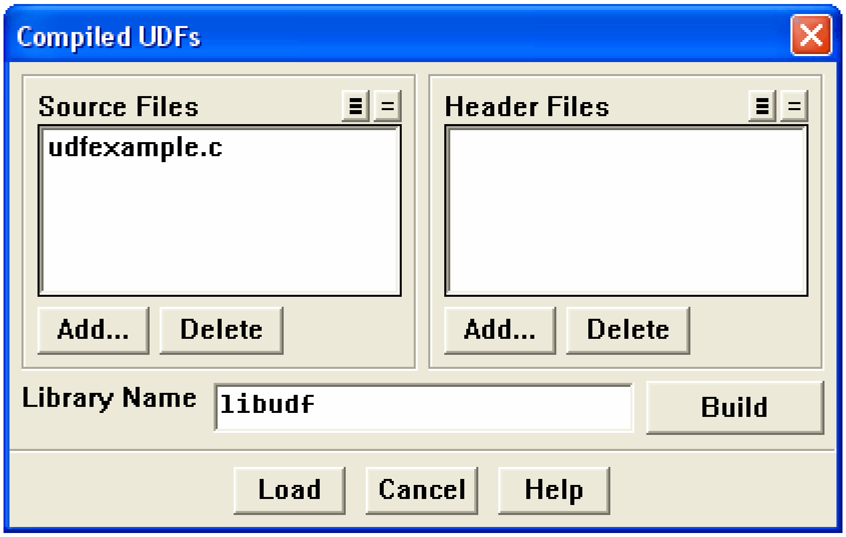
The procedure for developing a UDF and setting it to run with Fluent starts with the development of the UDF code in any standard text editor such as Notepad or WordPad. The written UDF is then compiled or interpreted in Fluent using a Windows compiling package such as Microsoft Visual Studio. It is recommended that UDFs be compiled rather than interpreted; UDFs containing inbuilt macros from Fluent have to be compiled. In order to compile a UDF we have to start the Visual Studio command prompt and invoke Fluent from there. Prior to this operation, we must also ensure that we have changed the environment setting for Fluent by using

Start -> Programs -> Fluent -> Set Environment

We follow a number of steps to create a shared library and compile the UDF using Microsoft Visual Studio after calling Fluent. The first step is to dynamically load the UDF in the Fluent simulation and store it in the form of a shared library that open whenever the particular case file is opened. This is done by using the 'Define UDF' panel in Fluent.

Define -> User-defined -> Functions -> Compiled -> Build -> Load

Figure 67 shows the 'Define UDF' panel in Fluent.



Compiled UDFs

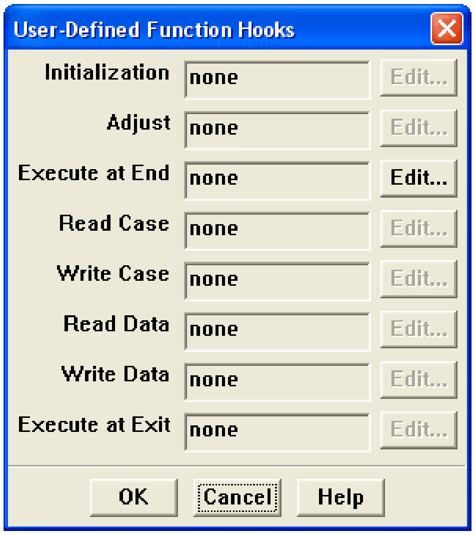
Source Files: udfexample.c

Header Files:

Library Name: libudf

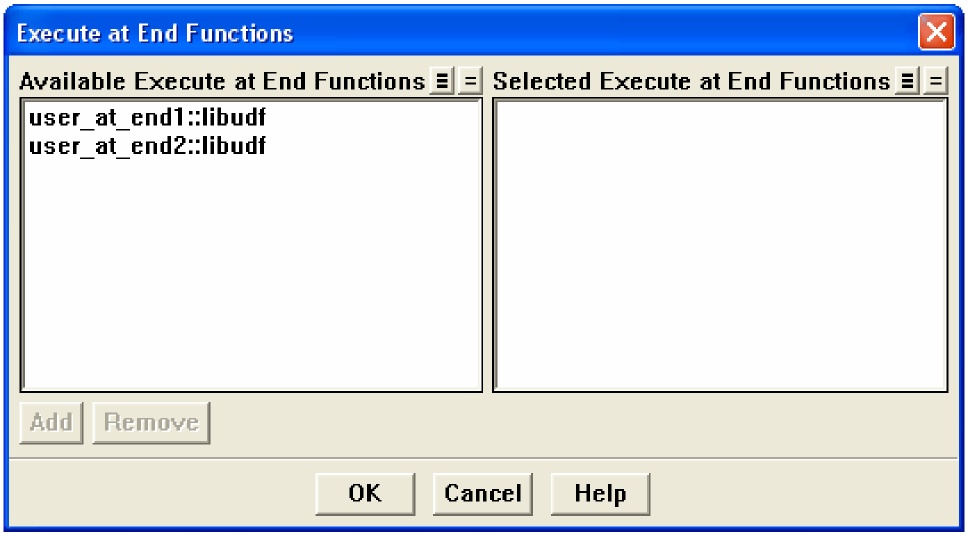
Figure 67: The 'Define UDF' panel in Fluent

The last step executed in integrating a UDF with Fluent is hooking it to the simulation which is done by using the user-defined hooks panel (Figure 68). We specify the UDF to be integrated with Fluent and save the case so that the UDF is loaded and hooked every time we reopen the case.



User-Defined Function Hooks

Initialization	none
Adjust	none
Execute at End	none
Read Case	none
Write Case	none
Read Data	none
Write Data	none
Execute at Exit	none


Execute at End Functions
Available Execute at End Functions
Selected Execute at End Functions
Add
Remove
OK
Cancel
Help

user_at_end1::libudf
user_at_end2::libudf

Figure 68: The user-defined hooks panels in Fluent

APPENDIX D

Discussion of issues encountered in GAMBIT and Fluent

Continuous phenomena modeled by a set of nonlinear partial differential equations possess a wide range of time scales. The best algorithms for approximating these continuous problems into discrete problems require the solution of very large sparse linear systems. This approximation requires us to take into account various sources of error such as under-relaxation factors and truncation values that are necessary to obtain convergent solutions. Software and hardware are also limited by the range of numbers they can compute. In a pure diffusion process such as in our microfabricated device it extends over extremely large time scales and results in values that vary over a logarithmic range. All these factors lead us to consider several issues while setting up a CFD simulation using GAMBIT and Fluent. These are either significant points that should be taken into consideration while setting up the simulation or they are enhancements to the process that provide greater accuracy.

The finite volume method provides us with the flexibility to use structured or unstructured meshes to perform calculations. There are distinct advantages and disadvantages to using both kinds of meshes. Unstructured meshes are mostly applied on complex geometries as they are not subject to as many rules as structured grids where each interior node has an equal number of adjacent elements. However, due to this relaxation of node valency, unstructured grids are not economical in terms of calculation time. On an average the ratio of the computational time required to run calculations on an unstructured grid compared to a structured one is 3:1 [47]. The most generally applicable structured meshing algorithms in GAMBIT are Map for 2D geometries and the Cooper scheme for 3D geometries. The Map scheme creates a regular structured grid of quadrilateral or mixed quadrilateral and triangular elements. In order for a face to be mappable, it must form a logical rectangle [48]

- It must have four end vertices and the remaining vertices should all be side vertices
- The number of mesh intervals on opposing sides of the rectangle must be equal

The Cooper scheme is the most flexible among the volume meshing schemes as it is associated with both hexahedral and wedge elements. In this scheme, GAMBIT treats the volume as a set of logical cylinders consisting of two end caps and a barrel. The caps act as the source faces and the barrels as the non-source faces. The sequence of operations in the Cooper meshing scheme is as follows [49]:

1. Create Map and/or Submap meshes on each of the non-source faces.
2. Imprint the source faces onto each other.
3. Mesh the source faces.
4. Project the source-face mesh node patterns through the volume

In our microfabricated passive diffusion device, we have to use a mixture of structured and unstructured mesh algorithms. The top surface of the chamber is a square with a circular boundary in the middle. The Cooper scheme treats two of the side faces as source faces and all other faces as non-source ones. The top surface of the chamber with a circular boundary in the middle is one of these non-source faces and therefore, has to be mappable or submappable. However, due to the circle in the middle, we cannot use the Map scheme in this case and have to mesh it first using an unstructured algorithm (pave) with quadrilateral elements. We can then mesh the rest of the volume using the Cooper scheme as it does not re-mesh the top face.

In the solver environment, Fluent, there are parameters that have to be optimized at particular steps and issues that have to be taken into consideration. Factors like the size of each timestep in a time-dependent simulation and by-products of computation such as numerical diffusion in data are examples of some of the relevant considerations in CFD calculations. The remaining part of this section discusses each such factor in detail and possible ways to circumvent them.

The most important factors to consider while setting up a CFD simulation in Fluent are the time discretization method and the timestepping method. Temporal discretization schemes are broadly classified into first-order and second-order categories. The fundamental objective is to replace the partial differential for time in continuous equations to a difference expression. Fluent uses backward differences to discretize time derivatives and using this method the first order method for the derivative $F(\phi) = \frac{\partial \phi}{\partial t}$ is [50]

$$\frac{\phi^{n+1} - \phi^{n}}{\Delta t}$$

The second order temporal discretization for the same expression is [51]

$$\frac{3\phi^{n+1} - 4\phi^{n} + \phi^{n-1}}{\Delta t}$$

Once the discretization has been done, the choice remains to choose the method by which to evaluate the values of $F(\phi)$, in particular the time level values of ϕ that should be used. The explicit method takes into account the following assumptions:

- If $\frac{\partial \phi}{\partial t}$ is calculated at instant t_n then the derivative for diffusion according to Fick's Second Law, $\frac{\partial^2 \phi}{\partial t^2}$, is also calculated at the same instant.
- The values ϕ_i^n at time t_n are known
- The resulting algebraic formulation has only one unknown: ϕ_i^{n+1}

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = D \frac{\phi_{i+1}^n + \phi_{i-1}^n - 2\phi_i^n}{\Delta x^2}$$

Explicit methods are not necessarily stable which implies that the timestep Δt cannot be chosen arbitrarily and has a limit up to which it can produce stable solutions. These methods are conditionally stable in time and their stability can be judged by two main criteria, the Courant Criterion and the von Neumann criterion [52].

Implicit methods take into account the following considerations:

- If the derivative $\frac{\partial \phi}{\partial t}$ is calculated at the instant t_{n+1} then the diffusion derivative $\frac{\partial^2 \phi}{\partial t^2}$ must also be calculated at the same time instant
- At t_{n+1} the values ϕ_i^{n+1} are known
- The resulting algebraic expression now has more unknowns: ϕ_{i+1}^{n+1} and ϕ_{i-1}^{n+1}

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = D \frac{\phi_{i+1}^{n+1} + \phi_{i-1}^{n+1} - 2\phi_i^{n+1}}{\Delta x^2}$$

Implicit methods are unconditionally stable in time and therefore, a good choice for time-dependent problems. Hence, although the choice of the timestep is not dependent on the user in the implicit method we still cannot choose it arbitrarily. A good way for judging the choice of Δt is to observe the number of iterations that Fluent needs to converge at each timestep. The ideal number of iterations per timestep is 5-10 and if Fluent requires substantially more timesteps (~5) to converge, then the timestep is too large. A time-dependent problem usually has a very high rate of producing transient values towards the beginning—a characteristic that also delays rapidly. The initial size of the timestep is crucial in this regard and needs to be conservatively small. It can gradually be increased to the maximum stable value, generally in the order of 10-15 seconds, as the calculation proceeds.

Fluent provides us with the capability to choose between two timestepping methods to execute our CFD simulations—fixed and adaptive. The fixed timestepping method, as suggested by the name, takes a single value for Δt that is constant throughout the calculation. The adaptive method provides us with a method to increase the size of the timestep in the course of the calculation. We specify the truncation error tolerance that denotes a threshold value to which the truncation error is compared, the minimum timestep size, the maximum timestep size and the number of fixed timesteps to execute. This allows us to have more flexibility in stabilizing the initial stage of the simulation and then increase the timestep size. In case of our device, although the adaptive timestepping algorithm can be used we determined through ten trial runs the optimum size of the timestep as 10 seconds. Moreover, the adaptive timestepping method also provides for ten seconds as the maximum timestep size.

Fluent, like all other software packages, has a computational limit in terms of the precision of numbers that it can calculate. The smallest machinable number that Fluent can calculate and store is in the order of 1E-35. Numbers in the range of 1E-26 are the maximum values around which stable solutions can be expected—beyond this range Fluent begins to enter the realm of numerical diffusion which has been discussed in further detail in the latter parts of this section. This issue of dynamic range and computational capability is common to all software and in this case, although we can scale up our initialization values to circumvent it, there is also a ceiling on the initial values; extremely high concentrations applied to a microscale device can cause overflow conditions in Fluent which (via software bug) results in the calculation of values in coordinates where there is no geometry.

Numerical diffusion is a dominant source of errors that we encounter in all multidimensional CFD simulations. In Eulerian simulations, time and space are divided into discrete grids and continuous differential equations such as the Navier-Stokes and convection-diffusion equations are discretized into finite-difference equations. This phenomenon is also referred to as false diffusion as the diffusion occurring in it is not a real phenomenon but its effect is analogous to that of increasing the real diffusion coefficient [53].

Numerical diffusion is employed by numerical models to control small-scale noise in the model that can arise from nonlinear instability, discontinuous physical processes and external forcing. However, this scheme itself introduces noise in the solution and can also lead to overshooting or undershooting of the solution. The second order diffusion term in Fick's Second Law has a Laplacian form and always involves the transfer from regions of higher values to regions of lower values. Therefore, it cannot create any new extrema where there are physically none present.

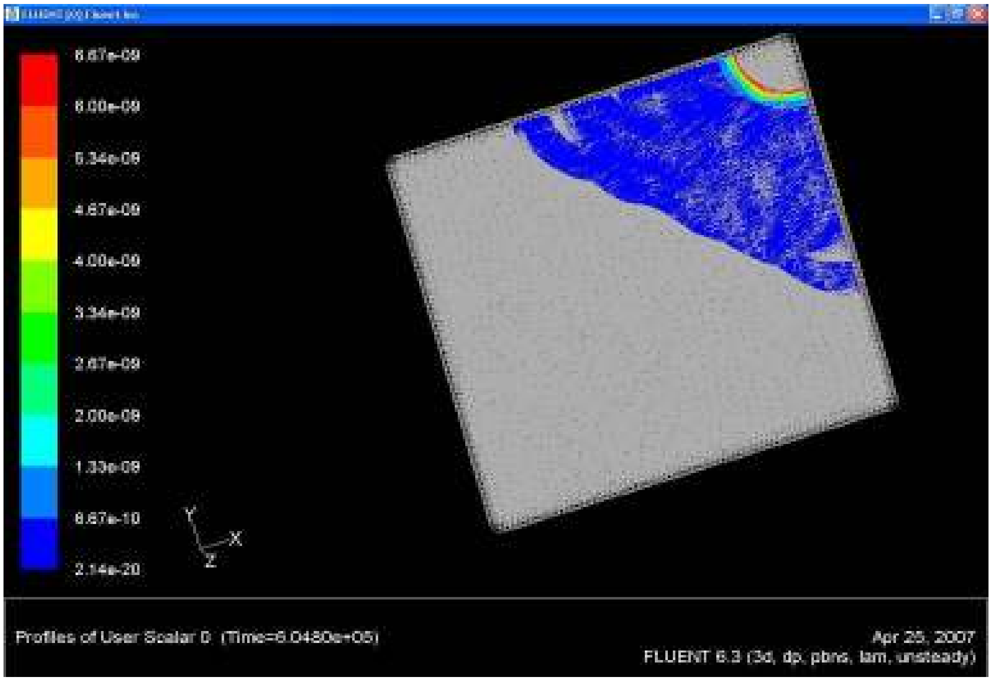


Figure 69: Numerical diffusion in a portion of the chamber of the passive diffusion device

The above figure demonstrates an example of the appearance of data when numerical diffusion takes place. The concentration of the diffused substance along a plane within the chamber of the device has been plotted as a series of contours. The initial levels beginning with red are in agreement with the data. Further along the plane though, we see discontinuous patches of blue which look as if diffusion has been forced in those regions—this has happened due to numerical diffusion. A portion of the discrepancies are also attributed to the contour algorithms in software packages—a topic we have discussed later in this section. However, we must remember that during discretization on regular grids, flow is aligned conventionally in along the direction of the coordinate axes rather than in the direction of the grid. If fluid flows along a diagonal then any numerical scheme can, at best, transfer the fluid in vertical and horizontal directions. After some time, the fluid spreads through the grid due to this sideways transfer. The results are seen as a numerical effect corresponding to higher diffusion rates. Contour plots display patches of fluid in areas of the geometry where the fluid has not physically propagated when the range of values drops below the order of 1E-20m^2/s.

Based on the above discussion, we know the following facts about numerical diffusion:

- Numerical diffusion is most noticeable when the real diffusion is small, i.e., the diffusion coefficient is very small.
- All practical computational schemes for solving fluid flow contain a finite amount of numerical diffusion as it arises from truncation errors that are a result of representing fluid flow equations in discrete form.
- The second order discretization scheme can help reduce the effects of numerical diffusion
- Numerical diffusion is minimized when the flow is aligned with the grid and this is determined by the choice of the grid. Triangular grids are never aligned with the flow while in hexahedral or quadrilateral meshes the flow is aligned to the mesh only when the surface or volume within which the flow is taking place is a regular shape such as a rectangle or hexagon.

Contour plots are extremely useful tools in visualizing any type of flow and are a significant part of data processing. While there exist a number of programs to read data extracted from Fluent and arrange it in regularly spaced contours, all of these programs are limited in the range of data that they can process and plot. Firstly, all software programs are limited by the smallest machinable value that they can read and plot. In Matlab, this value is in the order of 1E-16 while in more specialized programs such as Tecplot, this value increases to 1E-21. Our limit of control over the contour levels is also restricted as the number of levels that we can set to get accurate contour lines for a range of data is all interrelated. The fundamental method for setting contour levels in all software programs is to calculate the range of the data, divide it into the required number of levels and assign each level a number (Matlab). This method is adequate for data with short ranges but in order to visualize a large range of data completely, we either need to visualize it in parts or visualize the logarithms of the original values. In our case, we have used the original concentration or gradient values in Tecplot.

In Fluent, we simulate diffusion in the entire device but our area of interest does not encompass the complete geometry. In such cases, we need to employ the data extraction methods provided in Fluent to obtain the data from the part of the geometry that is relevant to us. Fluent provides us with the capability to extract data from various geometrical entities from straight lines to iso-surfaces. The number of points in these entities can be specified or can be extracted along the original grid where the entity passes through it. In cases where we need visualize properties such as concentration contours, it is best to extract the points along the original grid. When we specify the number of points to extract on a plane, the values are interpolated if the number of points specified is greater that the number of actual grid points on that particular surface. This leads to loss in accuracy; we also cannot export this data as a grid to programs like Tecplot since these points do not actually constitute a grid. Similarly, when we plot the logarithmic values of concentration of diffused material along half the diagonal of the chamber, we use the lines along grid points rather than specifying the number of points in order to avoid interpolation.

APPENDIX E

Matlab and C codes used in this model

1. Matlab code for plotting concentration logarithms

```
%Program to make logarithmic pots of concentration
%close all; clear all;
format long;
fname = input('Enter the line file you want to open')
A = dlmread(fname);

%Defining the arrays
x = A (:,6);
con = A(:,5);
y = A(:,3);

%Getting rid of undershoots and conversion to actual concentration values
for i = 1:length(con)
    if con(i,:)<0
        con(i,:)=(-1000)*con(i,:)
    else
        con(i,:)=1000*con(i,:)
    end
end
con;

% Defining the arrays for logarithmic plotting
% Catching exceptions in the concentration values

[r,c,v]=find(con)
v;
diff = length(con)-length(v)
diff;
for j=1:length(v)
    x1(j,:)=x(diff+j,:)
end
x1;
semilogy(x,con,'o','MarkerFaceColor','g');
xlabel('X (m)');
ylabel('LOG[C] (M)');
title('[channel-width] micron channel device-[time] hrs');
```

2. Matlab code for creating sampling plots

```
%Program to make logarithmic pots of concentration
close all; clear all;
fname = input('Enter the line file you want to open')
A = dlmread(fname);

%Defining the arrays
X = A (:,2);
Y = A(:,3);

plot(X,Y,'*','MarkerEdgeColor','r');

xlabel('X(m)');
ylabel('Y(m)');
title('Sampling of points in the [channel-width] micron channel device at
[time] hrs');
```

3. Matlab code for creating concentration and gradient plots

```
%Program to make plots of linear data
close all; clear all;
format long;
filename = input('Enter the file name to open')
A = dlmread(filename)
format long;
%Setting the arrays
x = A(:,2)
c = A(:,5)
%Getting rid of undershoots in concentration and conversion to moles
for i=1:length(c)
    if c(i,:)<0
        c(i,:)=(-1000)*c(i,:)
    else
        c(i,:)=1000*c(i,:)
    end
end
c;
for j=2:length(c)
    g(j,:)=c(j-1,:)-c(j,:)
end
g;
for k=1:(length(g)-1)
    cg(k,:) = g(k+1,:)
end
cg;
for l=1:(length(x)-1)
    xg(l,:)=x(l+1,:)
end
```

```
xg;

for m=1:length(xg)
    cgrad(m,:)=cg(m,:)/(xg(1,1)-xg(2,1))
end
cgrad;
[AX,H1,H2]=plotyy(x,c,xg,cgrad);
set(get(AX(1),'Ylabel'),'String','C (M)');
set(get(AX(2),'Ylabel'),'String','Gradient (M/m)');
xlabel('X (m)');
title('20micron channel device-264hrs');
```

4. Matlab Code for creating conclusion plots

```
%Program to make simultaneous plots

f1 = input('Enter the first file');
A = dlmread(f1);
f2 = input('Enter the second file');
B = dlmread(f2);
f3 = input('Enter the third file');
C = dlmread(f3);
f4 = input('Enter the fourth file');
D = dlmread(f4);
f5 = input('Enter the fifth file');
E = dlmread(f5);

%Defining the matrix columns
x1 = A(:,2);
c1 = A(:,5);
x2 = B(:,2);
c2 = B(:,5);
x3 = C(:,2);
c3 = C(:,5);
x4 = D(:,2);
c4 = D(:,5);
x5 = E(:,2);
c5 = E(:,5);

%Getting rid of undershoots and conversion to actual concentration values
for i = 1:length(c1)
    if c1(i,:)<0
        c1(i,:)=(-100)*c1(i,:)
    else
        c1(i,:)=100*c1(i,:)
    end
end
for i = 1:length(c2)
    if c2(i,:)<0
        c2(i,:) = (-10)*c2(i,:)
    else
        c2(i,:) = 10*c2(i,:)
    end
end
```

```
for i = 1:length(c3)
    if c3(i,:)<0
        c3(i,:) = (-1000)*c3(i,:)
    else
        c3(i,:) = 1000*c3(i,:)
    end
end
for i = 1:length(c4)
    if c4(i,:)<0
        c4(i,:) = (-1000)*c4(i,:)
    else
        c4(i,:) = 1000*c4(i,:)
    end
end
for i = 1:length(c5)
    if c5(i,:)<0
        c5(i,:) = (-1000)*c5(i,:)
    else
        c5(i,:) = 1000*c5(i,:)
    end
end

%Making the actual plots
semilogy(x1,c1,'-',x2,c2,'--',x3,c3,'-*',x4,c4,'-o',x5,c5,'-x');
xlabel('X (m)');
ylabel('LOG[C] (M)');
```

5. UDF in C for computing cell gradients

```
#include "udf.h"
#include <stdio.h>

DEFINE_EXECUTE_AT_END(execute_at_end)
{

/* Declaration of variables*/

Domain *d;
Thread *t;
cell_t c;

d = Get_Domain(1);  /* single-phase flow therefore, one domain */

/* Looping over all the cell threads in the domain */
thread_loop_c(t,d)
{
        if(FLUID_THREAD_P(t))
        {
                begin_c_loop(c, t)
                {
```

```
            C_UDMI(c,t,0) = C_UDSI_G(c,t,0)[0];
            C_UDMI(c,t,1) = C_UDSI_G(c,t,0)[1];
            C_UDMI(c,t,2) = C_UDSI_G(c,t,0)[2];
        }
        end_c_loop(c,t)
    }
}
/*printf("Gradient of UDS: %d\n", Grad);*/
fflush(stdout);
}
```

REFERENCES

[1] Lauffenberger, Douglas A. and Horwitz, Alan F., 1996, *Cell Migration: A Physically Integrated Molecular Process*, Cell, 84, pp. 359-369.

[2] Dimilla, P.A., Barbee, K and Lauffenberger, D.A., 1991, *Mathematical Model for the Effects of Adhesion and Mechanics on Cell Migration Speed*, Biophysical Journal, 60(1), pp.15-37.

[3] Horwitz et al., 2003, *Cell Migration: Integrating Signals from Front to Back*, Science, 302, pp. 1704-1709.

[4] Attinger, C.E., Janis, J.E. and Broughton, G., 2006, *The Basic Science of Wound Healing*, The Journal of Plastic and Reconstructive Surgery, 117, pp. 12S-34S.

[5] Butterfield, T.A., Best, T.M. and Merrick, M.A., 2006, *The Dual Roles of Neutrophils and Macrophages in Inflammation: A Critical Balance between Tissue Balance and Repair*, Journal of Athletic Training, 40(1), pp. 457-465.

[6] Angelo, L.S. and Kurzrock, R, 2007, *Vascular Endothelial Growth Factors and its relationship to Inflammatory Mediators*, Clinical Cancer Research, 13(10), pp. 2825-2830.

[7] Park, J. and Barbul, A., 2004, *Understanding the role of immune regulation in wound healing*, The American Journal of Surgery, 187(5, S1), pp. S11-S16.

[8] Raulet et al., 2000, *Ligands for the murine NKG2D receptor: expression by tumor cells and activation of NK cells and macrophages*, Nature Immunology, 1, pp. 119-126.

[9] Sahai, Erik, 2005, *Mechanisms of cancer cell invasion, Current Opinion in Genetics and Development*, 15, pp. 87-96.

[10] Eccles, Suzanne A., 2005, *Targeting key steps in cancer metastatic progression*, Current Opinion in Genetics and Development, 15, pp. 77-86.

[11] Donner, David B., et al., 1995, *Vascular Endothelial Cell Growth Factor promotes tyrosine phosphorylation of mediators of signal transduction that contain SH2 domains*, Journal of Biological Chemistry, 270 (12), 6729-6733.

[12] Gallagher, J.T., 2006, *Multiprotein Signaling Complexes: Regional Assembly on Heparan Sulphate*, Biochemical Society Transactions, 34, pp. 438-441.

[13] Balkwill, Fran, 2004, *Cancer and the chemokine network*, Nature Reviews| Cancer, 4, pp. 540-550.

[14] Bailey, C, Negus, R, Morris, A et al., 2007, *Chemokine Expression is associated with the accumulation of Tumor-Associated Macrophages (TAMs) and Progression in Human Colorectal Cancer*, Clinical and Experimental Metastasis, 24(2), pp. 121-130.

[15] Newgreen, Donald F., McKeown, Sonja J. and Farlie, Peter G., *The Neural Crest: Basic biology and clinical relationships in the craniofacial and enteric nervous systems*, Birth Defects, 72, pp. 173-189.

[16] Thiery, Jean Paul, Duband, Jean Loup and Tucker, Gordon C., 1985, *Cell migration in the vertebrate embryo: Role of cell adhesion and tissue environment in pattern formation*, Annual Reviews in Cell Biology, 1, pp. 91-113.

[17] Kanzler, B., Foreman, R.K., Labosky, P.A. and Mallo, M., 2000, *BMP signaling is essential for the development of skeletogenic and neurogenic cranial neural crest*, Dvelopment, 127(5), pp. 1095-1104.

[18] Monsoro-Burq, Anne-Helene, Fletcher, Russell B. and Harland, Richard M., 2003, Neural crest induction by paraxial mesoderm in Xenopus embryos requires FGF signals, Development, 130, pp. 3111-3124.

[19] Buehr, Mia, 1997, The primordial germ cells of mammals: some current perspectives, Experimental Cell Research, 232, pp. 194-207.

[20] Zigmond, Sally H. and Hirsch, James G., 1973, Leukocyte locomotion and chemotaxis: New methods for evaluation and demonstration of a cell-derived chemotactic factor, The Journal of Experimental Medicine, 137, pp. 387-410.

[21] Zicha, D., Dunn, D.A. and Brown, A.F., 1991, A new direct-viewing chemotaxis chamber, Journal of Cell Science, 99, pp. 769-775.

[22] Kanegasaki, S et al., 2003, A novel optical assay system for the quantitative measurement of chemotaxis, Journal of Immunological Methods, 282(1-2), pp. 1-11.

[23] Jeon et al., 2002, Neutrophil chemotaxis in linear and complex gradients of interleukin-8 formed in a microfabricated device, Nature Biotechnology, 20, pp. 826-830.

[24] Dertinger, S.K.W. et al., 2001, Generation of gradients having complex shapes using microfluidic networks, Analytical Chemistry, 73, pp. 1240-1246.

[25] Lin et al., 2002, Extension of devices for linear and non-linear gradients, Lab on a Chip, 9, pp. 29-33.

[26] Fluent v 6.2, subsidiary of ANSYS Inc., 2004, New Hampshire, USA.

[27] GAMBIT, preprocessor, Fluent v 6.2, 2004, New Hampshire, USA.

[28] Cussler, E.L., 1997, Diffusion: Mass Transfer in Fluid Systems, 2nd Edition, Chapter I, p. 2.

[29] Rubinstein, Issac and Rubinstein, Lev, 1998, Partial Differential Equations in Classical Mathematical Physics 2nd Edition, Chapter XIII, p. 296.

[30] Hobbie, Russell K., 1997, Intermediate physics for medicine and biology, 3rd Edition, Chapter IV, p. 85.

[31] Hobbie, Russell K., 1997, Intermediate physics for medicine and biology, 3rd Edition, Chapter IV, p. 78.

[32] Crank, John, 1979, The Mathematics of Diffusion, 2nd Edition, Chapter I, p. 3.

[33] Bohle et al. and Oertel, Herbert, 2004, Prandtl's Essentials of Fluid Mechanics, 2nd Edition, Chapter V, p. 217.

[34] White, Frank M., 2004, Fluid Mechanics, 5th Edition, Chapter I, p. 14.

[35] Batchelor, G.M., 2000, An Introduction to Fluid Mechanics, 3rd Edition, Chapter 2.1, p.71.

[36] Fletcher, C.A.J, 1991, Computational Techniques for Fluid Dynamics, 2nd Edition, 1, Chapter 1.1, p.3.

[37] Thom, A., 1933, The flow past circular cylinders at low speeds, Proceedings of the Royal Society of London, 141A, 651-666.

[38] Smith, G.D., 1985, Numerical Solution of Partial Differential Equations, 3rd Edition, Chapter II, p. 11-43.

[39] Thomee, V, 1997, Galerkin Finite Element Method for Parabolic Problems, Springer Series in Computational Mathematics, Chapter I, p. 1-20.

[40] Donea, Jean and Huerta, Antonio, 2003, Finite Element Methods for Flow Problems, 2nd Edition, Chapter I, p. 4.

[41] Knabner, P and Angermann, L, 2003, Numerical Methods for Elliptical and Parabolic Partial Differential Equations, Springer Texts in Applied Mathematics, Chapter VI, pp. 255-262.

[42] Barth, Timothy and Ohlberger, Mario, 2004, Finite volume methods: foundation and analysis, Encyclopedia of Computational Mechanics, pp. 1-57.

[43] Gropp, W.D., Kaushik, D.K., Keyes, D.E. et al., 2000, Performance Modeling and Tuning of an Unstructured Mesh CFD Application, Proceedings of the IEEE, pp.1-15.

[44] Tecplot Inc, 2006, v10, Redmond, Washington, USA.

[45] Björck, A, 1996, Numerical Methods for Least Squares Problems, Chapter 2.2, pp. 42-51.

[46] Fluent Inc., Ansys Ltd., 2007, UDF Manual.

[47] Hokr, Milan and Maryska, Jiri, 2005, *Influence of Mesh Geometry to Numerical Diffusion in Upwind Scheme for Porous Media Solute Transport*, Proceedings of Algoritmy, pp.123-131.

[48] Ansys Inc., GAMBIT Modeling Guide, 2006, GAMBIT v 2.3.6, *MG03*, pp.84-86.
[49] Ansys Inc., GAMBIT Modeling Guide, 2006, GAMBIT v 2.3.6, *MG03*, pp.189-190.

[50] Turek, Stefan, 1996, *A Comparative Study of Time-Stepping Techniques for the Incompressible Navier-Stokes Equations: from Fully Implicit Non-Linear Schemes to Semi-Implicit Projection Methods*, International Journal for Numerical Methods in Fluids, 22, pp. 987-1011

[51] Ferziger, Joel H. and Peric, Milovan, 1996, *Further Discussion of Numerical Errors in CFD*, International Journal for Numerical Methods in Fluids, 23, pp. 1263-1274.

[52] Hager, J.O. and Lee, K.D., 1996, *Effects of Implicit Preconditioners on Solution Acceleration Schemes in CFD*, International Journal for Numerical Methods in Fluids, 22, pp. 1023-1035.

[53] Xue, Ming, 2000, *High-Order Monotonic Numerical Diffusion and Smoothing*, American Meteorological Society, 128, pp. 2853-2863.