2018-01-01

Mathematical Modeling Of Tumor Growth For Free-Boundary Problem By Enhanced Finite Volume Method

Mashriq Ahmed Saleh
University of Texas at El Paso, priyomq@gmail.com

MATHEMATICAL MODELING OF TUMOR GROWTH FOR

FREE-BOUNDARY PROBLEM BY ENHANCED

FINITE VOLUME METHOD

MASHRIQ AHMED SALEH

Master's Program in Computational Science

APPROVED:

Xianyi Zeng, Chair, Ph.D.

Vladik Kreinovich, Ph.D.

Rodrigo Romero, Ph.D.

Charles Ambler, Ph.D.
Dean of the Graduate School

MATHEMATICAL MODELING OF TUMOR GROWTH FOR FREE-BOUNDARY PROBLEM BY ENHANCED FINITE VOLUME METHOD

by

MASHRIQ AHMED SALEH

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Program in Computational Science

THE UNIVERSITY OF TEXAS AT EL PASO

August 2018

Acknowledgements

I would like to show my deepest gratitude to all my family members for their encouragement and support in my higher studies. They have always been there for me, when I needed them. My biggest appreciation goes to my mentor and committee chair Dr. Xianyi Zeng who gave me the opportunity to work with him. I am very grateful and lucky to have an awesome professor like him whose guidance, vast knowledge and experience motivated me in moving forward to achieve my goals. His excellent mentor-ship and encouragement helped me to complete research projects and grow as a future researcher. It was a great pleasure to work under his mentorship. I am also thankful to Dr. Vladik Kreinovich and Dr. Rodrigo Romero for agreeing to serve on my thesis committee and Dr. Jianjun Paul Tian, Dr. Ben Niu for their support to complete this work. I am also very grateful to all the members of Computation Science Program, specially Dr. Ming-Ying Leung and Cindy Davis for their support and guidance. I also gratefully acknowledge the co-operation and help of all my friends and faculty members.

NOTE: This thesis was submitted to my Supervising Committee on the July 20, 2018.

Abstract

Modeling tumor growth due to infiltration of immune cells presents several challenges in numerical computations. First, it involves multiple cell species whose total number should be a constant, due to the incompressibility assumption; second, by mapping the Eulerian coordinate of the free-boundary problem onto a fixed logical domain, geometric source terms appear and they need to be addressed properly in numerical methods. In this work, we use a simplified model that contains two species and prescribed infiltration velocity and to show that the conventional finite volume methods fail to preserve the trivial (constant) solutions. To this end, we introduce the totality conservation law (TCL) and the geometric law (GCL) as the two criterions to address the incompressibility and property of preserving constant solutions on changing Eulerian domains, respectively. The classical Godunov-type finite volume methods are enhanced to satisfy these conditions, and performance improvements are verified by numerical tests with arbitrary infiltration velocities.

Table of Contents

List of Figures

Chapter 1

Introduction

Tumor-infiltrating immune cells consist of a significant component of many cancers [26, 27]. A plethora of studies have been conducted to explore the impacts of tumor-infiltrated immune cells on associated tumors [28, 29]. Numerous studies have showed that direct contact between infiltrating immune cells and tumor cells correlates with destruction of cancer cells, reduction of tumor sizes, and improved clinical prognosiscite [36, 37]. On the other hand, a large number of studies showed that increased infiltration of immune cells may promote tumor progression and invasion [30, 31, 33] and many clinical data have also indicated the tumor infiltration of certain immune cells associates with poor prognosis of patients with cancers [38, 39, 13]. The primary reasons for these contradictory observations have remained elusive; in this work, we develop a numerical tool to study how the infiltration of immune cells into the tumors is regulated, in the hope that it will revealing the mystery in the future.

Modeling the tumor growth due to immune cell infiltration using the partial differential equations (PDE) has been an active research area in recent years. One of the earliest papers addressing this phenomenon from a mathematical point of view is by Evelyn F. Keller and Lee A. Segel [19], who modeled the cell movements by Brownian motion and concluded that they generally move towards a region with high chemotaxis concentration. The Patlak-Keller-Seger (PKS) chemotaxis system, which describes the interaction between the cell and the chemotaxis, is then studied both theoretically and numerically by various author [12, 17, 15, 16]. Existing literature focuses on solving the PKS system on a fixed domain; hence they are suitable for describing the cell movements inside the tumor but not for modeling how the tumor grows. Recently, Ben Niu et al. [9] proposed a free boundary

problem that models the growth of tumor due to immune cell infiltration. In this model, the immune cells are attracted by the chemotaxis that usually has higher concentration inside the tumor and enter the tumor boundary; the mean cell movement velocity is derived by assuming the cells are incompressible, hence the total cell number per unit volume is assumed to be constant. Thus the tumor grows (or shrinks) as immune cells enter (exit) through the tumor boundary.

In this study, the spherically symmetric free-boundary problem is considered and solved numerically by first mapping the physical coordinates onto a fixed logical one and then discretization using the classical finite volume method (FVM). FVM is robust and it captures the shock of the solution but the conventional FVM fails to contain the overshoots of our solution. Also, geometrical source term appears when coordinate is changed and conventional finite volume methods cannot balance them out, even while having constant solutions.

To resolve these issues, we use a simplified model that represents the characterization of the tumor growth model described in [9] for our investigation. The totality conservation law (TCL) and the geometric conservation law (GCL) are defined and justified as the criteria for a finite volume method method to maintain constant solutions and satisfy the incompressibility condition required. The theoretical development of our finite volume method that satisfy both TCL and GCL contains three stages. First, we extend the TCL and GCL properties to a general finite volume discretization, namely DTCL and DGCL, respectively where D stands for discrete. Second, we introduce several consistency properties, so that for any numerical flux that satisfies these properties, the resulting method will satisfy both DTCL and DGCL. Finally, the classical and first-order upwind method and the second order MUSCL scheme are extended to satisfy these proposed consistency properties.

The remainder of the paper is organized as follows. In Chapter 2, we briefly describe the previous models such as the Platelet-Derived Growth factor (PDGF)- driven Glioma model and the Osteopontin (OPN) model in Gliomas. Later, a simplified model that captures the most important features is introduced Chapter 3. Chapter 4 covers the proposed method where we define the DTCL and DGCL conditions and prove necessary conditions for the

numerical method to satisfy these criteria. In Chapter 5, we solve the simplified model and assess the performance of the enhanced methods to the conventional FVM. Finally, we conclude with Chapter 6 explaining the significance of the work.

Chapter 2

Mathematical Models and Dynamics

The understanding of the regulation of immune cell infiltration into tumor cells is important to characterize the role of tumor-infiltrated cells. In this chapter, we briefly review the models, for more indepth discussion, readers are refered to [9]. Using different values of the maximum of chemoattractant (a substance that attracts motile cells of a particular type) production rate, this model differentiates between wtIDH1 (wild-type IDH1) and muIDH1 (mutant IDH1). How the wtIDH1 tumors reach death volume earlier than the muIDH1 can be revealed by this model. It also shows that with the increase of tumor volume in both types of gliomas (a type of tumor that occurs in the brain and spinal cord) in time, the net increase rate of immune cells infiltrated into the tumor increases while the percentage of these cells decreases. Analyzing this model, we observe that the proportion of immune cells infiltrated into the tumor is decreasing as the tumor volume increases. The more immune cells infiltrated in to the tumor, the earlier the tumor reaches death volume [9].

2.1 The PDGF-driven Model

Considering a radially symmetric tumor where the distance from the center point of the tumor is denoted by r and $R(t)$ is the tumor boundary, let $G(r,t)$, $H(r,t)$, $N(r,t)$ and $A(r,t)$ are the number density of glimoa cells, the number density of necrotic cells, the number density of infiltrated immune cells, and the concentration of chemoattractants, respectively. The movement inside the tumor cell is caused by the proliferation and removal of cells and the velocity is denoted by $V(r,t)$. The chemoattractants undergo diffusion. The immune cells move along the gradient field which is generated by the movement of chemoattractants

and then they undergo the similar convection process within the tumor.

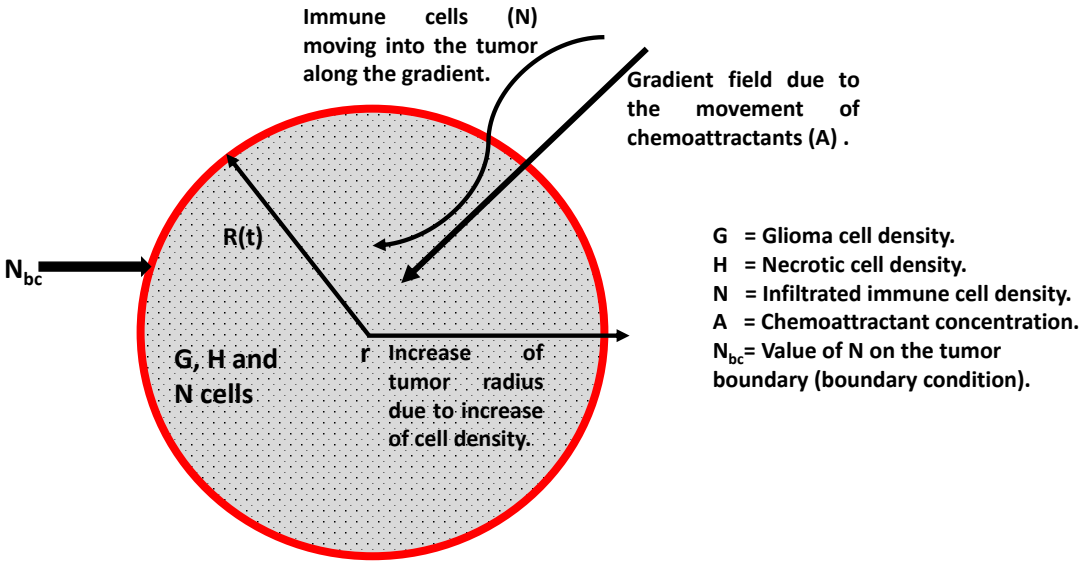


Figure 2.1: Diagram of tumor growth due to inflitrated immune cells

Figure 2.2 explains the tumor growth and immune cells movement into the tumor. The following equations can be derived by mass conservation law:

$$\frac{\partial G(r,t)}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}[R^2 G(r,t)V(r,t)] = \lambda G(r,t) - \mu H(r,t), \qquad r \in [0, R(t)) \tag{2.1}$$

$$\frac{\partial H(r,t)}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}[R^2 H(r,t)V(r,t)] = \mu G(r,t) - \delta H(r,t), \qquad r \in [0, R(t)) \tag{2.2}$$

$$\frac{\partial A(r,t)}{\partial t} = D\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2\frac{\partial A(r,t)}{\partial r}\right] + \frac{mG(r,t)}{\beta + G(r,t)} - \gamma A(r,t), \qquad r \in [0, \infty) \tag{2.3}$$

$$\begin{aligned}&\frac{\partial N(r,t)}{\partial t}+\frac{1}{r^2}\frac{\partial}{\partial r}[r^2N(r,t)V(r,t)]\\&=-\alpha\frac{1}{r^2}\frac{1}{\partial r}\left[r^2N(r,t)\frac{\partial A(r,t)}{\partial r}\right]-\rho N(r,t),\qquad r\in[0,R(t))\end{aligned}\tag{2.4}$$

It is assumed here that all the cells have the same size. As the number density of the tumor tissue is constant, it can be written as $G(r,t)+H(r,t)+N(r,t)=\theta$ inside the tumor region. After combining the aforementioned equations, the final equation which is for the velocity is:

$$\frac{\theta}{r^2}\frac{\partial}{\partial r}[r^2V(r,t)]=\lambda G(r,t)-\delta H(r,t)-\alpha\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2N(r,t)\frac{\partial A(r,t)}{\partial r}\right]-\rho N(r,t)\tag{2.5}$$

Also, the free boundary condition is given by $\frac{dR(t)}{dt}=V(R(t),t)$. The initial conditions are specified as $R(0)=\epsilon$, where ϵ is in general a small number. The boundary conditions for the chemoattractant $A(r,t)$ are specified as $\frac{\partial A}{\partial r}(0,t)=0$ and $A(r,t)$ vanishes at the infinite and $V(0,t)=0$ for $t\geq 0$.

The parameter details are summarized in the table below:

Table 2.1: Model parameters and their unit

Parameter List		
Parameters	Descriptions	Dimensions
λ	Proliferation rate of glioma cells	day^{-1}
μ	Glioma cell lysis rate	day^{-1}
δ	Removal rate of necrotic cells	day^{-1}
D	Diffusion coefficient of chemoattractant	mm^2/day
m	Maximum of chemoattractant production rate	$10^5 pg/ml \cdot day$
β	Michalis constant	$cells/mm^3$
γ	Chemoattractant degradation rate	$10^2/day$
α	Chemotatic coefficient	$mm^2 \cdot ml/day \cdot pg$
ρ	Clearance rate of immune cells	day^{-1}
θ	Cell density of tumor issue	$cells/mm^3$

2.1.1 The Growth Dynamics

The tumor growth dynamics can be described by this mathematical model. The glioma cell population exponentially increases as soon as the tumor starts to grow. The chemoattractants secreted from the tumor cell also increases nonlinearly. They start diffusing within the host body and start forming a dynamic chemotatic gradient field. Along this gradient

field, the immune cells start to migrate into the tumor.

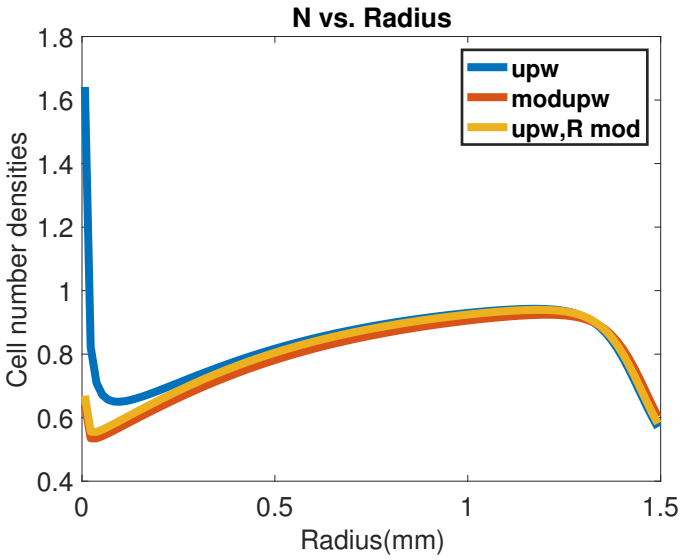


Figure 2.2: Solution of immune cells against the tumor radius

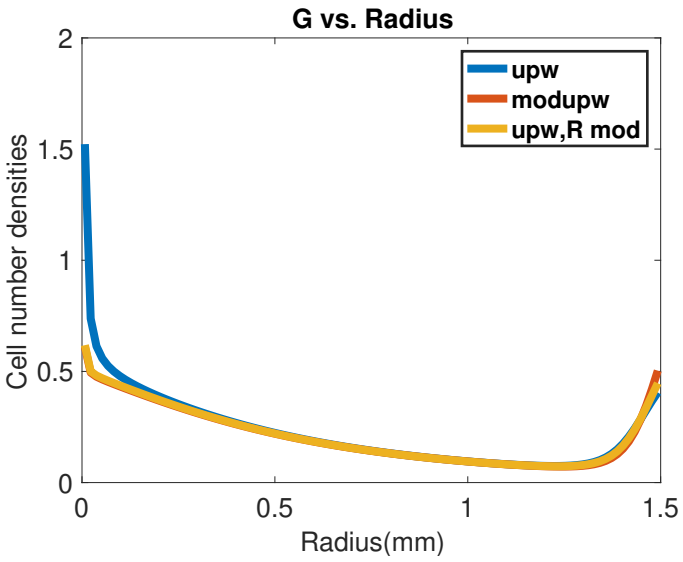


Figure 2.3: Solution of glioma cells against the tumor radius

From Figure (2.2), we see that immune cell density inside the tumor approximately follows a linear pattern. As the tumor volume increases throughout the time, the model

shows distinct growth patterns for immune cells and glioma cells (Figure (2.3)). So, it can be said that this model mathematically describes the dynamics of the immune cell infiltration into the tumor body. The different plots in every figure is obtained by implementing FVM to calculate the flux using different upwind methods.

2.2 The Osteopontin Dynamics Model in Gliomas

We can extend the previous model to Osteopontin (OPN) model of tumor cell growth. OPN, a secreted non-collagenous, sialic-acid-rich, chemokine-like protein that is also a member of the small integrin-binding ligand N-linked glycoproteins family, plays a crucial role in determining the oncogenic potential of various cancers [3, 4]. The pivotal role of OPN in tumor metastasis has been highlighted by gene-transfer experiments [5]. Overexpression of OPN results in an increase in the malignant phenotype, whereas transfection with antisense oligonucleotides yields a population with a reduced malignant potential [6]. Since its first identification in bone, the multifaceted roles of OPN have been an area of intense investigation. Extensive research has elucidated the pivotal role of OPN in regulating the cell signaling that controls tumor progression and metastasis [5].

2.2.1 Model Dynamics

OPN dynamics in glioma can be described as follows. Glioma cells secrete chemoattractanrs to attract immune cells, microglia and monocytes. Microglia, a type of neuronal support cell (neuroglia) occurring in the central nervous system of invertebrates and vertebrates, functions primarily as an immune cell. As the name microglia suggests, these cells are small — the smallest of all the neuroglia [7]. Another type of immune cell that is made in the bone marrow and travels through the blood to tissues in the body where it becomes a macrophage. Macrophages surround and kill microorganisms, ingest foreign material, remove dead cells, and boost immune responses [8]. These glioma cells also secrete osetopontin for the enhancement of their proliferation. Similar as the previous PDGF model,

immune cells move into tumor site along the gradient field of chemoattracttants and they also secrete osteopontin to inhibit tumor growth.

2.2.2 The Mathematical Model

Let r be a generic point in the tumor location at any time t, $G(r,t)$, $H(r,t)$, $A(r,t)$ and $N(r,t)$ are exactly defined as in the PDGF model. In addition, $P_1(r,t)$ and $P_2(r,t)$ denote the concentration of osteopontin secreted by the immune cells and concentration of osteopontin secreted by glioma cells. Also, $V(r,t)$ is the velocity and $\Gamma(r,t)$ is the tumor boundary. Both the glioma immune cells and also the necrotic cells convect with the fluid within the tumor. The attractants and osteopontin diffuse instead of advection since they are small in size compared to cells. Considering one of the simplest case, where the tumor is viewed as a symmetric sphere and based on the previous PDGF-model, the following system is proposed for describing the dynamics of OPN:

$$\begin{aligned}\frac{\partial G(r,t)}{\partial t}+\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2G(r,t)V(r,t)\right] &= \lambda\left[1+\frac{P_2(r,t)}{b+P_2(r,t)}\right]G(r,t)\\ &\quad -\mu\left[1+\frac{P_1(r,t)}{a+P_1(r,t)}\right]G(r,t), \quad r\in[0,R(t))\end{aligned} \tag{2.6}$$

$$\begin{aligned}\frac{\partial H(r,t)}{\partial t}+\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2H(r,t)V(r,t)\right] &= \mu\left[1+\frac{P_1(r,t)}{a+P_1(r,t)}\right]G(r,t)\\ &\quad -\delta_1 H(r,t), \quad r\in[0,R(t))\end{aligned} \tag{2.7}$$

$$\frac{\partial A(r,t)}{\partial t}=D_3\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2\frac{\partial A(r,t)}{\partial r}\right]+\frac{\alpha_3 G(r,t)}{\beta_3+G(r,t)}-\gamma_3 A(r,t), \quad r\in[0,\infty) \tag{2.8}$$

$$\begin{aligned}\frac{\partial N(r,t)}{\partial t}+\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2N(r,t)V(r,t)\right] &= -\rho\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2N(r,t)\frac{\partial A(r,t)}{\partial r}\right]\\ &\quad -\delta_2 N(x,t), \quad r\in[0,R(t))\end{aligned} \tag{2.9}$$

$$\frac{\partial P_1(r,t)}{\partial t}=D_1\frac{\partial}{\partial r}\left[r^2\frac{\partial P_1(r,t)}{\partial r}\right]+\frac{\alpha_1 N(r,t)}{\beta_1+N(r,t)}-\gamma_1 P_1(r,t), \quad r\in[0,\infty) \tag{2.10}$$

$$\frac{\partial P_2(r,t)}{\partial t}=D_2\frac{\partial}{\partial r}\left[r^2\frac{\partial P_2(r,t)}{\partial r}\right]+\frac{\alpha_2 G(r,t)}{\beta_2+G(r,t)}-\gamma_2 P_2(r,t), \quad r\in[0,\infty) \tag{2.11}$$

$$\begin{aligned}\frac{\theta}{r^2}\frac{\partial}{\partial r}\left[r^2V(r,t)\right] =&\lambda\left[1+\frac{P_2(r,t)}{b+P_2(x,t)}\right]G(r,t)-\delta_1 H(r,t)\\ &-\alpha\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2N(r,t)\frac{\partial A(r,t)}{\partial r}\right]-\delta_2 N(r,t)\end{aligned} \tag{2.12}$$

The initial conditions are given as $G(r,0), H(r,0), N(r,0), P_1(x,0), P_2(x,0)$ within the initial tumor and $A(x,0)$ is given in the whole domain. The boundary conditions are given as $\frac{\partial P_1}{\partial n}|_\Gamma = 0, \frac{\partial P_2}{\partial n}|_\Gamma = 0$. Also the free boundary is given as

$$\frac{dR(t)}{dt} = V(R(t),t), \quad R(0)=\epsilon;$$

Normalizing the G, H, N by $G/\theta, H/\theta, N/\theta$ and considering $G+H+N=1$, we get

$$\begin{aligned}\frac{\partial G(r,t)}{\partial t}+\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2G(r,t)V(r,t)\right] =&\lambda\left[1+\frac{P_2(r,t)G(r,t)}{b+P_2(r,t)}\right]\\ &-\mu\left[1+\frac{P_1(r,t)G(r,t)}{a+P_1(r,t)}\right]\end{aligned} \tag{2.13}$$

$$\frac{\partial H(r,t)}{\partial t}+\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2H(r,t)V(r,t)\right] = \mu\left[1+\frac{P_1(r,t)G(r,t)}{a+P_1(r,t)}\right]-\delta_1 H(r,t) \tag{2.14}$$

$$\frac{\partial A(r,t)}{\partial t} = D_3\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2\frac{\partial A(r,t)}{\partial r}\right]+\frac{\alpha_3 G(r,t)}{\beta_3+G(r,t)}-\gamma_3 A(r,t) \tag{2.15}$$

$$\frac{\partial N(r,t)}{\partial t}+\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2N(r,t)V(r,t)\right] = -\rho\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2N(r,t)\frac{\partial A(r,t)}{\partial r}\right]-\delta_2 N(r,t) \tag{2.16}$$

$$\frac{\partial P_1(r,t)}{\partial t} = D_1\frac{\partial}{\partial r}\left[r^2\frac{\partial P_1(r,t)}{\partial r}\right]+\frac{\alpha_1 N(r,t)}{\beta_1+N(r,t)}-\gamma_1 P_1(r,t) \tag{2.17}$$

$$\frac{\partial P_2(r,t)}{\partial t} = D_2\frac{\partial}{\partial r}\left[r^2\frac{\partial P_2(r,t)}{\partial r}\right]+\frac{\alpha_2 G(r,t)}{\beta_2+G(r,t)}-\gamma_2 P_2(r,t) \tag{2.18}$$

$$\begin{aligned}\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2V(r,t)\right] =&\lambda\left[1+\frac{P_2(r,t)}{b+P_2(r,t)}\right]G(r,t)-\delta_1 H(r,t)\\ &-\alpha\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2N(r,t)\frac{\partial A(r,t)}{\partial r}\right]-\delta_2 N(r,t)\end{aligned} \tag{2.19}$$

Other boundary conditions are still the Neumann type at the origin and A is defined in $[0,+\infty)$. Initial conditions are same as those in the previous model. Moreover, the new functions defined at zero time with $P_1(r,0) = P_2(r,0) = 0$.

To transform the governing equations (2.13 - 2.19) to the conservation law, we multiply

them with r^2. Next to convert them to a fixed domain instead of a free-boundary problem, we use transformation of coordinates and later the chain rule:

$$(r,t) \longmapsto (\eta,\tau) = \Big(\frac{r}{R(t)}, t\Big)$$

Chain rule:

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \tau} - \frac{\eta R'}{R}\frac{\partial}{\partial \eta}, \quad \frac{\partial}{\partial r} = \frac{1}{R}\frac{\partial}{\partial \eta} \tag{2.20}$$

Also, for any species X, considering $\eta^2 R^2 X = \widetilde{X}$, we get the final form of the governing equations in normalized coordinates:

$$\frac{\partial \widetilde{G}}{\partial \tau} + \frac{\partial}{\partial \eta}\left[\left(\frac{V}{R} - \frac{\eta R\prime}{R}\right)\widetilde{G}\right] = \frac{\lambda \widetilde{G} P_2}{b+P_2} - \frac{\mu \widetilde{G} P_1}{a+P_1} - \frac{R'\widetilde{G}}{R} + (\lambda-\mu)\widetilde{G} \tag{2.21}$$

$$\frac{\partial \widetilde{H}}{\partial \tau} + \frac{\partial}{\partial \eta}\left[\left(\frac{V}{R} - \frac{\eta R\prime}{R}\right)\widetilde{H}\right] = \frac{\lambda \widetilde{G} P_1}{a+P_1} - \delta_1 \widetilde{H} - \frac{R'\widetilde{H}}{R} + \mu \widetilde{G} \tag{2.22}$$

$$\frac{\partial \widetilde{A}}{\partial \tau} + \frac{\partial}{\partial \eta}\left[\Big(-\frac{\eta R'}{R}\Big)\widetilde{A}\right] = D_3 \frac{\partial}{\partial \eta}\left(\eta^2 \frac{\partial A}{\partial \eta}\right) + \frac{\alpha_3 \widetilde{G}}{\beta_3 + G} - \gamma_3 \widetilde{A} - \frac{R'\widetilde{A}}{R} \tag{2.23}$$

$$\frac{\partial \widetilde{N}}{\partial \tau} + \frac{\partial}{\partial \eta}\left[\left(\frac{V}{R} - \frac{\eta R\prime}{R} + \frac{\rho}{R^2}\frac{\partial A}{\partial \eta}\right)\widetilde{N}\right] = -\delta_2 \widetilde{N} \tag{2.24}$$

$$\frac{\partial \widetilde{P_1}}{\partial \tau} + \frac{\partial}{\partial \eta}\left[\Big(-\frac{\eta R'}{R}\Big)\widetilde{P_1}\right] = D_1 \frac{\partial}{\partial \eta}\left(\eta^2 \frac{\partial P_1}{\partial \eta}\right) + \frac{\alpha_1 \widetilde{N}}{\beta_1 + N} - \gamma_1 \widetilde{P_1} - \frac{R'\widetilde{P_1}}{R} \tag{2.25}$$

$$\frac{\partial \widetilde{P_2}}{\partial \tau} + \frac{\partial}{\partial \eta}\left[\Big(-\frac{\eta R'}{R}\Big)\widetilde{P_2}\right] = D_2 \frac{\partial}{\partial \eta}\left(\eta^2 \frac{\partial P_2}{\partial \eta}\right) + \frac{\alpha_2 \widetilde{G}}{\beta_2 + G} - \gamma_2 \widetilde{P_2} - \frac{R'\widetilde{P_2}}{R} \tag{2.26}$$

$$\frac{1}{\eta^2 R}\frac{\partial}{\partial \eta}\left[\eta^2\left(V + \frac{\alpha}{R} N \frac{\partial A}{\partial \eta}\right)\right] = \frac{\lambda P_2 \widetilde{G}}{b+P_2} - \delta_1 \widetilde{H} - \delta_2 \widetilde{N} + \lambda \widetilde{G} \tag{2.27}$$

2.3 Numerical Solution

We have used finite volume method (FVM) for solving the model instead of a typical finite difference scheme. The reason behind using finite volume method is its conservation properties. It is derived on the basis of the integral form of conservation law, a starting point that turns out to have many advantages and is closely related to a finite difference method.

2.4 Classical Finite Volume Method

In practice many interesting solutions contain discontinuities such as shock waves instead of being smooth. A fundamental feature of nonlinear conservation law is that these discontinuities can easily develop even though smooth initial data is used, and so they must be dealt with both mathematically and computationally and is troublesome for accurate approximation of such solutions. For this method, the primary problem is to determine a good numerical flux function which can correctly approximate the flux values of the cell average or over the whole cell.

Instead of approximation at grid points, FVM breaks the computation domain into grid cells and then approximate the total integral of flux over each grid cell or actually cell average of the flux, which can be measured by dividing the total integral over the flux by the volume of each cell. At each time step, these values are modified by calculating the flux through the edges of the grid cells.

For simple notations, we define:

$$\widetilde{G} = \eta^2 R^2 G, \quad \widetilde{H} = \eta^2 R^2 H, \quad \widetilde{N} = \eta^2 R^2 N.$$

For the velocity variable, however, we adopt nodal approximation:

$$V_i \approx V(\eta_i),$$

where $\eta_i = ih, i = 0, ..., L_\eta$.

We denote the solutions by $\mathcal{S}$, at any given time τ, $\mathcal{S}$ should contain the below information:

- The radius R.
- The cell numbers $\widetilde{G}_i$,$\widetilde{H}_i$ and $\widetilde{N}_i$ for $1 \leq i \leq L_\eta$.
- The velocity V_i, where $0 \leq i \leq L_\eta$.
- The chemotaxis attractant concentration A_i for $1 \leq i \leq mL_\eta$.

- The osteopontin concentration P_1 and P_2 for $1 \leq i \leq L_\eta$ (for the OPN model).

Here $m > 1$ is a positive number that denotes the truncated domain for computing A. Particularly, because A is defined on $\eta \in \overline{\mathbb{R}}^+$, it is not easy to solve A on the entire half line. Nevertheless, because the source term involving G vanishes for $\eta \geq 1$, we expect A to decay exponentially as $\eta \mapsto \infty$. Hence we approximate this unbounded differential equation by setting the boundary condition $A(\eta = m, \tau) = 0$ and solve for A on the truncated interval $[0, m]$, which is discretized using mn intervals of the same size $\triangle\eta$.

2.4.1 The General Workflow

Considering space and time discretizations separately and a first-order-in-time method to update the solutions from τ^k to $\tau^{k+1} = \tau^k + \triangle\tau^K$ is denoted by $\mathcal{S}^{k+1} = \mathcal{M}(\mathcal{S}^k, \triangle\tau^k)$. For constructing the operator $\mathcal{M}$, we follow the below steps in the following order:

1. Update R^{k+1}.
2. Update the cell numbers $\widetilde{G}_i^{k+1}$, $\widetilde{H}_i^{k+1}$ and $\widetilde{N}_i^{k+1}$, $1 \leq i \leq L_\eta$.
3. Update the chemotaxis attractant concentration A_i^{k+1}, $1 \leq i \leq mL_\eta$.
4. Update the osteopontin concentrations P_{1i}^{k+1} and P_{2i}^{k+1} for $1 \leq i \leq L_\eta$ (for the OPN model).
5. Update the velocity V_i^{k+1}, $0 \leq i \leq L_\eta$.
6. Calculate the time step $\triangle\tau^k$.

The first and last steps will remain the same for different methods.

- **Step 1: Updating tumor radius**

  The forward Euler method is used to update the radius of the tumor. There is a correlation among the numbers of data points at successive time intervals [11].

$$R^{k+1} = R^k + \triangle\tau^k V_{L_\eta}^k$$

- **Step 2: Computing the cell numbers**

  After deriving the finite volume formulation for G, H and N, we get :

$$\begin{aligned}\frac{d\widetilde{G}_i}{d\tau}+\frac{1}{\triangle\eta}[F_i^G-F_{i-1}^G]=\lambda'\left(1+\frac{P_{2i}}{b+P_{2i}}\right)\widetilde{G}_i(1-l'G_i)\\ -\mu\left(1+\frac{P_{1i}}{a+P_{1i}}\right)\widetilde{G}_i-\omega\widetilde{G}_iN_i-\frac{R'}{R}\widetilde{G}_i\end{aligned} \tag{2.28}$$

$$\frac{d\widetilde{H}_i}{d\tau}+\frac{1}{\triangle\eta}[F_i^H-F_{i-1}^H]=\mu\left(1+\frac{P_{1i}}{a+P_{1i}}\right)\widetilde{G}_i-\delta_1\widetilde{H}_i-\frac{R'}{R}\widetilde{H}_i \tag{2.29}$$

$$\frac{d\widetilde{N}_i}{d\tau}+\frac{1}{\triangle\eta}[F_i^H-F_{i-1}^H]=-\delta_2\widetilde{N}_i-\frac{R'}{R}\widetilde{N}_i \tag{2.30}$$

  Here $\lambda'=\lambda\eta^q$ and F_j^X where $X=G,H,N$ is a numerical flux approximating $\left(\frac{V}{R}-\frac{\eta R'}{R}\widetilde{X}\right)$ at η_j. In general, a p-th order flux function leads p-th order spatial discretization.

- **Step 3: Updating the Chemotaxis Attractant Concertation**

  The equation for chemotaxis attractant concentration A involves both advection and diffusion. In order to avoid a tiny time step size in explicit time-integration, we adopt a fully implicit method for both advection and diffusion here. Particularly, a first-order method can be constructed by using the first-order upwind flux for advection and central difference for diffusion. Integrating (2.24) over a cell C_i where $1\le i\le mL$ leads to the following semi-discrete formula for A:

$$\begin{aligned}&\frac{d(\eta_i^2R^2A_i)}{d\tau}+\frac{1}{\triangle\eta}[F_i^A-F_{i-1}^A]=\frac{D_3}{\triangle\eta}\left[\eta_i^2\frac{A_{i+1}-A_i}{\triangle\eta}-\eta_{i-1}^2\frac{A_i-A_{i-1}}{\triangle\eta}\right]\\ &+\chi_{i\le L}\frac{\alpha_3\widetilde{G}_i}{\frac{\beta_3}{\theta}+G_i}-\left(\gamma+\frac{V_L}{R}\right)\eta_i^2R^2A_i\end{aligned}$$

  Here $\chi_{i\le L_\eta}$ is an indicator function showing that the source term containing G should only be included for $1\le i\le L_\eta$, i.e., when the cell is in the tumor.

- **Step 4: Updating the Osteopontin Concentrations (for OPN model)**

  The osteopontin concentrations P_1 and P_2 are updated the same way as the chemotaxis attractant concentration A, by implicit first-order upwind/central-difference

method or IMEX [41] second-order MUSCL/central-difference method. The only difference is that the two equations are defined on $1 \leq i \leq L_\eta$, and we enforce the Neumann boundary condition at the right boundary- therefore the finite differencing stencil needs to be biased for both $i = 1$ and $i = L_\eta$.

- **Step 5: Computing velocity**

  Computing the velocity V_i^{k+1} makes use of the integral equation that can be derived from (2.27) with the boundary condition $V_0^{k+1} = V(\eta = 0, \tau^{k+1}) = 0$. The integral of the velocity is given by:

$$V = \frac{R}{\eta^2}\int_0^\eta \xi^2 \left[\lambda\left(1 + \frac{P_2}{b + P_2}\right)G(1 - l'G) - \delta_1 H - \delta_2 N - \omega G N\right] d\xi - \frac{\alpha}{R} N \frac{\partial A}{\partial \eta} \quad (2.31)$$

  where $l' \in [0, 1]$, λ and $q \in [0, 1]$ being constants introduces spatially inhomogeneous growth and a toggle for the logistic growth, respectively.

- **Step 6: Time step calculation**

  For computing the time step, $\triangle\tau^k$ is computed according to the Courant condition. Especially, because all the diffusion terms are treated implicitly, we set the Courant number according to the advection velocity:

$$\triangle\tau^k = \alpha_{cfl} \min_{1 \leq i \leq L} \frac{h}{max\left(\left|\frac{V_i^k - \eta_i V_L^k}{R^k}\right|, \left|\frac{V_i^k - \eta_i V_L^k}{R^k} + \frac{\rho}{(R^k)^2}\frac{A_{i+1}^k - A_i^k}{\triangle\eta}\right|\right)}$$

  where α_{cfl} is the Courant number specified by the user, which is typically smaller than 1 for partially explicit computations and can take any value when everything is implicit.

2.4.2 Result Analysis

A classical nonlinearly stable method such as explicit first-order upwind method is used here for updating the cell number densities G, H and N which are Glioma cells, Nectoric cells and Immune cells, respectively. The cell number densities are plotted against the growth of the tumor radius in the Figure (2.4).

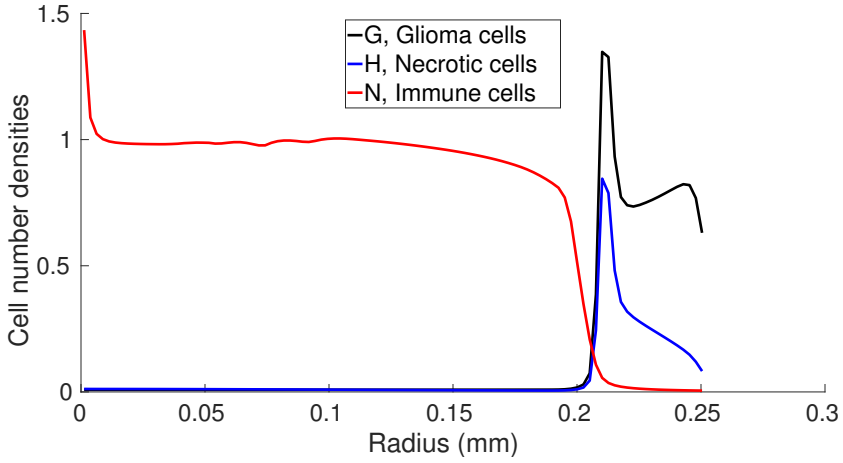


Figure 2.4: Solution of G, H and N against the tumor radius

From the above figure, we can see that around the origin and also at the boundary, overshoots are observed. At origin, immune cell density, N is 1.5 and also near the right boundary, their sum is again more than one. So, the non-compressibility assumption does not hold for the solution of the system. This might occur due to the discretization method that we have used in conventional finite volume method. In the next chapter we will consider a simplified model to work with and then introduce an enhanced FVM to solve the governing equations.

Chapter 3

A Simplified Model

In this chapter we address the numerical challenges we face while solving the free boundary problems described by spherically symmetric conservation laws. Here, we normalize the radial coordinate to transform the free boundary problem to a fixed boundary one and utilize a finite volume method to solve the equations. We show that classical finite volume method fails to restore the constant solutions and propose modifications to maintain the totality conservation law, the geometric conservation law and also balance the source terms.

3.1 A Model Problem

In this section, for simplicity, we consider a simplified model with only two species G and N. Then the governing equations in spherical coordinate are as below:

$$\frac{\partial(r^2 G)}{\partial t} + \frac{\partial}{\partial r}[r^2 G V] = r^2 f \tag{3.1}$$

$$\frac{\partial(r^2 N)}{\partial t} + \frac{\partial}{\partial r}[r^2 N(V+u)] = r^2 g \tag{3.2}$$

$$\frac{1}{r^2}\frac{\partial}{\partial r}[r^2(V+uN)] = f+g \tag{3.3}$$

For this model, the computational domain is:

$$(r,t) \in [0, R(t)) \times [0,T]$$

For some positive $T > 0$ and $R : [0,T] \mapsto \mathbb{R}^+$ denotes the radius of the spherical domain which satisfies:

$$\dot{R} = V(R(t),t). \tag{3.4}$$

The lower case letters in (3.1 - 3.3) are prescribed functions:

$$u(r,t) = \breve{u}(r,t,G,N), \qquad f(r,t) = \breve{f}(r,t,G,N), \qquad g(r,t) = \breve{g}(r,t,G,N)$$

where they represent the velocity difference between N and G, the source term of G, and the source term of N, respectively. We further require that $\breve{u} = 0$ at $r = 0$ for all $t \in [0,T]$. Now, because G and N represent the proportion of the two species whose sum is incompressible , it satisfies the initial condition:

$$G(r,0) + N(r,0) = 1, \quad \forall\ 0 \le r \le R(0). \tag{3.5}$$

We can define that the total number $\Theta(r,t) = G(r,t)+H(r,t)$; then the incompressibility assumption requires $\Theta \equiv 1$. If this holds, we actually have a very convenient way to estimate the growth of tumor.

Now, the only mechanism such that the new cells can enter the tumor is through the boundary condition for N at $r = R(t)$. Now, we can write:

$$\breve{N}(t) = \begin{cases} N(R(t),t), & u(R(t),t) \ge 0; \\ N_{bc}(t), & u(R(t),t) < 0, \end{cases}$$

where u is the prescribed infiltration velocity, $\breve{N}(t)$ is the flow out of/into the tumor and N_{bc} (boundary condition) is the prescribed ambient number of immune cells, N. We also obtain the ODE for $R(t)$:

$$R'(t) = u(R(t),t,)\breve{N}(t),$$

which will help to design numerical tests for which the exact tumor growth curve can be calculated.

3.2 The Totality Conservation Law in Radial Coordinate

Considering $G(r,t)+N(r,t)=\Theta(r,t)$ and adding (3.1) and (3.2) and substituting it to the right hand side of (3.3), we get:

$$\frac{\partial(r^2\Theta)}{\partial t}+\frac{\partial}{\partial r}[r^2\Theta V+r^2Nu]=r^2(f+g)=\frac{\partial}{\partial r}[r^2(V+uN)],$$

or equivalently:

$$\frac{\partial(r^2\Theta)}{\partial t}+\frac{\partial}{\partial r}[r^2(\Theta-1)V]=0, \tag{3.6}$$

From which it can be accepted that $\Theta(r,t)\equiv 1$ regardless of the other variables if the initial condition (3.5) holds.

By replacing one of G and N by Θ in either (3.1) or (3.2) with (3.6), the PDE system is essentially the same problem with identical solutions. We expect that $G(r,t)+N(r,t)=\Theta(r,t)\equiv 1$ for the solutions if the original system of equations.

As equation (3.6) explains the conservation of the sum of the two species, we can call it *the totality conservation law* or *TCL* with the context of the current work and expect the numerical method to satisfy a discrete version of *TCL*.

3.3 The Conservation Laws in Normalized Coordinate

Due to (3.4), this problem is a free boundary problem. Because of this, fixed grid discretization in (r,t) domain becomes difficult. To resolve this, we define the normalized coordinates (η,τ) and define them as described in chapter 2.

Here we suppress the dependence of R' and R on t (or τ) for simplicity. Before going into the details of changing the coordinates of (3.1 - 3.3), we apply the chain rule (2.20) to (3.6) and obtain:

$$\frac{\partial(\eta^2R^2\Theta)}{\partial\tau}-\frac{\eta R'}{R}\frac{\partial}{\partial\eta}\Big[\eta^2R^2\Theta\Big]+\frac{1}{R}\frac{\partial}{\partial\eta}\Big[\eta^2R^2(\Theta-1)V\Big]=0$$

or equivalently:

$$\frac{\partial(\eta^2 R^2\Theta)}{\partial\tau} + \frac{\partial}{\partial\tau}\Big[\eta^2 R(\Theta-1)V\Big] - \frac{\partial}{\partial\eta}\Big[\eta^3 R' R\Theta\Big] = -\eta^2 R' R\Theta \tag{3.7}$$

which is the TCL in the normalized coordinates. Unlike (3.6), setting $\Theta \equiv 1$ in (3.7) does not lead to a trivial identity; instead, we get:

$$\frac{\partial(\eta^2 R^2)}{\partial\tau} + \frac{\partial}{\partial\eta}\Big[-\eta^3 R' R\Big] = -\eta^2 R' R. \tag{3.8}$$

This equation is called the *geometric conservation law* or *GCL*, which clearly holds at the continuous level; but we also require the numerical method to satisfy a discrete version of *GCL*, or the *discrete geometric conservation law (DGCL)*, which poses further requirements on our spatial and temporal discretizations.

3.4 The Model Problem in Normalized Coordinates

Finally, we apply the chain rule to (3.1 - 3.3) to derive the governing equations in the normalized coordinates (η, τ):

$$\frac{\partial(\eta^2 R^2 G)}{\partial\tau} + \frac{\partial}{\partial\eta}\left[\left(\frac{V}{R} - \frac{\eta R'}{R}\right)\eta^2 R^2 G\right] = \eta^2 R^2 f - \eta^2 R' R G, \tag{3.9a}$$

$$\frac{\partial(\eta^2 R^2 N)}{\partial\tau} + \frac{\partial}{\partial\eta}\left[\left(\frac{V}{R} - \frac{\eta R'}{R} + \frac{u}{R}\right)\eta^2 R^2 N\right] = \eta^2 R^2 g - \eta^2 R' R N, \tag{3.9b}$$

$$\frac{1}{\eta^2}\frac{\partial}{\partial\eta}\left[\eta^2\left(\frac{V}{R} + \frac{u}{R}N\right)\right] = f + g, V(0,t) = 0; \tag{3.9c}$$

the computational domain in $(\eta, \tau) \in [0,1] \times [0,T]$ for some positive $T > 0$ and $R : [0,T] \mapsto \mathbb{R}^+$ denotes the radius of the spherical domain, which satisfies:

$$\dot{R}(\tau) = V(1,\tau). \tag{3.10}$$

The lower case latter in (3.9) represent the prescribed functions:

$$u(\eta,\tau) = \breve{u}(\eta R, \tau, G, N), \qquad f(r,t) = \breve{f}(\eta R, \tau, G, N), \qquad g(r,t) = \breve{g}(\eta R, \tau, G, N)$$

where $\breve{u}$, $\breve{f}$ and $\breve{g}$ are the same known functions. Clearly, adding (3.9a) and (3.9b) and comparing the resulting right hand side to that of (3.9c) leads to TCL (3.8).

Chapter 4

An Enhanced Finite Volume Method

Neither TCL nor GCL is automatically satisfied by finite volume discretization to solve (4.10). To fix the issue, we subdivide the domain for η by L_η cells with uniform cell size $\triangle\eta = 1/L_\eta$ and suppose that the discrete velocities are defined at the grid points: $V_i^n \approx V(\eta_i, \tau^n)$, $\eta_i = i \triangle \eta, 0 \leq i \leq L_\eta$. To motivate the issue, let us focus on the GCL and first consider a numerical method such that (3.8) is satisfied at the discrete level. Let the discretization of (3.8) and (3.10) for the time interval $[\tau^n, \tau^{n+1}]$ be :

$$\frac{1}{\triangle\tau^n}\left[\eta_{i-1/2}^2 (R^{n+1})^2 - \eta_{i-1/2}^2 (R^n)^2\right] + (R')^n R^n \mathcal{D}_{i-1/2}\left[-\eta^3\right] = -\eta_{i-1/2}^2 (R')^n R^n, \quad (4.1a)$$

$$(R')^n = V_{N_\eta}^n, \quad (4.1b)$$

$$R^{n+1} = R^n + \triangle\tau^n V_{L_\eta}^n. \quad (4.1c)$$

where (3.11a) collocates at the cell center $\eta_{i-1/2}$ and $\mathcal{D}_{i-1/2}$ is the spatial discretization of choice for ∂_η at $\eta_{i-1/2}$. The method (4.12) features the natural choice of the forward Euler time-integrator and approximation of $R'(\tau^n)$ by $V_{L_\eta}^n$, c.f. (3.10). However, it follows that:

$$\begin{aligned}\mathcal{D}_{i-1/2}\left[\eta^3\right] &= -\eta_{i-1/2}^2 - \frac{\eta_{i-1/2}^2}{\triangle\tau^n}\frac{(R^n + \triangle\tau^n V_{L_\eta}^n)^2 - (R^n)^2}{V_{L_\eta}^n R^n} \\ &= -\eta_{i-1/2}^2 - \eta_{i-1/2}^2\left[2 + \frac{\triangle\tau^n V_{L_\eta}^n}{R^n}\right]\end{aligned}$$

which is highly undesirably since the right hand side depends on the solutions V and R. An easy way to fix this issue is to approximate $(R')^n$ of (3.8) by:

$$(R')^n \approx \frac{(R^{n+1})^2 - (R^n)^2}{2 \triangle \tau^n R^n} = V_{L_\eta}^n\left(1 + \frac{1}{2} \triangle \tau^n \frac{V_{L_\eta}^n}{R^n}\right) \quad (4.2)$$

then (4.12a) reduces to :

$$\mathcal{D}_{i-1/2}(-\eta^3) = -3\eta_{i-1/2}^2, \tag{4.3}$$

which is independent of V and R as desired. The preceding case study indicates that we must design the time-integrator carefully to satisfy DGCL. Furthermore, the spatial discretization $\mathcal{D}_{i-1/2}$ needs to compute the derivative of third-degree polynomials exactly, hence we are motivated to use a numerical approximation that is at least third-order accurate in space. In the remainder of this section, we construct finite volume method that satisfy both the GCL and TCL discretely. To this end, our numerical approximations to the species are collocated at cell centers and velocities V and u at grid points. Thus the discrete versions of G and N represent cell averages:

$$G_i^n \approx \frac{1}{\triangle\eta\eta_{i-1/2}^2(R^n)^2}\int_{\eta_{i-1}}^{\eta_i} \eta^2(R^n)^2 G(\eta,\tau^n)d\eta, \tag{4.4a}$$

$$N_i^n \approx \frac{1}{\triangle\eta\eta_{i-1/2}^2(R^n)^2}\int_{\eta_{i-1}}^{\eta_i} \eta^2(R^n)^2 N(\eta,\tau^n)d\eta,\ 1 \le i \le L_\eta, \tag{4.4b}$$

where $\eta_{i-1/2} = (i-1/2)\triangle\eta = (\eta_{i-1}+\eta_i)/2$; and the discrete versions of V and u represent nodal values:

$$V_i^n \approx V(\eta_i.\tau^n),\ 0 \le i \le L_\eta, \tag{4.5a}$$

$$u_i^n \approx u(\eta_i.\tau^n),\ 0 \le i \le L_\eta, \tag{4.5b}$$

where no approximation is needed for u since the function $\check{u}$ is prescribed earlier.

The remainder of this chapter is organized as follows. A general finite volume formulation is provided and our main result is given in the next section, which provides sufficient conditions for the method to satisfy the conditions derived previously; and they are presented in the subsequent sections.

4.1 A General Finite Volume Discretization

The explicit first-order time-accurate finite volume formulation of (3.9a) and (3.9b) is obtained by integrating these equations over each cell $[\eta_{i-1},\eta_i]$ and then discretizing the

time-derivative by forward-Euler method :

$$\frac{\eta_{i-1/2}^2[(R^{n+1})^2G_i^{n+1}-(R^n)^2G_i^n]}{\triangle\tau^n}+\frac{1}{\triangle\eta}\Big[F_i^{G,n}-F_{i-1}^{G,n}\Big]=\eta_{i-1/2}^2[(R^n)^2f_i^n-(R')^nR^nG_i^n], \tag{4.6a}$$

$$\frac{\eta_{i-1/2}^2[(R^{n+1})^2N_i^{n+1}-(R^n)^2N_i^n]}{\triangle\tau^n}+\frac{1}{\triangle\eta}\Big[F_i^{N,n}-F_{i-1}^{N,n}\Big]=\eta_{i-1/2}^2[(R^n)^2g_i^n-(R')^nR^nN_i^n]. \tag{4.6b}$$

Here, $F_j^{G,n}$ and $F_j^{N,n}$ denote numerical fluxes for G and N at η_j, respectively:

$$F_j^{G,n}\approx\left(\frac{V}{R}-\frac{\eta R'}{R}\right)\eta^2R^2G\Big|_{\eta=\eta_j,\tau=\tau^n}, \tag{4.7}$$

$$F_j^{N,n}\approx\left(\frac{V}{R}-\frac{\eta R'}{R}+\frac{u}{R}\right)\eta^2R^2N\Big|_{\eta=\eta_j,\tau=\tau^n}. \tag{4.8}$$

The two source terms f_i^n and g_i^n are approximations to the functions f and g, respectively; how the source terms are computed is not critical here, as long as the same formula is used in the velocity equation. In conventional finite volume methods, the velocities $V/R-\eta R'/R$ and $V/R-\eta R'/R+u/R$ are used to compute the two fluxes F_j^G and F_j^N, respectively. For our problems, however, it is advantageous to consider each component of the velocity separately; namely, we segregate the numerical fluxes as:

$$F_j^{G,n}=F_{V,j}^{G,n}+F_{R',j}^{G,n},\qquad F_{V,j}^{G,n}\approx V\eta^2RG,\qquad F_{R',j}^{G,N}\approx-\eta^3R'RG; \tag{4.9}$$

$$F_j^{N,n}=F_{V,j}^{N,n}+F_{R',j}^{N,n}+F_{u,j}^{N,n},\quad F_{V,j}^{N,n}\approx V\eta^2RN,\quad F_{R',j}^{N,n}\approx-\eta^3R'RN,\quad F_{u,j}^{N,n}\approx u\eta^2RN; \tag{4.10}$$

There remains the velocity equation, which does not involve any time derivatives. Therefore, we derive the following discrete version by integrating $\eta^2(R^n)^2\times(4.10c)$ over $[0,\eta_i]$ for all $i=1,....,N_\eta$ and using $V_0^n=0$ to obtain:

$$\eta_i^2R^nV_i^n+\mathcal{F}_{u,i}^{N,n}=\sum_{k=1}^{i}\triangle\eta\Big(\eta_{k-1/2}^2(R^n)^2f_k^n+\eta_{k-1/2}^2(R^n)^2g_k^n\Big). \tag{4.11}$$

Here $\mathcal{F}_{u,j}^{N,n}$ approximates $u\eta^2R^2N$ at (η_i,τ^n); and the source terms on the right hand side use the same formula (hence the same notation) as those in (4.17a) and (4.17b).

4.2 Sufficient Condition for DTCL and DGCL

It is fair to assume that we use the same flux function to compute the numerical fluxes $\mathcal{F}_{V,j}^{G,n}$ and $\mathcal{F}_{V,j}^{N,n}$; to this end we suppose:

$$F_{V,j}^{X,n} = \mathcal{F}_j^n(X_{j+k}^n : -l \leq k \leq r, \mathcal{P}), \tag{4.12}$$

$$F_{R',j}^{X,n} = \widehat{\mathcal{F}}_j^n(X_{j+k}^n : -l \leq k \leq r, \widehat{\mathcal{P}}), \tag{4.13}$$

where $l \geq 0$ and $r \geq 1$ are fixed numbers denoting the stencil of the flux function. Here X represents either species and parameter sets $\mathcal{P}$ and $\widehat{\mathcal{P}}$ are placeholders for high-resolution fluxes that are described later.

We distinguish the flux functions $\mathcal{F}_j^n$ and $\widehat{\mathcal{F}}_j^n$ because the former approximates the fluxes due to a spatially varying velocity V whereas the latter approximates fluxes due to spatially constant velocity R'; furthermore, we maintain the subscript j and the subscript n in these generic functions to indicate their dependence on the spatial coordinates η, domain size R^n, as well as $(R')^n$, which are determined independent of this discretization herein.

Next, we note that both flux functions are in the form $\mathcal{F}(X_{j+k}^n : -l \leq k \leq r, ...)$ where the omitted quantities represent the inputs that are the same when the flux function is applied to compute fluxes for G and N, respectively.

Definition 4.1

The flux function $\mathcal{F}(X_{j+k}^n : -l \leq k \leq r, ...)$ is called *additive* if for all X, Y and $Z = X + Y$:

$$\mathcal{F}(X_{j+k}^n : -l \leq k \leq r, ...) + \mathcal{F}(Y_{j+k}^n : -l \leq k \leq r...) = \mathcal{F}(Z_{j+k}^n : -l \leq k \leq r, ...). \tag{4.14}$$

where the omitted inputs are kept the same in all the three function evaluations.

Furthermore, we define the V-consistency for the flux function $\mathcal{F}_j^n$ of (4.12) and cubic-preserving for the flux function $\widehat{\mathcal{F}}_J^n$ of (4.13) as follows.

Definition 4.2

The numerical flux function $\mathcal{F}_j^n$ of (4.12) is *V-consistent* if for all V_j^n:

$$F_{V,j}^{1,n} = \eta_j^2 R^n V_j^n, \tag{4.15}$$

that is, setting $X_{j+k} = 1, \forall k$ in the right hand side of (4.12) yields $R^n \eta_j^2 V_j^n$.

Definition 4.3

The numerical flux function $\widehat{\mathcal{F}}_j^n$ of (4.13) is *cubic-preserving* if

$$\frac{1}{\triangle\eta}\left(F_{R',j}^{1,n} - F_{R',j-1}^{1,n}\right) = -3\eta_{j-1/2}^2 (R')^n R^n. \tag{4.16}$$

Note that $\mathcal{F}_{R',j}^{X,n}$ can be treated as the flux for an advection equation with the spatially constant velocity $-(R')^n R^n$ and convected variable $\eta^3 X$. The purpose of this section is to derive sufficient conditions such that our method satisfies the discrete geometric geometric conservation law (DGCL) and the discrete totality conservation law (DTCL). To this end, we have the following definitions:

Definition 4.4

The method given by (4.6a),(4.6b) and (4.11) satisfies the DGCL if $N_j^m + G_j^m = 1$, $1 \leq j \leq L_\eta$ and $m = n, n+1$, there are (4.6a) and (4.6b) lead to (4.11).

Definition 4.5

The method given by (4.6a),(4.6b) and (4.11) satisfies the DGCL if they had a conservative discretization of (3.6).

Now we state the main theorem that will eventually guide us in the construction of the proposed numerical methods.

Theorem 4.6

The numerical method given by (4.6a),(4.6b) and (4.11) satisfies both DGCL and DTCL if : (1) $\mathcal{F}_j^n$ is additive and V-consistent, (2) $\widehat{\mathcal{F}}_j^n$ is additive and cubic-preserving, (3)$\mathcal{F}_{u,j}^{N,n} = F_{u,j}^{N,n}$,and (4)$(R')^n$ equals the right hand side of (4.2)

Proof. Adding (4.6a) and (4.6b) and then incorporating (4.11), we get:

$$
\begin{aligned}
&\frac{\eta_{i-1/2}^2}{\triangle\tau^n}\left[((R^{n+1})^2(G_i^{n+1}+N_i^{n+1}) - ((R^n)^2(G_i^n+N_i^n)\right] \\
&+ \frac{1}{\triangle\eta}\left(F_{V,i}^{G,n} + F_{V,i}^{N,n} - F_{V,i-1}^{G,n} - F_{V,i-1}^{N,n}\right) \\
&+ \frac{1}{\triangle\eta}\left(F_{R',i}^{G,n} + F_{R',i}^{N,n} - F_{R',i-1}^{G,n} - F_{R',i}^{N,n}\right) + \frac{1}{\triangle\eta}\left(F_{u,i}^{N,n} - F_{u,i-1}^{N,n}\right) \\
&= \eta_{i-1/2}^2(R^n)^2 f_i^n + \eta_{i-1/2}^2(R^n)^2 g_i^n - \eta_{i-1/2}^2(R')^n R^n(G_i^n+N_i^n) \\
&= \frac{R^n(\eta_i^2 V_i^n - \eta_{i-1}^2 V_{i-1}^n)}{\triangle\eta} + \frac{1}{\triangle\eta}\left(\mathcal{F}_{u,i}^{N,n} - \mathcal{F}_{u,i-1}^{N,n}\right) - \eta_{i-1/2}^2(R')^n R^n(G_i^n+N_i^n)
\end{aligned}
$$

Define $\Theta_i^n = G_i^n + N_i^n$ and $\Theta_i^{n+1} = G_i^{n+1} + N_i^{n+1}$ as before, following the additivity of the fluxes $\mathcal{F}_j^n$ and $\widehat{\mathcal{F}}_j^n$ we obtain:

$$
F_{V,i}^{G,n} + F_{V,i}^{N,n} = F_{V,i}^{\Theta,n} \qquad , \qquad F_{R',i}^{G,n} + F_{R',i}^{N,n} = F_{R',i}^{\Theta,n}
$$

Invoking in addition the assumption that $\mathcal{F}_{u,j}^{N,n} = F_{u,j}^{N,n}$, we obtain:

$$
\begin{aligned}
\frac{\eta_{i-1/2}^2}{\triangle\tau^n}\left[(R^{n+1})^2\Theta_i^{n+1} - (R^n)^2\Theta_i^n\right] + \frac{1}{\triangle\eta}\left(F_{V,i}^{\Theta,n} - F_{V,i-1}^{\Theta,n}\right) + \frac{1}{\triangle\eta}\left(F_{R',i}^{\Theta,n} - F_{R',i-1}^{\Theta,n}\right) \\
= \frac{R^n(\eta_i^2 V_i^2 - \eta_{i-1}^2)V_{i-1}^2}{\triangle\eta} - \eta_{i-1/2}^2(R')^n R^n \Theta_i^n.
\end{aligned}
\tag{4.17}
$$

Clearly, (4.17) represents a conservative finite volume discretization of the continuous totality conservation law (3.6) using the same flux functions $\mathcal{F}_j^n$ and $\widehat{\mathcal{F}}_j^n$; hence the method satisfy DTCL.

Now we move on to show the DGCL and to this end assume $\Theta_i^n \equiv 1$ and $\Theta_I^{n+1} \equiv 1$, then (4.17) reduces to:

$$\frac{\eta_{i-1/2}^2}{\triangle \tau^n}\Big[(R^{n+1})^2 - (R^n)^2\Big] + \frac{1}{\text{ffl}\triangle \eta}\Big(F_{V,i}^{1,n} - F_{V,i-1}^{1,n}\Big) + \frac{1}{\text{ffl}\triangle \eta}\Big(F_{R',i}^{1,n} - F_{R',i-1}^{1,n}\Big) = \frac{R^n(\eta_i^2 V_i^n - \eta_{i-1}^2 V_{i-1}^n}{\triangle \eta} - \eta_{i-1/2}^2 (R')^n R^n. \tag{4.18}$$

Since $(R')^n = ((R^{n+1})^2 - (R^n)^2)/(2 \triangle \tau^n R^n)$ according to (4.2), (4.18) is equivalent to :

$$\frac{1}{\triangle \eta}\Big(F_{V,i}^{1,n} - F_{V,i-1}^{1,n}\Big) + \frac{1}{\triangle \eta}\Big(F_{R',i}^{1,n} - F_{R',i-1}^{1,n}\Big) = \frac{R^n(\eta_i^2 V_i^n - \eta_{i-1}^2 V_{i-1}^n)}{\triangle \eta} - 3\eta_{i-1/2}^2 (R')^n R^n. \tag{4.19}$$

The last equation is trivial following the V-consistency of $\mathcal{F}_j^n$ and the cubic-preserving of $\widehat{\mathcal{F}}_j^n$. Hence we conclude that given all the assumptions as stated, and that $N_j^m + G_j^m = \Theta_j^m = 1$ and $m = n, n+1$, (4.6a) and (4.6b) give rise to (4.11). Thus the methods satisfies DGCL. ■

4.2.1 A Review of The Conventional Flux Functions

We briefly review the conventional finite volume method in the context of (4.10); particularly we consider the spherically symmetric conservation law for a generic species X in spherical coordinates and radial advective velocity U:

$$\frac{\partial(\eta^2 R^2 X)}{\partial \tau} + \frac{\partial}{\partial \eta}\big[U(\eta^2 R^2 X)\big] = 0, \tag{4.20}$$

where we omitted the right hand side since its approximation is generally independent of the finite volume discretizaions.

The conservative herein is $\widetilde{X} \stackrel{def}{=} \eta^2 R^2 X$ rather than X, particularly $\widetilde{X}_j = \eta_{j-1/2}^2 R^2 X_j$ and $\widetilde{X}_{j+1} = \eta_{j+1/2}^2 R^2 X_{j+1}$; hence (4.20) is simply the conservation advection equation for $\widetilde{X}$ by the velocity U:

$$\frac{\partial \widetilde{X}}{\partial \tau} + \frac{\partial}{\partial \eta}(U\widetilde{X}) = 0, \tag{4.21}$$

whose finite volume discretization (at the semi-discretized level) reads:

$$\frac{d\widetilde{X}_j}{d\tau} + \frac{1}{\triangle\eta}(F_i - F_{i-1}) = 0,$$

where $F_j \approx U\widetilde{X}|_{\eta=\eta_j}$. Because $\widetilde{X}_j$ represents the cell average over the cell $[\eta_{j-1}, \eta_j]$ and $\widetilde{X}_{j+1}$ represents that over $[\eta_j, \eta_{j+1}]$, we need to decide which one (or both) to use to approximate the flux at η_j. The widely used first-order flux is given by the upwind flux:

$$F_j = \mathcal{F}^{upw}(U_j, \widetilde{X}_j, \widetilde{X}_{j+1}), \tag{4.22}$$

where U_j is the nodal velocity at η_j and the flux function is given by:

$$\mathcal{F}^{upw}(U_j, Y_j, Y_{j+1}) = \begin{cases} U_j Y_j & U_j \geq 0; \\ U_j Y_{j+1} & U_j < 0. \end{cases} \tag{4.23}$$

Extension to higher accuracy is achieved by the limited polynomial reconstruction. One of the most widely used second-order extension is given by the high resolution MUSCL flux [24]:

$$F_j = \mathcal{F}^{muscl}(U_j, \widetilde{X}_{j-1}, \widetilde{X}_j, \widetilde{X}_{j+1}, \widetilde{X}_{j+2}, \phi_j, \phi_{j+1}) \tag{4.24}$$

where ϕ_j and ϕ_{j+1} are slope limiters and the MUSCL flux function is given by:

$$\begin{aligned} F_j &= \mathcal{F}^{muscl}(U_j, \widetilde{X}_{j-1}, \widetilde{X}_j, \widetilde{X}_{j+1}, \widetilde{X}_{j+2}, \phi_j, \phi_{j+1}) \\ &\stackrel{def}{=} \mathcal{F}^{upw}\left(U_j, Y_j + \frac{1}{2}\phi_j(Y_{j+1} - Y_j), Y_{j+1} - \frac{1}{2}\phi_{j+1}(Y_{j+2} - Y_{j+1})\right) \end{aligned} \tag{4.25}$$

The slope limiter $\phi_j \in [0, 1]$ usually depends on the solutions, but only weakly in the following sense. Slope limiters are introduced to reduce the magnitude if the slope such that the reconstruction will not introduce any new local extremum - a property called monotone preserving. Hence if $\phi_j = c_j$ satisfies the monotone preserving property, so is all slope limiters $\phi_j \in [0, c_j]$. For this reason the slope limiters are introduced as free (or more precisely semi-free) parameters.

The limiters are non-linear functions of the discrete solutions, for example, the minmod limiter computes:

$$\phi_j = \varphi^{minmod}(\triangle Y_{j-1}, \triangle Y_j) \begin{cases} 0, & \triangle Y_{j-1} \triangle Y_j \leq 0, \\ min\left(\dfrac{\triangle Y_{j-1}}{\triangle Y_j}, 1\right), & \triangle Y_{j-1} \triangle Y_j > 0. \end{cases} \tag{4.26}$$

Here $\triangle Y_k = Y_{k+1} - Y_k, \forall k$.

In the meantime, we show that these conventional flux functions are neither V-consistent nor cubic-preserving. The later is obvious since the upwind flux is only first-order accurate and the MUSCL flux is at most second-order; whereas cubic-preserving requires a third-order flux advection equations.

We will now focus on V-consistency and consider, for example, the upwind flux $\mathcal{F}^{upw}$. Then the V- consistency requires that $\mathcal{F}^{upw}(V_j/R, \eta^2_{j-1/2}R^2, \eta^2_{j+1/2}R^2) = \eta^2_j RV_j$; however, this equality does not hold either when $V_j \geq 0$, in which case according to (4.23) :

$$\mathcal{F}^{upw}\left(\frac{V_j}{R}, \eta^2_j R^2, \eta^2_{j+1} R^2\right) = \eta^2_{j-1/2} RV_j \neq \eta^2_j RV_j;$$

similarly if $V_j > 0$, we have

$$\mathcal{F}^{upw}\left(\frac{V_j}{R}, \eta^2_j R^2, \eta^2_{j+1} R^2\right) = \eta^2_{j+1/2} RV_j \neq \eta^2_j RV_j;$$

Nevertheless, the conventional methods that were just reviewed will the basis for our constructing fluxes according to our requirement.

4.2.2 Modified Fluxes for Transporting The Species. *V*-consistency:

In this section, V- consistent flux $\mathcal{F}^n_j$ by modifying the conventional upwind or MUSCL fluxes; then a synchronized limiter is introduced to ensure the additivity property as required by Theorem 4.5. The fluxes $F^{X,n}_{V,j}$ and $F^{X,n}_{u,j}$ will be constructed accordingly. To

construct a V-consistent flux $\mathcal{F}_j^n$, instead of applying the conventional flux functions to the conservative variables $\widetilde{X}_j$, we consider the primitive ones X_j instead. Particularly, a first-order upwind method for (4.13) can be constructed by setting $l = 0, r = 1$ and $\mathcal{P} = \emptyset$

$$\mathcal{F}j^n(\{X_j^n, X_{j+1}^n\}) = \eta_j^2 (R^n)^2 \mathcal{F}^{upw}\left(\frac{V_j^n}{R^n}, X_j^N, X_{j+1}^n\right). \tag{4.27}$$

Because $\mathcal{F}^{upw}(V_j^n/R^n, 1, 1) \equiv V_j^n/R^n$, the flux (4.27) is V- consistent.

Similarly, extension to higher-order accuracy can make use of the MUSCL flux (4.25) :

$$\begin{aligned}
&\mathcal{F}_j^n\left(\left\{X_{j-1}^n, X_j^n, X_{j+1}^n, X_{j+2}^n\right\}, \left\{\phi_j^{X,n}, \phi_{j+1}^{X,n}\right\}\right) \\
&= \eta_j^2 (R^n)^2 \mathcal{F}^{muscl}\left(\frac{V_j^n}{R^n}, X_{j-1}^n, X_j^n, X_{j+1}^n, X_{j+2}^n, \phi_j^{X,n}, \phi_{j+1}^{X,n}\right),
\end{aligned} \tag{4.28}$$

$$\phi_k^{X,n} = \varphi^{minmod}(\triangle X_{k-1}^n, \triangle X_k^n), \quad k = j, j+1. \tag{4.29}$$

Here clearly $\mathcal{P} = \left\{\phi_j^{X,n}, \phi_{j+1}^{X,n}\right\}$ and the limiter can be replaced by any other limiter of choice. It is not difficult to verify that $X_k^n \equiv 1$, the MUSCL flux $\mathcal{F}^{muscl}$ gives rise V_j^n/R^n regardless of the values of the limiters; hence the flux function (4.28) is V- consistent, no matter what the limiter we choose.

Next, the additivity of these fluxes are considered, which is essentially requiring that the fluxes are linear in the inputs X_j. Hence, the upwind fluxes are by nature additive; as an example, we consider $\mathcal{F}_j^n$ and suppose $V_j^n \geq 0$, then following (4.29):

$$\begin{aligned}
&F_{V,j}^{G,n} + F_{V,j}^{N,n} = \mathcal{F}_j^n\left(\{G_j^n, G_{j+1}^n\}\right) + \mathcal{F}_j^n\left(\{N_j^n + N_{j+1}^n\}\right) \\
&= \eta_j^2 (R^n)^2 \mathcal{F}^{upw}\left(\frac{V_j^n}{R^n}, G_j^n, G_{j+1}^n\right) + \eta_j^2 (R^n)^2 \mathcal{F}^{upw}\left(\frac{V_j^n}{R^n}, N_j^n, N_{j+1}^n\right) \\
&= \eta_j^2 (R^n)^2 \cdot \frac{V_j^n}{R^n} G_j^n + \eta_j^2 (R^n)^2 \cdot \frac{V_j^n}{R^n} N_j^n = \eta_j^2 (R^n)^2 \cdot \frac{V_j^n}{R^n} \Theta_j^n \\
&= \eta_j^2 (R^n)^2 \mathcal{F}^{upw}\left(\frac{V_j^n}{R^n}, \Theta_j^n, \Theta_{j+1}^n\right) = F_{V,j}^{\Theta,n}.
\end{aligned}$$

The argument for the case $V_j^n < 0$ is similar; hence $\mathcal{F}_j^n$ is additive.

Extension to MUSCL-based fluxes (4.28) is not straightforward, as the limiter function φ is generally nonlinear in the solutions. Following the discussion below equation (4.25),

we can circumvent by synchronizing the limiters G and N, that is:

$$F_{V,j}^{G,n} = \eta_j^2 (R^n)^2 \mathcal{F}^{muscl}\left(\frac{V_j^n}{R^n}, G_{j-1}^n, G_j^n, G_{j+1}^n, G_{j+2}^n, \phi_j^n, \phi_{j+1}^n\right) \tag{4.30}$$

$$F_{V,j}^{N,n} = \eta_j^2 (R^n)^2 \mathcal{F}^{muscl}\left(\frac{V_j^n}{R^n}, N_{j-1}^n, N_j^n, N_{j+1}^n, N_{j+2}^n, \phi_j^n, \phi_{j+1}^n\right) \tag{4.31}$$

where

$$\phi_k^n = min\big(\phi_k^{G,n}, \phi_k^{N,n}, \phi_k^{\Theta,n}\big), \qquad k = j, j+1. \tag{4.32}$$

Here, $\phi_k^{G,n}, \phi_k^{N,n}$ and $\phi_k^{\Theta,n}$ are obtained by (4.29) with $X = G, X = N$ and $X = \Theta = G + N$, respectively. Note that the same limiters are used to compute the two fluxes. To show the additivity, we assume again without loss of generality that $V_j^n \geq 0$, then:

$$\begin{aligned}
F_{V,j}^{G,n} &= \eta_j^2 (R^n)^2 \cdot \frac{V_j^n}{R^n}\left(G_J^n + \frac{1}{2}\phi_j^n (G_{j+1}^n - G_j^n)\right), \\
F_{V,j}^{N,n} &= \eta_j^2 (R^n)^2 \cdot \frac{V_j^n}{R^n}\left(N_J^n + \frac{1}{2}\phi_j^n (N_{j+1}^n - N_j^n)\right) \\
\Rightarrow F_{V,j}^{G,n} + F_{V,j}^{N,n} &= \eta_j^2 (R^n)^2 \cdot \frac{V_j^n}{R^n}\left(\Theta_j^n + \frac{1}{2}\phi_j^n (\Theta_{j+1}^n - \Theta_j^n)\right) \\
&= \eta_j^2 (R^n)^2 \mathcal{F}^{muscl}\left(\frac{V_j^n}{R^n}, \Theta_{j-1}^n, \Theta_j^n, \Theta_{j+1}^n, \Theta_{j+2}^n, \phi_j^n, \phi_{j+1}^n\right).
\end{aligned}$$

Finally, we complete the fluxes for the species by computing $F_{u,j}^{N,n}$ as:

$$F_{u,j}^{N,n} = \eta_j^2 (R^n)^2 \mathcal{F}^{upw}\left(\frac{u_j^n}{R^n}, N_j^n, N_{j+1}^n\right) \tag{4.33}$$

for first-order accuracy or:

$$F_{u,j}^{N,n} = \eta_j^2 (R^n)^2 \mathcal{F}^{muscl}\left(\frac{u_j^n}{R^n}, N_{j-1}^n, N_j^n, N_{j+1}^n, N_{j+2}^n, \phi_j^n, \phi_{j+1}^n\right) \tag{4.34}$$

for the second-order accuracy.

4.2.3 Modified Fluxes for Transporting The Species.Cubic-preserving:

To construct a cubic-preserving flux $F^{X,n}_{R',j}$, however, we cannot follow the same strategy in previous section. Indeed, if this flux was defined as:

$$F^{X,n}_{R',j} = \eta_j^2 (R^n)^2 \mathcal{F}^{upw}\left(-\frac{\eta_j (R')^n}{R^n}, X^n_j, X^n_{j+1} \right);$$

supposing $(R')^n \geq 0$ and setting the $X^n_k \equiv 1$ we have:

$$\begin{aligned} & F^{1,n}_{R',j} = -\eta^3_{j-1}(R')^n R^n, \quad F^{1,n}_{R',j} = -\eta^3_j (R')^n R^n \\ & \Rightarrow \frac{1}{\triangle \eta}\left(F^{1,n}_{R',j} - F^{1,n}_{R',j-1} \right) = \left(3\eta^2_{j-1/2} + \frac{1}{4} \triangle \eta^2 \right) (R')^n R^n. \end{aligned}$$

To proceed, we recognize that a higher-order and non-linearly stable flux can be obtained by the reconstruct solutions on each cell such that the total variation does not increase, and apply the upwind flux to the two reconstructed values on both sides of the cell face. In the MUSCL scheme, the reconstruction is achieved by limiting the slope of a linear reconstruction that preserve cell average; in this section, we adopt the average -preserving and monotone cubic reconstruction of the Piecewise Parabolic Method (PPM) [14],but construct the flux differently.

Let the variable be Y as before, the PPM reconstruction is normalized coordinates $\xi = (\eta - \eta_{j-1})/\triangle \eta$ on the cell $[\eta_{j-1}, \eta_j]$ reads:

$$Y(\xi) = Y_{j,-} + \xi\big(Y_{j,+} - Y_{j,-} + Y_{6,j}(1-\xi)\big), \quad Y_{6,j} \stackrel{\triangle}{=} 6Y_j - 3(Y_{j,-} + Y_{j,+}). \tag{4.35}$$

Here $Y_{j,-}$ and $Y_{j,+}$ are the two end values that are defined as:

$$Y_{j,-} = Y_j + \phi^Y_{j,-}(Y_{j-1/2} - Y_j), \quad Y_{j-1/2} \stackrel{\triangle}{=} \frac{7}{12}(Y_{j-1} + Y_j) - \frac{1}{12}(Y_{j-2} + Y_{j+1}), \tag{4.36}$$

$$Y_{j,+} = Y_j + \phi^Y_{j,+}(Y_{j+1/2} - Y_j), \quad Y_{j+1/2} \stackrel{\triangle}{=} \frac{7}{12}(Y_j + Y_{j+1}) - \frac{1}{12}(Y_{j-1} + Y_{j+2}). \tag{4.37}$$

The value $Y_{k+1/2}, k = j-1, j$ are third-order reconstruction of the face values when the data is smooth; and the two limiters $\phi_{j,\pm}$ are decided as follows:

(a) if $(Y_{j+1/2}-Y_j)(Y_{j-1/2}-Y_j)\geq 0$, we have a local extrema and set:

$$\phi^Y_{j,-}=\phi^Y_{j,+}=0. \tag{4.38}$$

(b) if (a) is not true, and if $\big|Y_{j+1/2}-Y_j\big|>2\big|Y_{j-1/2}-Y_j\big|$ or $\big|Y_{j-1/2}-Y_j\big|>2\big|Y_{j+1/2}-Y_j\big|$, the reconstructed profile is n ot monotone or the interval; $[\eta_{j-1},\eta_j]$ and we compute:

$$\phi^Y_{j,+}=-\frac{2(Y_{j-1/2}-Y_j)}{Y_{j+1/2}-Y_j} \tag{4.39}$$

in the former case, and

$$\phi^Y_{j,-}=-\frac{2(Y_{j+1/2}-Y_j)}{Y_{j-1/2}-Y_j} \tag{4.40}$$

in the latter case.

(c) Otherwise, set the $\phi^Y_{j,-}$, or $\phi^y_{j,+}$, or both, to one.

Finally denoting $\overline{X}^n_j=\eta^3_{j-1/2}X^n_j$, we define the flux $\check{\mathcal{F}}^n_j$ of (4.13) as:

$$\begin{aligned}&\check{\mathcal{F}}^n_j\big(\{X^n_{j-2},X^n_{j-1},X^n_j,X^n_{j+1},X^n_{j+2},X^n_{j+3}\},\{\phi^{X,n}_{j,+},\phi^{X,n}_{j+1,-}\big)\\&\overset{def}{=}\mathcal{F}^{upw}\big(-(R')^nR^n,\overline{X}_{j,+},\overline{X}_{j+1,-}\big).\end{aligned}$$

Here $j_{k,\pm}$ are computed according to (4.36) and (4.37) with data $\overline{X}_k$; and the limiters $\phi^{X,n}_{j,\pm}$ are computed as $\phi^{\overline{X}}_{j,\mp}$ according (a-c) mentioned before.

Theorem 3.7

The flux $F^{X,n}_{R',j}=\check{\mathcal{F}}^n_j=(\{X^n_{j-2},X^n_{j-1},X^n_j,X^n_{j+1},X^n_{j+2},X^n_{j+3}\},\{\phi^{X,n}_{j,+},\phi^{X,n}_{j+1,+}\})$ *as described before cubic-preserving for* $j\geq 3$. *Proof:* Let $X^n_k\equiv 1,\forall k$, we first compute the reconstructed values $\overline{X}_{k+1/2}$, where $k\geq 2$:

$$\overline{X}_{k+1/2}=\frac{7}{12}\bigg(\eta^3_{k-1/2}\eta^3_{k+1/2}\bigg)-\frac{1}{12}\bigg(\eta^3_{k-3/2}-\eta^3_{k+3/2}\bigg)=\eta^3_k-\frac{1}{4}\eta_k\triangle\eta^2.$$

Now we compute the limiters $\phi^{X,n}_{k,\pm}$ for $k\geq 3$. First of all:

$$\begin{aligned}&\overline{X}_{k+1/2}-\overline{X}_k=\eta^3_k-\frac{1}{4}\eta_k\triangle\eta^2-\eta^3_{k-1/2}=\frac{3}{2}\eta^3_{k-1/2}\triangle\eta+\frac{1}{2}\eta_{k-1/2}\triangle\eta^2>0,\\&\overline{X}_{k-1/2}-\overline{X}_k=\eta^3_{k-1}-\frac{1}{4}\eta_{k-1}\triangle\eta^2-\eta^3_{k-1/2}=-\frac{3}{2}\eta^3_{k-1/2}\triangle\eta+\frac{1}{2}\eta_{k-1/2}\triangle\eta^2<0.\end{aligned}$$

thus the condition in (a) does not hold. Continuing to check the conditions in (b), we get:

$$2\left|\overline{X}_{k+1/2}-\overline{X}_k\right|-\left|\overline{X}_{k-1/2}-\overline{X}_k\right|=\frac{3}{2}\eta_{k-1/2}^3\triangle\eta+\frac{3}{2}\eta_{k-1/2}\triangle\eta^2>0,$$
$$2\left|\overline{X}_{k-1/2}-\overline{X}_k\right|-\left|\overline{X}_{k+1/2}-\overline{X}_k\right|=\frac{3}{2}\eta_{k-1/2}^3\triangle\eta+\frac{3}{2}\eta_{k-1/2}\triangle\eta^2>0.$$

hence neither condition in (b) is true. Thus we conclude that $\phi_{k,\pm}^{X,n}$ for all $k\geq 3$; consequently $\overline{X}_{k,-}=\overline{X}_{k-1/2}$ and $\overline{X}_{k,+}=\overline{X}_{k+1/2}$.

Finally we can calculate the flux $F_{R',j}^{1,n}$ for all $j\geq 3$. Because $\overline{X}_{j+1,-}=\overline{X}_{j+1/2}=\overline{X}_{j,+}$, regardless of the sign of $(R')^n$, there is:

$$F_{R',j}^{1,n}=-(R')^nR^n\overline{X}_{j+1,-}=-(R')^nR^n\overline{X}_{j+1}=-(R')^nR^n\left(\eta_j^3-\frac{1}{4}\eta_j\triangle\eta^2\right)$$

hence for $j\geq 4$:

$$\frac{1}{\triangle\eta}\left(F_{R',j}^{1,n}-F_{R',j-1}^{1,n}\right)=-\frac{(R')^nR^n}{\triangle\eta}\left[\left(\eta_j^3-\frac{1}{4}\eta_j\triangle\eta^2\right)-\left(\eta_{j-1}^3-\frac{1}{4}\eta_{j-1}\triangle\eta^2\right)\right]$$
$$=\left[\left((\eta_{j-1/2}+\frac{1}{2}\triangle\eta)^3-(\eta_{j-1/2}-\frac{1}{2}\triangle\eta)^3\right)-\frac{1}{4}(\eta_j-\eta_{j-1})\triangle\eta^2)\right]$$
$$=-\frac{(R')^nR^n}{\triangle\eta}\left(3\eta_{j-1/2}^2\triangle\eta+\frac{1}{4}\triangle\eta^3-\frac{1}{4}\triangle\eta^3\right)=-3\eta_{j-1/2}^2(R')^nR^n.$$

Hence the constructed flux is cubic-preserving.

Theorem 3.7 addresses the cubic-preserving for cells far away from the origin. More specially, we need to consider the cubic-preserving on the first three cells $[0,\triangle\eta]$, $[\triangle\eta,2\triangle\eta]$ and $[2\triangle\eta,3\triangle\eta]$. Following a similar procedure in the preceding proof, cubic-preserving on these cells amount to:

$$F_{R',2}^{1,n}=-(R')^nR^n\left(\eta_2^3-\frac{1}{4}\eta_2\triangle\eta^2\right),\tag{4.41}$$
$$F_{R',1}^{1,n}=-(R')^nR^n\left(\eta_1^3-\frac{1}{4}\eta_1\triangle\eta^2\right),\tag{4.42}$$
$$F_{R',0}^{1,n}=0.\tag{4.43}$$

Note that (3.53) is enforced automatically as the boundary condition at the origin; hence we focus on the first two, which are guaranteed if:

$$\overline{X}_{2+1/2}=\frac{15}{2}\triangle\eta^3,\quad \overline{X}_{1+1/2}=\frac{3}{4}\triangle\eta^3,$$

and that $\phi^{X,n}_{2,\pm} = \phi^{X,n}_{1,\pm} = 0$ when all X's are 1. The required $\overline{X}_{2+1/2}$ is precisely given by the formula in (4.36) or (4.37) and we then focus on $\overline{X}_{1+1/2}$ and the limiters. To this end, utilizing the knowledge that when Y is given by $\eta^3 X$ we expect $Y = 0$ at $\eta = 0$ and define $\overline{X}_{1+1/2}$ by the generic formula:

$$Y_{1+1/2} = \frac{7}{12}(Y_1 + Y_2) - \frac{1}{12}(Y_3 - Y_1). \tag{4.44}$$

According to this definition $\phi^{X,n}_{2,\pm} = 1$ when $X \equiv 1$ as desired; but on the first cell we have $\phi^{X,n}_{1,-} = 1$ and $\phi^{X,n}_{1,+} = 2/5$ following the criterion before, particularly (4.39). Hence we need to relax it. A simple fix can be derived by the following reconstruction of the cell $[0, \triangle\eta]$ utilizing the knowledge that $Y \sim \eta^3$ near $\eta = 0$:

$$\begin{aligned} &Y(\xi) = \xi^3(Y_{1,+} + Y_{6,1}(1 - \xi^2)), \qquad Y_{6,1} \stackrel{\triangle}{=} 12Y_1 - 3Y_{1,+}, \\ &Y_{1,+} = Y_1 + \phi^Y_{1,+}(Y_{1+1/2} - Y_1). \end{aligned}$$

As before $Y_{6,1}$ is defined such that the mean of $Y(\xi)$ is Y_1 or any right end value $Y_{1,+}$. Because $Y_{1,-} \equiv Y_{1-1/2} \equiv 0$, there is no way to design the limiter such that $Y(\xi)$ is monotone if $Y_1 \neq 0$ and $Y_{1+1/2}$ is too close to Y_1 . To this end, we design $\phi^Y_{1,+}$ to prevent the non-monotone due too large $Y_{1+1/2}$:

(a) If $Y_1 Y_{1+1/2} \leq 0$ or $3\big|Y_{1+1/2}\big| \leq 8\big|Y_1\big|$, we set $\phi^Y_{1,+} = 0$

(b) if (a) is not true, and if $\big|Y_{1+1/2}\big| > 6\big|Y_1\big|$, we define

$$\phi^Y_{1,+} = \frac{5}{Y_{1+1/2}/Y_1 - 1} \tag{4.45}$$

(c) Otherwise, $\phi^Y_{1,+} = 1$. Following this modified definition, when $\overline{X}_1 = \eta^3_{1/2} = \frac{1}{8} \triangle \eta^3$, the limiter is precisely $\phi^{X,n}_{1,+} = 1$. Finally we ensure the additivity of the fluxes $F^{G,n}_{R',j}$ and $F^{N,n}_{R',j}$ by constructing the limiter $\phi^n_{j,\pm}$ for both species as follows. For $j \geq 2$:

(A) If $(\overline{N}_{j+1/2} - \overline{N}_j)(\overline{N}_{j-1/2} - \overline{N}_j) \geq 0$, or $(\overline{G}_{j+1/2} - \overline{G}_j)(\overline{G}_{j-1/2} - \overline{G}_j) \geq 0$, or $(\overline{\Theta}_{j+1/2} - \overline{\Theta}_j)(\overline{\Theta}_{j-1/2} - \overline{\Theta}_j) \geq 0$, we set:

$$\phi^n_{j,-} = \phi^n_{j,+} = 0. \tag{4.46}$$

(B) Otherwise, we compute:

$$\alpha_1 = min\left(\frac{2\left|\overline{N}_{j-1/2} - \overline{N}_j\right|}{\left|\overline{N}_{j+1/2} - \overline{N}_j\right|}, \frac{2\left|\overline{G}_{j-1/2} - \overline{G}_j\right|}{\left|\overline{G}_{j+1/2} - \overline{G}_j\right|}, \frac{2\left|\overline{\Theta}_{j-1/2} - \overline{\Theta}_j\right|}{\left|\overline{\Theta}_{j+1/2} - \overline{\Theta}_j\right|}\right),$$

and

$$\alpha_2 = max\left(\frac{\left|\overline{N}_{j-1/2} - \overline{N}_j\right|}{2\left|\overline{N}_{j+1/2} - \overline{N}_j\right|}, \frac{\left|\overline{G}_{j-1/2} - \overline{G}_j\right|}{2\left|\overline{G}_{j+1/2} - \overline{G}_j\right|}, \frac{\left|\overline{\Theta}_{j-1/2} - \overline{\Theta}_j\right|}{2\left|\overline{\Theta}_{j+1/2} - \overline{\Theta}_j\right|}\right).$$

(B1) If $\alpha_2 > \alpha_1$, use (4.46)

(B2) Otherwise $\alpha_1 < 1$, set: $\phi^n_{j,-} = 1, \quad \phi^n_{j,+} = \alpha_1$; and if $\alpha_2 > 1$, set: $\phi^n_{j,-} = \alpha_2^{-1}, \quad \phi^n_{j,+} = 1$.

Similarly for the first cell, we have:

(A) if $\overline{X}_1\overline{X}_{1+1/2} \leq 0$ or $3\left|\overline{X}_{1+1/2}\right| \leq 8\left|\overline{X}_1\right|$ is true for any of $X = G, X = N, or X = \Theta$, we set $\phi^n_{1,+} = 0$.

(B) Otherwise set:

$$\phi^n_{1,+} = min\left(1, \frac{5}{\overline{G}_{1+2}/\overline{G}_1 - 1}, \frac{5}{\overline{N}_{1+2}/\overline{N}_1 - 1}, \frac{5}{\overline{\Theta}_{1+2}/\overline{\Theta}_1 - 1}\right).$$

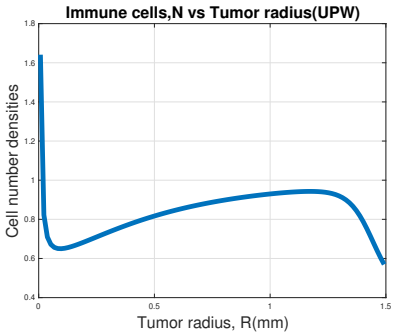
To this point, all the fluxes that are required to update the species have been specified. Now we extend the designed method to the modeling of infiltration of immune cells in a spherical tumor.

In the next chapter, we evaluate the performance of the numerical method derived in this chapter and by solving the simplified model for two species.

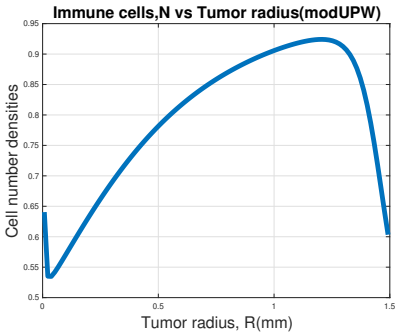
Chapter 5

Result

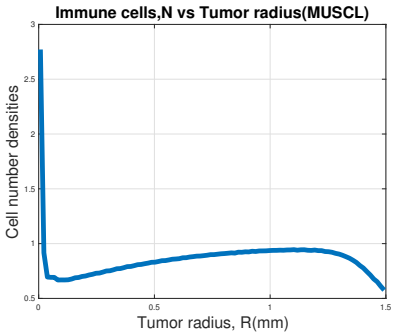
In this chapter, we show the numerical solution for the simplified model solved for only two species G and N. For calculating flux due to V/R and u/R, we have used upwind method, modified upwind method, MUSCL method and also modified MUSCL method. For flux due to $\eta R'/R$, in addition to the previous methods, we have also used piecewise parabolic method to satisfy the sufficient conditions described in the previous chapter.



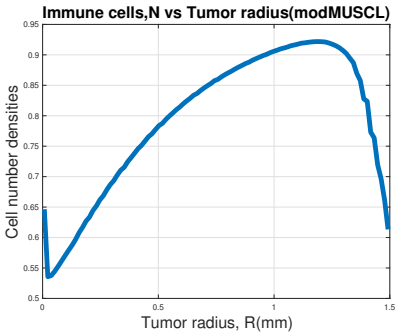
(a) Regular upwind method



(b) Modified upwind method



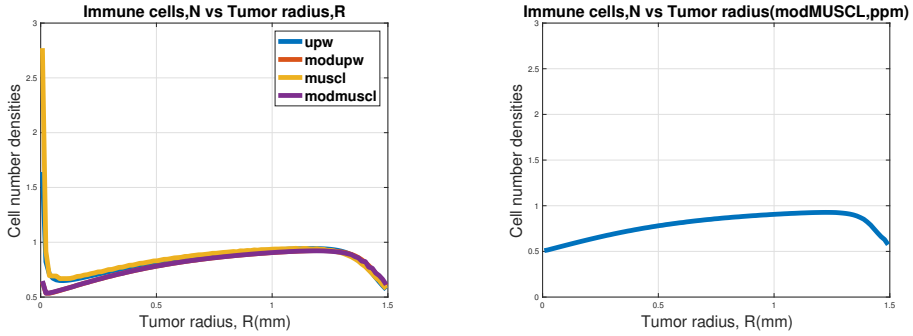
(c) Regular MUSCL method



(d) Modified MUSCL methods

Figure 5.1: Solution of immune cells N against tumor radius R over time

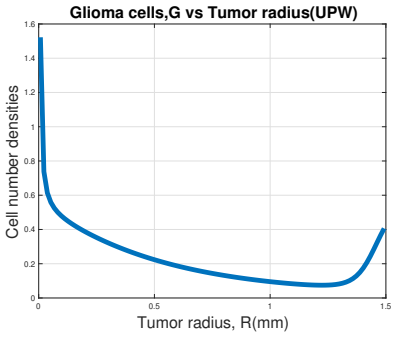
In Figure (5.1), solution of N was obtained. From Figures (5.1a) and (5.1c), we can see that the problem with overshoot around the the origin still persists. In Figures (5.1b) and (5.1d), we can see that the overshoot around the origin is more contained in the latter two.



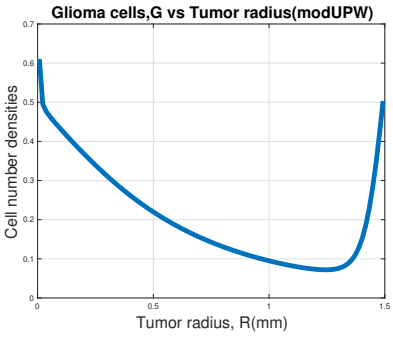
(a) Method comparison (b) Modified MUSCL and PPM

Figure 5.2: Comparison of solutions for N

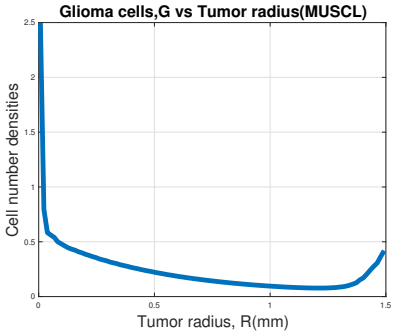
In Figure (5.2a), solutions of all the previous methods for solving N are compared. We can say that modified MUSCL method generates the least overshoot in the four methods. Later, in Figure (5.2b), we used modified MUSCL for flux due to V/R and u/R and PPM for flux due to $\eta R'/R$ and observe no visible oscillation around the origin.



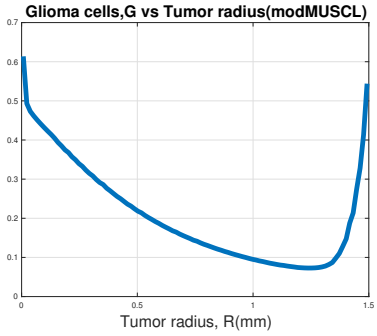
(a) Regular upwind method



(b) Modified upwind method



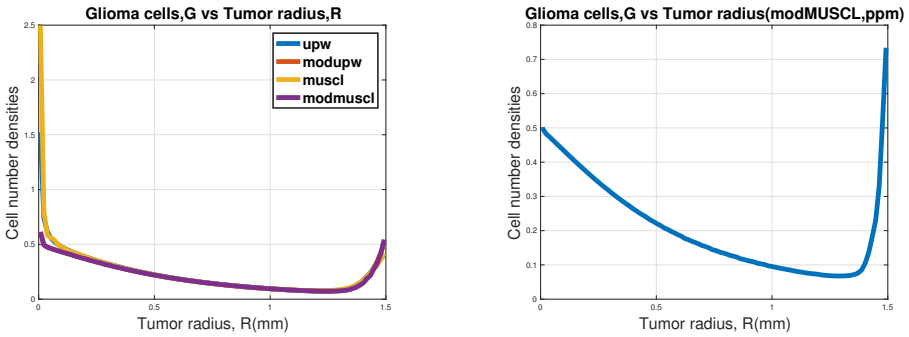
(c) Regular MUSCL method



(d) Modified MUSCL methods

Figure 5.3: Solution of Glioma cells G against tumor radius R over time

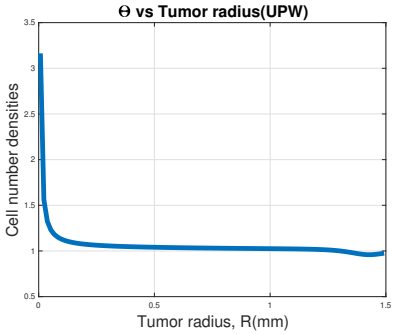
In Figure (5.3), solutions of G are showed. From Figures (5.3a) and (5.3c), we can see that the problem with incompressibility around the the origin still persists. In Figures (5.3b) and (5.3d), we can see that the overshoot around the origin is less visible than of the previous graphs of G.



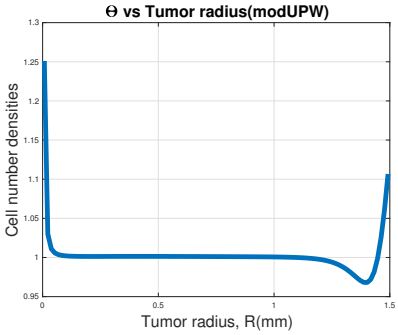
(a) Method comparison (b) Modified MUSCL and PPM

Figure 5.4: Comparison of solutions for G

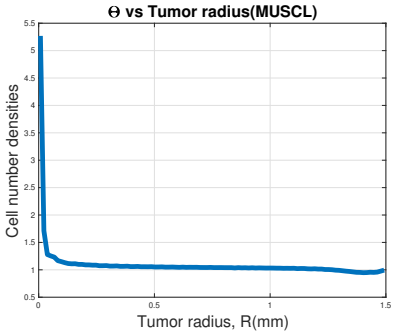
As observed for N, similarly for G, modified MUSCL gives the least overshoot. From Figure (5.4a) between the two conventional and modified methods and Figure (5.4b), we see no visible oscillation which means most of it has been removed from the solution where we have used PPM for calculating the flux due to $\eta R'/R$.



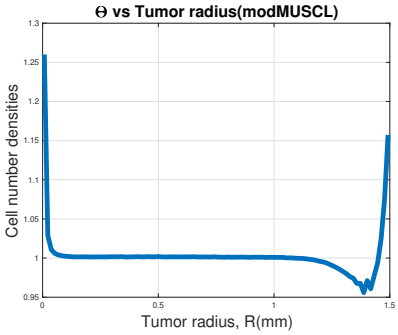
(a) Regular upwind method



(b) Modified upwind method



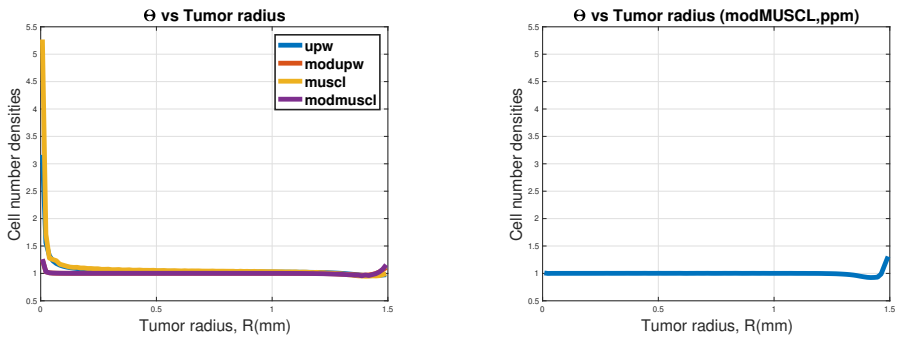
(c) Regular MUSCL method



(d) Modified MUSCL methods

Figure 5.5: Solution of Θ against tumor radius R over time

In Figure (5.5), solution of $\Theta = G + N$ are showed. From Figures (5.5a) and (5.5c), we can see that it does not follow the conservation law at the origin. In Figures (5.5b) and (5.5d), we can see that the sum is close to 1 which is due to second-order approximation.



(a) Method comparison (b) Modified MUSCL and PPM

Figure 5.6: Comparison of solutions for Θ

From Figure (5.6a) between the four methods, modified MUSCL method follows the conservation. In Figure (5.6b), we see that the solution follows conservation law throughout the whole tumor growth except the final point which is 1.5 mm which is really close to 1. From the results above, it can be established that the enhanced finite volume method introduced earlier generates stable trivial solution for both first-order and second order methods, numerically by satisfying the DTCL and DGCL.

Chapter 6

Conclusion

Summary

In this paper, we first briefly described the Platelet-Derived Growth Factor (PDGF)-driven Glioma mathematical model and then we extend the model to Osteopontin model. We showed that using the conventional Gudonov-type finite volume, we can not preserve the trivial solution. We then introduce a new simplified model which has two species G and N, which are Glioma cells and Immune cells, respectively and also a prescribed infiltration velocity. Next, we derived some conditions which are sufficient to satisfy the totality conservation law and the geometric conservation law.

After plotting the solutions using upwind method, modified upwind method, MUSCL method, modified MUSCL method and Piecewise-Parabolic Mehtod (PPM) (only for flux due to $(\eta R'/R)$), we see that using first-order or second-order method does not influence the stability unless we use the newly introduced modified method for both order. We also see a drastic difference when we use the PPM. Using second-order MUSCL method for calculating flux due to (V/R) and (u/R) and Piecewise-Parabolic Method for $(\eta R'/R)$, we get the most stable and constant solution which also satisfies both GCL and TCL.

References

[1] Boilly, B., Vercoutter-Edouart, A. S., Hondermarck, H., Nurcombe, V., & Le Bourhis, X. (2000). FGF signals for cell proliferation and migration through different pathways. Cytokine and Growth Factor Reviews. http://doi.org/10.1016/S1359-6101(00)00014-9

[2] Elliott, R. L., & Blobe, G. C. (2005). Role of transforming growth factor Beta in human cancer. Journal of Clinical Oncology : Official Journal of the American Society of Clinical Oncology. http://doi.org/10.1200/JCO.2005.02.047

[3] Weber, G. F. (2001). The metastasis gene osteopontin: A candidate target for cancer therapy. Biochimica et Biophysica Acta - Reviews on Cancer. http://doi.org/10.1016/S0304-419X(01)00037-3

[4] Rittling, S. R., & Chambers, A. F. (2004). Role of osteopontin in tumour progression. British Journal of Cancer. http://doi.org/10.1038/sj.bjc.6601839

[5] Rangaswami, H., Bulbule, A., & Kundu, G. C. (2006). Osteopontin: Role in cell signaling and cancer progression. Trends in Cell Biology. http://doi.org/10.1016/j.tcb.2005.12.005

[6] Gardner,H. A., Berse, B., & Senger, D. R. (1994). Specific reduction in osteopontin synthesis by antisense RNA inhibits the tumorigenicity of transformed Rat1 fibroblasts. Oncogene.

[7] Rogers, K. (n.d.). Microglia. Retrieved May 27, 2018, from https://www.britannica.com/science/microglia

[8] National Cancer Institute. (n.d.). Monocytes. Retrieved May 27, 2018, from https://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0022057/

[9] Ben Niu, Xianyi Zeng, Frank Szulzewsky, Sarah Holte, Philip K. Maini, Eric C. Holland, and Jianjun Paul Tian (2018). Mathematical modeling of PDGF-driven glioma reveals the infiltrating dynamics of immune cells into tumors. Submitted.

[10] Senger, D. R., Wirth, D. F., & Hynes, R. O. (1979). Transformed mammalian cells secrete specific proteins and phosphoproteins. Cell. http://doi.org/10.1016/0092-8674(79)90103-X

[11] Mariani, M. C., Bhuiyan, M. A. M., & Tweneboah, O. K. (2018). Estimation of stochastic volatility by using Ornstein–Uhlenbeck type models. Physica A: Statistical Mechanics and Its Applications. http://doi.org/10.1016/j.physa.2017.08.153

[12] Calvez, V., & Carrillo, J. A. (2006). Volume effects in the Keller-Segel model: energy estimates preventing blow-up. Journal Des Mathematiques Pures et Appliquees. http://doi.org/10.1016/j.matpur.2006.04.002

[13] Chertock, A., Epshteyn, Y., Hu, H., & Kurganov, A. (2018). High-order positivity-preserving hybrid finite-volume-finite-difference methods for chemotaxis systems. Advances in Computational Mathematics. http://doi.org/10.1007/s10444-017-9545-9

[14] Colella, P., & Woodward, P. R. (1984). The Piecewise Parabolic Method (PPM) for gas-dynamical simulations. Journal of Computational Physics. http://doi.org/10.1016/0021-9991(84)90143-8

[15] Espejo, E., Vilches, K., & Conca, C. (2013). Sharp condition for blow-up and global existence in a two species chemotactic Keller-Segel system in. European Journal of Applied Mathematics. http://doi.org/10.1017/S0956792512000411

[16] Farhat, C., Geuzaine, P., & Grandmont, C. (2001). The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. Journal of Computational Physics. http://doi.org/10.1006/jcph.2001.6932

[17] Filbet, F. (2006). A finite volume scheme for the Patlak-Keller-Segel chemotaxis model. Numerische Mathematik. http://doi.org/10.1007/s00211-006-0024-3

[18] Gottlieb, S., & Shu, C.-W. (1998). Total variation diminishing Runge-Kutta schemes. Mathematics of Computation of the American Mathematical Society. http://doi.org/10.1090/S0025-5718-98-00913-2

[19] Keller, E. F., & Segel, L. A. (1971). Model for chemotaxis. Journal of Theoretical Biology. http://doi.org/10.1016/0022-5193(71)90050-6

[20] LeVeque, R. J. (2002). Finite Volume Methods for Hyperbolic Problems. Cambridge University Press. http://doi.org/10.1017/CBO9780511791253

[21] Morton, K. W., & Sweby, P. K. (1987). A comparison of flux limited difference methods and characteristic galerkin methods for shock modelling. Journal of Computational Physics. http://doi.org/10.1016/0021-9991(87)90114-8

[22] López Ortega, A., & Scovazzi, G. (2011). A geometrically-conservative, synchronized, flux-corrected remap for arbitrary Lagrangian-Eulerian computations with nodal finite elements. Journal of Computational Physics. http://doi.org/10.1016/j.jcp.2011.05.005

[23] Gottlieb, S., & Shu, C.-W. (1998). Total variation diminishing Runge-Kutta schemes. Mathematics of Computation of the American Mathematical Society. http://doi.org/10.1090/S0025-5718-98-00913-2

[24] van Leer, B. (1979). Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. Journal of Computational Physics. http://doi.org/10.1016/0021-9991(79)90145-1

[25] Xianyi Zeng.(2016) A general approach to enhance slope limiters in MUSCL schemes on nonuniform rectilinear grids. SIAM J. Sci. Comput., 38(2):A789–A813

[26] Mantovani A., B. F. (2001). Inflammation and cancer: back to. Lancet. http://doi.org/10.1016/S0140-6736(00)04046-0

[27] Quail, D., & Joyce, J. (2013). Microenvironmental regulation of tumor progression and metastasis. Nature Medicine. http://doi.org/10.1038/nm.3394.Microenvironmental

[28] Kitamura, T., Qian, B. Z., & Pollard, J. W. (2015). Immune cell promotion of metastasis. Nature Reviews Immunology. http://doi.org/10.1038/nri3789

[29] Shalapour, S., & Karin, M. (2015). Immunity, inflammation, and cancer: An eternal fight between good and evil. Journal of Clinical Investigation. http://doi.org/10.1172/JCI80007

[30] Gannot, G., Gannot, I., Vered, H., Buchner, A., & Keisari, Y. (2002). Increase in immune cell infiltration with progression of oral epithelium from hyperkeratosis to dysplasia and carcinoma. British Journal of Cancer. http://doi.org/10.1038/sj.bjc.6600282

[31] Hald, J., Rasmussen, N., & Claesson, M. H. (1995). Tumour-infiltrating lymphocytes mediate lysis of autologous squamous cell carcinomas of the head and neck. Cancer Immunology Immunotherapy. http://doi.org/10.1007/BF01516999

[32] Calzascia, T., Di Berardino-Besson, W., Wilmotte, R., Masson, F., de Tribolet, N., Dietrich, P.-Y., & Walker, P. R. (2003). Cutting edge: cross-presentation as a mechanism for efficient recruitment of tumor-specific CTL to the brain. Journal of Immunology (Baltimore, Md. : 1950). http://doi.org/10.4049/jimmunol.171.5.2187

[33] Jochems, C., & Schlom, J. (2011). Tumor-infiltrating immune cells and prognosis: The potential link between conventional cancer therapy and immunity. Experimental Biology and Medicine. http://doi.org/10.1258/ebm.2011.011007

[34] Zhang, Q., Liu, L., Gong, C., Shi, H., Zeng, Y., Wang, X., ... Wei, Y. (2012). Prognostic Significance of Tumor-Associated Macrophages in Solid Tumor: A Meta-Analysis of the Literature. PLoS ONE. http://doi.org/10.1371/journal.pone.0050946

[35] Templeton, A. J., McNamara, M. G., Šeruga, B., Vera-Badillo, F. E., Aneja, P., Ocaña, A., ... Amir, E. (2014). Prognostic role of neutrophil-to-lymphocyte ratio in solid tumors: A systematic review and meta-analysis. Journal of the National Cancer Institute. http://doi.org/10.1093/jnci/dju124

[36] Boon, T., Coulie, P. G., & Van Den Eynde, B. (1997). Tumor antigens recognized by T cells. Immunology Today. http://doi.org/10.1016/S0167-5699(97)80020-5

[37] Nosho, K., Baba, Y., Tanaka, N., Shima, K., Hayashi, M., Meyerhardt, J. A., ... Ogino, S. (2010). Tumour-infiltrating T-cell subsets, molecular changes in colorectal cancer, and prognosis: cohort study and literature review. J Pathol. http://doi.org/10.1002/path.2774 [doi]

[38] Zhang, Q., Liu, L., Gong, C., Shi, H., Zeng, Y., Wang, X., ... Wei, Y. (2012). Prognostic Significance of Tumor-Associated Macrophages in Solid Tumor: A Meta-Analysis of the Literature. PLoS ONE. http://doi.org/10.1371/journal.pone.0050946

[39] Templeton, A. J., McNamara, M. G., Šeruga, B., Vera-Badillo, F. E., Aneja, P., Ocaña, A., ... Amir, E. (2014). Prognostic role of neutrophil-to-lymphocyte ratio in solid tumors: A systematic review and meta-analysis. Journal of the National Cancer Institute. http://doi.org/10.1093/jnci/dju124

[40] Noushmehr, H., Weisenberger, D. J., Diefes, K., Phillips, H. S., Pujara, K., Berman, B. P., Aldape, K. (2010). Identification of a CpG Island Methylator Phenotype that Defines a Distinct Subgroup of Glioma. Cancer Cell. http://doi.org/10.1016/j.ccr.2010.03.017

[41] Pareschi, L., & Russo, G. (2005). Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation. In Journal of Scientific Computing. https://doi.org/10.1007/s10915-004-4636-4

Appendix A

Matlab Code

The below code is for selecting the method for calculating the flux by the FVM and also setting the values for the parameters.

sample_run.m

```
Nx  = 100;
Na  = Nx;
T   = 1.0;
R0  = 1.0;
Nbc = 0.5;
f = @(time,x_array,radius,g_array,n_array)zeros(size(x_array));
g = @(time,x_array,radius,g_array,n_array)zeros(size(x_array));
u = @(time,x_array)-x_array.^2;

opt{1} = 0.4;     % CFL
opt{2} = 0;       % Initial condition
opt{3} = 1;       % dR
opt{4} = 0.1;     % dt_max
opt{5} = 0;       % 0 (upw) 1 (mod) 2 (MUSCL) 3 (mod MUSCL) for _V
opt{6} = 1;       % 0 (upw) 1 (mod) 2 (ppm) 3 (mod MUSCL)
...4 (MUSCL) for _R
opt{7} = 0;       % 0 (upw) 1 (mod) 2 (MUSCL) 3 (mod MUSCL) for _U
```

```
opt{8} = 1e-8;  % Tolerance

[sol_R, sol_G, sol_N,sol_V,csrv_G,csrv_N,x_c]= tumor_run(Nx,f,g,...
u,T,R0,Nbc,opt);
radius = x_c*sol_R(1,end);
gpn = (sol_G(end,:)+sol_N(end,:));% Theta
```

tumor_run.m

The below routine evaluates the values passed on to it from the sample_run.m file and calls the functions as set previously.

```
function [sol_R, sol_G, sol_N,sol_V,csrv_G,csrv_N,x_c] =...
tumor_run(Nx,f,g,u,T,R0,Nbc,opt)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Nx:  Number of uniform cells
% T:   Terminal time
% R0:  Initial radius
% f:   Handle for source term function of G
%      A generic function that depends on:
%       t, x, R, G, and N
% g:   Handle for source term function of N
%      A generic function that depends on:
%       t, x, R, G, and N
% u:   Handle for infiltration velocity
%      u needs to be zero at x=0
% Nbc: Boundary data for N
% opt: opt{1} -- cfl number
```

```
%         opt{2} -- initial condition index
%         opt{3} -- method to compute dR
%         opt{4} -- maximum allowed dt
%         opt{5} -- flux for _V
%         opt{6} -- flux for _R
%         opt{7} -- flux for _U
%         opt{8} -- tolerance
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  cfl      = opt{1};
  dx       = 1.0/Nx;

  % (1,:) -- current solution in each loop
  % (2,:) -- updated solution in each loop
  % (3,:) -- temporary storage
  G        = zeros(3,Nx);   % Arrays for storing G at cells
  N        = zeros(3,Nx);   % Arrays for storing N at cells
  A        = zeros(3,Nx);   % Arrays for storing A at cells
  R        = zeros(3,1);    % Arrays for storing R
  phi_V    = zeros(1,Nx+1);%Arrays for storing Limiter for MUSCL
  phi_U    = zeros(1,Nx+1);
  phi_R    = zeros(1,Nx+1);
  x_f = linspace(0,1,Nx+1); % Normalized coords at faces
  x_c = 0.5*(x_f(1:end-1)+x_f(2:end)); % Normalized cell centers

  V  = zeros(1,Nx+1); % Array for storing V at nodes
  U = zeros(1,Nx+1); % Array for storing U at nodes
```

```
S_F = zeros(1,Nx); % Array for storing f at cell centers
S_G = zeros(1,Nx); % Array for storing g at cell centers

csrv_G    = zeros(2,Nx);
csrv_N    = zeros(2,Nx);
flux_G_V = zeros(1,Nx+1);
flux_G_R = zeros(1,Nx+1);
flux_N_V = zeros(1,Nx+1);
flux_N_R = zeros(1,Nx+1);
flux_N_U = zeros(1,Nx+1);

%
% Initial setup
%
t_cur = 0.0;
% Initial data for G, N
switch opt{2}
case 0
  G(1,:) = (1.0-Nbc)*ones(1,Nx);
  N(1,:) = Nbc*ones(1,Nx);
end
% Initial radius
R(1,1) = R0;
% Initial sources
S_F(:) = f(t_cur,x_c,R0,G(1,:),N(1,:));
S_G(:) = g(t_cur,x_c,R0,G(1,:),N(1,:));
% Initial velocities
U(:) = u(t_cur,x_f);
```

```
V(:) = calc_v(R0,U,N(1,:),S_F,S_G,x_c,x_f,Nbc);
dt   = calc_dt(dx,cfl,R0,V,U,x_f,opt{4});

% Output
itime           = 1;
time(1)         = t_cur;
sol_R(1)        = R0;
sol_G(1,1:Nx)   = G(1,:);
sol_N(1,1:Nx)   = N(1,:);
sol_V(1,1:Nx+1) = V(:);

while t_cur < T
  %
  % Update the solutions G and N using
  % S_F, S_G, U, and V that are calculated
  % in the previous time step
  %
  % Only for forward Euler
  %

  % Step 1: update dR and R
  R(2,1) = R(1,1) + dt*V(end);
  cur_R  = R(1,1);
  switch opt{3}
  case 0
    dR = V(end);

  case 1
```

```
  dR = V(end) * (1+0.5*dt*V(end)/cur_R);

end

% Step 2: compute conservative G and N
for i = 1:Nx
  csrv_G(1,i) = x_c(i)*x_c(i)*cur_R*cur_R*G(1,i);
  csrv_N(1,i) = x_c(i)*x_c(i)*cur_R*cur_R*N(1,i);

end

% Step 3: compute the fluxes _V
switch opt{5}
case 0 % Classical upwind FVM applied to csrv_*
 flux_G_V(:) = calc_flux_upw( V, 1.0/cur_R,...
 csrv_G(1,:), 1, 0.0 );
 flux_N_V(:) = calc_flux_upw( V, 1.0/cur_R,...
 csrv_N(1,:), 1, 0.0 );
case 1 % Modified upwind

 flux_G_V(:) = calc_flux_upw( V, 1.0/cur_R, G(1,:), 1, 0.0 );
 flux_N_V(:) = calc_flux_upw( V, 1.0/cur_R, N(1,:), 1, 0.0 );
  for i = 1:Nx+1
    flux_G_V(i) = flux_G_V(i)*x_f(i)*x_f(i)*cur_R*cur_R;
    flux_N_V(i) = flux_N_V(i)*x_f(i)*x_f(i)*cur_R*cur_R;
  end
  case 2 % MUSCL
      phi_V = calc_limit(V,G(1,:),N(1,:));
```

```
  flux_G_V = calc_muscl_new(V, 1.0/cur_R,...
  csrv_G(1,:),phi_V, 1, 0.0);
  flux_N_V = calc_muscl_new(V, 1.0/cur_R,...
  csrv_N(1,:),phi_V, 1, 0.0);

case 3 % Modified MUSCL
    phi_V = calc_limit(V,G(1,:),N(1,:));
  flux_G_V= calc_muscl_new(V, 1.0/cur_R, G(1,:),phi_V, 1, 0.0);
  flux_N_V= calc_muscl_new(V, 1.0/cur_R, N(1,:),phi_V, 1, 0.0);
    for i = 1:Nx+1
      flux_G_V(i) = flux_G_V(i)*x_f(i)*x_f(i)*cur_R*cur_R;
      flux_N_V(i) = flux_N_V(i)*x_f(i)*x_f(i)*cur_R*cur_R;
    end


end


% Step 4: compute the fluxes _R
switch opt{6}
case 0 % Classical upwind FVM applied to csrv*
  flux_G_R(:) = calc_flux_upw( x_f, -dR/cur_R,...
  csrv_G(1,:), 1, 0.0 );
  flux_N_R(:) = calc_flux_upw( x_f, -dR/cur_R,...
  csrv_N(1,:), 1, 0.0 );

case 1 % Modified upwind
```

```
  flux_G_R(:) = calc_flux_upw( x_f, -dR/cur_R,...
  G(1,:), 1, 0.0 );
  flux_N_R(:) = calc_flux_upw( x_f, -dR/cur_R,...
  N(1,:), 1, 0.0 );

  for i = 1:Nx+1
    flux_G_R(i) = flux_G_R(i)*x_f(i)*x_f(i)*cur_R*cur_R;
    flux_N_R(i) = flux_N_R(i)*x_f(i)*x_f(i)*cur_R*cur_R;

  end
case 2 % PPM

  [flux_G_R, flux_N_R] = calc_flux_sync_ppm(dx,dR,...
  cur_R,x_c,x_f, G(1,:),N(1,:),opt{8},Nbc);
  case 3 %Modified MUSCL
      phi_R = calc_limit(x_f,G(1,:),N(1,:));
  flux_G_R(:) = calc_muscl_new( x_f, -dR/cur_R, G(1,:),...
  phi_R, 1, 0.0);
  flux_N_R(:) = calc_muscl_new( x_f, -dR/cur_R, N(1,:),...
  phi_R, 1, 0.0);

  for i = 1:Nx+1
    flux_G_R(i) = flux_G_R(i)*x_f(i)*x_f(i)*cur_R*cur_R;
    flux_N_R(i) = flux_N_R(i)*x_f(i)*x_f(i)*cur_R*cur_R;
  end

case 4 %MUSCL
  phi_R = calc_limit(x_f,G(1,:),N(1,:));
```

```
  flux_G_R(:) = calc_muscl_new( x_f, -dR/cur_R, ...
  csrv_G(1,:),phi_R, 1, 0.0);
  flux_N_R(:) = calc_muscl_new( x_f, -dR/cur_R,...
   csrv_N(1,:),phi_R, 1, 0.0);

end

% Step 5: compute the fluxes _U
switch opt{7}
case 0 % Classical upwind FVM applied to csrv_*
  flux_N_U(:) = calc_flux_upw( U, 1.0/cur_R, csrv_N(1,:),...
  0, Nbc*(x_c(Nx)+dx)*(x_c(Nx)+dx)*cur_R*cur_R );
case 1 % Modified upwind
  flux_N_U(:) = calc_flux_upw( U, 1.0/cur_R, N(1,:), 0, Nbc );
  for i = 1:Nx+1
    flux_N_U(i) = flux_N_U(i)*x_f(i)*x_f(i)*cur_R*cur_R;
  end

  case 2 % MUSCL
  phi_U = calc_limit(U,csrv_G(1,:),csrv_N(1,:));
  flux_N_U(:) = calc_muscl_new( U, 1.0/cur_R,csrv_N(1,:),...
  phi_U, 0,Nbc*(x_c(Nx)+dx)*(x_c(Nx)+dx)*cur_R*cur_R);

case 3  % mod MUSCL
phi_U = calc_limit(U,G(1,:),N(1,:));
[flux_N_U(:)] = calc_muscl_new( U, 1.0/cur_R,...
 N(1,:),phi_U, 0, Nbc);
  for i = 1:Nx+1
```

```
    flux_N_U(i) = flux_N_U(i)*x_f(i)*x_f(i)*cur_R*cur_R;
  end
end
% Step 6: update conservative variables
for i = 1:Nx %
  csrv_G(2,i) = csrv_G(1,i) + dt * ( x_c(i)*cur_R*x_c(i)*...
  cur_R*S_F(i) - x_c(i)*x_c(i)*dR*cur_R*G(1,i));
  csrv_G(2,i) = csrv_G(2,i) - (dt/dx) * ...
  (flux_G_V(i+1) - flux_G_V(i) + flux_G_R(i+1) - flux_G_R(i));

  csrv_N(2,i) = csrv_N(1,i) + dt * ( x_c(i)*cur_R*x_c(i)...
  *cur_R*S_G(i) - x_c(i)*x_c(i)*dR*cur_R*N(1,i) );
  csrv_N(2,i) = csrv_N(2,i) - (dt/dx) * ( flux_N_V(i+1) - ...
  flux_N_V(i) + flux_N_R(i+1) - flux_N_R(i) ...
  + flux_N_U(i+1) - flux_N_U(i) );
end

% Step 7: covnert to primitive G and N
for i = 1:Nx
  G(2,i) = csrv_G(2,i)/(x_c(i)*R(2,1))/(x_c(i)*R(2,1));
  N(2,i) = csrv_N(2,i)/(x_c(i)*R(2,1))/(x_c(i)*R(2,1));

end

t_cur = t_cur + dt;
% Update source terms, U, and V
S_F(:) = f(t_cur,x_c,R(2,1),G(2,:),N(2,:));
S_G(:) = g(t_cur,x_c,R(2,1),G(2,:),N(2,:));
```

```
    U(:)    = u(t_cur,x_f);
    V(:)    = calc_v(R(2,1),U,N(2,:),S_F,S_G,x_c,x_f,Nbc);
    dt      = calc_dt(dx,cfl,R(2,1),V,U,x_f,min(opt{4},T-t_cur));

    % Copy data
    R(1,1) = R(2,1);
    G(1,:) = G(2,:);
    N(1,:) = N(2,:);

    % Output
    itime = itime+1;
    time(itime)          = t_cur;
    sol_R(itime)         = R(1,1);
    sol_G(itime,1:Nx)    = G(1,:);
    sol_N(itime,1:Nx)    = N(1,:);
    sol_V(itime,1:Nx+1) = V(:);
  end
end
```

calc_dt.m

Next, for calculating the time step $\triangle t$, we use the code calc_dt.m file and updating the value for Velocity, we call the function calc_v.m

```
function dt = calc_dt(dx,cfl,R,V,U,Xf,dt_max)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute time step size
% Using
```

```
% dx:   Cell size
% cfl:  Courant number
% R:    Radius
% V:    Mean velocity 1xNx+1
% U:    Infiltration velocity 1xNx+1
% Xf:   Array of normalized cell faces
% dt_max: Maximum allowed dt
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  v_max = 0.0;
  Nx    = length(Xf)-1;

  for i = 2:Nx+1
    loc_v = V(i)/R;
    v_max = max(v_max,abs(loc_v));
    loc_v = Xf(i)*V(end)/R;
    v_max = max(v_max,abs(loc_v));
    loc_v = U(i)/R;
    v_max = max(v_max,abs(loc_v));
  end

  if v_max == 0.0
    dt = dt_max;
  else
    dt = min(cfl * dx / v_max, dt_max);
  end

end
```

```
function V = calc_v(R, U, N, S_F, S_G, Xc, Xf, NBC)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute array V of size 1xNx+1
% Using
% R:    Radius
% U:    Infiltration velocity 1xNx+1
% N:    Array of N: 1xNx
% S_F: Array of source f: 1xNx
% S_G: Array of source g: 1xNx
% Xc:   Array of normalized cell centers
% Xf:   Array of normalized cell faces
% NBC: Boundary value for N
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  Nx = length(Xc);
  V  = zeros(1,Nx+1);

  % Step one:
  for i = 1:Nx
    V(i+1) = V(i) + Xc(i)*Xc(i)*(S_F(i)+S_G(i));
  end

  % Step two:
  for i = 2:Nx+1
    if i<Nx+1
      loc_N = 0.5*(N(i-1)+N(i));
    else
      loc_N = NBC;%0.5*(N(i-1)+NBC);
```

```
    end
    V(i) = R*((V(i)/Xf(i)/...
      Xf(i))-U(i)*loc_N/R);
  end

end
```

Now for updating the solution for G and N, we call the functions calc_flux_upw.m when we calculate flux by first order upwind method.

calc_flux_upw.m

```
function flux = calc_flux_upw( V, ooR, VAL, FIXBC, BCVAL )
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute the upwind flux according to velocity
% V/R
% V:    Nodal velocity array 1xNx+1
% ooR: One over Radius
% VAL: Cell value array 1xNx
% FIXBC: Whether to fix bc flux (1:yes) (0:no)
% BCVAL: FIXBC=1 -- fixed bc flux
%          FIXBC=0 -- boundary condition
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Nx   = size(V,2)-1;
flux = zeros(1,Nx+1);
flux(1) = 0.0;
```

```
for i = 2:Nx
  loc_vel = V(i)*ooR;
  if loc_vel >= 0.0
    flux(i) = loc_vel* VAL(i-1);
  else
    flux(i) = loc_vel* VAL(i);
  end
end

% BC
if FIXBC == 1
  flux(Nx+1) = BCVAL;
end
if FIXBC == 0
  loc_vel = V(Nx+1)*ooR;
  if loc_vel >= 0.0
    flux(Nx+1) = loc_vel* VAL(Nx);
  else
    flux(Nx+1) = loc_vel* BCVAL;
  end
end


end
```

When we calculate flux by second order MUSCL method, we use the MATLAB function, calc_muscl_new.m. To use MUSCL method, we need to calculate the limiters first. The function for calculating limiters is given below:

calc_limit

```
function phi = calc_limit(V,G,N)

Nx  = size(V,2)-1;
 G(Nx+1)= 0;
 G(Nx+2)= 0;
 N(Nx+1)= 0;
 N(Nx+2)= 0;

sum = G+N;
    phi_G   = zeros(1,Nx+1);
    phi_N   = zeros(1,Nx+1);
    phi_tht = zeros(1,Nx+1);
    phi_t   = zeros(1,Nx+1);
    phi     = zeros(1,Nx+1);

    for i = 2:Nx+1

     %%%%Limiter_G calculation
     if ((G(i) - G(i-1))*(G(i+1)-G(i))) <= 0
     %%Calculating phi_G(i)
      phi_G(i) = 0.0;
     else
      phi_G(i) = min((G(i) - G(i-1))/...
      (G(i+1)-G(i)), 1 );
     end
    end
```

```
for i = 2:Nx+1

   %%%%Limiter_N calculation
   if ((N(i) - N(i-1))*(N(i+1)-N(i))) <= 0
   %%Calculating phi_N(i)
    phi_N(i) = 0.0;
   else
    phi_N(i) = min((N(i) - N(i-1))/...
     (N(i+1)-N(i)), 1 );
   end
end
for i = 2:Nx+1

   %%%%Limiter_Theta calculation
   if ((sum(i) - sum(i-1))*(sum(i+1)...
   -sum(i))) <= 0
    %%Calculating phi_tht(i)
    phi_tht(i) = 0.0;
   else
    phi_tht(i) = min((sum(i) - sum(i-1))/...
     (sum(i+1)-sum(i)), 1 );
   end

end

for i = 2:Nx+1
```

```
        phi_t(i) = min(phi_N(i),phi_tht(i));
        phi(i) = min(phi_G(i),phi_t(i))  ;

    end

end
```

After calculating the limiters, we call the function calc_muscl_new.m and pass value of the limiters too.

calc_muscl_new

```
function [flux] = calc_muscl_new(V, ooR, VAL, phi, FIXBC, BCVAL)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute the MUSCL flux according to velocity
% V/R
% V:    Nodal velocity array 1xNx+1
% ooR: One over Radius
% VAL: Cell value array 1xNx
% FIXBC: Whether to fix bc flux (1:yes) (0:no)
% BCVAL: FIXBC=1 -- fixed bc flux
%         FIXBC=0 -- boundary condition
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Nx = length(VAL);
new_val = zeros(1,Nx+2);
VAL(Nx+1)= 0;
VAL(Nx+2)= 0;
for i = 2:Nx
```

```
    new_val(i) = VAL(i) + 0.5 * phi(i) * ...
    (VAL(i+1) - VAL(i));
    new_val(i+1) = VAL(i+1) - 0.5 *...
    phi(i+1) * (VAL(i+2) - VAL(i+1));
    flux = calc_flux_upw(V,ooR,...
     new_val,FIXBC,BCVAL);
```

end

end

Now, for improving the flux value due to $\eta R^{prime}/R$, we use piecewise-parabolic-method or PPM. For that, we used the function calc_flux_sync_ppm.m

calc_flux_sync_ppm.m

```
function [flux_G, flux_N] = calc_flux_sync_ppm(dx,dR,...
    R,Xc,Xf,G,N,TOL,NBC)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute the synchronized PPM fluxes
% dR:  R'
% R:   Radius
% Xc:  Array of cell centers: 1xNx
% Xf:  Array of cell faces: 1xNx+1
% G:   Array of G: 1xNx
% N:   Array of N: 1xNx
% TOL: Tolerance
% NBC: Boundary condition for N
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  Nx = length(Xc);
```

```
flux_G = zeros(1,Nx+1);
flux_N = zeros(1,Nx+1);

% Original bar version
GBar = G.*Xc.*Xc.*Xc;
NBar = N.*Xc.*Xc.*Xc;
TBar = GBar+NBar;
% Constant extrapolation at the right end (ghost values)
% May change to linear extrapolation or quadratic extrapolation
GBar(Nx+1) = (1.0-NBC)*(Xc...
(end)+dx)^3;
GBar(Nx+2) = (1.0-NBC)*(Xc...
(end)+2*dx)^3;
NBar(Nx+1) = NBC*(Xc...
(end)+dx)^3;
NBar(Nx+2) = NBC*(Xc...
(end)+2*dx)^3;
TBar(Nx+1) = (Xc(end)+dx)^3;
TBar(Nx+2) = (Xc(end)+2*dx)^3;

hGBar = zeros(1,Nx+1);
hNBar = zeros(1,Nx+1);
hTBar = zeros(1,Nx+1);

sot = 7.0/12.0;
oot = 1.0/12.0;

hGBar(2) = sot*(GBar(1)+GBar(2))...
```

```
- oot*(GBar(3)-GBar(1));
hNBar(2) = sot*(NBar(1)+NBar(2))...
- oot*(NBar(3)-NBar(1));
hTBar(2) = sot*(TBar(1)+TBar(2))...
- oot*(TBar(3)-TBar(1));

for i = 3 : Nx+1
  hGBar(i) = sot*(GBar(i-1)+GBar(i))...
  - oot*(GBar(i+1)+GBar(i-2));
  hNBar(i) = sot*(NBar(i-1)+NBar(i))...
  - oot*(NBar(i+1)+NBar(i-2));
  hTBar(i) = sot*(TBar(i-1)+TBar(i))...
  - oot*(TBar(i+1)+TBar(i-2));
end

%% Limiters (1,:) is - version, (2,:) is + version
phi = zeros(2,Nx);
% Limiter in the first cell (only need phi(2,1))
if abs(NBar(1))<=TOL && ...
abs(hNBar(2))<=TOL
  phi_n = 1.0;
elseif NBar(1)*hNBar(2)<=0.0 ||...
 3*abs(hNBar(2))<=8*abs(NBar(1))
  phi_n = 0.0;
else
  phi_n = min(1, 5.0/(hNBar(2)...
  /NBar(1)-1.0));
end
```

```
if abs(GBar(1))<=TOL &&...
        abs(hGBar(2))<=TOL
  phi_g = 1.0;
elseif GBar(1)*hGBar(2)<=0.0 ...
|| 3*abs(hGBar(2))<=8*abs(GBar(1))
  phi_g = 0.0;
else
  phi_g = min(1, 5.0/(hGBar(2)/...
GBar(1)-1.0));
end

if abs(TBar(1))<=TOL &&...
 abs(hTBar(2))<=TOL
  phi_t = 1.0;
elseif TBar(1)*hTBar(2)<=0.0 ||...
 3*abs(hTBar(2))<=8*abs(TBar(1))
  phi_t = 0.0;
else
  phi_t = min(1, 5.0/(hTBar(2)/...
  TBar(1)-1.0));
end

phi(2,1) = min([phi_n phi_g phi_t]);

% Limiters away from x=0
for i = 2:Nx
  diffNp = hNBar(i+1)-NBar(i);
```

```
diffNm = hNBar(i)   -NBar(i);
diffGp = hGBar(i+1)-GBar(i);
diffGm = hGBar(i)   -GBar(i);
diffTp = hTBar(i+1)-TBar(i);
diffTm = hTBar(i)   -TBar(i);
if diffNp*diffNm>=0.0 ||...
diffGp*diffGm>=0.0 || diffTp*diffTm>=0.0
  phi(1,i) = 0.0;
  phi(2,i) = 0.0;
else
  alpha1 = 2.0*min([abs(diffNm/diffNp)...
  abs(diffGm/diffGp) abs(diffTm/diffTp)]);
  alpha2 = 0.5*max([abs(diffNm/diffNp)...
  abs(diffGm/diffGp) abs(diffTm/diffTp)]);
  if alpha2 > alpha1 + TOL
    phi(1,i) = 0.0;
    phi(2,i) = 0.0;
  elseif alpha1<1.0
    phi(1,i) = 1.0;
    phi(2,i) = alpha1;
  elseif alpha2>1.0
    phi(1,i) = 1.0/alpha2;
    phi(2,i) = 1.0;
  else
    phi(1,i) = 1.0;
    phi(2,i) = 1.0;
  end
end
```

```
  end

  flux_G(1) = 0.0;
  flux_N(1) = 0.0;
  loc_vel   = -dR*R;
  if loc_vel >= 0.0
    for i = 2:Nx
      flux_G(i) = loc_vel * (GBar(i-1)...
      +phi(2,i-1)*(hGBar(i)-GBar(i-1)));
      flux_N(i) = loc_vel * (NBar(i-1)...
      +phi(2,i-1)*(hNBar(i)-NBar(i-1)));
    end
  else
    for i = 2:Nx
      flux_G(i) = loc_vel * (GBar(i)...
      +1.0*(hGBar(i)-GBar(i)));
      flux_N(i) = loc_vel * (NBar(i)...
      +1.0*(hNBar(i)-NBar(i)));
    end
  end
  flux_G(Nx+1) = 0.0;
  flux_N(Nx+1) = 0.0;

end
```

Curriculum Vitae

Mashriq Ahmed Saleh, the youngest son of Saleh Ahmed Khan, was born on July 01, 1989 in Bangladesh. He completed his Bachelor in Electrical and Electronic Engineering from North South University. His undergraduate research was "Stability of $\mathcal{PT}$-symmetric Behavior and Normal Coupling with Optical Waveguides". After graduation, he worked in the telecommunication industry for around 4 years. He has been a Graduate Research Assistant since Jan, 2017 at University of Texas at El Paso.

Mashriq's research is focused on Mathematical Modeling and Numerical simulation. He is working on developing a mathematical modeling of tumor growth for free-boundary problem.