Dissertation

submitted
to the

Combined Faculty for the Natural Sciences and Mathematics

of the

Ruperto-Carola University of Heidelberg

for the degree of

Doctor of Natural Sciences

Put forward by

M.E., Fei Xing
Born in: China

Date of oral examination: .....

Towards Agent-based Multi-scale Tumor Growth Modeling: Software Environment and Computational Complexity

Advisors: Prof. Dr. Peter Bastian
Prof. Dr. Dieter W. Heermann

Abstract

Understanding tumor development crossing multiple spatial-temporal scales is of great practical importance to better fighting against cancers. It is hard to attack this problem with pure biological means. In recent decades, computer-based modeling and simulation techniques have been playing an increasingly important role in addressing it. After reviewing the literature, however, we notice that existing tumor models are either highly simplified or too complicated to be scaled to large tumor systems.

In light of these problems, we have developed a software environment TUGME to facilitate the multi-scale modeling and simulation of tumor development based on the agent-based method. The most important feature of this software environment is its flexibility which enables straight-forward model reuse and extension. Tumor models of TUGME are hybrid as discrete and continuous approaches are coupled to model the discrete and continuous nature of the tumor system. TUGME is highly modularized, thus changing one module only requires few or no modifications to the others.

Using TUGME, we have simulated the avascular growth of a multicellular tumor spheroid system of the tumor cell line, EMT6/Ro. Our tumor models treat individual tumor cells as single agents. The cell morphology and topology are represented by a 3D Voronoi tessellation. Cell motion, which is driven by mechanical interactions between a cell and its surroundings, is modeled using Newton's second law. Oxygen and glucose are treated as nutrients for cell energy production. Their transport and metabolism by cells are described by reaction-diffusion equations. Cell proliferation is defined considering the availability of both oxygen and glucose as well as the availability of space as its controllers. Based on these models, a series of simulations have been carried out. Good agreements between our simulations and experiments indicate the applicability of TUGME and the validity of our tumor models. In addition, the investigation of the invasive tumor morphology under different nutrient conditions shows that a lower nutrient concentration gives rise to a rougher tumor surface.

One of the key challenges of agent-based multi-scale cancer modeling and simulation is the sharp increase of the computational cost of model solving with increasing system size (the number of tumor cells). According to our tests, the main computational bottleneck of our tumor models consists in solving the linear system of cell motion. To better understand this problem, we look into the properties of the matrix of the linear system. Our conclusion is that its matrix is extremely sparse, symmetric and positive-definite. These properties can help find a more efficient solver for the linear system. This work can be important reference for people who intend to work on individual-cell-oriented cancer modeling.

"A journey of thousands of miles may not be achieved through accumulation of each single step, just as the enormous ocean may not be formed gathering every brook or stream."

XUN Kuang

Zusammenfassung

Das Verständnis der Entwicklung von Tumoren auf verschiedenen räumlich-zeitlichen Ebenen ist von herausragender praktischer Bedeutung für die Bekämpfung von Krebs. Es ist schwierig, dieses Problem alleine mit biologischen Mitteln in Angriff zu nehmen. In den letzten Jahrzehnten haben computergestützte Modellierung und Simulationstechniken eine zunehmend wichtige Rolle dabei gespielt, das Problem anzugehen. Nach Sichtung der Literatur haben wir allerdings festgestellt, dass bereits vorhandene Tumormodelle entweder in hohem Maße vereinfacht oder andererseits zu kompliziert sind, um auf große Tumorsysteme skaliert werden zu können.

Im Lichte dieser Probleme haben wir eine Software-Umgebung, TUGME, entwickelt, die die Multiskalen-Modellierung und die agentenbasierte Simulation der Tumorentwicklung erleichtert. Das wichtigste Merkmal dieser Software-Umgebung ist ihre Flexibilität, die die einfache Wiederverwendbarkeit und Erweiterung der Tumormodelle ermöglicht. Die TUGME zugrunde liegenden Tumormodelle sind hybrid, da diskrete und kontinuierliche Methoden gekoppelt wurden, um die sowohl diskrete als auch kontinuierliche Natur von Tumorsystemen zu modellieren. TUGME ist in hohem Grade modularisiert, sodass Änderungen an einem Modul nur wenige bis keine Modifikationen an den anderen erfordern.

Unter Verwendung von TUGME haben wir das avaskuläre Wachstum eines mehrzelligen Tumorsystems der Tumorzelllinie EMT6/Ro simuliert. In unseren Tumormodellen betrachten wir einzelne Tumorzellen als Agenten. Die Zellmorphologie und Zelltopologie werden durch eine dreidimensionale Voronoi-Tessellation abgebildet. Bei der Modellierung der Zellbewegung, die durch mechanische Interaktionen zwischen einer Zelle und ihrer Umgebung bestimmt ist, wird vom zweiten Newtonschen Gesetz ausgegangen. Sauerstoff und Glukose werden als Nährstoffe für die Zellenergieerzeugung betrachtet. Ihr Transport und Metabolismus durch Zellen werden durch Reaktionsdiffusionsgleichungen beschrieben. Die Zellproliferation wird durch die Verfügbarkeit von Sauerstoff, Glukose sowie ausreichend Platz bestimmt. Auf diesen Modellen basierend wurde eine Reihe von Simulationen durchgeführt. Übereinstimmungen zwischen unseren Simulationen und Experimenten belegen die Anwendbarkeit von TUGME und die Gültigkeit unserer Tumormodelle. Darüber hinaus zeigt die Untersuchung der invasiven Tumor-Morphologie unter verschiedenen Nährstoffbedingungen, dass eine niedrigere Nährstoffkonzentrationen eine rauere Tumoroberfläche verursacht.

Eine Schlüsselherausforderung der agentbasierten Multiskalen-Simulation von Krebs sind die mit zunehmender Systemgröße (Anzahl an Tumorzellen) einhergenden extrem

steigenden Rechenkosten. Der ausschlaggebende Engpass in den numerischen Berechnungen besteht unseren Tests zufolge darin, das lineare Gleichungssystem der Zellbewegung zu lösen. Um dieses Problem besser zu verstehen, untersuchen wir die Eigenschaften der Matrix des linearen Gleichungssystem. Unsere Schlussfolgerung ist, dass diese Matrix extrem dünn besetzt, symmetrisch und positiv-definit ist. Diese Eigenschaften können dabei helfen, einen leistungsfähigeren Löser für das lineare System zu finden. Diese Arbeit stellt eine wichtige Referenz für Forscher dar, die beabsichtigen, im Bereich der Modellierung von Krebs auf der Skala einzelner Zellen zu arbeiten.

Acknowledgements

I would like to express my deepest appreciation to my thesis advisor Prof. Dieter W. Heermann. His continuous support and encouragement have helped me overcome numerous difficulties. Without his guidance, this dissertation would not have been possible.

I would like to give my sincere thanks to my thesis advisor Prof. Peter Bastian whose group have been working on developing DUNE, with which I have saved a lot of programming efforts in solving both of the reaction-diffusion problems and the linear systems. Besides, discussing with him has always been a good chance for me to learn new things.

Thanks to Prof. Jürgen Hesser at the Department of Radiation Oncology, University Medical Center Mannheim and Heidelberg University. Discussing with him and the researchers at his group has inspired me many new ideas. His comments on our publications have been of great value.

Many thanks to Dr. Felix Heimann and Adrian Ngo at Prof. Peter Bastian's group for their help with DUNE.

Special thanks to Prof. Yiping Yao for his support and advice to my PhD study.

Special thanks to Prof. Chris H. Rycroft at the School of Engineering and Applied Sciences, Harvard University. His help with Voro++ is highly appreciated.

Thanks to the scientific director of the Institute for Theoretical Physics (ITP), Dr. Eduard Thommes at Heidelberg University. And thanks to all the secretaries at ITP, particularly Melanie Steiert, Sonja Bartsch, Cornelia Merkel, Franziska Binder, Tina Birch and Manuela Wirschke. Thank you all very much for the help during my study. Thanks to Jörg Eisele who has contributed to some basic ideas and the programming implementations of this project. Special thanks to Dr. Gabriell Mate for proofreading part of the theis. Special thanks to Andreas Hofmann for his help with the German version of the abstract of this thesis. Thanks to group members, Dr. Songling Li, Dr. Yang Zhang, Dr. Lei Liu, Wei Xiong, Min Chu. It is my pleasure to study, do research and discuss with them. Thanks to Dr. Bing Wang, Dr. Zongqian Hu, Feng Zhu and Huilong Chen for their help during my PhD study.

Thanks to the China Scholarship Council (CSC) for the financial support to my study.

Thanks to my parents, sisters, brothers, girlfriend and her parents for their persistent support to my study.

To my parents.

Contents

Abstract i

Zusammenfassung iii

Acknowledgements v

Contents vii

List of Figures x

List of Tables xvi

Abbreviations xvii

1 Aim and Structure of the Thesis 1
- 1.1 Motivation . . . 1
- 1.2 Structure of the Thesis . . . 3

2 Cancer Modeling 5
- 2.1 Physical Models of Cancer . . . 5
- 2.2 Computational Cancer Models . . . 8
  - 2.2.1 Brief Introduction to Computer-based Modeling and Simulation . . . 8
  - 2.2.2 Benefits of Modeling and Simulation . . . 9
  - 2.2.3 Tumor Models and Cell Representation . . . 11
  - 2.2.4 The Agent-based Method and Software Tools . . . 12

3 Introduction to Cancer Development 16
- 3.1 Tumors and Cancer . . . 16
- 3.2 Multi-stages of Tumorigenesis . . . 17
- 3.3 Cancer Biology . . . 19
  - 3.3.1 The Randomness of Cancer Incidence . . . 19
  - 3.3.2 The Cancer Stem Cell Theory . . . 21
  - 3.3.3 Genotypes and Phenotypes of Cancer Cells . . . 22
  - 3.3.4 The Aberrant Cancer Cell Proliferation . . . 24
  - 3.3.5 Cancer Cell Metabolism . . . 26
  - 3.3.6 Cancer Cell Biophysics and Biomechanics . . . 28

4 The Design and Implementation of TUGME 31
- 4.1 General Perspectives of TUGME . . . 31

4.2 Basic Design Principles of TUGME . . . 32
4.3 Modules of TUGME . . . 33
4.3.1 Brief Introduction to DUNE . . . 33
4.3.2 Cell Agents: Cell Types and States . . . 35
4.3.3 Cell Morphology and Topology: Cell Shape Deformation and Neighborhood . . . 42
4.3.3.1 Introduction to Voronoi Tessellations . . . 42
4.3.3.2 Calculating the Radical Voronoi Tessellations in Three Dimensions . . . 44
4.3.4 Cell Mechanics: Cellular Mechanical Interactions and Motion . . . 47
4.3.4.1 Force Types . . . 47
4.3.4.2 Force Models . . . 48
4.3.4.3 Solving the Linear System of Cell Motion Using DUNE 53
4.3.5 Nutrients and Metabolic Products: the Transport and Metabolism of Biochemicals . . . 53
4.3.5.1 The RDE Model . . . 53
4.3.5.2 Solving the RDEs Using DUNE . . . 53
4.3.5.3 Up- and Down-scaling of Biochemicals . . . 56
4.3.6 Cell Cycle: Cell Proliferation and Death . . . 60
4.3.7 Interdependences Between Modules of TUGME . . . 64
4.4 Conducting Simulations Using TUGME . . . 66
4.4.1 Initialization of Simulations . . . 66
4.4.2 Control of Simulations . . . 67
4.4.3 Data Output and Visualization . . . 69

5 Case Study: Modeling the Growth of the EMT6/Ro MTSs 71
5.1 Introduction . . . 71
5.2 The Model . . . 72
5.3 Parameter Settings . . . 73
5.4 Results and Discussion . . . 76
5.4.1 Population Growth . . . 76
5.4.2 Distribution of Nutrients . . . 77
5.4.3 Invasive Morphology . . . 77
5.4.4 Discussion . . . 79
5.5 Conclusion . . . 83

6 Computational Complexity of Agent-based Multi-scale Cancer Modeling 84
6.1 Introduction . . . 84
6.1.1 Multi-scale Complexity of Tumor Growth . . . 84
6.1.2 Simulation Time and Wall-clock Time . . . 85
6.2 The Model . . . 86
6.3 Evaluating the Computational Cost of Model Solving . . . 86
6.3.1 Calculating the 3D Radical Voronoi Tessellation . . . 87
6.3.2 Solving the Linear System of Cell Motion . . . 87
6.3.3 Solving RDEs Using FEM . . . 93
6.3.4 Performing Cell Proliferation . . . 95

6.3.5 Computational Bottlenecks 95
6.4 Summary 96

7 Summary and Future Work **97**
7.1 Summary 97
7.1.1 Development of a Tumor Growth Modeling and Simulation Software Environment 97
7.1.2 Modeling the Growth of the EMT6/Ro MTSs 98
7.1.3 Evaluating the Computational Cost of Agent-based Multi-scale Tumor Modeling 98
7.2 Future Work 99
7.2.1 Testing Different Hypotheses About Cell Dynamics 99
7.2.2 Modeling Dynamics of Advanced Tumors 99
7.2.3 Modeling Drug Delivery 99
7.2.4 Extending TUGME for Parallel Simulations 100

A Derivation of the Linear System of Cell Motion **101**

B Positive-definiteness of the Matrix of the Linear System **105**

C Brief Introduction to Finite Element Method **108**

D 3D Radical Voronoi Cells Versus Realistic Tumor Cells **113**

E Data Representation in TUGME **115**

Bibliography **117**

Selected Publications **134**

List of Figures

2.1 Diagram illustrating the relationship between dry-lab and wet-lab experiments. On the one hand, models should be built based on the basic knowledge (theories) of the real system, which is abstracting its traits of relevance. On the other hand, models are used to test hypotheses and more importantly to make predictions on the modeled system. By comparing simulation results with experimental data, models get validated, meanwhile, the knowledge about the studied system gets accumulated. Image adapted from Ref. [1]. 9

2.2 Diagram illustrating agent-agent and agent-environment interactions. . . 13

3.1 Liver metastatic colon carcinomas (white). Colon carcinoma cells colonize the liver and trigger several secondary tumors in it. Image adapted from Ref. [2] 17

3.2 Schematic illustration of the spatial-temporal scales involved in tumorigenesis. Image adapted from Ref. [3] 19

3.3 Some regulators (cytokines) closely relevant to tumorigenesis. Image adapted from Ref. [4] 23

3.4 Schematic illustration of the general steps of a cell signaling pathway. Particularly, decision making is basically to determine which genes are to be expressed. 24

3.5 Diagram illustrating the phases of an entire cell division cycle. $G1$, S, $G2$ and M separately represent the gap-1, the synthesis, the gap-2 and the mitosis phases. $G0$ stands for the state that cell cycle is suspended. Healthy cells temporarily rest in the $G0$ state and can be waken up to proliferate again when it is necessary. Tumor cells keep this ability, however, $G0$ state tumor cells can undergo a special death programme, tumor necrosis, for some reasons not fully understood. Necrotic tumor cells are dead, hence, they can't resume cell cycle anymore. 25

3.6 Diagram illustrating the phases of an entire cell cycle and the corresponding checkpoints. Checkpoints impose cell cycle control to guarantee that a cell has correctly completed all the requisite steps of one phase before entering the next one. 26

3.7 A tumor growth curve replenishing several tumor cell lines. The red diamond and the green triangle represent prostate and breast tumors from patients. Image adapted from Ref. [5]. 27

3.8 Diagram illustrating the three-layer structure of solid tumors consisting of the necrotic core (N) of dead cells, the quiescent rim (Q) of resting cells and the proliferating front layer (P) of actively growing and dividing cells. The outermost layer represents the healthy tissue. For avascular tumors, nutrients can be merely transported by molecular diffusion from the healthy tissue. So does the tumor cell metabolic products which are transported from the interior to the outer space of tumors. 27
3.9 Metabolism of glucose in normal and cancer cells: aerobic versus anaerobic respiration. (A) With adequate oxygen, glucose is broken down into carbon dioxide in healthy cells via aerobic respiration, generating as many as 36 ATP molecules per glucose molecule. (B) In cancer cells and healthy cells short of oxygen, glucose is broken down into the lactate via anaerobic respiration, generating 2 ATP molecules per glucose molecule. Image adapted from Ref. [2] 29

4.1 The hierarchical structure of the C++ classes of agent models of single cells. All cell agent classes in TUGME have to inherit from the basic class `cell` directly (e.g., `voroCell` and `xCell`) or indirectly (e.g., `voroStemCell`). Although C++ allows multiple inheritance, for example, `zCell` in this figure, this isn't allowed in TUGME, since it can be seen clearly from this figure that `zCell` inherits `cell` repeatedly from both `voroCell` and `xCell`. `CSCellProperties` defines the CSC features. It is separated from `voroStemCell` for the possibility that one needs to model CSC properties without considering the morphology and topology of tumor cells. `xCell`, `yCell` and `zCell` don't exist actually, they are just general representations of possible cell agent classes that may be implemented in the future. 36
4.2 The shape similarity between 3D radical Voronoi cells and human kidney epithelial cells. The figure on the left is a section cutting through a 3D radical Voronoi tessellation, where the convex polygon of green edges is a Voronoi cell. The figure on the right is an image of Madin-Darby canine kidney (MDCK) epithelial tissue cells obtained with confocal microscopy adopted from reference [6]. The MDCK epithelial tissue cell shows the polyhedron-like shape. 37
4.3 The Voronoi cell agent class `voroCell`. 37
4.4 Diagram illustrating the basic idea of avoiding element shift when removing an element from a C++ `vector` by element replacement. 39
4.5 Pseudo-code showing the algorithm of adding and deleting elements from the C++ `vector` storing the single cell agents. V_1 and V_2 stores the agents of existing and newly generated cells separately. 40

4.6 The performance test of three different C++ standard data structures, namely the `vector`, the `list` and the `map`. Two operations are tested, which are sequential inserting and random deleting. For the `vector` and the `list`, inserting is done only to the end in stead of to a random position, since the stored position of cell agent object in `vector` doesn't matter in TUGME. Theoretically, the cost of inserting an element to any position of the `list` would be no difference, however, it varies significantly for the `vector` ($O(1)$ versus $O(n)$ averagely, where n is the size of the `vector`). The upper left panel shows the CPU time to add certain numbers, 10^3, 10^4, 10^5 and 10^6, of elements to the `vector`, the `list` and the `map`. The rest three panels (the upper right, the lower left, and the lower right) show the CPU time of randomly deleting different numbers, 0.1%, 1% and 5%, of elements from different size, 10^3, 10^4, 10^5 and 10^6, of these data structures. `vector_1` and `vector_2` separately represent our optimized deleting method and the standard C++ deleting mechanism. 41

4.7 Image illustrating a 2D common Voronoi tessellation. White dots stand for the position of the Voronoi cell (the spherical center of tumor spheres in three dimensions). Convex polygons bounded by green line segments represent Voronoi cells (convex polyhedra in three dimensions). The green line segment (edges of polygons) stands for the contact interface between two directly neighboring cells. In three dimensions, these contact interfaces are replaced by convex polygonal areas. A 3D radical Voronoi tessellation with two cells can be seen in figure D.1 in appendix E of this thesis. 42

4.8 Diagrams comparatively illustrating a single polygonal Voronoi cell of the common and radical Voronoi tessellation in two dimensions. The green areas bounded by solid red line segments represent the Voronoi cell and the dashed yellow line segments indicate the neighborhood between cells. Blue circles indicate the default spherical periphery of single cells. For the common Voronoi cell (on the left), the solid red line segments and its corresponding dashed yellow line segments equally bisect each other perpendicularly. In contrast, for the radical Voronoi cells (on the right), solid red line segments have been shifted towards the smaller radius cells though they are still perpendicular to the dashed yellow line segments and equally bisected by them. 43

4.9 Illustration of 3D radical Voronoi tessellation with non-periodic (the leftmost panel), periodic (the middle panel) boundaries and Voronoi cells of dodecahedral shapes (the rightmost panel). 45

4.10 The `voroPP` interface. This class wraps the Voro++ source code. The data structure types for storing the Voronoi information are declared in a traits named `voronoiTratis` (not shown here). `wall_initial_shape` is an inner class of `voroPP`. It defines the default shape (a dodecahedron) for individual radical Voronoi cells. 46

4.11 Diagram illustrating the overlap h_{ij} between two contact spheroids where R_i and R_j are the radii of the two spheroids separately. d_{ij} measures the distance between cells. 49

4.12 An approximated JKR force curve versus the virtual overlap h_{ij} between two neighboring cells. The portion of the curve above the light gray line indicates a repulsive effect, while the below counterpart indicates the adhesion effect. The idea is that contact cells tend to attract each other when they are in contact, meanwhile, they also push each other when they undergo shape deformation. For small overlaps, the adhesion (viscosity) force dominates, while with the decrease of cell-cell distance (h_{ij} increases). Cell elastic repulsion force sharply increases with the increment of cell shape deformation. The right cross point between the force curve and the x-axis indicates a zero JKR force, which is considered to be a result of balancing between the adhesion and the repulsion forces, which can be thought as a preferred distance between cells within tissues. Larger than this h_{ij}, the repulsion force starts to dominate. 50
4.13 Diagram illustrating the interface for modeling the mechanical forces of single tumor cells. 52
4.14 Diagram illustrating the relationship between the C++ classes of RDE models in TUGME. 55
4.15 A snapshot of part of the TUGME source code for evaluating the concentration of a biochemical molecule at a give position. 58
4.16 Diagram illustrating the up- and down-scaling of biochemical molecules. 59
4.17 Illustration of the piecewise characteristic of the reaction function $\overline{r_i}(x,t)$ averaged over space and time. 59
4.18 Diagram illustrating the non-linear interaction between tumor cells and nutrients. Most cell life behaviors need the support of energy, which is generated by biochemical reactions consuming nutrients, such as oxygen and glucose. Cells uptake nutrients from the extracellular environment and release metabolic products back. If molecular diffusion can not compensate the loss of consumption of nutrients, the nutrient concentration will continuously decline. Tumor cells suspend cell cycle and enter a so-called quiescent state (tumor cell quiescence) for short of energy when the nutrient concentration below certain thresholds. Doing so, tumor cells may survive for its nutrient consumption rate reduces in the quiescent state. If the nutrient concentration gets even lower, tumor cells die (tumor cell necrosis, which is energy-independent) for energy depletion. . 60
4.19 Diagram illustrating the interface and several implementations of cell cycle models. The symbol "#" indicates a protected field of a C++ class. 62
4.20 Flowchart illustrating the phase transitions of an entire cell cycle. 63
4.21 Flowchart illustrating the state transitions in the $G1$ phase. 64
4.22 Flowchart illustrating the state transitions in the $G0$ state. 64
4.23 Diagram illustrating the relative position of two newly generated daughter cells relative to their mother cell. 65
4.24 Diagram illustrating the interdependences between modules of TUGME. The dashed gray arrows represent the theoretical data flowing directions, while the solid orange arrows stand for the actual ones. 65
4.25 A snapshot of part of a configuration file. 66
4.26 A snapshot of part of an initial file. 67
4.27 Flowchart illustrating the main work flow of simulations using TUGME. 68

4.28 Visualization of the radical Voronoi polyhedra and the distribution of biochemical molecules using Paraview (above) and Pov-Ray (below) in three dimensions. Blue line segments in this figure are the edges of the radical Voronoi polyhedra. Colors of the above right panel indicate the concentration gradient (from red (high) to blue (low)) of biochemical molecules within simulation domain. Three panels below from left to right are tumor cell spheres, corresponding radical Voronoi polyhedra and the merger of them. 70

5.1 Flowchart illustrating the state transitions of an entire cell cycle considering oxygen, glucose and cell volume as controllers. A Tumor cell becomes necrotic if one of the concentrations of oxygen and glucose drops down to the defined necrotic thresholds. Similar checks on cell quiescence are done after the necrotic conditions have been passed (necrotic conditions are not met). If the quiescent requirements on the concentrations of both oxygen and glucose are satisfied (quiescent conditions are not met either), the volume of the cell is checked to guarantee that it can proceed its cell cycle only when it has actually obtained enough volume for its division. Once passing this final checkpoint, it can definitely finish the rest cell cycle phases and finally divides into two daughter cells which immediately start the next cell cycle from the $G1$ phase. Since CSC features are taken into account. The daughter cells may be a CSC or a common tumor cell, which depends on the type of the mother cell. After using up the proliferative potential, a common tumor cell executes apoptosis. 74

5.2 The cell population growth curves of tumors under different conditions of oxygen and glucose concentrations, namely 0.28 and 16.5 (case a), 0.07 and 16.5 (case b), 0.28 and 0.8 (case c), 0.07 and 0.8 (case d) (the unit is $mmol \cdot L^{-1}$). The dashed red lines are the simulation results. The solid blue diamonds and green circles are the corresponding experimental results taken from Ref. [7]. 76

5.3 Molecular distributions of glucose and oxygen during simulations. Four lines from the above to the bottom (solid blue, dashed green, dashed red, and dashed cyan) in the 8 panels of this figure represent the concentrations of oxygen and glucose on day 0, 9, 16 and 23 separately. 78

5.4 Illustration of the growth morphology of the simulated MTSs under different conditions of oxygen and glucose. Cases (a), (b), (c), (d) from the top to the bottom represent the four conditions of oxygen and glucose introduced in figure 5.2. The light gray region stands for the MTS necrotic core, which is surrounded by the dark brown rim of quiescent cells. The green layer consists of actively proliferative cells. 79

5.5 Comparison the MTS surface roughness under different nutrient conditions. Panels (a), (b), (c) and (d) represent four simulated cases. The dashed red lines are the simulation results averaged over 10 simulation runs under the same condition but different number of initial tumor cells. The solid blue lines are the corresponding curves fitted using the hyperbolic tangent formula $A \cdot tanh(x/\delta + B) + C$. A larger δ indicates a rougher surface. The averaged values of δ are plotted in figure 5.6. . . . 80

5.6 Evaluation of the MTS surface roughness. The averaged values of δ are 18.8, 27.1, 26.2, 42.8 (μm) in the cases (a), (b), (c) and (d) with standard deviations 2.9, 3.9, 2.2, 5.5. 81

6.1 Performance test of the Voro++ package. Red dots are the results of our test. The solid green line is the linear least-square fitting. The slop of this line is about $6.176 * 10^{-5}$. 87
6.2 Performance test of four linear system numerical solvers, namely BiCGStab, CG, Gradient and Loop preconditioned by GS, Jac, (ILU(0) and ILU(1)), SOR, SSOR. Corresponding data can be found in table 6.1. 89
6.3 Illustration of the matrix sparsity of the linear system of cell motion. The two panels represent different matrix sizes, 75×75 and 3000×3000, which correspond to the tumor systems of 25 and 1000 cells. Blue square dots, which look much larger in size in the left panel than that in the right panel because of the rescale of the figures according to the computational domain size, except those diagonal ones, in this figure represents a 3×3 matrix, indicating the neighboring relationship between cells of i (row) and j (column). 91
6.4 Number of iterations (averaged over 10 samples) needed by the numerical solver (BiCGStab) to converge with increasing the system size (the number of tumor cells). 94
6.5 CPU time required by model solving. VT stands for calculating the 3D radical Voronoi tessellation. LS stands for solving the linear system of cell motion. RD is solving the RDEs of oxygen and glucose. CC represents performing cell proliferation. The number of tumor cells is $4 * 10^5$. The FEM grid applied to solve the RDEs is a $30 \times 30 \times 30$ lattice. The condition for numerical solvers to converge is theoretically described by $\|\mathbf{b}_{i+1} - \mathbf{b}\|_2 < \|\mathbf{b}\|_2 \cdot 10^{-10}$. 96

C.1 Hat functions $\varphi(x)$ in 1D case. I_i is the sub-interval i. 109

D.1 Radical Voronoi tessellations of two spheroids of different radii. These spheroids in different groups have the same coordinates but different radii. 114

List of Tables

2.1 Software tools for agent-based modeling and simulation. 14

4.1 Data members of the cell agent class (`cell`) in TUGME. `STT` and `BTT` are its template arguments. `BTT` represents the `basicTraitsType`, where the very fundamental data types are declared. `STT` stands for `superTraitsType`, which dynamically integrates all traits of the C++ classes of a tumor model in TUGME. `UIDType` is actually a C++ `unsigned long int` type. `coordinate3DType` is a triple of numbers of `NumberType`, and `NumberType` is the C++ `double` type. 35

4.2 The Voronoi information of single cells. 47

4.3 Tissue and cell mechanical forces. 48

5.1 Parameters used in simulations. 75

6.1 Performance test of preconditioned linear system numerical solvers. . . . 89

Abbreviations

WHO	the **W**orld **H**ealth **O**rganization
DNA	the **D**eoxyribonucleic **A**cid
GF	the **G**rowth **F**actor
TUGME	the **TU**mor **G**rowth **M**odeling **E**nvironment
DUNE	the **D**istributed and **U**nified **N**umerics **E**nvironment
FEM	the **F**inite **E**lement **M**ethod
CSC	the **C**ancer **S**tem **C**ell
ATP	the **A**denosine **T**riphosphates
ECM	the **E**xtracellular **M**atrix
ECF	the **E**xtracellular **F**luid
MTS	the **M**ulticellular **T**umor **S**pheroid
HPC	the **H**igh **P**erformance **C**omputing
DEVS	the **D**iscrete **E**vent **S**ystem **S**pecification
ABM	the **A**gent-based **M**ethod
MAS	**M**ultiagent **S**ystems
MCMC	**M**arkov **C**hain **M**onte **C**arlo
PDE	the **P**artitial **D**ifferential **E**quation
RDE	the **R**eaction **D**iffusion **E**quation
CA	**C**ellular **A**utomata

Chapter 1

Aim and Structure of the Thesis

1.1 Motivation

Cancer has been longtime the second killer to human beings right after the cardiovascular diseases. According to the latest statistics by the World Health Organization (WHO), 8.5 million people died of cancer in 2012, and the annual cancer cases may increase from 14 million in 2012 to 22 million within 2 decades. Nowadays, worldwide people face the threat of a variety of cancers. Hence, there is no need to emphasize the importance of cancer research.

Numerous key advances have been made over the more than two century history of cancer research (reviewed in [8]). Some significant discoveries have profoundly reshaped our understandings of cancers. For example, the discovery of viruses causing the avian cancer by Peyton Rous in 1911 [9] has led us to the finding that the incidence of cancers is closely related to cell chromosomes, where the genetic information is encoded and passed from cells to cells via cell replication. Alterations in a certain fragment of a DNA sequence (gene mutations) are particularly responsible for the incidence of a cancer. These DNA fragments were termed genes whose expression synthesizes a variety of proteins that are vital for living organisms. Cancer related genes were later classified into proto-oncogenes and cancer suppressor genes. Furthermore, some substances (carcinogens) have been found having a strong capability to cause damages to cell genomes, hence, avoiding exposure to them can help decrease the possibility of developing cancers. However, it is not possible to eradicate cancer completely from our life and many cancers remain causing high morbidity and mortality, even though, substantial breakthroughs in medical equipments enable us to detect a cancer in its much earlier stage, which has been helping both avoid a proportion of fatal cancers today and improve the life-span and quality of persons with a cancer. As a result, cancer

research has been much more motivated for innovations of caner therapy strategies nowadays.

Understanding the working mechanisms of cancer cells can be of great help to better fighting against them. Furthermore, more and more investigations have gradually uncovered the fact that the development of a cancer (tumorigenesis) is multi-factorial and multi-scale by nature [2–4, 10, 11]. Hence, a systematic understanding [11–14], crossing multiple spatial-temporal scales, becomes critical. Unfortunately, current experiments in the laboratory are restricted to very finite spatial-temporal scales of the tumor world for the imitations of spatial-temporal resolution of current apparatuses for both carrying out experiments and conducting measurements. Pure experimental investigation (or the wet-lab experimentation) is difficult to shed light on the micro-scale biological, biochemical and biophysical processes of single cells.

Computer-based modeling and simulation (the dry-lab experimentation) is believed to be a potential auxiliary to the traditional biological experiments for systematically investigating complex systems like tumors in systems biology. With the fast boost of both computer hardware and software in the past two decades, computer-based modeling and simulation has been playing a more and more important role in the research of challenging biological problems, since computer models are much more flexible and controllable, and it can cover a much wider temporal-spatial scale of the investigated problem compared with the traditional experimental means.

However, modeling and simulation of complex multicellular cancer systems currently is also very challenging. The challenges come from two general aspects, namely model establishment and model solving. Model establishment involves modeling methodology choosing and model fidelity control. Model solving includes the requirements of computer hardware like memory as well as the time needed for carrying out simulations. These two aspects are actually closely coupled. It is obviously that model complexity determines the computational cost of model solving. Inversely, the computer hardware limits the complexity of a model, since a model can not be solved or takes too long time to solve is practically useless. Therefore, model establishment is the key to multi-scale cancer modeling.

After reviewing abundant cancer models on the market, we notice that some problems are important but have not or well addressed by now. They include: 1) reusing of current models is almost as difficult as establishing a new one; 2) most models are either oversimplified for reducing the computational cost of model solving or too complex to model large tumor systems; and 3) there was basically no easy to use software environment specially tailored for multi-scale cancer modeling and simulation at the time (in 2011) when we started this project. Obviously, reusing models based on a

software environment is very helpful for carrying out computer-model-based cancer research in many aspects, including decreasing the risks of making errors during model development, reducing various costs of creating a new model, avoiding the waste of building similar models and so on.

In light of these questions, this thesis is aimed at developing a software environment to facilitate agent-based multi-scale cancer modeling and simulation by enabling easy model reuse and extension. Furthermore, under the help of this environment, it is expected to find some of the underlying working mechanisms of tumor cells and to give predictions of possible death/survival conditions of them, which would help both broaden our views of complex cancer systems and promote the innovation of cancer treatment strategies.

1.2 Structure of the Thesis

In this project, an agent-based multi-scale tumor growth modeling and simulation environment TUGME (TUmor Growth Modeling Environment) is proposed. TUGME is designed for easy model reuse and extension. It allows users to carry out simulation experiments to test hypotheses about caner and cancer cells by either directly using the default models or implementing the model interfaces provided.

Chapter 2 begins with a brief introduction to the experimental (in vivo and in vitro cancer models) means of cancer research in the laboratory. After that, the main advantages and disadvantages of these means are summarized. In the second section of this chapter, we introduce the conceptions of computer-based modeling and simulation and summarize its advantages compared with the traditional experimental approaches. Then, prevalent cancer modeling approaches are comparatively introduced. Finally, we end this chapter with a short review of some existing software tools that could be applied to agent-based cancer modeling.

In chapter 3, fundamental conceptions of cancers and tumors are introduced. Then, tumorigenesis is briefly introduced by highlighting some of the key transformations essential for the final incidence of a cancer. The final section of this chapter is focused on reviewing some of the critical biological, biochemical and biophysical traits of cancers and cancer cells.

In chapter 4, we introduce the details of the design and implementation of TUGME. First, the general perspectives of TUGME are summarized. Second, the basic design principles of TUGME are highlighted. Third, TUGME is highly modularized. The

functionalities of its modules and their inter-dependences are introduced in details. Finally, the basic steps using TUGME to carry out a simulation is introduced.

In chapter 5, a concrete tumor model is developed based on TUGME. Then, the multi-cellular tumor spheroid (MTS) growth of the cell line EMT6/Ro is simulated with this model. Three properties of the modeled EMT6/Ro tumors are particularly investigated in our simulations, namely the cell population, the distribution of nutrients and the invasive morphology of MTS tumors. Good agreements between our simulations and published experiments indicate the validity of our tumor model as well as the practical applicability of TUGME.

In chapter 6, the computational complexity of carrying out agent-based multi-scale cancer modeling and simulation is discussed mostly based on the model developed in chapter 5. We evaluate the computational bottleneck of our model and try to find out a better solution by testing different options for solving the modular parts of our tumor model. Finally, we end this chapter by concluding our results and the expecting the further endeavors of agent-based multi-scale cancer modeling.

In chapter 7, a summary of the thesis is given and a discussion of possible avenues for further research based on TUGME are prospected.

Chapter 2

Cancer Modeling

In this chapter, firstly, popular cancer research means are introduced, including the in vivo, in vitro and in silico models. Particularly, the general advantages and disadvantages of these approaches are briefly introduced. After that, the advantages of computer-based cancer modeling and simulation are briefly summarized. Thirdly, general approaches, including the continuous, the discrete and the hybrid methods, for establishing computational models are comparatively reviewed. Specifically, single cell representation methods are highlighted. Finally, the agent-based cancer modeling method is introduced and some existing software libraries, which can be applied to agent-based cancer modeling, are briefly reviewed.

2.1 Physical Models of Cancer

Basic cancer research by biologists is mainly carrying out experiments in the laboratory (the so-called wet-lab experiments [15]). In general, the experimental materials are various types of cancer cells, which are either injected into living animals like mice to induce tumors within them (in vivo tumors) or cultivated in culture medium with properly supplied nutrients like glucose (in vitro tumors). Research directly based on in situ and metastasis tumors within the body of persons with a cancer can seldom done as a set of very strict regulations have to be passed.

In vivo environment (the body of animals) is obviously mostly similar to human bodies. Hence, experimental results drawn based on the in vivo tumors are usually believed more reliable. However, many aspects of the in vivo environment are hardly controllable to researchers, especially the individual-tumor-dependent factors as well as the intrinsic randomness. Compared with in vivo environment, the environment of in vitro tumors

can be better controlled, but it is relatively less real for the absence of normal tissue cells that surround in vivo tumor tissues. In general, easy to control and limited effects of secondary factors of in vitro environment allow a more direct investigation of individual factors (univariate analysis), which makes the in vitro tumors popular among experimental oncologists.

Tumor monolayer is prevalent in vitro, where tumor cells grow on a Petri dish with necessary nutrients for sustaining cell growth and proliferation. One important characteristic of tumor monolayers is that all cultured cells have basically the even accessibility to nutrients, hence, distinct tumor cell dynamics are considered as the result of all other factors except nutrients. It is an important experimental means for investigating and analyzing the growth and invasion mechanisms of tumors [16, 17]. Besides, tumor monolayers have been widely used as test systems to investigate the curative effects of anti-cancer drugs, radiotherapy, and chemotherapy etc [18, 19].

Unfortunately, tumor monolayers can not represent many aspects of actual tumor cell aggregates, for example, the 3D structure and the biological and biophysical properties of closely related to the 3D structure as it has been discussed in [20]. Multicellular tumor spheroids (MTSs) first used to investigate the effects of radiotherapy on tumor cells by Sutherland et al. in 1971 [21] are now prominently applied to cancer research. MTSs are thought to be more real and suitable in vitro tumor models for preserving the 3D structure of real tumor cell aggregates. Furthermore, significant differences or even contradictory phenomena have been indeed observed by conducting comparative experiments using tumor monolayers and MTSs [22–24]. MTSs provide an alternative with intermediate complexity between tumor monolayers and in vivo tumors, and more importantly, they can be used to model the avascular growth of real tumors that are too small to detect clinically. In addition, quantitative measurements of MTSs are very important references for validating the in silico cancer models.

No matter the in vivo or the in vitro tumors, they are basically sort of physical models of actual tumors. Hence, inevitable random factors as well as the differences of the biological, biochemical and biophysical environment between the body of human beings and animals or between the human body and the culture medium may significantly affect the credibility of experimental results. Examples are the rare successful cases of the tryouts of anti-cancer drugs with persons with a cancer for various reasons like drug toxicity [25–27] or inefficient drug delivery [28, 29], while these drugs may have shown a false image of promising curative effects in the tests with in vivo and in vitro tumors. Understanding the failures of anti-tumor drugs is now a hot research area in cancer research.

Anyway, there is no doubt that both the in vivo and in vitro approaches have played and will continue playing an irreplaceable role in cancer research. However, cancer is such complicated a system that a systematic understanding needs comprehensive insights into multiple spatial-temporal scales of the tumor world. Unfortunately, it is getting increasingly clear that traditional biological means face severe challenges in doing so for the limitations of resolution in space and time, especially at the micro level, of current apparatus in both carrying out experiments and conducting measurements [30]. As a matter of fact, it has become a common problem for many research areas in biology [12, 31, 32].

In recent decades, with the boost in both computer hardware and software, computer-based modeling and simulation techniques have been playing a more and more important role in the research of many scientific problems including complex biological systems like cancer. Compared with the traditional biological means, computer modeling and simulation has intrinsic flexibility, which allows the possibility of shedding light on the micro-world of tumors. Since Zeigler et al. systematically introduced for the first time the theory of modeling and simulation in 1976 [33], it has underwent an exponential growth in last two decades in numerous disciplines, such as military, economics, biology.

Considering the long-term characteristic of the tumorigenesis process, much more research has focused on a specific growth stages (avascular, vascular and tumor metastases) of tumors. This project is initially started with the modeling and simulation of the avascular growth stage. This is not to say that avascular tumor growth is more important than the other two stages. As a matter of fact, the other two stages are of more practical relevance to cancer therapy. Nevertheless, when attempting to investigate complex systems like cancers, it is wise to first try to understand parts of the entire system as well as possible before linking them together as a whole, especially when the parts are already complex too. Avascular tumor growth is relatively simple to model and is thought to influence the later two more important stages significantly. Moreover, more experimental data about avascular tumors are available, which facilitates quantitatively validating in silico tumor models. While experimental data about vascular tumor growth and tumor metastases are relatively lacking for various difficulties in obtaining vascular tumors in vivo and in vitro and conducting experiments and measurements on them. Finally, the computational cost of solving tumor models on computers is worthy of special considerations. As tumorigenesis lasts long, too complicated models can be impossible to solve within a reasonable time using single desktop computers.

2.2 Computational Cancer Models

2.2.1 Brief Introduction to Computer-based Modeling and Simulation

The conception of model in computer-based modeling and simulation is completely different from that of a physical model which is a physical replica of the corresponding real-world system. In computer-based modeling and simulation, models conceptually abstract the features or traits of relevance of the real-world system. They are usually described by mathematical equations and are called theoretical models. A theoretical model should be established exactly fulfilling goals the modeler. Specifically, all and only features that are of particular interest to the modeler should be extracted from the target system, and those of irrelevance should be neglected. For example, features like the weight and the height of individuals are usually not considered for a population model, since they are considered to be not critically relevant.

Actual physical systems are of two general types, namely the static systems and the dynamic systems. The state of a static system does not change over time, hence, static systems are relatively easier to understand. The state of a dynamic system varies over time. The mechanisms regulating the dynamics of a complex dynamic system can involve a lot of nonlinearly coupled factors. Furthermore, some dynamic systems involve intrinsic randomness, for example, Brownian motion [34]. The randomness of dynamic systems is modeled based on pseudo-random numbers that are generated by computer algorithms. Markov Chain Monte Carlo (MCMC) is a typical type of method to model the dynamic this kind of systems [35]. Models are static and dynamic correspondingly depending on the type of the modeled systems.

Some models are hard to solve by hand. For example, many time-dependent PDEs (RDE models) can be very hard to solve analytically, however, numerical solutions can be got much easily with algorithms that can be implemented and executed on computers. Using computers to solve theoretical models is a process termed computer-based simulations (or computer simulations).

Computer-based simulations (dry-lab experimentations) and wet-lab experimentations are closely related. Behavioral rules of theoretical models are basically assumptions on the workings of the investigated system. Since computer programs should be unambiguous, these rules should be definite. However, these rules should be proposed based on the known knowledges or theories of the studied system rather than random conjectures. Simulations allow a layout of the evolving process of the modeled system,

which may be never or hardly observed directly with experimental means in the laboratory. For this kind of systems, simulations, on the one hand, can help us better understand their dynamics. On the other hand, new insights gained from simulations can be used to direct the wet-lab experiments. Figure 2.1 illustrates the interplay between computer-based modeling and simulation and the wet-lab experimentation.

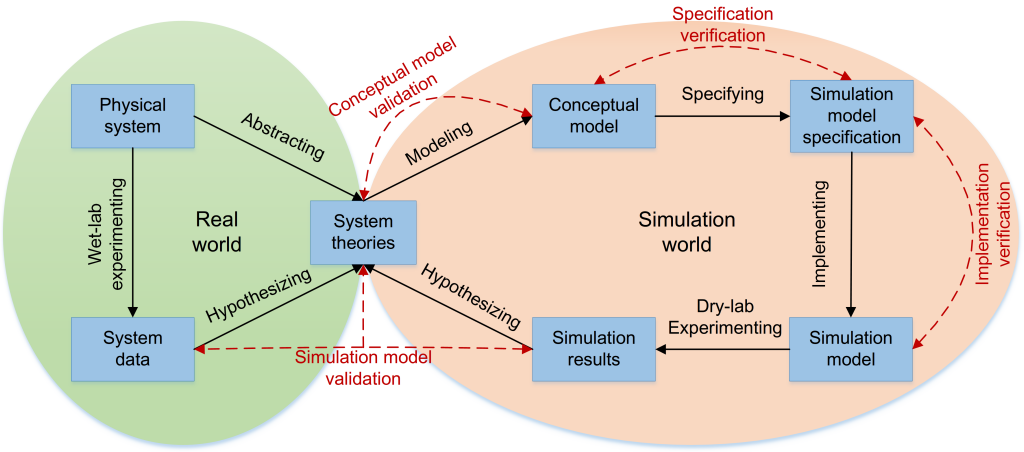


FIGURE 2.1: Diagram illustrating the relationship between dry-lab and wet-lab experiments. On the one hand, models should be built based on the basic knowledge (theories) of the real system, which is abstracting its traits of relevance. On the other hand, models are used to test hypotheses and more importantly to make predictions on the modeled system. By comparing simulation results with experimental data, models get validated, meanwhile, the knowledge about the studied system gets accumulated. Image adapted from Ref. [1].

A simulation is to solve a theoretical model under the help of computers. The theoretical model has to be implemented using either basic programming languages like C, C++, Jave etc or domain-oriented modeling languages, such as SIMULA [36] and GPSS [37]. A detailed introduction about domain-oriented modeling languages can be found in [38]. Basic programming languages allow full flexibility of any applications, however, the programming skills are required mastering. Domain-oriented languages are usually easy to learn, whereas, they are tailored for certain application domains.

2.2.2 Benefits of Modeling and Simulation

Though several obvious advantages of conducting computer-based cancer modeling have been mentioned early in this thesis, systematically summarizing all possible benefits would better motivate this project. These benefits include:

- Theoretical models are highly simplified. A neatly-defined theoretical model should encapsulate only features of significantly relevance, neglecting those of

irrelevance or weakly related. Obviously, neat models are much easier to control and analysis.

- Theoretical models are fully-controllable. All mechanisms that regulating dynamics of a theoretical model should be definitely defined by the modeler. These mechanisms can be changed by the modeler for testing different hypotheses or other purposes. More importantly, singular analysis of a specific parameter excludes the effects of secondary factors, which may significantly blur the results of the wet-lab experiments, in computer-based modeling and simulation. Besides, simulation results are completely repeatable even for models with randomness.

- The costs of developing a new theoretical model usually is much lower than that of creating a new physical model or making a real-world system. Furthermore, an actual system may not be available or allowed to conduct experiments with it. For example, it has a lot of very strict procedures to go through before testing an anti-cancer drug on persons with cancers.

- Theoretical models can be reused and modified, while this is basically impossible for physical models or real-life tumors. Theoretical models are conceptually described and exist as computer programs. Reusing them is less expensive than creating a new one, not to mention creating a new physical model or a real-life tumor.

- Simulations could take a much shorter period of time than that of the physical models and the real-life systems. Hence, long-term dynamics of a system, for example, the star forming process [39], which is very difficult to be directly investigated by experimentations in the laboratory, could be studied within a much shorter period of time via computer-based simulations. Tumor growth can evolve for years in reality. Simulations of tumor growth taking days or even weeks would be of great practical value.

- High performance computing (HPC) based on high performance computers, such as super computers and cloud computers, provides techniques that can be applied to computer-based modeling and simulation to speed up the simulation process. Though HPC-based cancer modeling is still in its infant stage, wide applications and remarkable success of HPC-based simulations in many domains, such as military, industry and scientific research, indicate that this is a promising direction. A first try done by Kang et al. shows the feasibility of this idea [40].

- With the advances in computer-based modeling and simulation, the methodology for model development and description has formed up a number of the so-called specifications of formalisms, such as discrete event system specification

(DEVS) [33, 41] and the agent-based method (ABM) [42–44]. These diverse formalisms are of great help to efficient model construction. ABM is prevalent in single-cell-oriented tumor modeling.

2.2.3 Tumor Models and Cell Representation

Generally there are three classes of approaches for cancer modeling: namely the continuum, the discrete and the hybrid [3, 45–47]. Each type of approach has it own characteristics which make it proper for investigating certain features of tumors and tumor cells.

Continuum models are generally realized by Ordinary differential Equations (OREs) or Partial Differential Equations (PDEs). They are usually applied to study the large scale properties, such as the population and the volume of tumor tissues [48] or the density of tumor cells [49]. Reaction-Diffusion Equations (RDEs) are commonly adopted to model the transport and metabolism of nutrients [50–53], where the diffusion term and the reaction term correspondingly model the molecular diffusion and consumption by cells.

Continuum models have many valuable advantages. First of all, they can be scaled to very large tumors without substantially increasing the computational cost of modeling solving. Secondly, they can be solved efficiently on computers, since there are many classical methods particularly for solving complex PDEs numerically. The disadvantage of continuum models is that the discreteness of individual tumor cells is difficult to explicitly model. As the basic building units of a tumor tissue, cells are discrete by nature. Cellular membranes separate the inner cell world from the surroundings. Behaviors of cells, such as growth, proliferation, movement and death, are individual-cell-based. Furthermore, tumor cells are heterogeneous [54]. To take into account the discrete nature, the discrete approach has been proposed naturally.

The discrete approach enables much higher flexibility in representing individual tumor cells compared with the continuum approach. Its basic idea is to treat each tumor cell as an individual object, where cell growth, proliferation, motion, death, interactions can be explicitly modeled as the behaviors of the individual cell objects. ABM is naturally adopted. However, the discrete approach isn't versatile in representing all the aspects of tumors or tumor cells. For example, the transport and metabolism of biochemical molecules are too inefficient to realize with the discrete approach. The hybrid approach, which naturally integrates the continuum and discrete approaches, gradually becomes the favor of tumor modelers in recent years [3, 44–47, 55].

The hybrid approach is far more complicated than mechanically combining the continuum and the discrete approaches. Its key is to couple the continuous processes with the discrete processes properly because these processes are quantified by continuous and discrete variables. As the tumor growth involves multiple spatial-temporal scales, these variables have to be mapped consistently.

As tumor cells persistently interact with each other, cell-cell topology and three dimensional cell geometry are important for high fidelity hybrid models. Different strategies can be adopted to represent individual cells with respect to cell topology and geometry. Generally, they can be classified into two types, namely the lattice-based method and the lattice-free method.

Cellular automaton (plural form cellular automata, CA) [56–58] is a lattice-based method mostly used. CA partition the 3D space (two-dimensional (2D) area) into many much smaller compartments, lattice cells, using a variety of grids. Lattice cells usually are identical in shape, for example, cubes in three dimensions (squares in two dimensions), when regularly structured grids are applied. Besides, Each tumor cell can occupy one or several single lattice cells depending on the resolution of single cell representation. CA models neatly represent individual cells in the shape, the size, the neighborhood etc, on which many cell behavior rules could be defined easily based. For example, cell motion could be defined as a discrete jump from one lattice cell to its direct neighbors. However, this is unrealistic. On the one hand, the neighborhood of lattice cells is artificially determined by the structure of the grid adopted; On the other hand, cells move continuously rather than jump discretely in reality.

In contrast, the lattice-free approach frees the single cell representation from the lattice grids, which enables continuous varying the shape, the size and the location etc of single tumor cells. The lattice-free approach increases the model fidelity in terms of more precisely representing many sing cell properties which are believed to influence and be influenced by tumor growth dynamics [59–61].

2.2.4 The Agent-based Method and Software Tools

ABM is a powerful technique for model design in computer-based simulations. On the one hand, simulation is often utilized to investigate agent systems, for example multi-agent systems (MAS) [62]. On the other hand, ABM has been widely used as a standard model design method for a wide range of applications in computer-based simulations [62–64].

An agent is a computer system capable of autonomous action in its environment. Agents can be thought as objects with strong notion of autonomy [65]. Normal objects of systems encapsulate states and corresponding state-updating operation methods. In contrast, an agent has the ability to actively sense the changes in its environment, to deal with the perceived information and to make decision for its further actions (see figure 2.2). In a word, an agent is not only passively affected by its environment, but also actively change the environment for its own preference.

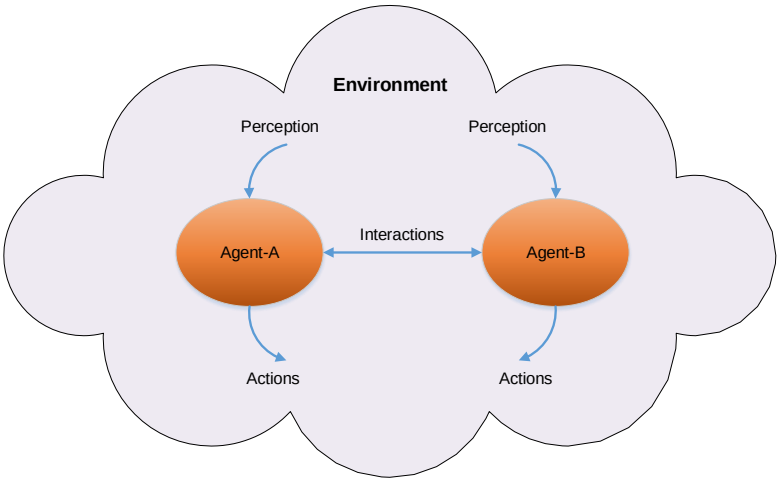


FIGURE 2.2: Diagram illustrating agent-agent and agent-environment interactions.

Agent-based applications basically consist of a common environment and a set of agents within it (see figure 2.2). In an agent-based system, an agent interacts with other agents (its neighbors) as well as its environment. Like a society, an agent-based system allows agents to achieve collective goals via cooperations and coordinations, and to achieve individual aims through competitions. ABM is very powerful for model description. Agents of an agent-based system may share some properties and also can vary significantly in some properties and behaviors. Besides, a complex agent can be further decomposed into sub-agents too.

With such high flexibility and strong description capability, ABM has a wide range of applications, which has stimulated the emergence of software environments or toolkits to facilitate construction of agent-based models. Here, some of them are briefly reviewed from the perspective of the possible application in agent-based cancer modeling.

The software tools listed in table 2.1 are representative with respect to the way of constructing an agent-based model. One may find more software tools for general applications of agent-based modeling and simulation like FLAME or SWARM. However,

Name	Main features	Ref.
Biocellion	Biocellion works more as a framework specially emphasizing on speeding up multicellular systems simulations by parallelization using high performance computing (HPC) algorithms. Interfaces are left for defining single cell properties and cell behavior, such as cell sizes and shapes, cell growth, division, death and movement, physical interactions and cell-environment exchange of molecules. In addition, PDEs are used for modeling the extracellular environment evolution, where corresponding solvers are provided.	[40]
CellSys	Individual tumor cells are treated as biophysically isotropic, elastic and adhesive spheroids in three dimensions (circles in two dimensions). Each spheroid is mapped to an agent undergoing cell movement, division, physical interactions with its ECM (extracellular matrix), neighbors and environment. Diffusion and metabolism (consumption or production) of nutrients, growth factors, metabolic products etc are modeled by mathematical equations. Besides, CellSys allows different types of cells.	[66]
FLAME	FLAME is a general framework for conducting agent-based modeling and simulation. Agents are defined following the concept of X-Machine, which is characterized by state, state transition functions, and input and output messages. FLAME is redesigned for being capable of handling larger-scale systems by exploring several levels of parallelism.	[67]
SWARM	SWARM models are hierarchically defined. Each SWARM model is a swarm of multiple types of agents. A model is a super agent, which consists of smaller agents. Besides, agents interact with others by scheduling discrete events as what is done in discrete event simulation (DES) [68].	[69]

TABLE 2.1: Software tools for agent-based modeling and simulation.

we are more interested in those tools that are particularly tailored for agent-based cancer modeling like CellSys and Biocellion, since more general-purpose tools usually ask for more programming endeavors for specialized applications. CellSys was initially published right in the year (in 2011) when we started this project. We design and implement a new software environment particularly for agent-based multi-scale cancer modeling and simulation instead of using CellSys directly for the reasons: 1) we want to explicitly model the dynamic changes of single cell morphology as well as its effects on the development of tumors using lattice-free methods; 2) we want to explicitly model the tumor cell motion driven by the mechanical interactions between cells and their surroundings; 3) we want a software environment that its models can be reused and extended easily for testing different hypotheses. Unfortunately, CellSys doesn't fulfill these perspectives very well, especially model reuse. Biocellion is designed to

take the advantages of high performance computers to speed up multicellular system simulations, however, it has just been published recently. Furthermore, it doesn't explicitly take cell morphology and cell topology into account.

Chapter 3

Introduction to Cancer Development

This chapter starts with a brief introduction to the basic conceptions of tumors and cancer as well as their relationship. Secondly, the development of a tumor is introduced by highlighting several critical transformations during this process. Finally, we review some important aspects of cancer cell biology that are particularly interesting to most cancer modelers.

3.1 Tumors and Cancer

A tumor is a lump or a mass of tissue characterized by out-of-controlled cell division. Tumor tissues are neoplastic, since they do not produce meaningful functionalities to the host as what normal (or healthy) tissues do. Tumors, most likely staying localized and being separated by a interface from the neighboring healthy tissues, are usually considered to be benign. Benign tumors are normally harmless except those at critical host sites, for example, a benign brain tumor may cause functional disruptions to neighboring nerves because of its physical constriction on them. In contrast, tumors, consisting of cells capable of invading surrounding normal tissues and spreading to distant host sites, forming secondary tumors (cancer metastases) (see figure 3.1) wherever preferable, are thought to be malignant. A malignant tumor is fatal to the host because of its uncontrolled growth and metastases, which cause functional failures to the in situ as well as the secondary-tumor-colonized tissues or organs. Only malignant tumors can be called cancer while benign ones cannot.

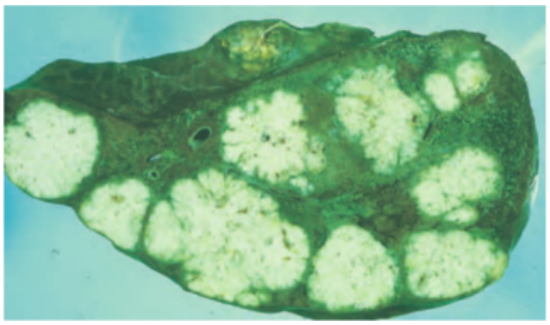


FIGURE 3.1: Liver metastatic colon carcinomas (white). Colon carcinoma cells colonize the liver and trigger several secondary tumors in it. Image adapted from Ref. [2]

Cancer is actually a general name of a group of diseases. By now, more than ten general types of cancers have been discovered in human beings. A malignant tumor is a cancer, whereas, not every cancer necessarily grows into a malignant tumor. For example, leukemia is a kind of cancer that only causes an abnormal high number of white blood cells in the blood stream. To avoid misunderstanding, we mean malignant tumors when using the term cancer and vice versa in the rest of this thesis.

3.2 Multi-stages of Tumorigenesis

The development (the onset and the progression) of a cancer has to undergo several critical transformations, which include, 1) the variation from healthy cells to abnormal cells, 2) the tumor-driven growth of blood vessels (angiogenesis), and 3), the local invasion and distant migration of tumor cells (tumor metastases, the malignant transformation of tumors). In general, these series of transformations indicate increasing degrees of malignancy of a tumor. Theoretically, stopping any of these transformations may prevent the occurrence of a cancer.

The onset of a tumor is characterized by various types and degrees of gene lesions, which transform normal cells to abnormal ones that deviate from the normal cell proliferation routine to a certain extend. Because of this, cancer is widely accepted as a gene disease [70, 71]. As a matter of fact, the starting point of almost all actual cancers are not clear to us, since early stage tumors are small and without evident symptoms, hence, they are rarely noticed in this stage, and most clinically detectable tumors are of the size with about one hundred millions (10^8) cells. Investigations on cells of mature tumors show that they have possessed some common capabilities, the hallmarks of cancer cells [4, 54], such as losing some of the normal cell proliferative regulating mechanisms and intentionally neglecting the apoptosis signals.

Early stage tumors are avascular, which means no particular blood vessels exist for them. Hence, nutrients like glucose (metabolic products) of cells are mainly transported by molecular diffusion from (to) the surrounding normal tissues. An avascular solid tumor normally can not grow too large. Although, its cell proliferation is out of control, which theoretically enables them unlimited proliferative potential, many actual solid tumors are found with inner necrosis (a type of cell death different from apoptosis [72, 73]), which may be due to the inefficient supply of energy-producing nutrients like glucose and oxygen, since the availability of ATP may change the way of cell death from apoptosis to necrosis [74]. When cell proliferation and cell death balance out each other, the avascular tumor growth reaches a saturation [7, 75].

It seems that tumor cells are not "satisfied"with the growth saturation. They try to find ways to survive and to expend. The aberrant cell proliferation actually facilitates the accumulation of further genetic damages and epigenetic variations. As a result, tumor cells get more and more uncontrollable and aggressive during the tumor growth. In the later stage of avascular tumor growth, tumor cells start to release some growth factors (GFs) to promote the growth of blood vessels (angiogenesis). With the stimulus of these GFs, blood vessels of normal tissues grow to tumor cells. By doing so, on the one hand, a tumor can acquire its own blood vessels that of course can help its transport of nutrients and metabolic products more efficiently. On the other hand, some tumor cells try to get rid of the bondage of the in situ tumor and to seek a new place of residence via the blood circular system (the distant migration). Meanwhile, some tumor cells escape from the in situ tissue and invade the neighboring normal tissues (local invasion).

Tumor cell local invasion and distant migration (tumor metastases) are the essential and most critical transformations (malignant transformations) of a tumor. Before this transformation, a tumor is actually still benign. However, the benign stage tumor of a cancer is radically different from the common benign tumors introduced in section 3.1, since it is almost destined to cause a cancer sooner or later, while a common benign tumor grows very slow and will be very likely keeping benign all its life-time. even it also has a very low probability of malignant transformation.

Tumorigenesis a process can never be finished over night but usually last for years or even decades. It has been conjectured that the duration of the actual tumor growth duration is always much shorter than the interval between the carcinogenic stimulus and the clinical occurrence of the tumor [76]. Statistics on the cancer incidence rate of different ages show a sharp increase after certain age, which may be explained by the long-term characteristic of tumorigenesis [2].

3.3 Cancer Biology

Tumorigenesis is multi-factorial, however, most research of oncologists has concentrated on genetic factors that contribute to the survival and proliferation of individual tumor cells. Tumorigenesis is actually an extremely complicated process, involving complex biochemical reactions as well as biomechanical interactions at multiple spatial-temporal scales of the tumor world [4, 10, 54, 77] (see figure 3.2). From the point of view of cancer treatment, understanding these biochemical reactions and biomechanical interactions may also help promote innovation of new cancer therapies.

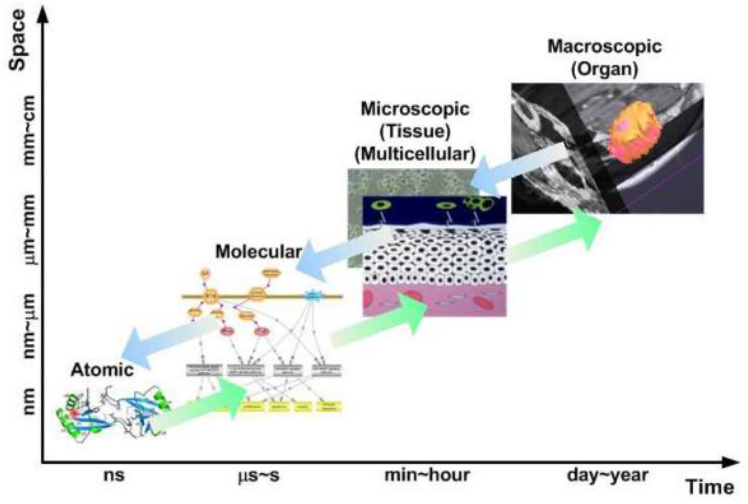


FIGURE 3.2: Schematic illustration of the spatial-temporal scales involved in tumorigenesis. Image adapted from Ref. [3]

3.3.1 The Randomness of Cancer Incidence

Since the revolutionary discovery of genes (proto-oncogenes) in normal avian genomes that have the potential to cause a cancer [78], it has been getting more and more clear that some mutations in certain genes, such as the activation of oncogenes like KRAS or the deactivation of cancer-suppressor genes like the p53 gene, are essential to the incidence of some cancers. Besides, genes, maintaining the genomic integrity of cells and sustaining a normal stromal environment for cells, are also responsible for the occurrence of cancers [79].

Mutations in genes may provide cells with new capabilities, however, most of these abilities hardly facilitate the survival of a species in a constantly changing environment. Whereas, many new traits of tumor cells actually facilitate tumorigenesis. For example, uncontrolled cell replication is a new ability for tumor cells, which helps the

accumulation of both tumor cells population growth and further gene mutations in the offspring tumor cells. The former gives rise to the increase of tumor size and the later may mount tumor cells further capabilities to survive and to spread.

Although, a variety of carcinogens have a high probability to cause various types of gene mutations to mammalian cells, which may be responsible to few cases of cancer incidence, most current human cancers are thought as an unlucky result of random hit of gene lesions to cell genomes.[2].

For normal somatic cells, the gene mutation probability (or mutation rate) of a single gene is very low (about 2.5×10^{-8} mutations per nucleotide site or 175 mutations per diploid genome per generation [80]). Furthermore, only very small parts of human genomes (about 3.5% [2]) are actually going to be expressed as genes (dominant genes), and only certain genes (proto-oncogenes and tumor-suppressor genes) are particularly relevant to human cancers. Hence, random gene mutations can only rarely hit the cancer related genes. Take human breast and colorectal cancers as examples, most of the identified gene mutations are not harmful in either of the cases, and mutant genes responsible for these two cancers are quite different too [81].

Our body is equipped with various self-protecting mechanisms that normally prevent us from developing cancers or other diseases. Cells are mounted with an elaborate repair apparatus, which constantly monitors cell genomes and fix the damages in DNA sequences as well as it can. Even if the self-repair procedure fails, most cells would initialize a suicide programme that is part of the cell cycle controlling mechanisms. For example, in the intestinal crypts of mouse, stem cells that have suffered genetic damages rapidly go into apoptosis, without halting their proliferation or attempting to fix the damage[2]. Besides, our immune system is in charge of preventing the development of various diseases by protecting against invading or infectious pathogens and eliminate damaged cells. It is believed that it plays dichotomous roles of possibly both antagonizing and enhancing tumor development and progression [82].

However, no protecting system can be infallible. These protecting mechanisms of cells could make mistakes of but certainly not limited to: 1) catastrophic damages to cell genomes that could possibly exceed the repair capacities of cells; 2) damages could happen to the genes related to the repair mechanisms themselves; 3) some gene mutations could enable cells to disguise themselves to escape from the hunt of the immune system agents.

All of these reasons leave the possibility of developing a tumor. The scenario could be, 1) a random mutation in a certain gene gives a normal cell an initial growth advantage to replicate itself more times than usual; 2) then, another gene mutation might provide

one of the survival offspring of this cell with further reproductive advantages; 3) with iterations of this process over many generations of cell division, a lot of gene mutations can be accumulated in the survival offspring cells. Though, most of them will die out finally, it is possible that one of the offspring happens to accumulate such mutations in certain critical genes that it becomes immortal.

3.3.2 The Cancer Stem Cell Theory

CSCs (Cancer Stem Cells) represent those special cancer cells that show specially the stem-cell-like capabilities [83, 84]. The inspiration of this theory is the experimental observation that only a minority subset of cells in many cancers have the capability to induce new tumors in vivo and in vitro. The CSC hypothesis can be treated as a part of the theory of cancer cell heterogeneity [83, 85] which generally declares that not all cells within a tumor are equal and individual tumor cells show distinct traits from others, for example it has been found that different cells inherited from a same origin may have very distinct invasive capabilities [86]. During the development of a tumor, genetic and epigenetic changes are both accumulated [87, 88] in its cells. Heterogeneous types of cells within one tumor is of no surprise considering the random hit of gene damages. However, tumor heterogeneity shows surprisingly grouped characteristics, which can never be explained by pure randomness. A pioneer work on tumor heterogeneity was done by Gloria H. Heppner in 1984 [89]. In recent decades, cell lineage analysis enables us to look into cell genes, which provides us an molecular level understanding of the tumor cell heterogeneity.

Stem cells have two common but critical traits, namely the ability to self-renew and to differentiate. Self-renewal means that at least one of the daughter cells of a stem cell division still keeps its stem cell properties. Stem cell differentiation means the non-stem daughter cell becomes mature cells building the organ within which they reside. Stem cells are usually thought as the reservoir of mature cells of organs. Cell telomere is well-known for marking the proliferative capacity of mammalian cells. It is shortened during cell proliferation for normal cells, which regulates their the mitosis and apoptosis. The telomerase activity, which maintains the length of cell telomeres, is very active, which gives common stem cells indefinite proliferative potential. The interesting thing is that CSCs have been found an active telomerase activity like common stem cells.

The CSC theory has changed our view of tumorigenesis and cancer chemotherapy profoundly [90]. With this theory, only the CSCs are proper to be called as tumorigenic cells, since the other cells differentiated from CSCs have very definite proliferative potential and will die out finally in some days or weeks [83], even though, they may acquire

some genetic and epigenetic mutations during this period of time. These differentiated cells may be better called nontumorigenic cells, since it is CSCs that basically give rise to the growth of a tumor tissue. CSCs population takes very small portion of the overall cell population of a tumor, hence, the tumor tissue bulk basically consists of these nontumorigenic cells, while the point is killing these cells instead of CSCs is not able to eliminate cancers, which may explain the failures of many anti-cancer drugs, which have successfully shrunk tumor size initially, however, eventually failed to cure a cancer for gradually acquired drug resistance. CSCs are able to differentiate. This means that they may adapt the cell proliferation according the drugs for survival, which may explain the acquired drug resistance of tumor cells during cancer treatments with drugs. In a word, therapies targeting CSCs would be very promising for curing cancer completely.

Although, challenging voices come consistently from researchers observing incompatible or even contradictory experimental results [91–99] since the born of CSC hypothesis, more and more people increasingly accept it an important theory explaining tumorigenesis, since even more positive evidences have been obtained in extensive experiments of various cancer cells lines. By now, CSCs have been identified by monitoring some putative CSC markers, such as CD133 and CXCR4 [100, 101] or testing their tumor-inducing capabilities, within a lot of cancers, including human brain tumors [102], breast cancer [103], head and neck squamous cell carcinoma (HNSCC) [104], lung cancer [105], colorectal cancer [106, 107], pancreatic cancer [108], hepatocellular carcinoma [109], melanomas [110], nasopharyngeal carcinoma [111] and etc.

3.3.3 Genotypes and Phenotypes of Cancer Cells

Genes and gene mutations are just part of the story of the tumorigenesis, since they just characterize the genotype of individual cells. The actual phenotype of a tumor and its cells is a joint result of cellular genes, cellular epigenetic traits and the influence of the environment. The relationship among them is very interesting. Genotypes indicate genes that may be expressed, the cellular environment affects which genes are to be expressed and epigenetic traits regulate how genes are expressed.

Gene expression is the process through which protein molecules are synthesized according to the genetic information encoded in cell genomes. The synthesized proteins are the main cell-life regulators. First, proteins are essential building blocks of components of cells and tissues; Second, most chemical reactions in intra- and extra-cellular volume can be performed only under the catalysis of various types of enzymes of proteins; Third, proteins regulate (e.g. as promoters or inhibitors shown in figure 3.3 for tumor

cells) various biological processes of cells. An altered gene may result in a protein with complete different functions that vary the behavior of a cell more or less. Epigenetic traits themselves are sheer consequences of varied gene expressions. Although epigenetic traits are not caused by gene mutations, they can be inherited by the offspring cells.

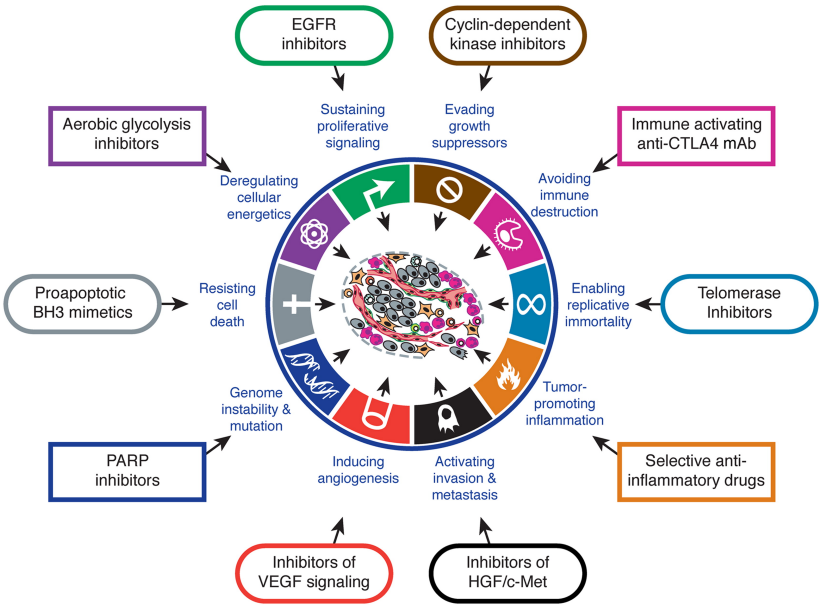


FIGURE 3.3: Some regulators (cytokines) closely relevant to tumorigenesis. Image adapted from Ref. [4]

Since a cell varies its behavior according to changes in the extracellular environment, it must be able to sense the changes of the environment, to deal with the perceived signals and finally to decide its further actions accordingly. Research has reported that there are some particular protein molecules residing on the cellular membranes, and they function as receptors of information transmitted by the corresponding molecules in the extracellular environment. The whole process that a cell receives, processes a signal, and decides its further actions is referred as a signaling pathway in molecular cell biology (see a schematic illustration of a cell signaling pathway shown in figure 3.4). Lots of cell signaling pathways that have been reported particularly up- or down-regulated (activated or deactivated) in human tumor cells [54, 112–117].

Tumor phenotypes vary significantly compared with that of normal tissues. Normal tissue cells are designated to cooperate with each other to perform certain functions. The cooperation is performed via cell-cell signal transduction. The fate of a normal cell, to proliferate, to pause proliferation, to differentiate, to die (apoptosis) etc, are not actually determined by the cell itself but its neighbors or more specifically the whole tissue or the organs [118, 119]. Cell-cell cooperations is the basis of the biological homeostasis of a normal tissue or organ, which is essential for them to perform normal

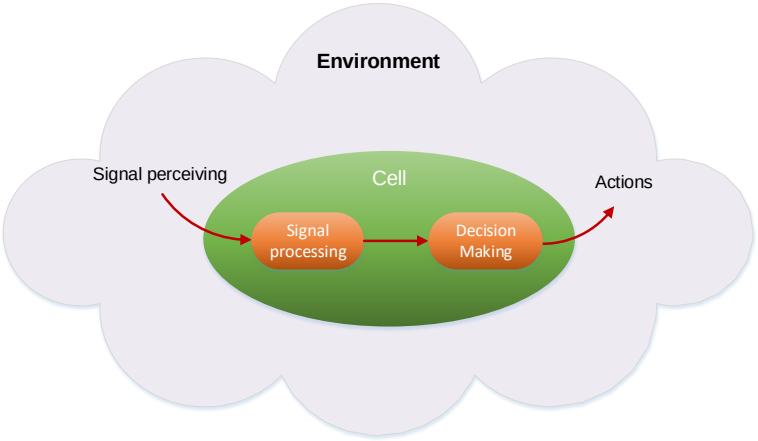


FIGURE 3.4: Schematic illustration of the general steps of a cell signaling pathway. Particularly, decision making is basically to determine which genes are to be expressed.

functions. However, tumor cells are found lost partially or fully these capabilities [118]. Their neoplastic growth results in a multicellular aggregate. However, this does not mean that tumor cells do not cooperate at all, since at least tumor angiogenesis indicates that they can work together for the survival of themselves.

3.3.4 The Aberrant Cancer Cell Proliferation

The cell proliferation is a process, through which a mother cell grow and divide into two daughter cells. The daughter cells can further divide by repeating basically the same proliferation programme as their mother cells do. Hence, the cell proliferation process is termed a cell cycle.

The normal cell cycle is generally divided into two phases, the interphase and mitosis. The interphase can be further divided into sub-phases, which are the $G1$ phase (Gap-one), the S phase (Synthesize) and the M phase (Mitosis). Figure 3.5 shows an entire cell cycle. A cell cycle starts from the $G1$ phase. During which, it grows larger nearly doubling its original volume and prepares materials for substance synthesis in the next phase, the S phase. During this phase, a cell mainly synthesizes various proteins and doubles tis DNAs. After this phase, it enters the $G2$ (Gap-two) phase, during which it grows to the size and synthesizes substances for cell division. Following this phase, the cell enters the M phase, through which it finally divides into two. As it is shown in this figure, there is a special phase, $G0$ phase, which stands for a cell cycle pause. For most somatic cells, they rest at this state until receiving a command to perform cell proliferation, which means they are alive and able to resume cell cycle when it is needed.

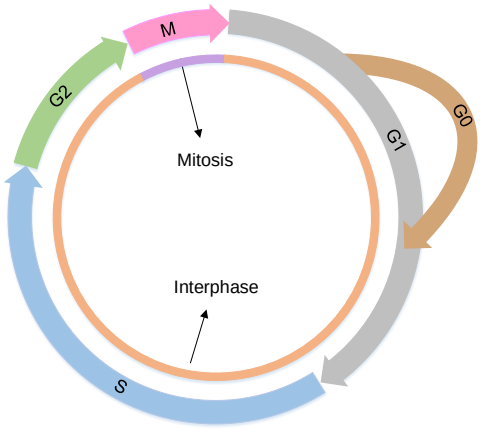


FIGURE 3.5: Diagram illustrating the phases of an entire cell division cycle. $G1$, S, $G2$ and M separately represent the gap-1, the synthesis, the gap-2 and the mitosis phases. $G0$ stands for the state that cell cycle is suspended. Healthy cells temporarily rest in the $G0$ state and can be waken up to proliferate again when it is necessary. Tumor cells keep this ability, however, $G0$ state tumor cells can undergo a special death programme, tumor necrosis, for some reasons not fully understood. Necrotic tumor cells are dead, hence, they can't resume cell cycle anymore.

The cell cycle of normal cells is well regulated by the so-called cell cycle control mechanisms [120, 121]. In each phase of a cell cycle, there is a so-called checkpoint at which cell cycle correctness is checked (see figure 3.6). At the end of the $G1$ phase, a cell checks (the $G1$-phase checkpoint) whether there are damages in its DNA and determines to proceed its cell cycle or suspend it to fix the errors. If everything is correct, it leaves the phase by entering the S phase. During the S phase, it needs to check the correctness (the S-phase checkpoint) of the synthesized DNA. Similarly, if an error is found, its cycle control mechanisms perform the according fixing programme. At the end of the $G2$ phase, the cell readies itself for mitosis by finally checking whether all DNAs has been correctly replicated (the $G2$-phase checkpoint). If the self-fixing programme fails to fix the cell cycle errors, the cell usually initializes a suicide programme.

The control mechanisms decide cell division and cell fate. Normal cells can divide for only a limited number of times before entering the programmed death procedure, apoptosis, even no error occurs. Besides, the control mechanisms only allow normal cells to divide when it is needed, for example, to compensate the loss of cell death or other reasons like a lesion to a tissue. Under the regulation of these mechanisms, the number of cells of a normal tissue keeps dynamically balanced, which is important for maintaining the biological homeostasis of the tissue.

Though much is known about the cell cycle process and its regulating mechanisms for normal cells, there is still much to be uncovered for the variations of cell cycle of cancer cells in regards to the development of cancers, since it is found that tumor cells lost some of the normal cell cycle control mechanisms because of various forms of genetic

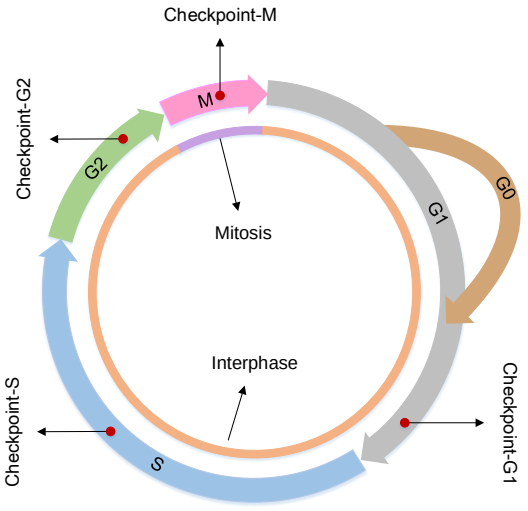


FIGURE 3.6: Diagram illustrating the phases of an entire cell cycle and the corresponding checkpoints. Checkpoints impose cell cycle control to guarantee that a cell has correctly completed all the requisite steps of one phase before entering the next one.

lesions [70], which makes their proliferation aberrant from the normal procedures. In other words, the tumor cell cycle is out of control, which is treated as one of the most important hallmarks of tumor cells [4, 54]. Out-of-control cell cycle is obviously a very general description of the traits of the cancer cell cycle. It would be more clear to list the specific cell cycle behavior manifested by tumor cells. The $G1$ phase is particularly interesting to cancer researchers, since cancer cells are found still sensitive to various stimulus in this phase [2] and mistakes in it may lead to cancer [121]. Some cell cycle checkpoints may be defective hence be skipped by tumor cells [122]. Some tumor cells can avoid entering the apoptosis programme of normal cells and therefore they obtain indefinite proliferative potential. The famous finding on the normal cell cycle times control is that cell cycle shortens the length of cell telomeres, which makes the proliferated somatic cells gradually lose the potential to proliferate and finally die (apoptosis) [123]. For these cells, an enzyme, telomerase, whose activity maintains the length of cell telomeres, is repressed with low activities, while the activity telomerase is highly activated in many cells of advanced tumors [124–126], which is similar to that of immortal cells [127]. And this observation provides a molecular level explanation to the aberrant tumor cell cycle (immortality of CSCs).

3.3.5 Cancer Cell Metabolism

The avascular tumor growth usually reach a growth saturation, which means mainly that the population (or the tumor size) stays relatively stable after a certain stage of

its progression. Figure 3.7 shows an experimental tumor (volume) growth curve with saturation. As one can see from the figure, tumors initially grow exponentially, then the growth slows down and finally reach a a platea.

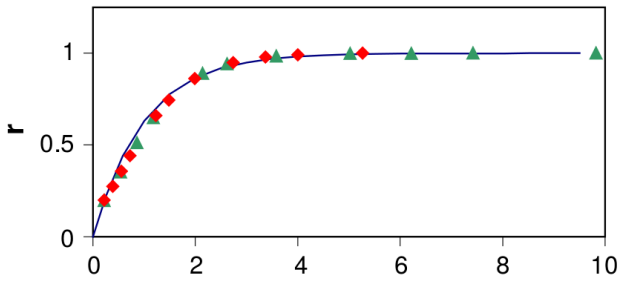


FIGURE 3.7: A tumor growth curve replenishing several tumor cell lines. The red diamond and the green triangle represent prostate and breast tumors from patients. Image adapted from Ref. [5].

Avascular tumors in growth saturation often show layered structures with cells of distinct proliferating states. As it is shown in figure 3.8, a core of necrotic cells resides inside a tumor, surrounding it is a rim of quiescent cells (in the $G0$ phase), which is further covered by an out layer of actively proliferating cells.

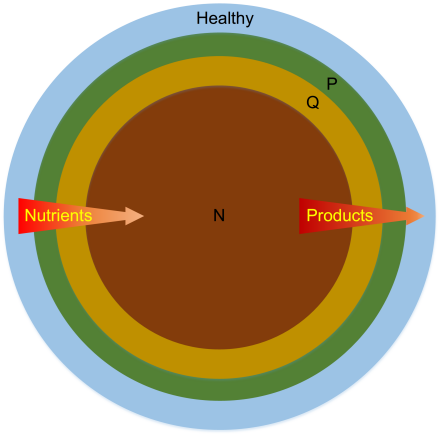


FIGURE 3.8: Diagram illustrating the three-layer structure of solid tumors consisting of the necrotic core (N) of dead cells, the quiescent rim (Q) of resting cells and the proliferating front layer (P) of actively growing and dividing cells. The outermost layer represents the healthy tissue. For avascular tumors, nutrients can be merely transported by molecular diffusion from the healthy tissue. So does the tumor cell metabolic products which are transported from the interior to the outer space of tumors.

There are a lot of possible reasons responsible for this phenomenon. The most popular one is the inefficient supply of nutrients for lack of tumor-own blood vessels. Cell behavior like growth, proliferation, differentiation, death, motion etc consumes a variety of biochemical molecules as nutrients for synthesizing new cellular components (proteins, DNA and etc) and producing energy. Driven by a variety of overactive growth

factors, tumor cells show much faster cell proliferation rate compared with that of the normal [128]. One of the consequences of this is nutrient depletion in the interior of a tumor, for example, inner hypoxia (depletion of oxygen) is commonly detected in most solid tumors. One possible explanation to this is that vascular supply of oxygen is insufficient to sustain the fast proliferation of tumor cells, since there is no special blood vessel for early tumor tissues. To get understand this phenomenon, understanding the energy metabolism of tumor cells has been a hot direction in cancer research.

The energy metabolism of tumor cells have been found varies significantly compared with that of normal cells [129–131]. This trait of tumor cells was first reported by Otto Warburg and named as the Warburg effect [129]. Normal cells under sufficient oxygen experience aerobic respiration oxygen, which breaks down glucose into carbon dioxide and water via the biochemical reaction 3.1 (see figure 3.9 (A)). Under the condition of insufficient oxygen, however, normal cells are limited to using only glycolysis (anaerobic respiration), which breaks down glucose into lactate through the biochemical reaction 3.2 (see figure 3.9 (B)). The aerobic respiration is much more energy production efficient than the anaerobic respiration, since it produce 36 ATP molecules consuming one glucose molecule while glycolysis only produces 2 ATP molecules in comparison (see figure 3.9). The point is that Warburg discovered that tumor cells produce energy relying largely on glycolysis even when they are exposed to adequate oxygen. This unusual energy metabolism of tumor cells seems to be contradictory to the reality that fast tumor growth needs more energy. However, tumor cells have found another way to partially compensate the inefficiency of energy production by consuming glucose in an extremely high rate.

$$C_6H_{12}O_6 + 6O_2 -> \ 6H_2O + 6CO_2 \tag{3.1}$$

$$C_6H_{12}O_6 -> \ 2C_3H_6O_3 \tag{3.2}$$

3.3.6 Cancer Cell Biophysics and Biomechanics

The development of a tumor involves a variety of changes of biomechanical and biophysical properties of tumor cells, e.g. tumor cell are more stiffer [132] that normal cells, which is closed related to the mechanical interactions between tumor cells and their surroundings, such as neighboring cells, the ECM and the extracellular fluid (ECF). Furthermore, this alternations are found in turn may significantly influence complex physiology of tumors and tumor cells, such as the survival, proliferation, invasion, migration tumor cells and tumor metastases. Besides, measurements of cell biophysical

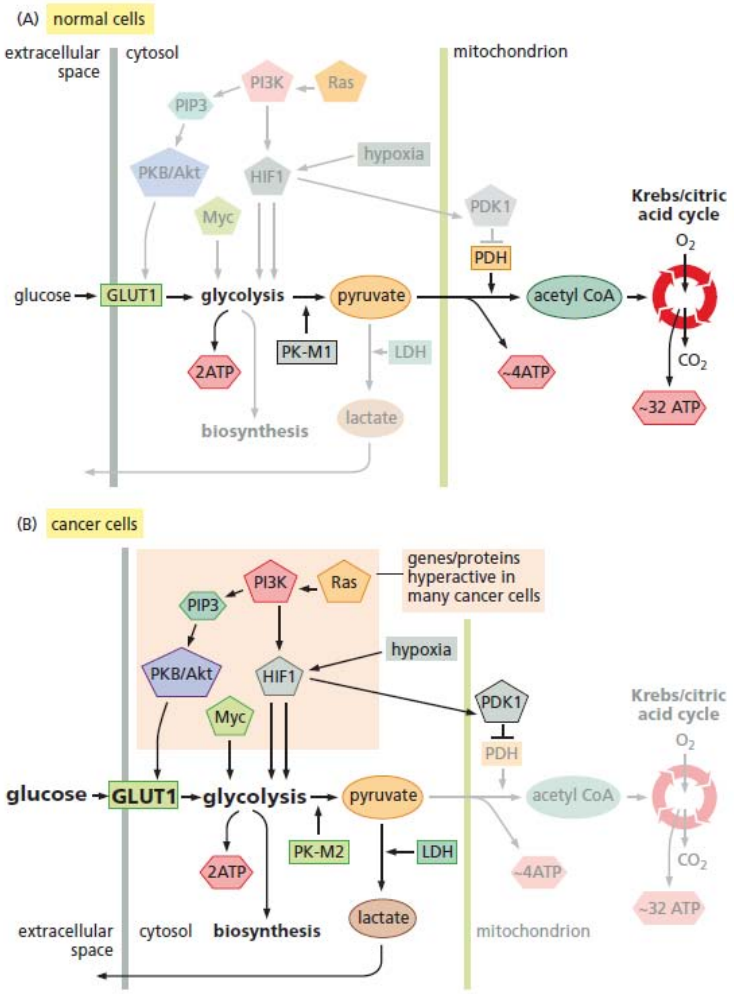


FIGURE 3.9: Metabolism of glucose in normal and cancer cells: aerobic versus anaerobic respiration. (A) With adequate oxygen, glucose is broken down into carbon dioxide in healthy cells via aerobic respiration, generating as many as 36 ATP molecules per glucose molecule. (B) In cancer cells and healthy cells short of oxygen, glucose is broken down into the lactate via anaerobic respiration, generating 2 ATP molecules per glucose molecule. Image adapted from Ref. [2]

property variances can be used as indicators of the cell biological state to identify types and stages of a tumor [2, 132].

In vitro, cultivated normal cells need to anchor to the solid substrate to survive and grow (anchorage dependence), and they stop proliferating when they get in contact (contact inhibition). These control mechanisms are essential to normal tissues for maintaining their biological homeostasis, which is essential for a normal tissue to perform its functions. However, tumor cells show a much stronger survival and proliferation ability compared with that of normal cells. They divide much faster and can gather together forming up a lump of multi-layer cells on culture dishes, that means contact inhibition of normal cells does not work for them. It seems that they have much stronger willing

to gather together, however, their gathering is obvious not for performing a specifically function meaningful to the host but rather mostly harmful.

Some proteins synthesized by cells are released into the extracellular environment. Part of these proteins assemble and form a network structure, the ECM, which builds connections among cells. Some other part of theses proteins reside on the cellular membranes functioning as receptors, like cell signaling pathway receptors, that bind with corresponding ligands (another type of proteins) on the neighboring cell membrane forming up connections between cells. Cell-cell and cell-ECM connections help form and sustain complex structures of tissues. Tumor cell transformation involves significant variances in the ECM. For example, they are found lost the ability to assemble a functional tissue structure as normal cells do. This variances are thought to facilitate tumor cell local invasion and metastases [10, 16, 17, 133].

Some other types of proteins assemble in inner cellular space creating polymeric biomolecules, the cytoskeleton. The cytoskeleton contains several different forms including actin microfilaments, intermediate filaments, and microtubules [134]. In eucaryotic cells, they generally work together connecting the cellular membrane and the nucleolus to sustain the rigidity and morphology of single cells [10, 134, 135]. Besides, in concert with accessory proteins, the cytoskeleton also plays an important role in some cellular processes, such as mechanotransduction, mitosis and locomotion [2, 10, 136].

Since tissue cells are undergoing proliferating rather than in a equilibrium state, they change and are changed by the extracellular environment biophysically. During this process, the degree of malignancy of cells generally increase, which facilitate survival and progression of tumor cells. Cell-cell contact and the extracellular environment factors can affect the expression and distribution of some integrin molecules [137]. A typical alteration of tumor cells is that their cellular membranes become less rigid than that of normal cells, which allows them to undergo easier shape deformation [138]. Besides, strong competition for space has been detected between cells in tumor tissues [128], which could be caused by enhanced cell proliferation. This space competition indicates high stress tension among cells in tumor tissues, and it has been found that stress play a important role in determining the growth of tumors [139–142].

Chapter 4

The Design and Implementation of TUGME

In this chapter, the design and implementation of the software environment TUGME (TUmor Growth Modeling Environment) are introduced in details. In the first section, we summarize the general perspectives of this software environment. Then, basic design principles for fulfilling the goals of TUGME are highlighted in the second section. Thirdly and most importantly, modules of TUGME are separately introduced in details. Finally, key steps associated with conducting a simulation with TUGME are briefly introduced.

Jörg Eisele as the co-developer of this project has contributed to the design and implementation of TUGME for his master study. Specifically, he introduced the C++ traits features to our project, set up DUNE and implemented the module `Biomechanics` of TUMGE.

4.1 General Perspectives of TUGME

General Perspectives of TUGME include:

- TUGME allows multiple spatial-temporal scales modeling and simulation of the tumor growth process. As it has been emphasized several times in the early chapters, tumor growth is a multiple spatial-temporal scale problem. TUGME is expected to fill the gaps between some certain spatial-temporal scales. The spatial scale covers from micrometers to centimeters, and the time scale ranges from seconds to months.

- Tumor models of TUGME are radically course-grained. Though agent-based method flexibly allows one to create more fine-grained tumor models at sub-cellular scales, modeling mature tumors using these models is very challenging for the explosive increase of the computational cost of model solving. Specially, TUGME is designed to facilitate tumor growth modeling and simulation from the single cell level to a multicellular tumor tissue. Hence, sub-cellular details aren't explicitly considered.
- TUGME is expected to be a tumor growth modeling and simulation framework. Its most important feature is the flexibility for easy model reuse and extension. This feature is highly emphasized during its design and implementation.

4.2 Basic Design Principles of TUGME

Considering the complexity of designing and implementing a software environment like TUGME, it isn't wise to be too ambitious in the beginning but rather to split a complex problem into smaller ones that can be addressed one after another. Flexibility for easy model reuse and extension is the mostly emphasized feature of TUGME. To do so, several important techniques are adopted during the design and implementation of TUGME.

First of all, data structures and algorithms are separated. Doing so overall is of great help from the perspective of flexibility, however, it may need to sacrifice memory efficiency and performance of the code. An algorithm may be inherently coupled with a specific type of data structure to perform its best efficiency. However, it is very common that large data sets are shared by different algorithms, whereas, no one data structure can best meet the requirements of all of them. For example, the cell-cell neighboring information is a large data set that is needed by several modules of TUGME. In this case, separating data structures and algorithms makes the maintaining of the code easier. In fact, this technique is very commonly applied in large software projects in computer science.

Secondly, functionalities are split into modules (modularization). Modularization is also very commonly used in software development in computer science. Several very basic rules are important to follow for modularization. First, each module usually implements one specific functionality and a complex module can be further divided into sub-modules, which makes it easier to design, implement and maintain. Second, dependences between modules are usually minimized so that modifications to one module doesn't ask for modifications to the others related.

Thirdly, dynamic and static polymorphisms are widely used. In TUGME, complex functionalities are decomposed into functions and a hierarchic structure of C++ classes is introduced with multi-layer abstraction, which enables flexibility of different implementations inheriting from different super classes. The trick is known as dynamic polymorphism in software development. It is flexible, however, not very efficient with respect to program execution. In contract, static polymorphism (generic programming) is more efficient since the template arguments are specified at compiling time instead of running time. However, it makes the code more difficult to understand and pushes the users to specify the template arguments explicitly.

Finally, C++ traits is widely applied associated with generic programming to provide type information of the fields of C++ classes. With traits, one doesn't need to know the actual type (C++ `int`, `double`, `bool` etc.) of a field of a C++ class, since it is wrapped and normally termed `fieldNameType`, which helps keep the consistence of field types.

4.3 Modules of TUGME

TUGME currently consists of five main modules:

`Cells` defines the interface and implementation of single cell agents.

`Morphology&topology` provides the interface for calculating the morphological and topological information of cells.

`Biomechanics` models the cell-cell and cell-environment mechanical interactions as well as biomechanics-driven cell motion.

`Biochemicals` models the transport and metabolism of biochemical molecules.

`Cell_cycle` models cell proliferation and dynamics associated.

Generally, each module of TUGME models a separate aspect of tumor tissues or cells. In the next section, these modules are to be introduced in details one by one.

4.3.1 Brief Introduction to DUNE

TUGME is implemented using the computer programming language C++. It is basically built on a software tool named DUNE (the Distributed and Unified Numerics Environment) [143]. We begin this section with a brief introduction to DUNE.

DUNE is a C++ software tool for solving partial differential equations (PDEs) with grid-based methods. DUNE is highly modularized. Its modules used by TUGME include:

- dune-common contains implementations of basic data structure types, basic functionalities, some constants etc, on which the other modules are built based. Hence, most functions of this module normally aren't directly called by users.
- dune-geometry is a module mainly for coordinate mapping between local (mapped) and global (physical) coordinates.
- dune-grid contains implementations of a variety of grids, structured, unstructured, conforming, non-conforming, periodic boundary grids etc. A DUNE grid is a container of entities, which are constructed recursively from low to high levels. Iterators are provided for accessing but not storing the entities at different levels, hence, random accessing is basically impossible in most cases. In addition, data can be attached to grids and writers are provided for data output, for example, VTK format output files are supported and can be visualized using VTK and Paraview [144].
- dune-istl is the abbreviation of the Iterative Solver Template Library. This module provides implementations of block vectors, block matrix, sparse matrix etc. More importantly, many linear system solvers, such as the preconditioned BiCGStab (stabilized bi-conjugate gradient method), the preconditioned CG (the conjugate gradient method), superLU, are implemented or integrated in it.
- dune-localfunctions provides a separate library of shape functions that can be reused by different FEM implementations. Specially, general C++ interfaces of shape functions are contained.
- dune-pdelab provides discretization schemes and solvers for PDE problems based on DUNE. This module is built based on the modules introduced above of DUNE. Hence, it calls the functions of these modules directly.

DUNE has other modules, for example, dune-fem [145], however, TUGME currently only uses the modules introduced above. In the next sections, where specific functionalities of DUNE are used, we will introduce more details about how the involved modules of DUNE are adopted in TUGME.

4.3.2 Cell Agents: Cell Types and States

In TUGME, each single cell is modeled as an individual agent, which is characterized by a set of variables and the corresponding state-updating methods.

To facilitate model reuse in TUGME, the agent model of single cells is split into multiple levels of C++ classes, which indicates different degrees of abstraction of cell models. The data members of cell agents basically mark the state of a cell or distinguish it from others. Some data members are shared by different models, while some others are model-dependent. A cell model at a higher abstraction level holds data members shared by the lower level ones, which is realized via C++ class inheritance. For example, cell position (`coordinate`) is a common state variable for all cell models in TUGME, while the variable (`isCSC`), distinguishing CSCs from common tumor cells, is necessary only by cell models considering CSC features. Hence, `coordinate` is a member of the cell model at the highest abstraction level, while `isCSC` is a member of the cell models at a lower abstraction level.

In TUGME, `cell` is the C++ class of the agent model of single cells at the highest abstraction level, as it is shown in figure 4.1. Currently, it has these data members listed in table 4.1. The `UUID`, as it is named, is to provide an identifier that uniquely marks each cell universally, which is left for parallel simulations. The `UID` is a local unique identifier which can only guarantee the uniqueness in sequential simulations. As many agent-based tumor models do [60, 61, 146–148], each tumor cell is treated as a sphere in the absence of biomechanical stimuli in TUGME, where the sphere is characterized by its center of mass (`coordinate`) and its radius. In addition, cell volume is monitored during cell growth and division. Obviously, the volume of a cell with no shape deformation is its spherical volume, while its actual volume varies for deformation on the cellular membrane and cell growth.

Field names	Data types
`UUID`	`BOOST_UUID`
`UID`	`STT::UIDType`
`coordinate`	`STT::coordinate3DType`
`radius`	`BTT::NumberType`
`celltype`	`cellType`: healthy or tumor

TABLE 4.1: Data members of the cell agent class (`cell`) in TUGME. `STT` and `BTT` are its template arguments. `BTT` represents the `basicTraitsType`, where the very fundamental data types are declared. `STT` stands for `superTraitsType`, which dynamically integrates all traits of the C++ classes of a tumor model in TUGME. `UIDType` is actually a C++ `unsigned long int` type. `coordinate3DType` is a triple of numbers of `NumberType`, and `NumberType` is the C++ `double` type.

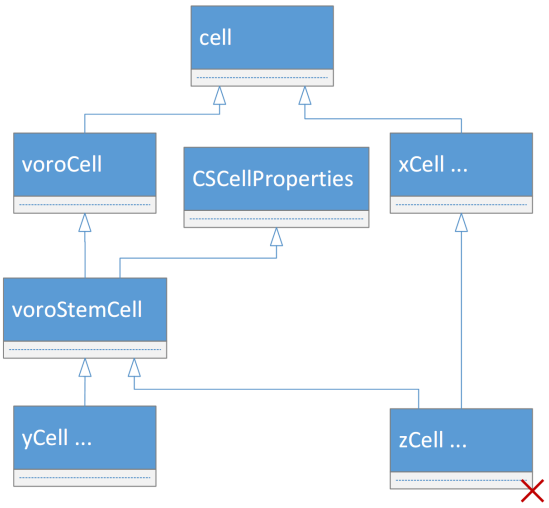


FIGURE 4.1: The hierarchical structure of the C++ classes of agent models of single cells. All cell agent classes in TUGME have to inherit from the basic class `cell` directly (e.g., `voroCell` and `xCell`) or indirectly (e.g., `voroStemCell`). Although C++ allows multiple inheritance, for example, `zCell` in this figure, this isn't allowed in TUGME, since it can be seen clearly from this figure that `zCell` inherits `cell` repeatedly from both `voroCell` and `xCell`. `CSCellProperties` defines the CSC features. It is separated from `voroStemCell` for the possibility that one needs to model CSC properties without considering the morphology and topology of tumor cells. `xCell`, `yCell` and `zCell` don't exist actually, they are just general representations of possible cell agent classes that may be implemented in the future.

Since `cell` is at the highest abstraction level, all other cell agent classes have to inherit from it in TUGME. Currently, there are two cell agent classes actually provided, namely `voroCell` and `voroStemCell` as it is shown in figure 4.1, both inheriting from `cell` directly or indirectly. One of the basic assumptions of `voroCell` is to represent a multicellular tumor tissue using a 3D radical Voronoi tessellation. This assumption is based on an observation of a striking similarity in shape between the 3D radical Voronoi tessellation cells and some real tissue cells as it is shown in figure 4.2. As a matter of fact, Voronoi tessellations have been used in the topological analysis in cell biology [149–151], epithelial tissue renewal [152] as well as cancer modeling [47, 148, 153, 154].

Figure 4.3 is a brief version of `voroCell`. For complex data members, for example a C++ `vector`, "`getAsRefXXX()`" methods are provided for data access. There are actually two overloaded methods for each data member. One of them returns a reference to the corresponding data member with full modification authority. This design actually breaks the information hidden rule of object-oriented programming (OOP). The underlying idea is that calling this type of methods in the cases when data modification is necessary can avoid copying larger data sets, which can be very time-consuming. For example, copying a C++ `vector` with a large number of elements. In the case, when one only needs to access the data without modifications, another type of methods

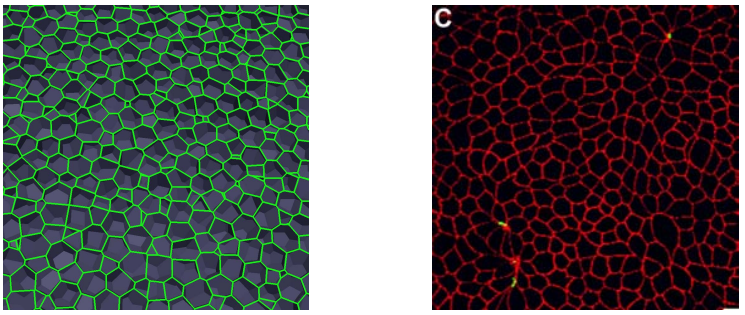


FIGURE 4.2: The shape similarity between 3D radical Voronoi cells and human kidney epithelial cells. The figure on the left is a section cutting through a 3D radical Voronoi tessellation, where the convex polygon of green edges is a Voronoi cell. The figure on the right is an image of Madin-Darby canine kidney (MDCK) epithelial tissue cells obtained with confocal microscopy adopted from reference [6]. The MDCK epithelial tissue cell shows the polyhedron-like shape.

should be called to get a constant reference to the target data. Theoretically, modularization asks for modules interacting with each other only through interfaces. Here, the agent class is a special module because it basically only stores single cell information, while all the functionalities of tumor cells are modeled in other modules (see figure 4.3) that inevitably need to access this module for the the formation of single cells. Obviously, this situation is due to modularization, and it can be avoided by integrating the functionalities into the cell agent class. However, doing so would definitely result in an extremely complex cell agent class which is much more difficult to maintain than the current one. The even worse thing is that doing so would almost lose all the flexibility of modularization.

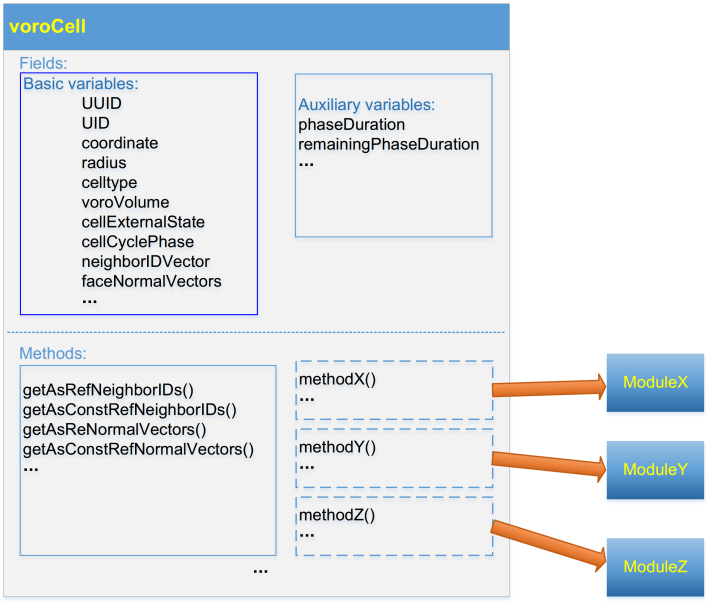


FIGURE 4.3: The Voronoi cell agent class `voroCell`.

In order to distinguish between CSCs and common tumor cells, `CSCellProperties` is introduced. Inheriting from it, the CSC features can be modeled, for example, `voroStemCell` in figure 4.1. In `CSCellProperties`, CSCs are assumed to have an indefinite proliferative potential, and they differentiate with a certain probability during their proliferation, which is modeled by setting at least one of the daughter cells of a CSC to be still a CSC. In contrast, the proliferative potential of a common tumor cell is very limited. If a common tumor cell is the daughter of a CSC, it is endued with the highest proliferative potential, and the proliferation potential reduces by one after each cell division. When using up its proliferative potential, a common tumor cell undergoes apoptosis. Besides, the daughter cells of a common tumor cell are currently assumed to be only common tumor cells, since we assume that common tumor cells can not dedifferentiate into CSCs.

Beside the cell agent class, a C++ class named `systemState` is introduced in TUGME as a counterpart to the multicellular tumor system. This class basically maintains all the state information of the whole tumor system and the settings of simulations. First of all, all the objects of cell agents are stored in this class. Secondly, the computational domain boundary of the simulated tumor system and the boundary condition type are stored here too. Finally, the class also keeps part of the simulation log. `states` (data structure type: `vector<cells>`) is used to store the objects of cell agents. Since `vector` can be randomly accessed by its index which is treated as a temporary identifier (`TID`) of tumor cells. It is temporary because the position of the object of a tumor cell may be changed for inserting newly born and deleting dead ones.

By now, we have mentioned three different identifiers, including the `UUID`, the `UID` and the `TID`. This may be confusing. First, the `UUID` and the `UID` have basically the same functionality but for different situations. Specifically, the `UID` is to uniquely mark a cell in sequential simulations, while the `UUID` is for parallel and distributed simulations. Second, the `TID` is actually the index number indicating the position where the object of a tumor cell agent is stored in `states`. The `TID` instead of the `UID` is currently used to access the object of a cell agent when it is known. We strongly suggest to use the `TID` whenever it is known, since accessing an object of a cell agent using either the `UID` or the `UUID` instead of the `TID` is search-dependent, which isn't time efficient ($O(N)$) compared with that of using the `TID` ($O(1)$). Since the parallel version of this project is still in development, the `UUID` isn't used in current sequential simulations.

We use the C++ `vector` instead of other data structures, like the C++ `map` (the `map` or the `hash_map`) or the `list`, because of its high access efficiency. During the running of simulations, the cell agent object is accessed very frequently both sequentially and randomly. One case is that a cell needs to know the state information of its neighbors

for deciding its next-step actions. The other case is that a newly generated cell needs to be added to the system. And the third case is that dead cells need to be removed from the system. The `list` is suited to inserting or deleting elements at random positions, while it has to be accessed sequentially. The `map` can be accessed randomly by its keys, however, its accessing (inserting and deleting) efficiency depends largely on the hash function of the key. Besides, for each single element, the `vector` is the most memory efficient, since each of its element only needs the memory for the data of the element itself, while an element of the `map` needs extra memory for its key and an element of the `list` needs extra memory for its pointers. However, the `vector` needs to preallocate space (the `vector` capacity) for its elements, hence, extra memory space is required beyond what is actually needed. And when the capacity of the `vector` is depleted, reallocating space is needed. Theoretically, reallocating memory could happen at any time when an element is added, which can be expensive for the `vector` with a large number of elements. However, the default mechanism of C++ actually avoids doing so too frequently.

Inserting an element to the `vector` is restricted only to its end (`push_back()`), since the cost of doing so is constant $O(1)$. Deleting (`erase()`) an element from it normally needs to shift the elements after the given element, whose average cost is $O(n)$. The exception is to delete the last element (`pop_back()`), whose average cost is also $O(1)$. In order to reduce the cost of deleting some randomly locating elements from `states`, a tricky algorithm is developed. The basic idea is to avoid deleting an element locating within (not at the end of) `states` by element replacement to take the advantage of the constant cost of deleting the last element of the `vector`. As it is shown in figure 4.4, the to be deleted element is replaced by the last element that should not be deleted.

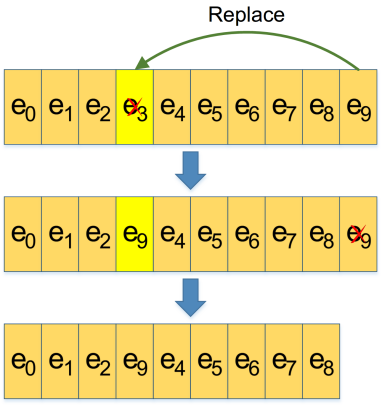


FIGURE 4.4: Diagram illustrating the basic idea of avoiding element shift when removing an element from a C++ `vector` by element replacement.

The trick is put forward by the co-developer, Jörg Eisele, of this project. It is extended

by taking into account the deleting operations of dead cells and inserting of newly born cells together in order to further reduce the deleting and inserting operations. The pseudo-code of the final algorithm is shown in figure 4.5.

```
1.  while V2 is not empty
2.      sequentially search for the first element Ei marked to delete in V1
3.      If found // not reach the end of V1
4.          replace Ei by the last element of V2, and
5.          remove the last element of V2
6.      else // indicate no element needs to delete from V1
7.          stop while loop // some elements in V2 may not be added
8.  If V2 is not empty
9.      while V2 is not empty // add the rest of elements of V2 to V1
10.         Insert the last element of V2 to the end of V1, and
11.         remove the last element of V2
12. else // some elements in V1 may not be deleted
13.     while there is an element Ei marked to delete in V1
14.         sequentially search for the first element should not be
            deleted from the end of V1 to the position of the Ei
15.         If found Ej // an element locating after Ei in V1
16.             replace Ei by Ej, and
17.             mark Ej to delete
18.         else // all elements locating after Ei are marked to delete
19.             stop while loop
20.     erase all the elements from Ei (including Ei) to the end of V1
```

FIGURE 4.5: Pseudo-code showing the algorithm of adding and deleting elements from the C++ `vector` storing the single cell agents. V_1 and V_2 stores the agents of existing and newly generated cells separately.

The overhead introduced by this algorithm comes from element replacement. Theoretically, all information of an element should be copied from the replacing cell to the one to be replaced. However, transient information, e.g. the Voronoi information that has to be recalculated in every simulation step, doesn't need to copy. Furthermore, copying this kind of information may be actually very time-consuming. To avoid this overhead, we introduce a specific method named `copyCellInfo()` that only copies the useful single cell information. Since, users may introduce new data members to single cell agent models, it is their responsibility to decide whether the newly introduced variables should be actually copied or not by rewriting the `copyCellInfo()` method.

The time cost for inserting and deleting some elements from the C++ `vector`, `list` and `map` has been tested in order to see more precisely their performance differences.

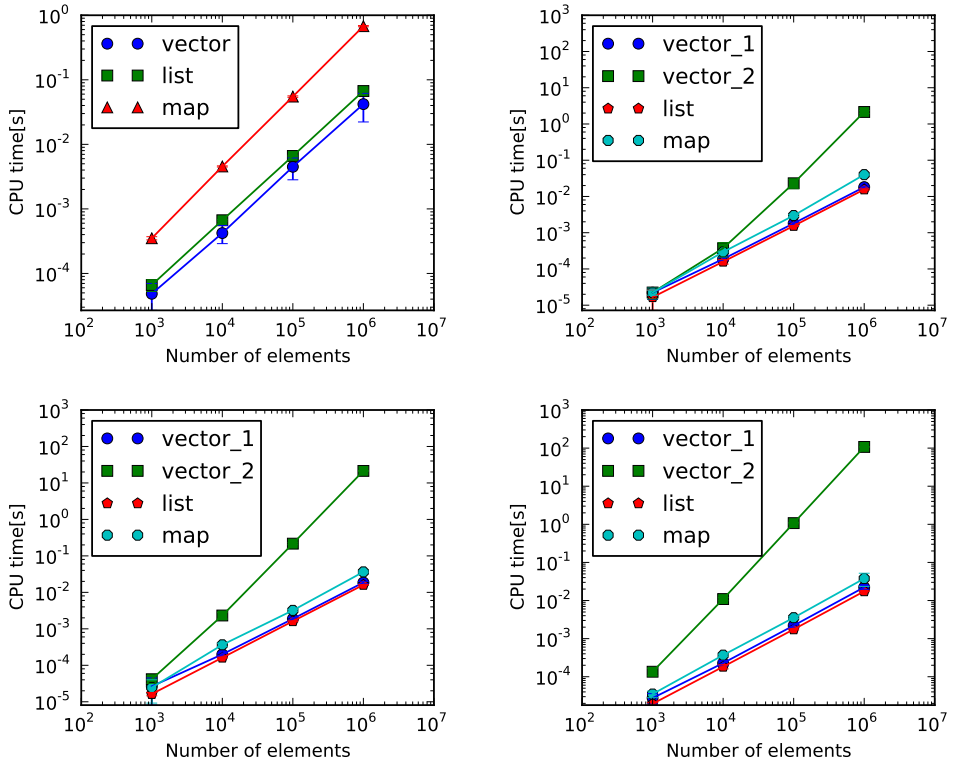


FIGURE 4.6: The performance test of three different C++ standard data structures, namely the `vector`, the `list` and the `map`. Two operations are tested, which are sequential inserting and random deleting. For the `vector` and the `list`, inserting is done only to the end in stead of to a random position, since the stored position of cell agent object in `vector` doesn't matter in TUGME. Theoretically, the cost of inserting an element to any position of the `list` would be no difference, however, it varies significantly for the `vector` ($O(1)$ versus $O(n)$ averagely, where n is the size of the `vector`). The upper left panel shows the CPU time to add certain numbers, 10^3, 10^4, 10^5 and 10^6, of elements to the `vector`, the `list` and the `map`. The rest three panels (the upper right, the lower left, and the lower right) show the CPU time of randomly deleting different numbers, 0.1%, 1% and 5%, of elements from different size, 10^3, 10^4, 10^5 and 10^6, of these data structures. `vector_1` and `vector_2` separately represent our optimized deleting method and the standard C++ deleting mechanism.

Results are shown in figure 4.6. As it is shown in this figure, the algorithm proposed by us significantly reduces the CPU time for randomly deleting elements from the `vector` compared with the default deleting mechanism of C++. The highest speed-up between them is about 10^4 (see the lower right panel of this figure). Obviously, the more elements to be deleted, the larger the speed-up will be. Though, the optimized method needs a little bit longer CPU time compared with that of the `list` for random deletion operations, whereas, it outperforms the `map`. In a word, our optimized method allows the `vector` a performance of random deletion similar to the `list` and the `map`, which definitely helps improve the simulation efficiency of TUGME.

4.3.3 Cell Morphology and Topology: Cell Shape Deformation and Neighborhood

As it has bee mentioned in section 4.3.2, the 3D radical Voronoi tessellation (a type of weighed Voronoi tessellations) is used to represent the multicellular tumor tissue. Specifically, each convex polyhedral Voronoi cell mimics a tumor cell in terms of tumor cell shape (the cell morphology) and the neighboring relationship with other cells (the cell topology).

4.3.3.1 Introduction to Voronoi Tessellations

For a given set of points P in a space S, a common Voronoi tessellation is defined as a partition that divides the space S into many connecting but non-overlapping sub-spaces ($subS_i$, $S = \bigcup_{i=0}^{N-1}\{subS_i\}$, N is the number of points in P, see a 2D example shown in figure 4.7) according to the neighboring relationship of the given points. The common Voronoi tessellation can be described theoretically by equation 4.1, where p_s stands for a single point within the sub-space $subS_i$, p_i and p_j are points from the point set P. As figure 4.7 shows, equation 4.1 defines an area in two dimensions (a volume in three dimensions). Any point p_s within $subS_i$ is closer to the point p_i than to any other points in P.

$$d(p_s, p_i) \leqq d(p_s, p_j), i \in N,\ \forall j \in N\ \&\ j \neq i,\ for\ \forall p_s \in subS_i \tag{4.1}$$

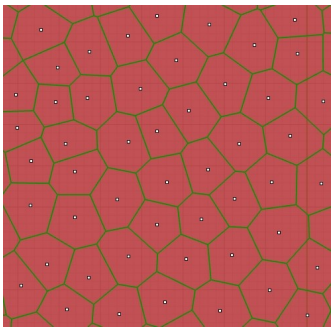


FIGURE 4.7: Image illustrating a 2D common Voronoi tessellation. White dots stand for the position of the Voronoi cell (the spherical center of tumor spheres in three dimensions). Convex polygons bounded by green line segments represent Voronoi cells (convex polyhedra in three dimensions). The green line segment (edges of polygons) stands for the contact interface between two directly neighboring cells. In three dimensions, these contact interfaces are replaced by convex polygonal areas. A 3D radical Voronoi tessellation with two cells can be seen in figure D.1 in appendix E of this thesis.

The radical Voronoi tessellation is one type of the so-called weighted Vovonoi tessellations, since the interface between two neighboring cells is weighted by their radii. One 3D scenario is that each point in the point set P represents the mass center of a tumor cell sphere with the radius R. Equation 4.1 is modified into equation 4.2 in order to describe the radical Voronoi tessellation, where the new introduced terms R_i and R_j stand for the radii of two neighboring cells. Figure 4.8 illustrates a single common Voronoi polygonal cell (on the left) and a radical Voronoi cell (on the right) in two dimensions. As it is shown in this figure: 1) both common and radical Voronoi cells are convex polygons; 2) edges of radical Voronoi ploygons are shifted from the cells with a large radius to the ones with a small radius compared with that of the common Voronoi polygons. One can easily understand this from their theoretical descriptions which are equations 4.1 and 4.2,

$$d^2(d_s, p_i) - R_i^2 \leqq d^2(d_s, p_j) - R_j^2, i \in N,\ \forall j \in N\ \&\ j \neq i,\ for\ \forall s \in subS_i \qquad (4.2)$$

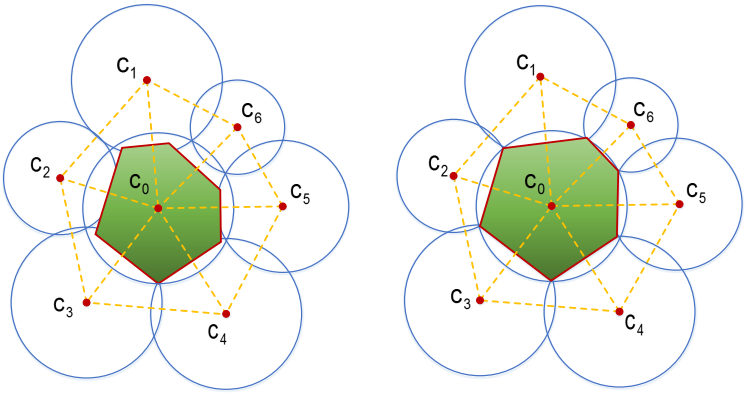


FIGURE 4.8: Diagrams comparatively illustrating a single polygonal Voronoi cell of the common and radical Voronoi tessellation in two dimensions. The green areas bounded by solid red line segments represent the Voronoi cell and the dashed yellow line segments indicate the neighborhood between cells. Blue circles indicate the default spherical periphery of single cells. For the common Voronoi cell (on the left), the solid red line segments and its corresponding dashed yellow line segments equally bisect each other perpendicularly. In contrast, for the radical Voronoi cells (on the right), solid red line segments have been shifted towards the smaller radius cells though they are still perpendicular to the dashed yellow line segments and equally bisected by them.

Another possibility for the weighted Voronoi tessellation could be theoretically described by equation 4.3, which isn't called radical Voronoi tessellations anymore, since it produces polygons of curved edges in two dimensions or curved surfaces in three dimensions for neighboring cells with different radius sizes. Hence, Voronoi cells aren't

always convex anymore according to this definition.

$$\frac{d^2(p_s,p_i)}{R_i^2} \leqq \frac{d^2(p_s,p_j)}{R_j^2}, i \in N,\ \forall j \in N\ \&\ j \neq i,\ for\ \forall s \in subS_i \tag{4.3}$$

This is quite easy to prove. Supposing that the coordinates of points p_s, p_i and p_j are (x,y), (x_i,y_i) and (x_j,y_j) in two dimensions, equations 4.2 and 4.3 can be reformulated into equations 4.4 and 4.5 correspondingly. Comparing these two derived equations, one can see that directly, equation 4.4 doesn't involve the second or higher order of the variable x or y. Hence, for two specific points p_i and p_j, the interface between the two neighboring Voronoi cells is a straight line. However, equation 4.5 leads to curved interfaces for the second order of both x and y ($R_i \neq R_j$) are involved. This conclusion can be generalized to the three dimensional case very easily, where polygons with curved edges in two dimensions are replaced by polyhedra with curved surfaces.

$$2xx_i + 2yy_i + R_i^2 \geqq 2xx_j + 2yy_j + R_j^2 \tag{4.4}$$

$$\frac{(x^2 - 2xx_i + y^2 - 2yy_i)}{R_i^2} - \frac{(x^2 - 2xx_j + y^2 - 2yy_j+)}{R_j^2} \leqq \frac{(x_j^2 + y_j^2)}{R_j^2} - \frac{(x_i^2 + y_i^2)}{R_i^2} \tag{4.5}$$

4.3.3.2 Calculating the Radical Voronoi Tessellations in Three Dimensions

The calculation of the radical Voronoi tessellation in three dimensions is complex and can be very time-consuming for a large number of points (cells). Most algorithms for the tessellation calculation is based on the idea that the Voronoi tessellation is the dual of the Delaunay triangulation [155]. By searching around software libraries for doing so, several packages are found on the market, such as QHull [156], Triangle [157], libvoronoi, Voronoi++ and Boost polygon library. However, they aren't suited to systems involved large number of points. Fortunately, a package named Voro++ [158], which is particularly designed for research problems in materials science, physics, and engineering that frequently involve large systems of particles, fulfills our perspectives very well.

Voro++ is adopted to calculate the three dimensional radical Voronoi tessellation in TUGME. It is an open source software library written in C++. Several features of Voro++ are particularly interested and useful with respect to the implementation of TUGME. First, wide use of the object-oriented and the generic programming techniques allows it to be easily integrated into other C++ programs like TUGME. Second, it

calculates each cell independently, which make it easier to be parallelized. Third, it is well optimized for high efficiency as the author has declared. Last but not the least, Voro++ flexibly allows three boundary types, the periodic, the non-periodic and the free boundary condition types (see figure 4.9).

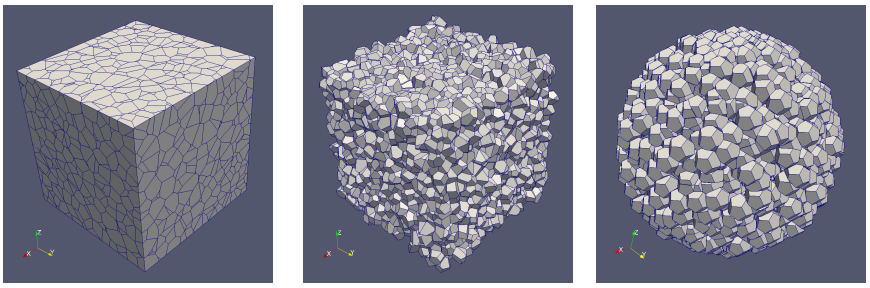


FIGURE 4.9: Illustration of 3D radical Voronoi tessellation with non-periodic (the leftmost panel), periodic (the middle panel) boundaries and Voronoi cells of dodecahedral shapes (the rightmost panel).

Different boundary condition types are important for the serious use of Voronoi tessellations, since the default Voronoi algorithms can not precisely deal with cells next to the computational domain boundary, which usually results in abnormal Voronoi cell shapes. This boundary condition problem can be attacked by imposing a static boundary, which cuts the boundary Voronoi cells stiffly. The leftmost panel in figure 4.9 illustrates a radical Voronoi tessellation of this case with a cubic computational domain. The other way to handle this problem is to adopt the periodic boundary condition. In this case, the whole computational space should be filled with roughly uniformly distributed cells, otherwise the volume of some Voronoi cells may be abnormally large. The middle panel in figure 4.9 shows a radical Voronoi tessellation of this case with a cubic domain too. The third scheme is to define a default shape for single cells, for example, a regular polyhedron. This scheme actually abandons the domain boundary and is more flexible to use for many problems like tumor modeling. The rightmost panel in figure 4.9 illustrates a radical Voronoi tessellation of this case, where each cell is a regular dodecahedron by default. All these three boundary types are supported by TUGME.

Voro++ allows installation on Linux operation systems like Ubuntu and Debian or direct use of its original source code [158]. In TUGME, we introduce a C++ class, `voroPP` (see figure 4.10), which wraps the original source code of Voro++. We do so instead of installing Voro++ on our systems for reasons including: 1) the functionalities provided by Voro++ aren't easy to use directly; 2) the data structures used by Voro++ are particularly optimized for high executive efficiency, however, they aren't direct to use and not suited to frequent access; and 3) part of the original source code may need

to be modified in certain conditions, for example, adjusting its computational tolerance.

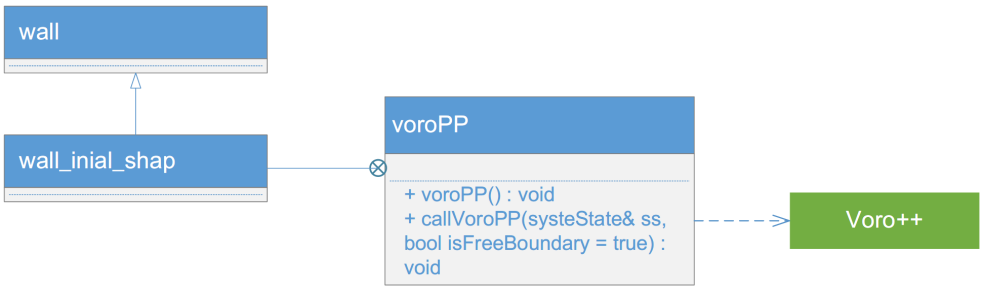


FIGURE 4.10: The `voroPP` interface. This class wraps the Voro++ source code. The data structure types for storing the Voronoi information are declared in a traits named `voronoiTratis` (not shown here). `wall_initial_shape` is an inner class of `voroPP`. It defines the default shape (a dodecahedron) for individual radical Voronoi cells.

The input of the voroPP interface is very simple, including the coordinate and the radius of tumor cells. The output Voronoi tessellation information is much more complicated, where data structures of TUGME for storing them are specially tailored for high accessing efficiency. Besides, the interface is responsible for converting the data structures of Voro++ into that of TUGME, which is done during calculating the Voronoi information. Two main points have to be emphasized here. First, the converting process is transparent to users. Hence, users just need to know how to access the data structures of TUGME instead of understanding the original code of Voro++. Second, data structure conversion doesn't introduce too much computational cost, since comparing to the time needed by Voronoi tessellation calculation, it can be neglected according to the our practical using experience with Voro++.

Table 4.2 shows the Voronoi information of each cell and the corresponding data structures in TUGME. The Voronoi information is part of the variables of the `voroCell` class. To access the Voronoi information correctly, one has to understand the data structures declared in `voronoiTraits`. For those who want to rewrite the cell class, the way of how the coordinate, the radius of tumor cells and the system boundaries and the boundary type are stored in `systemState` class should remain consistent with the existing implementation in order to make sure that `voroPP` can work correctly. Detailed explanations on how these data structures are associated and how they can be accessed correctly can be found in appendix E of this thesis.

Variable (set) types	Data structures
Voronoi cell volume	C++ `double`
Neighbors	C++ `vector`
Areas of contact faces	C++ `vector`
Normal vectors of contact faces	C++ `vector`
Vetices of contact faces	Matrix (C++ `vector` of C++ `vector`)
Vetices of cell polyhedra	C++ `vector`

TABLE 4.2: The Voronoi information of single cells.

4.3.4 Cell Mechanics: Cellular Mechanical Interactions and Motion

4.3.4.1 Force Types

Tumor cells are assumed to have the following fundamental physical properties:

- Isolated cells tend to be spherical in a environment absent of mechanical stimuli.
- Cells deform under loads (e.g. pressures from its neighbors).
- Cells are physically viscoelastic.
- Cells move in an over-damped manner.
- Cell rotation is neglected.

These properties characterize the cell-cell and cell-environment mechanical interactions. A tumor cell needs to increase its volume for division, doing which it pushes its contact neighbors through the cellular membrane. According to Newton's third law, both of the involved cells bare the force simultaneously. In reality, contact cells try to move away from each other to ease the tension of pressure of the cellular membranes. The actual moving direction of a cell is the joint result of forces from all the surroundings, including the neighboring cells, the ECM, the ECF etc [159–161]. Besides, isolated cells in culture solution have been observed conducting random motion in the absence of biochemical stimuli [162, 163], which indicates the existing of a random force. In a word, tumor cells constantly exert forces to each other during tumor growth.

To investigate tumor cell mechanics and its effects on the growth of multicellular tumors, several types of forces are explicitly considered in TUGME. Table 4.3 gives their names and the major biophysical sources.

The cell-cell contact force is due to the contact of two neighboring cells. Cytoskeletons that connect the cellular membrane and nucleolus (eucaryotic cells), keeps the rigidity of cells and at the same time they can reorganize themselves to ease the tension of

Force names	Symbols	Sources
Cell-cell contact force	$\mathbf{F}_{ij}^{c}$	cytoskeletons, the cellular membrane
Cell-cell friction force	$\mathbf{F}_{ij}^{f}$	cellular membranes
Cell-E friction force	$\mathbf{F}_{i}^{f}$	the ECM, the ECF etc.
Random force	$\mathbf{F}_{i}^{r}$	the ECF

TABLE 4.3: Tissue and cell mechanical forces.

stresses or pressures [142]. When a cell undergoes deformation, it gives a repulsive force back to the contact neighbor for recovering its original shape (probably spherical). The cellular membrane, consisting of the phospholipid bilayer with embedded proteins, also contributes to cell rigidity [164, 165]. Repulsion is one aspect of cell-cell mechanical interactions. Adhesion also exists between contact cells, which is especially apparent when they try to move away from each other. The contact force here includes both the repulsive and adhesive effects. Cell-cell friction force is to model the fiction effect between two contact cells moving relatively, as the cellular membrane is obviously not smooth. Cell-E friction force is the dragging effect from the extracellular materials like the ECM, the ECF. The ECM provides mechanical support to maintain the multicellular structures which may need to reorganize for cell motion.

4.3.4.2 Force Models

To establish physically reasonable models for the forces listed in table 4.3 is very challenging in reality. Taking the contact force $\mathbf{F}_{ij}^{c}$ as an example, on the one hand, it depends on mechanical properties of materials, such as geometry, face properties like roughness of contact objects [166]; On the other hand, tumor cells show complex biomechanical properties combining the elasticity, plasticity and viscosity. Cellular membranes deform elastically under small loads. The viscosity of cells is determined by many aspects, generally including the cytoplasm, the ECM connections, the ECF and bindings between receptors and ligands on cellular membranes [10, 161, 167].

Hertz firstly established a model for the contact force between a spheroid and a plane (or more general two spheroids) [168]. The most fundamental assumptions of the Hertzian theory includes:

- Surfaces of contact objects are continuous and non-conforming
- Surfaces of contact objects are frictionless
- No adhesion between contact objects exists.

- Loads exerted on objects are small (small deformation), hence, the deformation is elastic.

Considering these strict assumptions of the Hertzian theory and the mechanical properties of tumor cells, the Hertzian theory is thought to be too rough to model the contact force between tumor cells in reality. Johson Kendall and Roberts put forward a variation of the Hertzian theory by introducing the adhesion effect for the experimental observation of a contact area larger than what is predicted by the Hertzian theory [169]. The new theory is termed the JRK contact model, which can be described theoretically by the following equations:

$$\begin{aligned} |\mathbf{F}_{ij}^c| &= \frac{E_{ij}a_{ij}^3}{R_{ij}} - \sqrt{6\sigma\pi E_{ij}a_{ij}^3} \\ a_{ij}^3 &= \frac{R_{ij}}{E_{ij}}(|\mathbf{F}_{ij}^c| + 3\sigma\pi R_{ij} + \sqrt{6\sigma\pi R_{ij}|\mathbf{F}_{ij}^c| + (3\sigma\pi R_{ij})^2}) \\ \frac{1}{E_{ij}} &= \frac{3}{4}\left[\frac{1-\nu_i^2}{E_i} + \frac{1-\nu_j^2}{E_j}\right] \\ \frac{1}{R_{ij}} &= \frac{1}{R_i} + \frac{1}{R_j} \end{aligned} \tag{4.6}$$

where R_i and R_j are the radii of the contact objects, a_{ij} measures the radius of the contact area, σ is the surface contact energy density, E_i and E_j stand for the Young's module, ν_i and ν_j represent the Poisson ratio.

Unfortunately, the JKR model is very complex to treat two cases separately: contact objects approaching and leaving each other. To avoid this complexity, Beyer and Meyer-Hermann [170] approximated the JKR model based on the virtual overlap $h_{ij} = R_i + R_j - d_{ij}$ (see figure 4.11), which reads:

$$|\mathbf{F}_{ij}^c| \approx E_{ij}\sqrt{R_{ij}}h_{ij}^{3/2} - \sqrt{6\pi\sigma E_{ij}}(R_{ij}h_{ij})^{3/4} \tag{4.7}$$

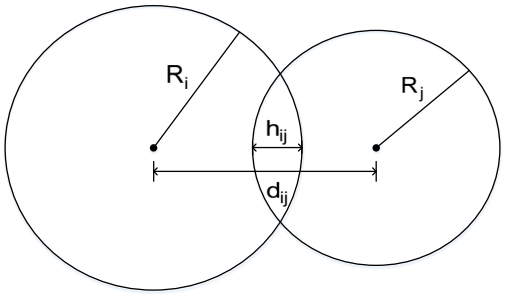


FIGURE 4.11: Diagram illustrating the overlap h_{ij} between two contact spheroids where R_i and R_j are the radii of the two spheroids separately. d_{ij} measures the distance between cells.

The approximated JKR model is applied to model the contact force $\mathbf{F}_{ij}^{c}$ between cells in our model. A possible JKR force curve versus the overlap between contact cells is shown in figure 4.12.

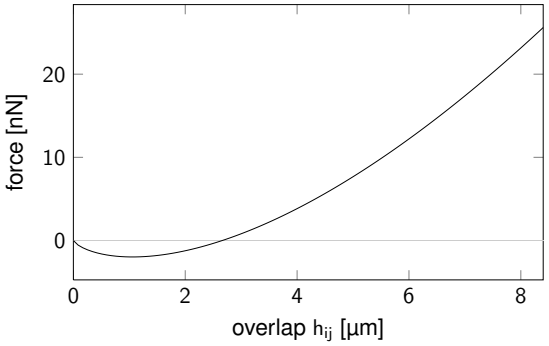


FIGURE 4.12: An approximated JKR force curve versus the virtual overlap h_{ij} between two neighboring cells. The portion of the curve above the light gray line indicates a repulsive effect, while the below counterpart indicates the adhesion effect. The idea is that contact cells tend to attract each other when they are in contact, meanwhile, they also push each other when they undergo shape deformation. For small overlaps, the adhesion (viscosity) force dominates, while with the decrease of cell-cell distance (h_{ij} increases). Cell elastic repulsion force sharply increases with the increment of cell shape deformation. The right cross point between the force curve and the x-axis indicates a zero JKR force, which is considered to be a result of balancing between the adhesion and the repulsion forces, which can be thought as a preferred distance between cells within tissues. Larger than this h_{ij}, the repulsion force starts to dominate.

The friction ($\mathbf{F}_{ij}^{f}$) between contact cells is modeled to be proportional to the relative velocity $\mathbf{v}_i - \mathbf{v}_j$ ($\mathbf{v}_i$ and $\mathbf{v}_j$ (see equation 4.8) designate the absolute velocity of cell i and cell j) of the involved cells but has the opposite direction. For the force ($\mathbf{F}_{i}^{f}$) of friction by the ECM and the ECF, a Stokes friction approach is applied (see equation 4.9). More details on the friction coefficients $\boldsymbol{\gamma}_i$ and $\boldsymbol{\gamma}_{ij}$ can be found in appendix A. The random force $\mathbf{F}_{i}^{r}$ is currently left unimplemented since its effect to the over-damped cell motion is believed to be such small that can be neglected. We keep the term in case a modeler is interested in modeling it in some particular cases.

$$\mathbf{F}_{ij}^{f} = -\boldsymbol{\gamma}_{ij} \cdot (\mathbf{v}_i - \mathbf{v}_j) \tag{4.8}$$

$$\mathbf{F}_{i}^{f} = -\boldsymbol{\gamma}_{i} \cdot \mathbf{v}_i \tag{4.9}$$

The mechanical interactions between cells and simulation domain boundaries have to be dealt with particularly, since the cells next to the boundaries directly affect their neighboring cells and so forth. We have mentioned three types of boundary conditions are supported by TUGME in section 4.3.3.2, namely the periodic boundary, the non-periodic boundary and the free boundary.

The default simulation domain is a rectangular box characterized by its lower left and upper right corners. In the non-periodic boundary case, the six faces of the box are the boundaries. In TUGME, the six faces are treated as virtual cells with the `TID`s from -1 to -6. For those cells next to one of these boundaries, the contact and friction forces are modeled as between them and their images treating the corresponding boundary face as a mirror. But, the velocity of the boundaries keeps zero. One may argue to simply specify these forces to be zero, however, our intention is to try our best to take into account the effect of boundaries instead of nothing considering the important roles of mechanical interactions between tumor cells and the surrounding healthy cells in reality. Of course, one can change our implementation very easily. As a matter of fact, in order to minimize the effects of the boundaries to the behavior of our tumor model, we usually make the simulation domain size much larger than the size of the modeled multicellular tumor system so that no tumor cells can reach the simulation domain boundaries during a simulation. This is done by filling outer tumor space with healthy cells when no default shape is defined for Voronoi cells. Hence, the boundaries only affect the healthy cells directly. In this case, if a healthy cell moves out of the simulation domain, it is removed from the system completely. The advantage of doing so is to provide an environment more realistically mimicking that of real tumors. The disadvantage is that one has to introduce proper rules to deal with the healthy cells.

In the case of the periodic boundary condition. problems get easier, since the contact and friction force models can be applied directly between a cell and its image. However, the periodic boundary condition only works for certain domain shapes, for example, rectangular boxes and spheres aren't applicable. Similar to the case of the non-periodic boundary condition, the outer tumor space is also filled with healthy cells. The key difference is that healthy cells will not be moved out of the simulation domain. With the growth of the tumor tissue, their space is compressed gradually by tumor cells, hence, rules of how healthy cells have be dealt with during tumor growth is even more important, otherwise, the tension between them and tumor cells can become unrealistically large.

One may think of that introducing healthy cells in the above two cases are too clumsy. In order to avoid doing that and to deal with the wired shape of cells next to the simulation domain boundary, the third type of domain boundary is provided in TUGME. In this case, a default shape for Voronoi cells is pre-defined, for example, the regular dodecahedron is used by default in TUGME. According to this design, an isolated cell has 12 neighbors by default. Currently, we set the `TID` of these cells to be -1 and do not model their mechanical interactions with tumor cells. Hence, this boundary results in a free biomechanical environment for the modeled multicellular tumors.

In order to allow different models for the forces considered in TUGME, an interface (see figure 4.13) is introduced. Currently, two implementations of this interface are provided, namely the Hertz model and the JRK model, which are distinguished according to how the cell-cell contact force is modeled.

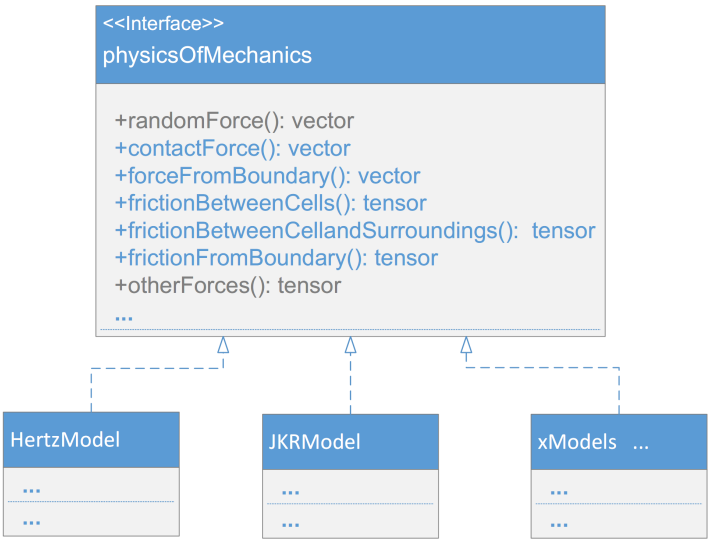


FIGURE 4.13: Diagram illustrating the interface for modeling the mechanical forces of single tumor cells.

Based on the forces given in table 4.3, the cell motion is modeled in an over-damped manner using Newton's second law (see equations 4.10, $\mathbf{F}_i^t$ is the total force on cell i, and N_i is the set of neighbors of cell i). By replacing force terms with their corresponding models, equation 4.10 can be reformulated into equation 4.11. For each cell, the only unknown variable in this equation is its velocity $\mathbf{v}(x, y, z)$.

$$\mathbf{F}_i^t = \sum_{j \in N_i} \mathbf{F}_{ij}^c + \mathbf{F}_i^f + \sum_{j \in N_i} \mathbf{F}_{ij}^f = m\mathbf{a} = m\dot{\mathbf{v}} = 0 \tag{4.10}$$

and

$$\sum_{j \in N_i} \mathbf{F}_{ij}^c = \left(\boldsymbol{\gamma}_i + \sum_{j \in N_i} \boldsymbol{\gamma}_{ij} \right) \cdot \mathbf{v}_i - \sum_{j \in N_i} \boldsymbol{\gamma}_{ij} \cdot \mathbf{v}_j \tag{4.11}$$

As one can see from equation 4.11, the motion equations of all cells are coupled via their velocity, which results in a system of linear equations. For a tumor system with N cells, the degree of freedom of the linear system is $3N$. More details about how the linear system of cell motion is derived can be found in appendix A of this thesis.

4.3.4.3 Solving the Linear System of Cell Motion Using DUNE

For a tumor system with millions of cells, the degree of freedom of the linear system is very large. It is usually too time-consuming and memory-expensive to solve such a large system analytically, while a numerical solution, fulfilling the expected accuracy, is more realistic to conduct. As it has been mentioned in section 4.3.1, DUNE provides a lot of solvers for numerically solving linear systems. The preconditioned BiCGStab solver is applied to solve our linear system of cell motion considering its performance and stability.

4.3.5 Nutrients and Metabolic Products: the Transport and Metabolism of Biochemicals

4.3.5.1 The RDE Model

In TUGME, the transport and metabolism of biochemical molecules are modeled by RDEs. Diffusion is the major way, through which many biochemical molecules are transport within tumor tissues. No matter for the avascular or the vascular tumors, this holds for cells locating far away from blood vessels. Similar RDE models have been widely used in lots of cancer models [3, 51, 55, 58, 146–148, 171, 172] for not only nutrients but also other biochemical molecules like GFs.

The RDE model takes the form:

$$\partial_t c(x,t) = \nabla \cdot [D(x,t) \cdot \nabla c(x,t)] - r(x,t),\ in\ \Omega \in R^d \times (0,T)\ (d=3) \tag{4.12}$$

where $c(x,t)$ represents the concentration of molecules over space and time. The diffusion coefficient $D(x,t)$ and the reaction (source) term $r(x,t)$ measures the diffusivity in extracellular space and the consuming (producing) rate by tumor cells of biochemical molecules. $\Omega \in R^d$ is the spatial domain that the equation works within, and d is the spatial dimension. $(0,T)$ is the time interval.

4.3.5.2 Solving the RDEs Using DUNE

DUNE, applied to solve our linear system of cell motion, provides the basic implementation of FEM in its module DUNE-pdelab, which significantly reduces the programming work of implementing FEM in TUGME. however, this doesn't mean that DUNE-pdelab has done everything for practical applications. First of all, random accessing the results of RDEs isn't properly supported by DUNE, which is a very important requirement

for TUGME, since tumor cells need to constantly evaluate their biochemical conditions such as the concentration of oxygen. Second, some implementation details are exposed to users, for example, inter-mapping between the physical computational domain (global) and the local computational domain, which increases the work of users. For these reasons, we introduce interfaces for TUGME wrapping DUNE-pdelab code.

An interface named `RCDProblem` is introduced for defining a reaction-diffusion problem of a molecule. As a tool, DUNE-pdelab is actually designed for general RDE problems, like reaction-convection-diffusion problems. `RCDProblem` basically keeps the generality of DUNE-pdelab at the same time tries to minimize the work of users during implementing a specific RDE model. As it is shown in figure 4.14, this interface integrates the basic aspects needed to define a reaction-convection-diffusion problem. Three different types of boundary conditions are supported, namely the Dirichlet boundary condition, the Neumann boundary condition, and the flow boundary condition. Currently, the model of oxygen, glucose, cell metabolites and GFs are provided. Implementation of a new model can be done very easily by referring to one of the four existing ones. From the figure, one may notice that `GeneriProblem`, which is a inner C++ class of the class `RCDProblemFEM`. It seems screwball to have both `GeneriProblem` and `RCDProblem`. The underlying idea of this design is to hide the implementation details of DUNE, since `GeneriProblem` is called directly by the DUNE routines involving the local domain which is mapped from the global domain (the physical domain), while `RCDProblem` is global-domain-oriented. The final thing needs to be emphasized is that its the user's responsibility to not only define a specific reaction-convection-diffusion problem, but also implement the class similar to the C++ class `multiRCDProblems`, since this class should be defined based on all the types of biochemical molecules considered.

When implementing a specific RDE model like `RDofOxygen`, one has to specify the parameters (terms) of the RDE. For a RDE with all its terms explicitly formulated, it is trivial to fill the corresponding terms of `RCDProblem` with the explicitly expressions. However, for the tumor system, some terms may too complex to be explicitly formulated as they are dynamically affected by both the properties (or the state) of tumor cells and the way of how a modeler deals with them.

First of all, the diffusion coefficient $D(x,t)$ basically depends both on physical properties of the molecule itself and the environment. In a free space, it can be described by the equation (the Stokes-Einstein equation):

$$D = \frac{KT}{6\pi\eta r} \tag{4.13}$$

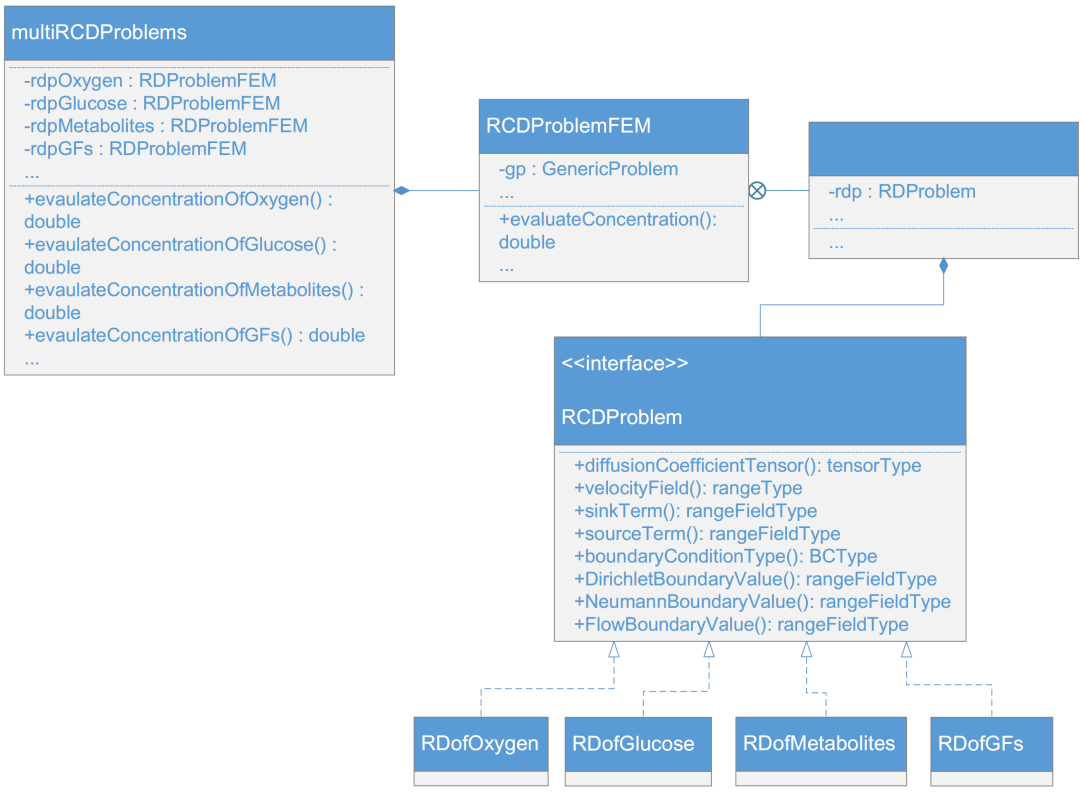


FIGURE 4.14: Diagram illustrating the relationship between the C++ classes of RDE models in TUGME.

where η represents the viscosity of the environment, r represents the radius of the spherical particle, K is the Boltzman constant and T measures the absolute temperature of the environment. According to this theory, the actual mobility of a molecule is affected by the viscosity of the environment, its size and the temperature of the environment. Unfortunately, the space of a tumor tissue isn't free for molecules like oxygen and glucose to diffuse, since the existence of the cellular membranes, the ECM, other molecules in the extracellular environment. The actual diffusion coefficients measured in experiments [173, 174] within tumor tissues of both oxygen and glucose are smaller compared with that in water. Obviously, the experimentally measured diffusion coefficient within tumor tissues is an averaged value at a large spatial-temporal scale. If one thinks about the diffusion process at the sub-cellular scale, the diffusion coefficient of a molecule regarding two directions, parallel and perpendicular to the cellular membrane, can differ significantly. Unfortunately, no measurement about this has been done by now. However, one can see our point from this argument. Similar problems will be also confronted for determining the reaction $r(x,t)$, which is to be discussed in the next section soon. The actual diffusion coefficient used in simulations with TUGME is the experimentally measured value within tumor tissues.

Secondly, The boundary conditions act as the source of nutrients or the drainage system of cell metabolites. It is well-known that the circulatory system plays the key role in transporting nutrients and cell metabolites. For avascular tumors the boundary condition of RDE should be specified according to the distribution of blood vessels in the normal healthy tissues surrounding a tumor. For a mature tumor, problems get much more complicated, for tumor-promoted angiogenesis should have formed blood vessel networks for it. Since we currently focus on avascular tumors only, the Dirichlet boundary condition is applied based on the assumption that the concentrations of nutrients in healthy tissues should keep at a dynamically stable level, which is essential for healthy cells to perform their normal functions. In addition, the molecular metabolism of healthy cells surrounding a tumor tissue isn't explicitly taken into account if healthy cells are taken into account, since we assume that their blood vessel network is efficient enough to maintain the biological equilibrium in terms of nutrients and metabolites given that an avascular tumor is usually still not so malignant to significantly affect its surrounding healthy tissues.

Finally, as FEM is grid-based, the size of the grid cells (elements) affects both the modeling accuracy and the time expense for solving RED. Theoretically, a finer grid can give a better modeling accuracy and the actual grid size is usually chosen as coarse as possible as long as the expected accuracy is fulfilled in order to save the computational cost. However, a tumor tissue is rather complex than free. It involves other processes, dynamically interacting with the reaction-diffusion process of molecules. In TUGME, a 3D lattice, named the Yasp-grid in DUNE, is applied. The Yasp-grid is one of the simplest grids provided by DUNE. The elements of the Yasp-grid are non-overlapping rectangular boxes. For the size of the grid element, we recommend that it should be larger than the largest volume of single tumor cells, the reason under the recommendation is to be introduced in details in the next section.

4.3.5.3 Up- and Down-scaling of Biochemicals

The metabolism of nutrients and metabolites happens at a smaller (upper) spatial-temporal scale compared with that of the RDE model. Hence, up- and down-scaling is necessary to fill the gaps between spatial-temporal scales.

Up-scaling is the process that tumor cells evaluate the biochemical condition of its environment, since its life behavior needs energy, which can only be produced via biochemical reactions (tumor cell aerobic or anaerobic respiration introduced in section 3.3.5) within cells. In TUGME, the biochemical conditions mean the concentrations of chemical molecules, which may be nutrients, metabolites, GFs etc. Specifically, the solution

of the RDE models is to be accessed. The concentration at a certain position (point) can be directly evaluated, however, how to properly do so over a volume (the volume of an entire tumor cell), is something different. The way we deal with this problem is to use the molecular concentration at the mass center of a tumor cell to represent the biochemical conditions of the whole cell volume. The underlying considerations include: 1) the RDE model is already a coarse-grained approximation; and 2) our experience with the concentrations of the involved chemical molecules is that there is a concentration gradient within the simulation domain (e.g. from the tumor periphery to its center) but no local sharp fluctuations, which indicates no significant variation of molecular concentrations within the volume of a single tumor cell. An alternative idea to do so may be to evaluate the molecular concentrations at a certain number of points uniformly distributed on the cellular surface or within the cellular space, then average them. However, it isn't supported by TUGME currently as we don't see significant benefits to the overall accuracy of our coarse-grained tumor models.

Another problem for evaluating the concentrations of biochemical molecules is associated with the implementation of FEM in DUNE. To evaluate the concentration at a given point in DUNE, one needs to know the the grid element, within which the given point locates. In other words, a scheme is needed to correctly map the coordinate of the given point to the corresponding grid element. For regularly structured grids like the Yasp-grid used by us, it is usually not very difficult to find such a mapping scheme, while doing so for irregularly structured grids is usually very difficult.

In TUGME, concentration evaluation is implemented in the C++ class named `duneYaspGridRandAccessHelper`. In this class, each grid element is given an unique ID (a non-negative integer), which corresponds to the index of the C++ `vector`, where pointers pointing to the grid elements are properly stored. The pointer is used instead of the grid element itself, because DUNE does not allow to store grid elements out of where the grid is declared. the `vector` instead of the `map` is used, since it is more efficient. However, we admit that the `vector` can only work for very few types of grids, otherwise the `map` is recommended. A snapshot of part of the original source code for evaluating molecular concentration at a given point (position) is shown in figure 4.15.

Down-scaling is the process that the reaction term $r(x,t)$ is determined for the RDE model, which can be very challenging. First of all, we want to declare that we approximate $r(x,t)$ at a spatial-scale larger than that of single tumor cells for two general reasons. First, evaluating $r(x,t)$ at a spatial-scale smaller than single cells asks for too much computational complexity. Take oxygen as an example, it can only be determined when the position of the mitochondria within cells is known, since oxygen metabolism is mainly performed within the mitochondria. Modeling the motion of the

```
806     // find the ID of the target grid element where the given point (the global coordinate) locates.
807     int id = dgrah.findEntityID(xGlobal[0], xGlobal[1], xGlobal[2]);
808     typedef Tissue::duneYaspGridRandAccessHelper<superTraitsType, LGV> gridHelperType;
809     typedef typename gridHelperType::EntityPointersVectorType EntityPointersVectorType;
810     const EntityPointersVectorType& v = dgrah.getAsRefEntityPointersVector();
811     ElementType& e = *(v[id]);
812     typename FS::DGF xdgf(fs.getGFS(), x);
813     xdgf.evaluate(e, e.geometry().local(xGlobal), r);
814
815     return r[0];
816 }
```

FIGURE 4.15: A snapshot of part of the TUGME source code for evaluating the concentration of a biochemical molecule at a give position.

mitochondria within a cell definitely increases much more computational cost. Second, a more precise $r(x,t)$ does make too much sense to significantly improving the general accuracy of our tumor growth model. A more precise $r(x,t)$ enables us to see the inhomogeneous distribution of biochemical molecules within a tumor cell, however, we have declared that before this difference is normally too small to significantly affect the overall behavior of individual tumor cells.

Basic assumptions of the method for evaluating $r(x,t)$ in TUGME include: 1) chemical molecules are uniformly distributed within a single tumor cell; 2) $r(x,t)$ is averaged over the volume that is large enough to accommodate at least one tumor cell; and 3) single cells are treated as the very basic (smallest) units with respect to the metabolism of biochemical molecules.

To do the space average for biochemical molecules, the entire simulation domain is partitioned into much smaller sub-volumes. According to our assumptions, the size of the sub-volume should be larger than the volume of single tumor cells. Besides, sub-volumes should be non-overlapping and fully divide the whole simulation domain. The grid for space partition can be just simply a lattice like the FEM Yasp-grid or finer. The grid we used is a regular lattice, which is defined by dividing each of the FEM Yasp-grid element into 8 equal rectangles. Figure 4.16 illustrates the FEM Yasp-grid (black) for and the grid for space average of biochemical molecules (blue) with respect to the radical Voronoi tessellation of the tumor tissue in two dimensions (noticing that the FEM Yasp-grid element is divided into 4 equal rectangles in two dimensions as it is shown in this figure).

Mathematically, the approximating method of $r(x,t)$ can be described by the equation,

$$\overline{r_i}(x,t) = \frac{\sum_{j=0}^{K_i-1} u_j}{V_{suv\_V_i} \cdot \Delta t} \tag{4.14}$$

where index i marks the sub-volume (sub_V) and $V_{suv\_V_i}$ stands for its volume. K_i is the number of tumor cells locating within the sub-volume. u_j stands for the amount of

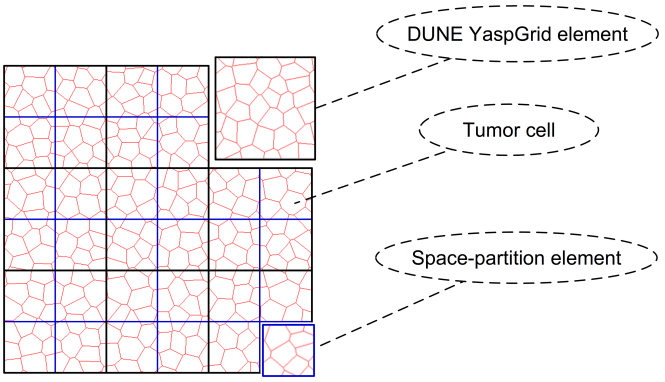


FIGURE 4.16: Diagram illustrating the up- and down-scaling of biochemical molecules.

nutrients consumed by cell j during the time period Δt. One could imagine that $\overline{r}(x,t)$ is actually a piecewise function, and each piece of function is constant with the sub-volume sub_V_i. The 1D case of the approximated reaction term $\overline{r_i}(x,t)$ is schematically illustrated in figure 4.17. It is a piecewise function (not continuous over space). To make a it continuous, one could use methods like interpolation or l_2-projection. However, this has not been done in TUGME, since it is actually not continuous as most chemical reactions can perform at certain organelles within cells. Besides, as a coarse-grained model, we think our approximation of $\overline{r_i}(x,t)$ should be reasonable enough.

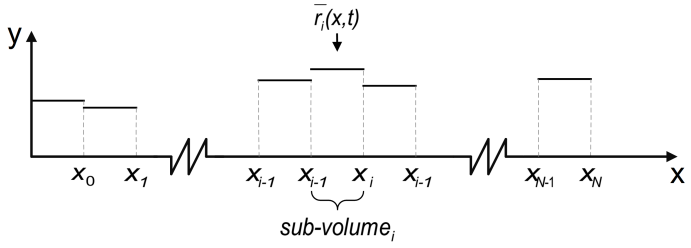


FIGURE 4.17: Illustration of the piecewise characteristic of the reaction function $\overline{r_i}(x,t)$ averaged over space and time.

If thinking further about equation 4.14, one may see that $\overline{r_i}(x,t)$ actually is a function of both the tumor cell density (number of cells K_i within the sub-volume sub_V_i) and the metabolic rate (the total amount of molecules divided by the time-step duration) of the involved biochemical molecules by tumor cells within the sub-volume. The fortunate thing is that the cell density can be measured directly via experiments in the laboratory. In our simulations, the tumor cell density is controlled by limiting the cell radius (or diameter) within a range that has been measured in experiments. One needs to find out a reasonable cell radius range for different types of tumors. The unfortunate thing is that the metabolic rate of most nutrients like oxygen and glucose seems to be very difficult to model using a simple rule, since the actual metabolism rate of biochemical

molecules may depend on many factors, such as: 1) the availability the biochemical molecule itself; 2) the availability of biochemical molecules that can react with the molecule; and 3) the state of cells (proliferating, quiescent, necrotic) [175, 176]. Take oxygen as an example, the interaction between tumor cells and oxygen forms a feedback loop as it is shown in figure 4.18. On the one hand, cell consumption decreases the concentration of oxygen; On the other hand, lower oxygen concentrations may change the behavior of cells, like the conversion from proliferation to quiescence for the short of energy, which in turn reduces the metabolic rate of oxygen. Tumor cells undergo this conversion is believed beneficial to cell survival. These reasons can reduce the accuracy of the measurements of the molecular metabolic rate, since the most prevalent way for measuring the metabolic rate used by current biological experimentalists in the laboratory is to average the amount of molecular consumption or production over a group of cells, however, the cell groups may actually consist of cells in distinct states, such as proliferating, quiescent and even necrotic. As far as we know, there is still no measurement carried out based on individual tumor cells. Besides, no experiment has been done to measure how much the concentration of a biochemical molecule can affect their metabolic rate by tumor cells of different states.

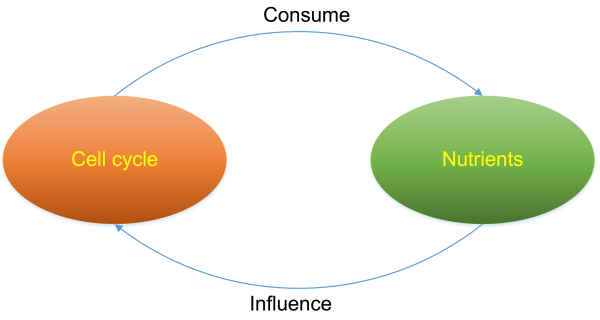


FIGURE 4.18: Diagram illustrating the non-linear interaction between tumor cells and nutrients. Most cell life behaviors need the support of energy, which is generated by biochemical reactions consuming nutrients, such as oxygen and glucose. Cells uptake nutrients from the extracellular environment and release metabolic products back. If molecular diffusion can not compensate the loss of consumption of nutrients, the nutrient concentration will continuously decline. Tumor cells suspend cell cycle and enter a so-called quiescent state (tumor cell quiescence) for short of energy when the nutrient concentration below certain thresholds. Doing so, tumor cells may survive for its nutrient consumption rate reduces in the quiescent state. If the nutrient concentration gets even lower, tumor cells die (tumor cell necrosis, which is energy-independent) for energy depletion.

4.3.6 Cell Cycle: Cell Proliferation and Death

In TUGME, tumor cell cycle is modeled by a series of phases, correspondingly mimicking the phases of an entire cell cycle in reality. Single cell behaviors, such as cell volume

growth, metabolism, division and the cell cycle control, are modeled by discrete events associated with the corresponding phases.

The cell cycle model is a key module in TUGME, since it directly determines the growth dynamics of the multicellular tumor systems, while the other modules affect the cell cycle dynamics. Basically modeling tumor growth is to investigate the cell behavioral mechanisms in terms of joint regulations of biological, biochemical and biophysical processes of high relevance. Hence, the cell cycle model may vary significantly for a variety of hypotheses. For example, if CSCs and common tumor cells are separately considered, the cell cycle model should at least distinguish the proliferation potential between these two types of cells. The cell cycle module of TUGME is expected to facilitate testing different hypotheses by enabling easy implementations of different cell cycle models.

In TUGME, the cell cycle model is hierarchically organized consisting of interfaces and implementations as it is shown in figure 4.19. In general, there are two types of cell cycle models in terms of how real cell cycle phases are modeled. The first type simply splits cell cycle into the $G1$ phase and the rest phases $SG2M$. The second type of models consists of exactly the real cell cycle phases, namely the $G1$, S, $G2$ and M phases.

The $G1$ phase is always explicitly modeled because it is particularly important to tumor cell proliferation, which has been introduced early in section 3.3.4 in chapter 2. This is also why we merge the rest S, $G2$ and M phases as one in the first general type of cell cycle models. We provide the second general cell cycle model type in case some users are interested in modeling some detailed dynamics of full cell cycle phases. Beside these phases, a state when $G1$-phase tumor cells suspend the proliferation temporarily is treated as a special phase named the $G0$ phase. It is important to model the mechanisms controlling transitions between the $G1$ and the $G0$ phases, since the period of time that tumor cells stay in the $G0$ phase directly affects the overall cell proliferation rate or the growth rate of a tumor. Besides, tumor cell necrosis is modeled as a separate state that differs to the apoptosis of normal somatic cells. In a word, a tumor cell can be in only one of the three states, namely the proliferating, the quiescent, and the necrotic states. If the CSC theory is considered, CSCs differentiate into common tumor cells which can undergo apoptosis too.

Defining the phases and/or the states for cells are just part of the work of establishing a cell cycle mode. The next step is to explicitly define the cell cycle rules. First of all, one needs to specify a distribution type that the duration of each cell cycle phase obeys, as the cell cycle phase duration isn't fixed but random. The phase duration distribution of all the phases are Gaussian distributions but with different means and

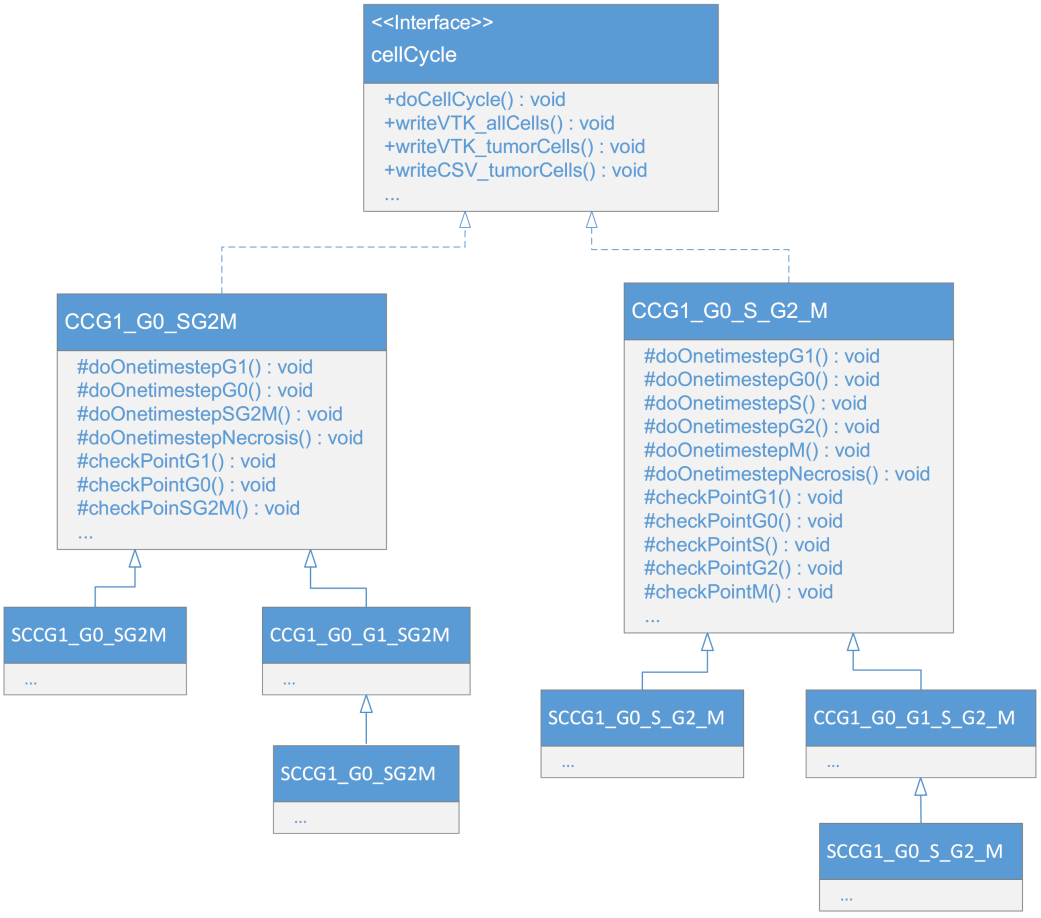


FIGURE 4.19: Diagram illustrating the interface and several implementations of cell cycle models. The symbol "#" indicates a protected field of a C++ class.

standard divinations in our cell cycle models. A Gaussian distribution allows negative values, which is obviously unrealistic for modeling the phase duration. This problem is handled by replacing all the negative random numbers with zero. Secondly, cell volume growth has to be modeled properly. It is defined by increasing the cell volume instead of its radius or diameter with a constant rate in our model. The growth rate is actually calculated by dividing the volume difference between the mature and the newly born tumor cells by the duration of the $G1$ phase. Since the radii of the mature and newly born tumor cells are parameters that do not change once specified for a model and τ_{G1} is a random number, the actual volume growth rate varies from cell to cell. The volume growth of tumor cells is restricted in the $G1$ phase, since an actual tumor cell undergoes significant volume growth in this phase and has slight volume growth in the rest phases except the M phase [2]. Theoretically, this growth model can be described by the equation $\frac{4}{3}\pi R_M^3 = 2 \cdot \frac{4}{3}\pi R_Y^3$ based on the assumption of spherical cell shape, where R_M and R_Y stand for the radii of the mature and newly born cells separately. Thirdly, one needs to define the mechanisms that control the behavior of tumor cells according to the biological, biochemical and biophysical conditions.

In TUGME, two cell cycle controlling rules are defined distinguished by how the conversions between the proliferating and quiescent state of tumor cells is modeled. The first rule is relatively simple, where a newly born cell enters the $G1$ phase directly and it can definitely complete this phase without disruptions by finally entering the $G0$ phase without checking any conditions. In the $G0$ phase, a checkpoint is implemented, where it can advance into three states: 1) switching to necrosis when the corresponding thresholds are met; 2) staying in the $G0$ state when the quiescent thresholds are satisfied; 3) otherwise, advancing into the S phase (resuming the cell cycle) when the cell manages to pass all requirements for proceeding its division. The general procedures of this version of the cell cycle model are shown in figure 4.20.

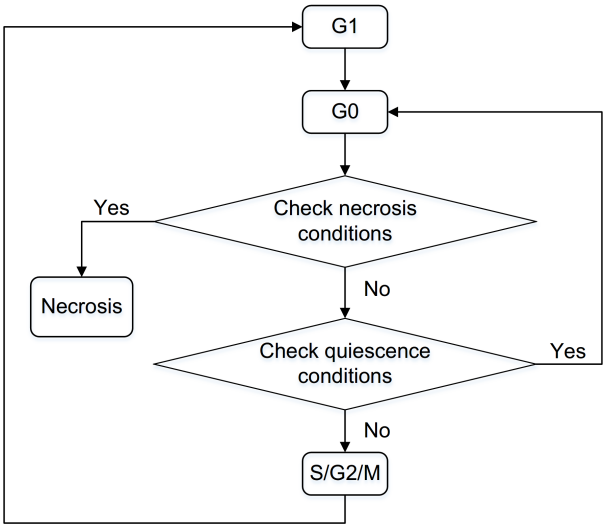


FIGURE 4.20: Flowchart illustrating the phase transitions of an entire cell cycle.

The second rule is characterized by allowing the inter-switch between the $G1$ phase and the $G0$ phase, which is impossible in the first case. According to this rule, a newly born cell is initially set in the $G1$ phase too. However, cells are designated to switch from the $G1$ phase to the $G0$ state once the quiescent conditions are met, and $G0$-state cells can switch back (resuming the cell cycle) if the corresponding conditions are satisfied. The main steps of the $G1$ phase and the $G0$ state of this cell cycle controlling rule are illustrated in figure 4.21 and 4.22.

Finally, a dividing cell is replaced by two daughter cells at the end of its division (the $SG2M$ or M phase). The radius of each daughter cell is set such that each of them has exactly half of the spherical volume of the mother cell. They are initially placed within the volume of the mother cell, which mimicking the daul-bell shape of a real cell at the end of its division. The daughter cells are assumed to have no preference in orientation and start to adjust their positions according to the biomechanical condition right after separating from each other. Hence, they are placed within the mother cell

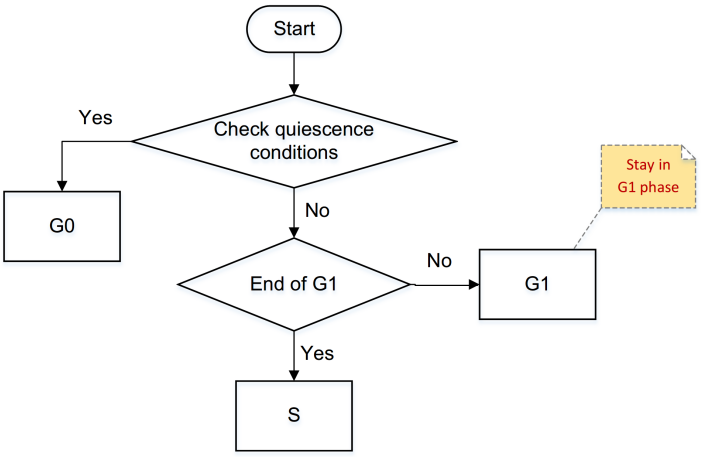


FIGURE 4.21: Flowchart illustrating the state transitions in the $G1$ phase.

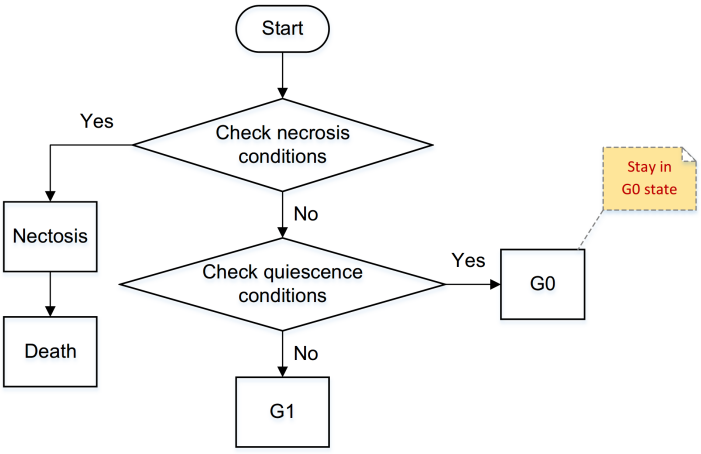


FIGURE 4.22: Flowchart illustrating the state transitions in the $G0$ state.

with a random orientation chosen from the uniform distribution in three dimensions. Figure 4.23 illustrates how the daughter cells are placed within the mother cell with a given orientation.

4.3.7 Interdependences Between Modules of TUGME

The modules of TUGME are coupled for data interdependences. First, the cell agent module stores the basic information, such as the position, the radius and the Voronoi information of all tumor cells. Hence, all the other modules of TUGME basically require to access it. Second, the Voronoi information can only be calculated when the positions and radii of cells are provided. Third, cell-cell and cell environment mechanical interactions and cell motion can be done only when the Voronoi information is available. Fourth, RDE models need to know the consumption or production of the considered

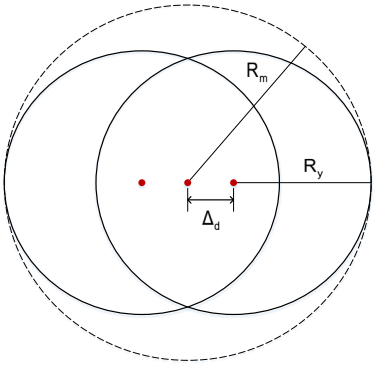


FIGURE 4.23: Diagram illustrating the relative position of two newly generated daughter cells relative to their mother cell.

biochemical molecules by tumor cells. Finally, cells decide their behavior based on the biochemical and biomechanical conditions.

Since the Voronoi information is treated as part of the basic cell information, it is stored in the cell agents instead of the `Morphology&Topology` module (the upper left module in figure 4.24). Hence, the explicit dependences between it and the modules `Biomechanics` (the lower left module) and `Cell_cycle` (the lower right module) become implicit and are replaced by the explicit dependences between the `Cells` module (the module in the center) and `Biomechanics` and `Cell_cycle` modules correspondingly. Since the amount of nutrients (or metabolites) consumed (produced) by tumor cells during cell cycle has been stored in the `Cell_cycle` module and the `Biochemicals` module provides interfaces for accessing the solution of RDEs, these two modules are explicitly interdependent in TUGME.

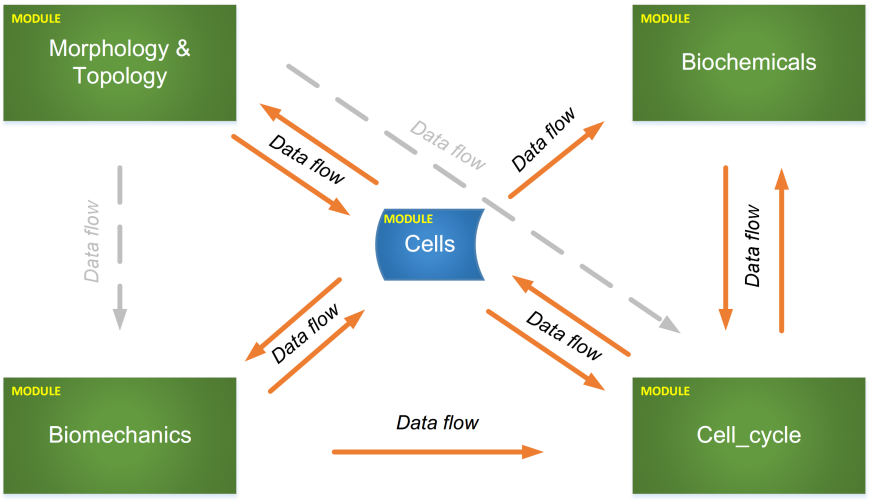


FIGURE 4.24: Diagram illustrating the interdependences between modules of TUGME. The dashed gray arrows represent the theoretical data flowing directions, while the solid orange arrows stand for the actual ones.

4.4 Conducting Simulations Using TUGME

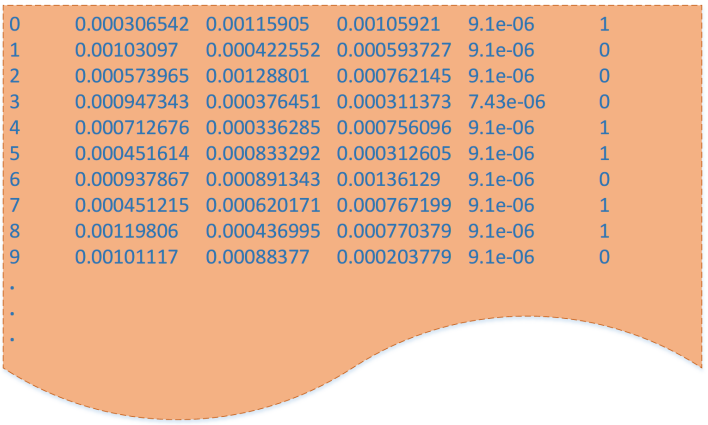
4.4.1 Initialization of Simulations

Parameter space sampling is often required in computer-based modeling and simulation. Varying model parameters sometimes requires changes to the source code, which is clumsy for programming languages like C++, since this results in recompiling of the source code. A common way to handle this problem is to introduce a configuration file and put the frequently changing parameters in this file so that changes can be made easily to it without modifying the source code. Furthermore, automatic configuration file generation is sometimes used for complex models with a large and complex sampling space of model parameters.

In TUGME, we also introduce a configuration file, for example, named config.ini, and a routine is provided to generate the configuration file with default parameter settings. Currently, users can change the parameter settings only manually. Figure 4.25 shows a snapshot of part of the configuration file. The indicating meaning of each parameter in this configuration file can be understood easily via their names, otherwise, one can turn to the source code, where comments on these parameters would be more helpful. Hence, detailed explanations to them need not be given here at all. As one can see, the configuration file is a typical INI formatted file. The INI format is a prevalent standard for configuration files for many software libraries. In TUGME, a parser is implemented based on the Property Tree Library of the boost library to parse this configuration file.

FIGURE 4.25: A snapshot of part of a configuration file.

Beside the configuration file, the initial state of the tumor cells and/or the surrounding healthy cells is put in a file too (see the snapshot of part of the initial file shown in figure 4.26). As it is shown in the figure, the first column represent the cell `UID`. The second to forth columns together designate the coordinate of cells in three dimensions. The fifth column gives the radius of cells. The final column specifies whether a cell is a CSC cell (1) or a common tumor cell (0). A special routine is provided in TUGME for automatically generating this file. The routine is to generate the given number of cells whose positions obey the uniform distribution within the given tumor tissue domain that is specified in the configuration file.



0	0.000306542	0.00115905	0.00105921	9.1e-06	1
1	0.00103097	0.000422552	0.000593727	9.1e-06	0
2	0.000573965	0.00128801	0.000762145	9.1e-06	0
3	0.000947343	0.000376451	0.000311373	7.43e-06	0
4	0.000712676	0.000336285	0.000756096	9.1e-06	1
5	0.000451614	0.000833292	0.000312605	9.1e-06	1
6	0.000937867	0.000891343	0.00136129	9.1e-06	0
7	0.000451215	0.000620171	0.000767199	9.1e-06	1
8	0.00119806	0.000436995	0.000770379	9.1e-06	1
9	0.00101117	0.00088377	0.000203779	9.1e-06	0
.					
.					
.					

FIGURE 4.26: A snapshot of part of an initial file.

4.4.2 Control of Simulations

Simulations using TUGME are advanced by simulation time in a time-step manner, where the time-step size can be fixed or adjusted depending on the schemes chosen. For the fixed time-step mechanism, as it is named, the time-step size is fixed during each simulation run, however, it can be changed for different simulation runs. It is specified at the very beginning of each simulation in the configuration file. In contrast, the variable time-step scheme allows for changing the time-step size over simulation steps.

The variable time-step size is actually self-adapting. Its basic idea can be described by the formula $min(T_{max},\ D_{max}/v_{max})$, where T_{max}, D_{max} and v_{max} separately represent the maximal time-step size, the maximal time-step displacement, both of which are specified in the configuration file, and the maximal cell velocity (v_{max}) which is got by solving the linear system of cell motion.

The time-step size can affect two important but theoretically incompatible aspects of simulations, namely the accuracy and the efficiency. The fixed time-step size scheme is obviously very simple to implement. For few systems evolving with a intrinsic temporal pace, one may be able to find a reasonable time-step size by testing different ones. However, for most real systems, a reasonable time-step size may be not feasible. In contrast, the self-adapting scheme is much more flexible. Although the implementation is more complex, it is often used.

The interdependences between the modules of TUGME determine the order, in which they are called in each simulation step. A simulator that advances simulations sequentially is provided in TUGME. Flowchart 4.27 illustrates the main executing procedures of the sequential simulator. From this flowchart, one can see the stop policy of simulations of TUGME is simulation time. In fact, the start and end time(s) of a simulation are specified in the configuration file. And the default unit for time is seconds.

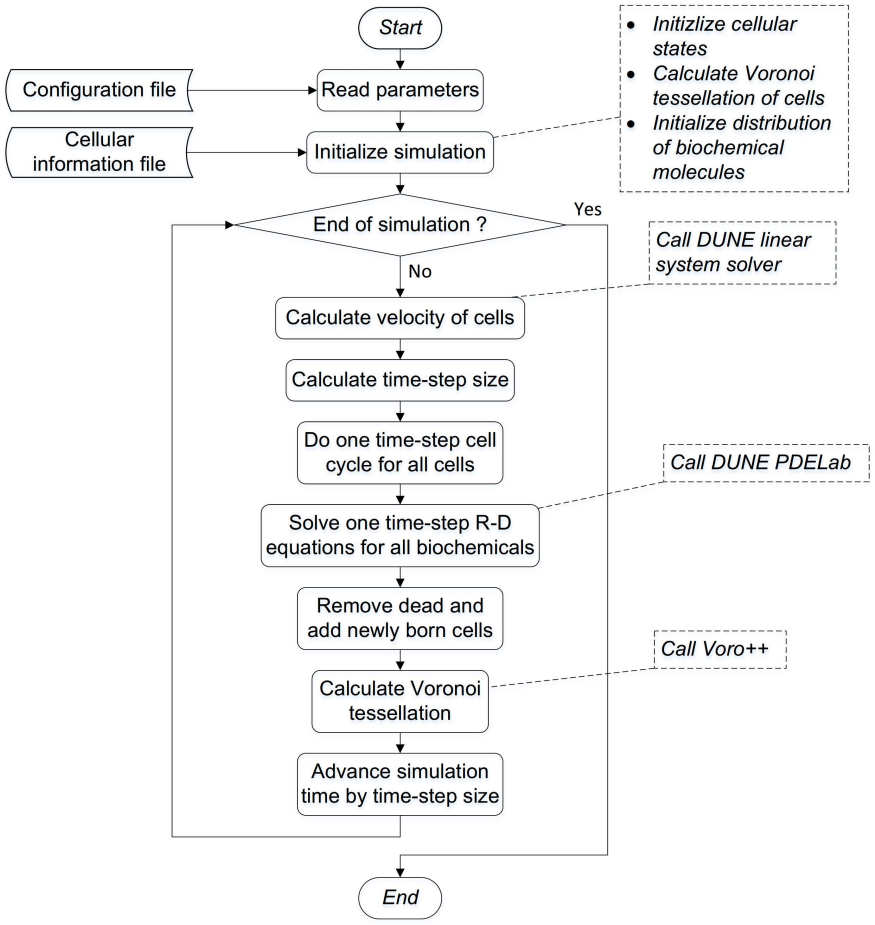


FIGURE 4.27: Flowchart illustrating the main work flow of simulations using TUGME.

4.4.3 Data Output and Visualization

For computer-based modeling and simulation, simulation data collection and visualization are important. One may do data analysis during the running of simulations and outputs the statistical results directly. For small and simple problems, this method works fine, since it usually would not increase the computational cost too much. However, it isn't proper to do so for problems involving complex statistics collections, since it may significantly drag simulations down. For this kind of problems, the values of variables of interest are usually written into files as raw simulation results, which allows one to analyze them flexibly without changing the original source code of models. The disadvantage is that outputting large data sets is very time-consuming.

Data visualization is necessary since it gives modelers the first impression of simulation results and it is needed for result presentation. Run-time data visualization is fine if visualization is to show the raw result files but not to directly access the memory of model variables. However, this may be not feasible for programs with a large amount of data sets, since accessing them may take long then that of the simulation itself.

TUGME provides methods to collect the information of models during the running of simulations. The information covers almost every model aspect, in which the modelers may be interested. First, one can access the basic information of single tumor cells, including the position (the spherical center), the radius, the type (tumor of healthy cells), the stem cell marker (yes or no), the cell cycle phase, the polyhedral volume and the number of neighbors etc. Second, cells can be visualized using currently two tools, namely Paraview [144] and Pov-Ray. As it is shown in figure 4.28, these two tools show some distinguish ideas for rendering objects. Paraview integrates the VTK for viewing the VTK format files, where points is the basic building unit of lines. Linking line segments end to end gives planes or polygons in two dimensions and connecting polygons in three dimensions produces polyhedra. However, curved surfaces, for example, a 3D sphere, are constructed by points in Paraview. The number of points used depends on the image resolution specified by users, which results in large files for high quality (resolution) images. Besides, it allows to generate videos directly with a set of figures whose names are ordered. In contrast, Pov-Ray supports defining a sphere by specifying its center and its radius, which allows users to create very high quality images with curved surfaces without substantially increasing its size. The interface of Paraview is much more user friendly, for example, it is easy to change the viewing orientation in Paraview, while the viewing orientation has to be specified by users in the input file of Pov-Ray and changes usually have to be made manually. Details about these two viewing tools are out of the discussion of this thesis. Finally, the concentration of biochemical molecules modeled by RDEs can also visualized by Paraview. Our molecular

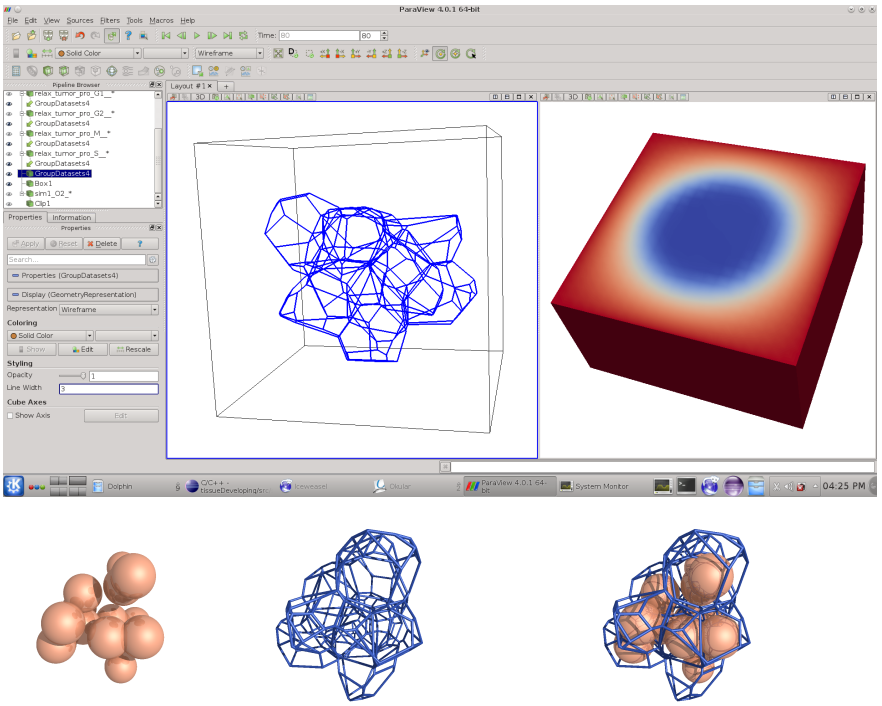


FIGURE 4.28: Visualization of the radical Voronoi polyhedra and the distribution of biochemical molecules using Paraview (above) and Pov-Ray (below) in three dimensions. Blue line segments in this figure are the edges of the radical Voronoi polyhedra. Colors of the above right panel indicate the concentration gradient (from red (high) to blue (low)) of biochemical molecules within simulation domain. Three panels below from left to right are tumor cell spheres, corresponding radical Voronoi polyhedra and the merger of them.

concentration file writer wraps the VTK format file writer (VTKWriter) provided by DUNE.

Chapter 5

Case Study: Modeling the Growth of the EMT6/Ro MTSs

This chapter introduces how to construct a concrete tumor model using TUGME. Furthermore, a series of simulations are carried out based on the models established by us to investigate the influences of biochemical and biomechanical factors on the overall growth of a multicellular tumor by treating them as the main cell cycle controllers. The parameters of our models are set specifically according to the EMT6/Ro mammary carcinoma cell line, since abundant experimental data about this cell line are available. We look at the population, the distribution of biochemical molecules as well as the morphology of tumor tissues under different oxygen and glucose conditions. Our simulations are compared with the experiments in the laboratory. In general, good agreements between our simulation results and the experimental data indicate the applicability of TUGME as well as the validity of our models.

5.1 Introduction

The ultimate goal of creating a computer model is to replace the real system which is difficult or impossible to study directly. Computer-model-based system research is generally to uncover the working mechanisms of the investigated system so that its behavior can be predicted. How well a computer model representing the physical system from the point of view of supporting its intended use indicates the fidelity of the model. Obviously, model fidelity is the key thing that concerns a modeler, since it directly influences the quality of simulation results.

In general, a set of standard programmes have been established for model quality control in computer-based modeling and simulation. These standards are termed VV&A [1, 177, 178]. VV&A is the abbreviation of (model) verification, validation and accreditation. Verification is mainly to check whether a model is correctly implemented as what it is designed by the modeler. Validation corresponds to the fidelity control of a model. Finally, accreditation is the process to decide based on both verification and validation information whether a model is practically useful according to the acceptability criteria.

Tumor model VV&A is still very challenging currently from the point of view of both model validation and accreditation. Validation is very hard for cancer modeling because of the shortage of the experimental data that can be actually used by cancer modelers, since most experimental oncologists seem to be more interested in the molecular working mechanisms within single cells. Besides, cells of different tumors usually show distinct phenotypes. Consequently, most current cancer modelers can only study few types of tumors. For cancer model accreditation, the problem is even worse, since there are hardly any criteria to the best of our knowledge. In a word, the VV&A of cancer computer models still in its very infant stage.

In order to validate our models, we simulate the growth of the EMT6/Ro mammary carcinoma, more specifically, the avascular growth of EMT6/Ro MTSs, since abundant experimental data are available about this cell line [7, 175, 176, 179–186].

5.2 The Model

As TUGME is designed to be a simulation framework, its interfaces can be implemented flexibly by users, which usually asks for modifications to some of its modules. For example, if one wants to introduce a new biochemical molecule as the cell cycle controller, the cell cycle model should be modified. In addition, the transport and metabolism of the molecule should be implemented too. The basic way to construct a tumor model in TUGME is to choose one implementation (if several ones are provided) for each modular interface and then assemble them together. Of course, the implementations of a modular interface can be the default ones provided by TUGME or those implemented by users themselves, however, it is the user's responsibility to guarantee the compatibility of the chosen implementations.

TUGME consists of five modules, as it has been detailedly introduced in section 4.3 in chapter 4. By now, two cell agent models are available, namely `voroCell` and `voroStemCell` which additionally takes into account the stem cell features of tumor

cells compared with `voroCell`. For representing the morphology of single cells and topology of multiple cells, the radical Voronoi tessellation (the `voroPP` interface) is provided with three types of boundary conditions. The cell motion is treated over-damped using the Newton's second law, where two cell-cell contact force models are provided, namely `HertzModel` and `JRKModel`. The transport and metabolism models of biochemicals, including oxygen, glucose, cell metabolites and GFs are provided. Finally, several cell cycle models are provided, and they are distinguished by how the cell cycle phases are modeled, whether CSC features are taken into account and how the switch between the $G1$ and $G0$ phases are handled. Based on the current models of each module in TUGME, many combinations of them may be reasonable to carry out simulations. Furthermore, some parameters of a model can be varied too. In a word, a lot of possible cancer models can be constructed by just directly assembling the existing implementations of the modules provided by TUGME.

The models used in our serious simulations consist of the following modular models. First, the cell agent model is `voroStemCell`. Second, the periodic boundary condition is applied. Third, the `JRKModel` model is chosen for modeling the cell-cell mechanics. Forth, oxygen and glucose are taken into account as the biochemical controllers of tumor cell cycle, and the physical volume of cells is treated as the biomechanical controller (see figure 5.1). Finally, based-on the defined biochemical and biomechanical cell cycle controllers, the cell cycle model that consists of the $G1$ and the $SG2M$ phases are embodied with the rule that a cell finishes the $G1$ phase by directly entering the $G0$ phase. Specifically, the thresholds for tumor cell quiescence and necrosis of both oxygen and glucose are given such that a tumor cell will change its cell cycle state at the $G0$ phase when one of the concentrations of oxygen and glucose drops below its corresponding thresholds. Satisfying all the biochemical requirements, a cell can only enter the $SG2M$ phase after its physical volume is no less then the given volume threshold for cell division. Figure 5.1 shows more details about this cell cycle model.

5.3 Parameter Settings

Most parameters of our models are summarized in table 5.1.

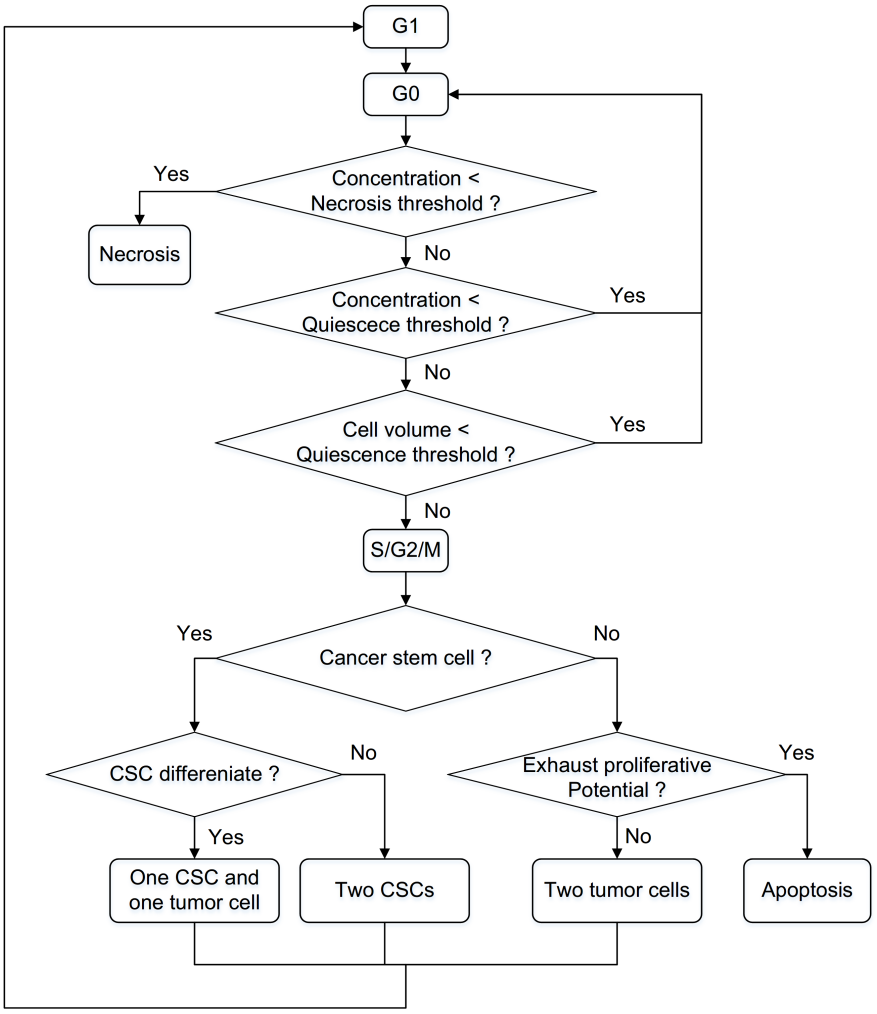


FIGURE 5.1: Flowchart illustrating the state transitions of an entire cell cycle considering oxygen, glucose and cell volume as controllers. A Tumor cell becomes necrotic if one of the concentrations of oxygen and glucose drops down to the defined necrotic thresholds. Similar checks on cell quiescence are done after the necrotic conditions have been passed (necrotic conditions are not met). If the quiescent requirements on the concentrations of both oxygen and glucose are satisfied (quiescent conditions are not met either), the volume of the cell is checked to guarantee that it can proceed its cell cycle only when it has actually obtained enough volume for its division. Once passing this final checkpoint, it can definitely finish the rest cell cycle phases and finally divides into two daughter cells which immediately start the next cell cycle from the $G1$ phase. Since CSC features are taken into account. The daughter cells may be a CSC or a common tumor cell, which depends on the type of the mother cell. After using up the proliferative potential, a common tumor cell executes apoptosis.

Parameter	Symbol	Unit	Value and Ref.
Oxygen diffusivity	D_{ox}	$\mu m^2 \cdot s^{-1}$	1750.0 [173]
Glucose diffusivity	D_{gl}	$\mu m^2 \cdot s^{-1}$	105.0 [187]
Oxygen uptake rate	U_{ox}	$amol \cdot cell^{-1} \cdot s^{-1}$	83±5.1 [175, 176]
Glucose uptake rate	U_{gl}	$amol \cdot cell^{-1} \cdot s^{-1}$	(180±7.1) [175])
$G1$-phase	τ_{G1}	h	8.0 ± 1.0(assumption)
$S/G2/M$-phase	τ_{SG2M}	h	(16.0 ± 3.0)(assumption)
Oxygen quiescence threshold	$Q_{THD_{O2}}$	$mmol \cdot L^{-1}$	0.016 (assumption)
Oxygen necrosis threshold	$N_{THD_{O2}}$	$mmol \cdot L^{-1}$	0.0082 [176]
Glucose quiescence threshold	$Q_{THD_{Glu}}$	$mmol \cdot L^{-1}$	0.14 (assumption)
Glucose necrosis threshold	$N_{THD_{Glu}}$	$mmol \cdot L^{-1}$	0.016 (assumption)
Mature cell radius	R_M	μm	9.1 [175, 179]
Newly born cell radius	R_Y	μm	7.2 [175, 179]
Necrotic cell radius	R_N	μm	6.75 [179]
Surface contact energy density	σ	$J \cdot m^{-2}$	1.0×10^{-4} [135]
Young's modulus	E	KPa	0.8 [167]
Poisson ratio	ν	-	0.499
Contact friction coefficient	η	$kg \cdot m^{-2} \cdot s^{-1}$	1.0×10^{11} [167]
Dynamic viscosity of the ECM	μ	$kg \cdot m^{-1} \cdot s^{-1}$	5.0×10^{3} [167]
Stem cell differentiation probability	P_D	-	0.5 (assumption)
Proliferative potential	PP	-	20 (assumption)
Cubic system size	CSS	μm	1000.0
Maximal time-step displacement	MTD	μm	4.0
Maximal time-step size	MTS	h	0.5

TABLE 5.1: Parameters used in simulations.

5.4 Results and Discussion

Based on the parameters given in table 5.1, a series of simulations of different conditions of oxygen and glucose are carried out. The simulation results are presented as follows.

5.4.1 Population Growth

We look at the growth dynamics of tumor cell population and compare our results with the experimental data by Freyer and Sutherland [7]. We neglect the tumor growth of the first 3 days by setting the initial tumor cell number according to the experiments. We do so because it is said that tumors undergo nothing particularly interesting but only exponential growth during this period of time [7].

Figure 5.2 illustrates the cell population growth curves and the corresponding experimental results under the four conditions introduced in the beginning of this section. Our simulation results agree with the experiments in all cases.

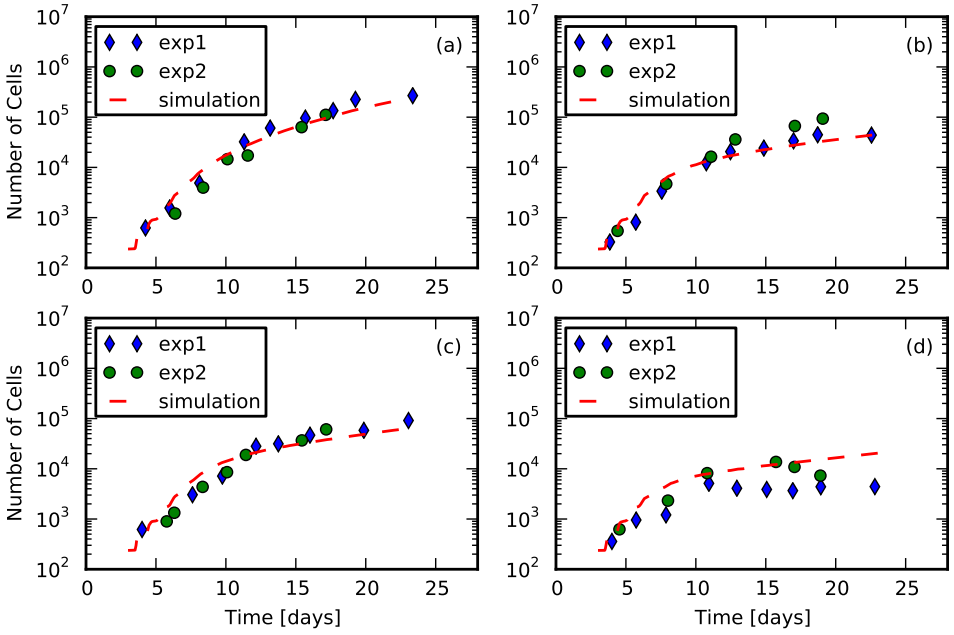


FIGURE 5.2: The cell population growth curves of tumors under different conditions of oxygen and glucose concentrations, namely 0.28 and 16.5 (case a), 0.07 and 16.5 (case b), 0.28 and 0.8 (case c), 0.07 and 0.8 (case d) (the unit is $mmol \cdot L^{-1}$). The dashed red lines are the simulation results. The solid blue diamonds and green circles are the corresponding experimental results taken from Ref. [7].

5.4.2 Distribution of Nutrients

To analyze the dynamics of oxygen and glucose, their distribution over the simulation space and time are presented in figure 5.3. This figure shows some important phenomena. First, clear concentration gradients for both oxygen and glucose have be obtained from the boundary to the geometric center of the simulation domain. Second, after some days (about 7-12 days), an inner lower plateau starts to form up inside tumor tissues, which corresponds to tumor necrosis (the gray inner regions of the tumor tissues shown in figure 5.4). Similar oxygen concentration gradients have been observed in experiments with the same tumor cell line but different oxygen and glucose conditions [183, 184, 188]. Third, the diffusion distance of oxygen obtained from our simulations are about 200-250 μm, which agrees well with that measured in experiments [184, 189].

5.4.3 Invasive Morphology

Besides, we also investigate the invasive morphology of growing tumors. The radical Voronoi tessellations of multicellular tumor tissues of our simulations are shown in figure 5.4. From this figure one can get a first impression of the morphological differences between tumors in different simulation conditions.

To quantitatively evaluate the surface roughness of tumor tissues, we calculate the cell number density, which is normalized and plotted (dashed red lines) in figure 5.5. For doing so, we assume a hyperbolic tangent function $(A \cdot tanh(x/\delta + B) + C)$. This functional form is expected for the interface between two densities. In particular, we estimate the half width δ form the fit (solid blue lines) to the normalized cell density curves. A larger δ indicates a rougher surface. The value of δ (see figure 5.6) in the four simulated cases shows that the tumor surface gets rougher with the boundary conditions getting lower for both oxygen and glucose.

We have not taken part of the cells inner tumors into account in evaluating the tumor surface roughness, since the cell density increases for tumor necrotic core, which is because necrotic cells are designated to undergo volume shrinkage in our models. We believe doing so does not affect the investigation on the tumor surface roughness.

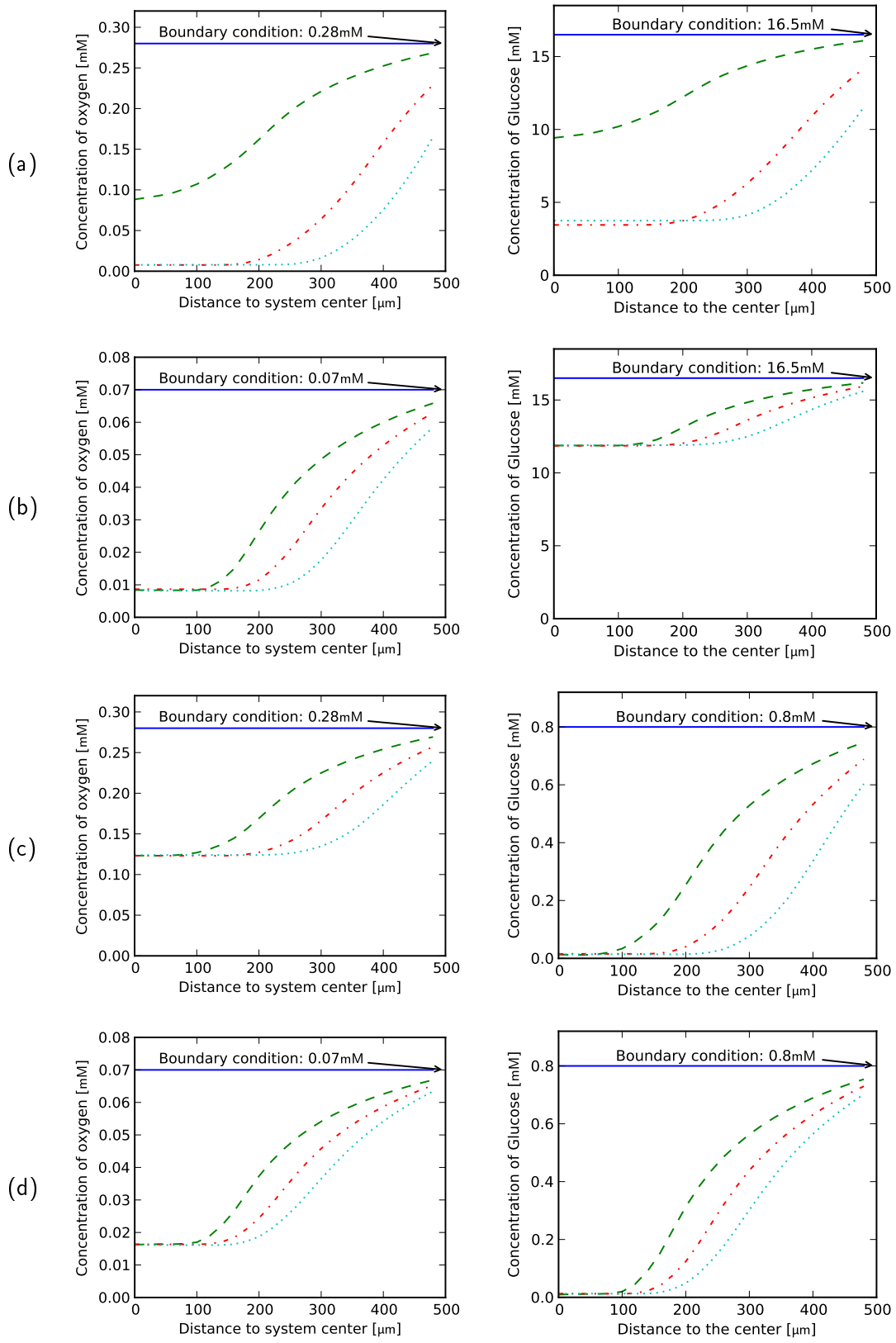


FIGURE 5.3: Molecular distributions of glucose and oxygen during simulations. Four lines from the above to the bottom (solid blue, dashed green, dashed red, and dashed cyan) in the 8 panels of this figure represent the concentrations of oxygen and glucose on day 0, 9, 16 and 23 separately.

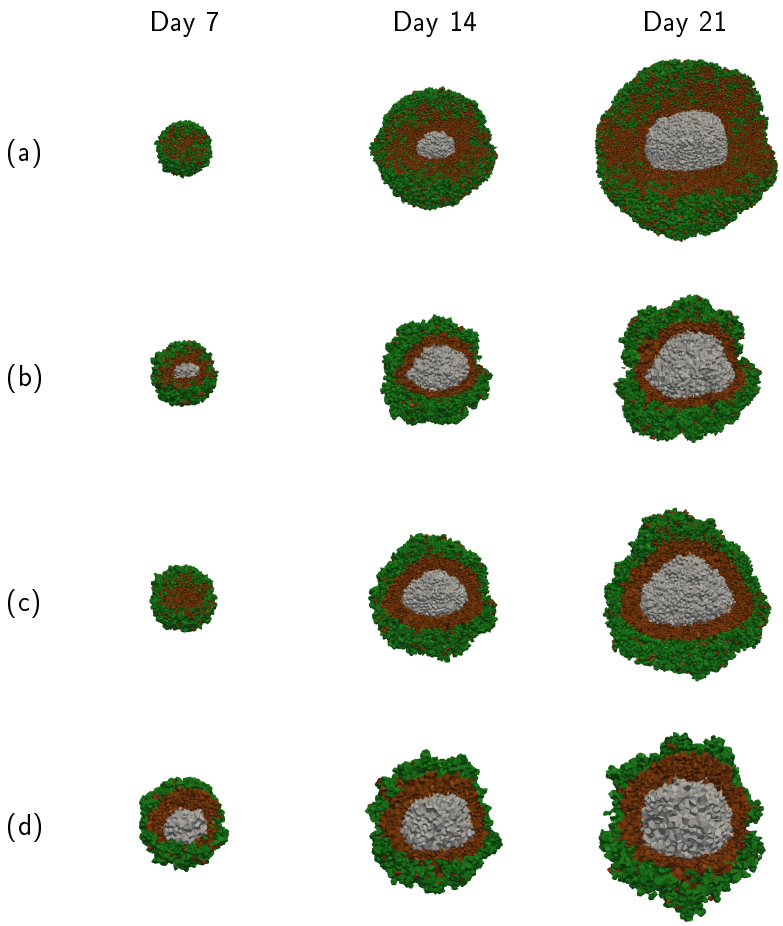


FIGURE 5.4: Illustration of the growth morphology of the simulated MTSs under different conditions of oxygen and glucose. Cases (a), (b), (c), (d) from the top to the bottom represent the four conditions of oxygen and glucose introduced in figure 5.2. The light gray region stands for the MTS necrotic core, which is surrounded by the dark brown rim of quiescent cells. The green layer consists of actively proliferative cells.

5.4.4 Discussion

We have constructed computational cancer models using TUGME. These models are characterized by considering dynamics at multiple spatial-temporal scales of the tumor-world using continuum and discrete approaches. In these models, individual cells undergo proliferation to grow and replicate themselves, which is controlled by the biophysical and biochemical conditions of the environment.

Figure 5.2 shows that tumors grow at a relatively fast rate for nearly 9-11 days in cases (a), (b) and (c) and for about 7 days in case (d). According the statistics of our simulation results, the cell number doubling times in our simulations are larger than the defined cell cycle duration (24 h) of our model, the reason is that some cells are temporarily trapped in the $G0$ phase because of not efficiently getting enough volume for cell division. This phenomenon reflects that the space competition between tumor

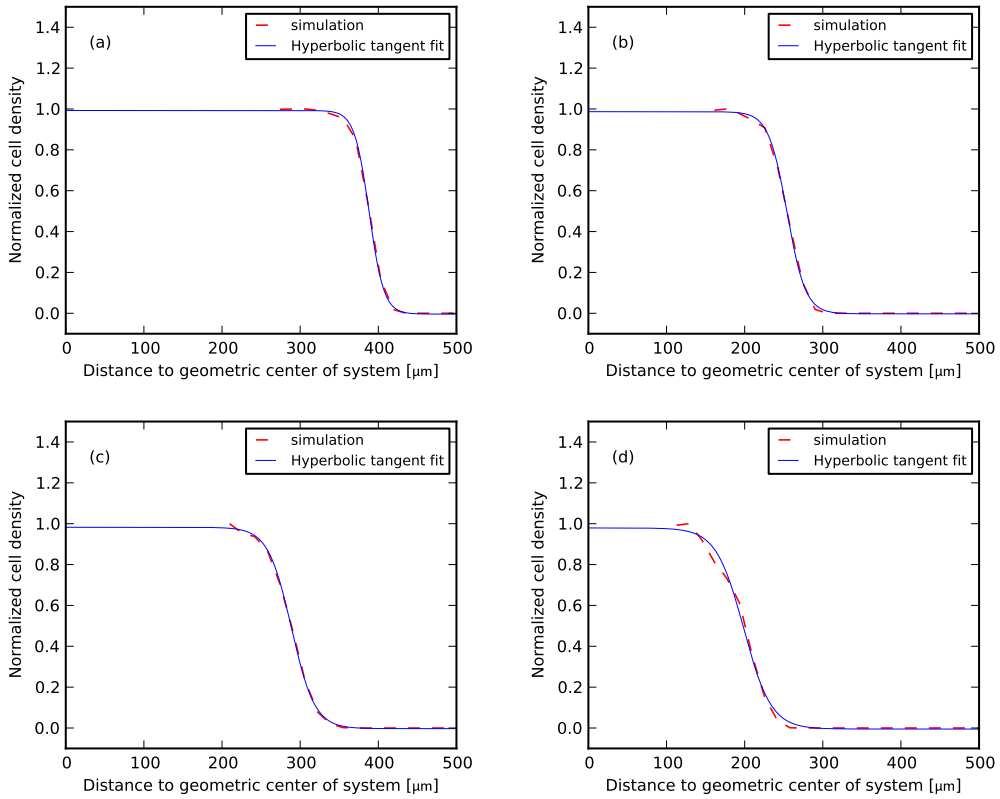


FIGURE 5.5: Comparison the MTS surface roughness under different nutrient conditions. Panels (a), (b), (c) and (d) represent four simulated cases. The dashed red lines are the simulation results averaged over 10 simulation runs under the same condition but different number of initial tumor cells. The solid blue lines are the corresponding curves fitted using the hyperbolic tangent formula $A \cdot tanh(x/\delta + B) + C$. A larger δ indicates a rougher surface. The averaged values of δ are plotted in figure 5.6.

cells can be so fierce during this fast growth stage that cell proliferation is affected. The initial cell number doubling times measured in experiments are 20-24 h [7], however, they are about 27-30 h in our simulations. Our explanation to this phenomenon is as follows. First, it can be easily seen from figure 5.2 that the cell population growth curves agree quite well with the experiment data, which indicates that the the cell number doubling times in our simulations are reasonable. The starting point of our simulation corresponds to roughly the third day of experiments, whose initial doubling times have been measured in the very beginning (the first day). Our simulation results conference with the conclusion that the growth rates slow as the tumor spheroids grow [7].

Our simulations indicate possible conditions for EMT6/Ro cells to undergo quiescence and necrosis. Specifically, the glucose concentration below the threshold 0.14 $mmol{\cdot}L^{-1}$ or the oxygen concentration below the threshold 0.016 $mmol{\cdot}L^{-1}$ lead to cell quiescence. Even lower concentrations such that glucose concentration is below the threshold 0.016 $mmol \cdot L^{-1}$ or oxygen concentration below the threshold 0.0082 $mmol \cdot L^{-1}$ result in cell necrosis. The value of the oxygen necrotic threshold (0.0082 $mmol \cdot L^{-1}$) is taken

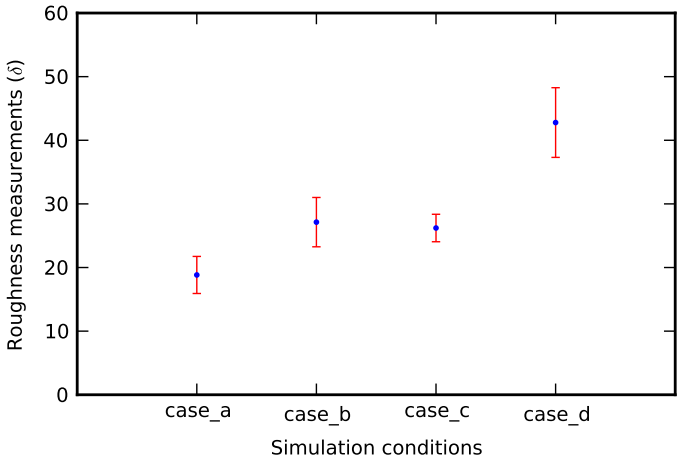


FIGURE 5.6: Evaluation of the MTS surface roughness. The averaged values of δ are 18.8, 27.1, 26.2, 42.8 (μm) in the cases (a), (b), (c) and (d) with standard deviations 2.9, 3.9, 2.2, 5.5.

from [176]. All other three parameters are assumed and adjusted in order to make our simulations agree with experiments.

For the nutrient that dominates tumor cell necrosis, its concentration level of the inner plateau (see figure 5.3) is below its corresponding necrotic threshold instead of zero, which indicates that the concentration of plateau of nutrients may be necrotic-threshold-dependent. Necrotic tumor cells are nutrient-independent, hence, there is a sharp decrease of nutrient consumption during the transition from quiescent to necrotic stat of cells, which results in a small fluctuation of nutrient concentrations during simulations. Since, the necrotic thresholds for oxygen and glucose are actually very small, their concentrations may drop below zero in our simulations, which is impossible in reality. We notice that this unrealistic phenomena only shortly occur at initial stage of tumor necrosis. After the plateau forming up, the concentration level of the plateau keeps above zero, which can be seen from figure 5.3 clearly.

The metabolic rates of nutrients by tumor cells directly determine their distribution over space and time. In the table 5.1, we have defined the averaged metabolic rates for oxygen and glucose. Obviously, they are valid only when both oxygen and glucose are sufficient for tumor cells, since the metabolic rate of oxygen and glucose varies significantly when one of them is not efficiently perfused [129, 175, 176]. Some tumor models assume that the metabolic rate of nutrients reduces by half for the quiescent cell compared with that of the proliferating cell. This assumption is too coarse compared with the roughly 100 times differences of the glucose metabolism found by Warburg in experiments [129]. In fact, we have tried this strategy in our early simulations,

which sometimes led to abnormal fluctuations to both oxygen and glucose concentrations during simulations. The side effect of the fluctuation is that some tumor cells are frequently switch between the actively proliferative and quiescent states, which is obviously not realistic. Our strategy to this problem is that the metabolic rates of oxygen and glucose by tumor cells are varied according the concentration levels of each other. For example, when glucose is sufficient, its metabolic rate may be doubled once the oxygen concentration level is extremely low (approaching the necrotic threshold of oxygen). In general, the metabolic rate may have at most 6 times variations ($\frac{U_{max}}{U_{min}}$, U_{max} and U_{min} stand for the largest and the smallest metabolic rate) for oxygen and 24 times for glucose in our model. And the rule is generally developed based on the experimental measurements published in [175].

Besides, one may also notice that tumor necrosis is triggered when one of the concentration of oxygen and glucose drops below the defined necrotic threshold in our simulations, which means that tumor cell necrosis actually happens even one of oxygen and glucose concentration is still much higher than their corresponding necrotic or even quiescent threshold (see the glucose concentration in cases (a) and (b) and the oxygen concentration in cases (c) and (d)). This is just our rule of interactions between tumor cells and nutrients. Some researchers may think that our rule is not realistic since they think tumor cells are able to survive as long as one of oxygen and glucose is still sufficient. Hence, only both of oxygen and glucose concentrations being critically low can lead to tumor cell necrosis. The underlying idea our rule is that tumor necrosis can actually happen when the oxygen level is high (at least not low enough to cause cell necrosis) for some solid tumors in reality. Hence, our inference would be that whether tumor cells can survive with only one of oxygen and glucose being sufficiently supplied may be cell-type-dependent. We do not declare that the rule used by us is universally applicable. One has to vary the rule accordingly. Furthermore, only the necrotic threshold of oxygen are taken from experiments, while all the other thresholds are assumed. We believe that experimental measurements of the critical concentration level of oxygen and glucose or other biochemicals for tumor cell quiescence and necrosis would be promising to improve the validity our tumor models.

Another observation of our simulation is that the correlation between nutrient supply conditions and the tumor growth morphology. Although the growth of tumors in our model was jointly influenced by both of the biomechanics and nutrients, the cases that nutrients dominate tumor growth obtained a better agreement with experiments in cell population dynamics (case (b), (c) and (d) shown in figure 5.2). The surface roughness of tumors in these cases points to the conclusion that a worse nutrient supply generally gives rises to a rougher the tumor surface. Our explanation to this phenomenon is that inefficient nutrient supply allows tumor cells different growth advantages, especially

those on the tumor surface (surface cells). Cells with larger growth advantages grow fast, which results in more fluctuations to the tumor surface. Furthermore, the further a branch of surface cells extends, the more growth advantages of the involved cells may get. The sparse distribution of proliferating cells on tumor surface shown in figure 5.4 partially proves the validity of this explanation. Besides, this explanation agrees with the finding that the tumor surface roughness is mainly due to contour cells that posses higher proliferating probability [190]. Our conclusion is not contradictory to that the tumor invasive morphology is affected by the biomechanical interactions between cells, which has been discussed in other models [60]. We believe that the invasive morphology of a tumor may vary if the biomechanical effects from normal cells surrounding the tumor tissue are not considered, since surface cells have more free space to move in this case.

5.5 Conclusion

We developed a multi-scale computational model for the multi-scale modeling of 3D tumor tissue growth, integrating continuum and discrete approaches for modeling different aspects of tumors. Cells in the model undergo proliferation interacting with each other and the environment biomechanically and biochemically. Simulation of the EMT6/Ro cell line showed the validity of the model as well as the joint effects of both biomechanics and nutrients on the tumor invasive morphology.

Angiogeneis is a critical step for the malignant transformation of tumors. The growth of tumors with a blood vessel network is of more practically relevance. Hence, modeling the blood vessel growth and specifying more real boundary conditions for RDEs of nutrients would be of interest. Another extension would be to introduce more cell cycle controllers, such as some cell metabolic products and some GFs.

Chapter 6

Computational Complexity of Agent-based Multi-scale Cancer Modeling

One of the main challenges of agent-based multi-scale cancer modeling is the explosive growth of the computational cost of model solving with increasing tumor system size and model complexity. Current models have to comprise a lot between model complexity and fidelity. We investigate the computational complexity of agent-based multi-scale cancer modeling and simulation using a model that has been introduced in chapter 5 of this thesis. We evaluate optional methods for attacking the computational bottleneck of our models and choose the best one in terms of the performance and the stability. This work can be important reference for researchers working on single-cell-oriented cancer modeling.

6.1 Introduction

6.1.1 Multi-scale Complexity of Tumor Growth

Cancer is widely accepted as a gene disease. Its development involves mutations or lesions to certain genes which are generally categorized into the proto-oncogenes and the tumor suppressor genes. The expression of these altered genes may give rise to new capabilities to tumor cells [4]. To perform its functionalities, a cell needs energy which is generated by biochemical reactions consuming nutrients like glucose and oxygen. Inner necrosis is common to solid tumors, which is believed to be tightly related to the

inefficient supply of oxygen [191, 192], since most tumor cell necrosis is found accompanied by hypoxia (the depletion of oxygen). In addition, cells interact mechanically with its nearest neighbors, the ECM and the extracellular materials. These biomechanical interactions are thought to influence the growth, invasion and metastases of tumor cells [10, 16, 17, 133, 137, 159].

Tumor systems are multi-scale by nature, involving the micro-scale dynamics of molecules and atoms to the macro-level dynamics of multicellular tissues and organs [4]. The dynamics of different spatial-temporal scales are not independent but rather nonlinearly coupled, producing the complex physiology of tumor systems. This complexity requires a holistic understanding [31], since single scale investigation is insufficient to uncover cross-scale mechanisms. Most research efforts by oncologists have focused on uncovering the molecular mechanisms regulating the behavior of individual tumor cells, for example, some cell signaling pathways have been reported particularly up- or down-regulated (activated or deactivated) in certain tumor cells [112, 113, 115–117], which has promoted developing new anti-cancer drugs by targeting these signaling pathways. However, molecular-scale mechanisms solely are not enough to explain the acquirement of drug resistance of cancer cells during chemotherapy. The mechanisms of how the factors except genetic reasons, like the cell-cell mechanical interactions, nutrients supply, pH values and various ions in intra- and inter-cellular space, affecting the growth and survival of tumor cells at multiple spatial-temporal scales are still not well understood.

It is difficult to address this problem with traditional biological means due to the limitations of spatial-temporal resolution of current apparatuses for both carrying out experiments and conducting measurements. As an auxiliary method, computer-based modeling and simulation has been playing an increasingly important role in investigating the problem with the rapid advances of both computer hardware and software in past two decades.

6.1.2 Simulation Time and Wall-clock Time

Two conceptions, namely simulation time and simulation wall-clock time, in computer-based modeling and simulation are worthy of special attention, when the computational cost of model solving particularly concerns a modeler. Simulation time is a measurement of the evolving time of the real system, while the time needed for running a simulation is termed simulation wall-clock time. These two conceptions are important, because wall-clock time can be much shorter, roughly the same as and longer than simulation time. These three cases correspond to three categories of simulations which

are faster-than-real-time simulations, real-time simulations and slower-than-real-time simulations.

This classification is important with respect to the purposes of simulations. Faster-than-real-time simulations allow to investigate processes whose evolution time can be too long (hundred years or even longer) to study directly with some other methods. For example, many cosmological problems fall into this category. Another application of faster-than-real-time simulations would be to see the long-term consequence of a dynamic process. For example, relatively short-term (months or several years) effects of low-dose ionizing radiations are practical to study in reality [193], however, long-term effects (decades or even longer) of even lower-dose ionizing radiations are still difficult to study directly in the laboratory. Real-time simulations are usually used to predict short term dynamics of systems. For example, real-time cancer simulations are used to test the effects of therapies and to predict the survival time of people with a cancer [194]. Slower-than-real-time simulations are commonly used to investigate very small spatial-temporal scale system dynamics, for example, various biochemical reactions, performing within tissue cells, which are too fast to monitor with current experimental apparatuses. Most molecular dynamics (MD) simulations fall into this category.

6.2 The Model

Tumor models used in the following tests have been detailedly introduced in chapter 5. And most parameters remain the same and those varied here for certain tests will be clarified particularly. As these models are coarse-grained, simulations based on TUGME are expected to be faster-than-real-time or at least to be real-time. In other words, the simulation wall-clock time should be no longer than the actual growth time (simulation time) of the simulated tumors.

6.3 Evaluating the Computational Cost of Model Solving

As a clinically detectable tumor usually comprises about 10^8 cells, our tumor models consist at least 10^4 cells.

All tests have been executed on a desktop computer with an Intel quad-core CPU, i5-3470, with 6M cache and up to 3.6 (base frequency 3.2) GHz single core frequency. The available RAM is 16 GB with designed frequency 1600 GHz. The operating system

(OS) is Debian GNU/Linux 7 with Linux kernel version number 3.2.0-4-amd64. The gcc version number is 4.6.3 (Debian 4.6.3-14).

6.3.1 Calculating the 3D Radical Voronoi Tessellation

The CPU time for calculating the 3D radical Voronoi tessellation using Voro++ has been tested under different numbers of cells. Figure 6.1 shows the results of this test. A linear relation between the CPU time and the cell number is shown clearly in this figure. This result is of no big surprise to us, since Voro++ calculates the Voronoi information of each cell independently, even though calculating the Voronoi information of two neighboring cells can be done together as the neighboring relationship is symmetric. By doing so, Voro++ doesn't store the Voronoi information of the cell calculated. The original goal of this design is to minimize the memory usage and to facilitate parallelizing this calculating process. As a result, Voro++ requires to recalculate the Voronoi information whenever it is needed.

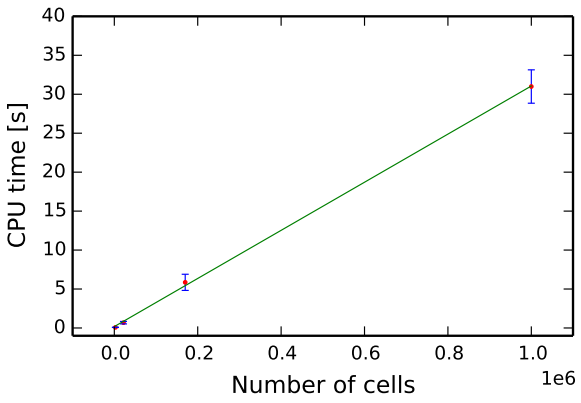


FIGURE 6.1: Performance test of the Voro++ package. Red dots are the results of our test. The solid green line is the linear least-square fitting. The slop of this line is about $6.176 * 10^{-5}$.

6.3.2 Solving the Linear System of Cell Motion

To find a proper numerical solver for our linear system of cell motion, the performance of several numerical solvers provided by DUNE has been comparatively tested. The tested solvers are the preconditioned stabilized bi-conjugate gradient method (BiCGStab), the preconditioned conjugate gradient method (CG), the preconditioned gradient method and the preconditioned loop method. The preconditioners include Gauss-Seidel (GS), Jacobi (diagonal) (Jac), incomplete LU factorization (decomposition) (ILU(0) and

ILU(1)), successive over-relaxation (SOR) and symmetric successive over-relaxation (SSOR).

Figure 6.2 and table 6.1 show the results of this test. Different solvers under different preconditioners (actually working ones) show distinct efficiency of convergence. In general, several conclusions can be drawn from the results. They are:

- CG and BiCGStab show similar efficiency and they clearly outperform the other solvers under the same preconditioner.
- The loop solver shows the worst efficiency of convergence under the same preconditioner compared with the other tested solvers.
- CG is not as stable as the other tested solvers. Its CPU time fluctuates largely especially when it is preconditioned by Jac for large systems (see the upper right panel in figure 6.2).
- ILU(1) outperforms the rest preconditioners for all tested solvers.
- SSOR shows the worst performance for all tested solvers.
- In many cases (see panels in figure 6.2 except the upper right one), GS and SOR show very similar performance for the same solver.
- For BiCGStab, ILU(0), GS and SOR show similar performance (see the upper left panel in figure 6.2).

Two preconditioners, GS and SOR, do not work with CG, which correspond to the empty columns in table 6.1 of CG. The reason is that CG requires a symmetric matrix of a linear system, however, these two preconditioners yield an asymmetric matrix when they are applied on a symmetric matrix like ours of the linear system of cell motion. The cases when solvers sometimes fail to get a solution are marked as "failed" in this table. As one can see from this table, these failures occur only for large systems. The failure is defined by failing to reach the given accuracy within 8,000 iterations.

Some properties of the matrix of a linear system are essential to guarantee the existence of a solution (the convergence of numerical solvers). And some properties may significantly affect the rate of convergence of numerical solvers. Take the CG solver as an example, it requires the matrix to be symmetric and positive-definite. Hence, theoretically, one needs to mathematically prove that the matrix of a linear system fulfills the requirements of a solver before applying it. In the next section, we are going to discuss some of the important properties of the matrix of our linear system of cell motion.

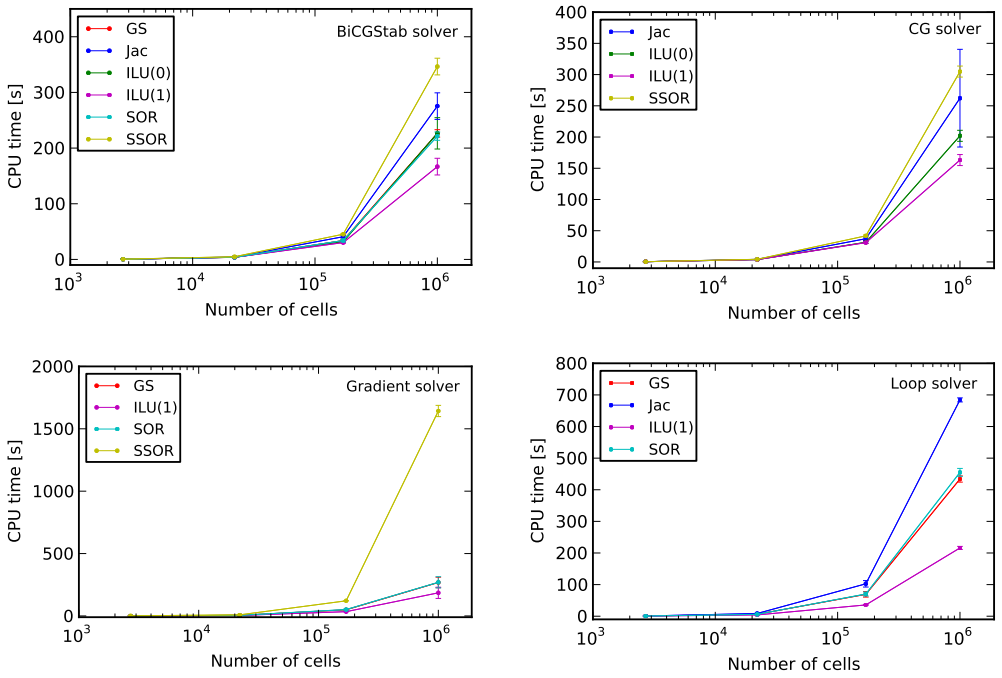


FIGURE 6.2: Performance test of four linear system numerical solvers, namely BiCGStab, CG, Gradient and Loop preconditioned by GS, Jac, (ILU(0) and ILU(1)), SOR, SSOR. Corresponding data can be found in table 6.1.

TABLE 6.1: Performance test of preconditioned linear system numerical solvers.

Solver		CPU time					
		GS	Jac	ILU(0)	ILU(1)	SOR	SSOR
BiCGStab	2.710^3	0,419	0.474	0.445	0.43	0.414	0.471
	2.210^4	3.933	4.229	3.768	3.717	3.863	4.811
	1.710^5	33.907	40.780	31.877	30.231	33.96	45.285
	1.010^6	226.323	275.398	226.515	166.705	221.469	346.464
CG	2.710^3	–	0.42	0.41	0.44	–	0.44
	2.210^4	–	4.17	3.72	3.73	–	4.26
	1.710^5	–	37.19	31.51	30.92	–	41.79
	1.010^6	–	262.23	201.94	163.14	–	304.83
Gradient	2.710^3	0.51	0.72	0.48	0.46	0.48	0.8
	2.210^4	4.93	8.32	4.20	3.99	4.97	8.67
	1.710^5	49.25	failed	40.03	34.25	51.20	121.19
	1.010^6	268.23	failed	failed	185.27	269.60	1642.72
Loop	2.710^3	0.64	0.79	0.53	0.45	0.55	3.85
	2.210^4	6.16	8.6	4.76	4.12	5.89	48.16
	1.710^5	69.19	102.5	35.58	35.67	70.19	601.32
	1.010^6	433.97	684.25	failed	216.01	454.98	failed

First of all, the matrix of our linear system is symmetric, since the neighboring relationship between tumor cells is symmetric. The matrix actually takes the form (detailed derivations can be found in appendix A)

$$\mathbf{M}_{ij} = \begin{cases} \boldsymbol{\gamma}_i + \sum_{k \in N_i} \boldsymbol{\gamma}_{ik} & \text{if } i = j \\ -\boldsymbol{\gamma}_{ij} & \text{if } i \neq j \text{ and if } j \in N_i \\ (0)_{3\times 3} & \text{otherwise} \end{cases} \tag{6.1}$$

where $\boldsymbol{\gamma}_i$ is the friction from the ECF on cell i, $\boldsymbol{\gamma}_{ij}$ is the friction from contact neighboring cell j and N_i is the number of contact neighbors.

However, the symmetry of the tumor cell neighborhood is not perfectly guaranteed by the Voro++ package for the numerical accuracy limitations of computers. As it is declared by the author of Voro++, this problem may happen when the contact area between two direct neighboring cells is extremely small. Since a numerical tolerance is given, where values smaller than this tolerance are replaced by zeros by Voro++. This problem indeed occurred in our initial test of Voro++. In order to overcome it, we made the tolerance of Voro++ much smaller than its default value. Afterwards, this problem has not occurred in our further tests. Furthermore, physical variables (like the contact force between tumor cells) related to the cell neighboring relationship usually vanish when the contact area between cells becomes extremely small according the the cell-cell contact force model. Thus, even though this problem happens in reality, it would not affect the overall accuracy of our model too much. In a word, we assume that the matrix of our linear system of cell motion is symmetric.

Secondly, our linear system matrix is very sparse. From equation 6.1, one can see that for any row i (it actually represents three rows of the three directions of cell velocity in three dimensions) of the matrix, only the elements, whose corresponding cells actually neighbor to cell i, are non-zero. Hence, the averaged number of neighbors of tumor cells indicates the sparsity of our linear system matrix.

To quantitatively see the sparsity of the matrix of our linear system, the averaged number of neighbors of single tumor cells of different cell radius sizes, changing within its theoretical size range, has been evaluated. The simulation domain size is fixed to be $400 \times 400 \times 400$ (μm). The number of cells increases from 18,000 to 64,000 (doubled twice), which roughly corresponds to the decrease of the cell radius size from 9.1 to 6.25 (μm). According to our test, the averaged number of neighbors of a single tumor cell is around 15 with a standard deviation of about 4, which means that there are on average about 15 non-zero elements in each row of the matrix of our linear system of cell motion. Another important point is that this number almost keeps unchanged

even if the cell density is doubled in our test. Comparing this number with the number of tumor cells (more than 10^4) in our serious simulation indicates that the matrix is actually extremely sparse. Figure 6.3 shows the matrix sparsity of two small tumor systems.

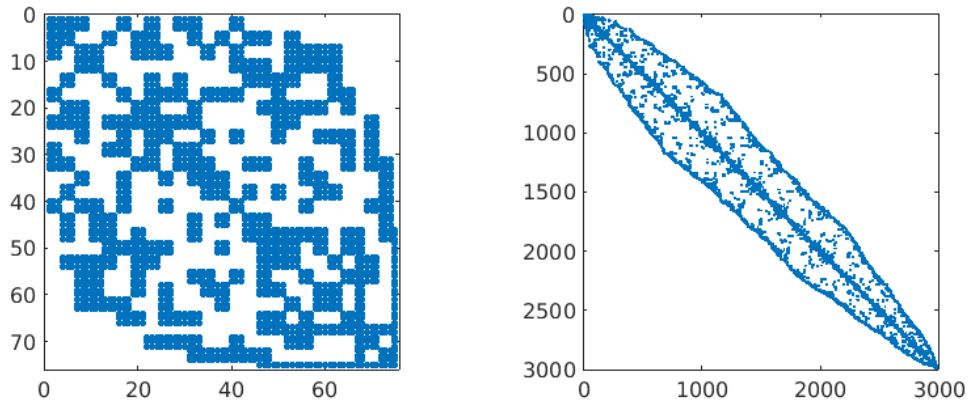


FIGURE 6.3: Illustration of the matrix sparsity of the linear system of cell motion. The two panels represent different matrix sizes, 75×75 and 3000×3000, which correspond to the tumor systems of 25 and 1000 cells. Blue square dots, which look much larger in size in the left panel than that in the right panel because of the rescale of the figures according to the computational domain size, except those diagonal ones, in this figure represents a 3×3 matrix, indicating the neighboring relationship between cells of i (row) and j (column).

Thirdly, the matrix of the linear system of cell motion is positive-definite. The detailed mathematical proof of this conclusion can be found in appendix B.

The larger the condition number of the matrix of a linear system, the worse it is conditioned. A bad conditioned matrix indicates that a small perturbation of $\mathbf{b}$ may lead to a large variation of the numerical solution of the system of the form $\mathbf{Ax} = \mathbf{b}$. Hence, it is important to evaluate the condition number of the matrix of a linear system.

The condition number of a matrix depends on the type of the norm ($\|\cdot\|$) applied. Here, we choose the l^2 norm which is usually noted as $\|\cdot\|_2$. As the matrix $\mathbf{M}$ of our linear system of cell motion is symmetric and positive-definite, its condition number can be formulated as:

$$\kappa(\mathbf{M}) = \frac{\lambda_{max}}{\lambda_{min}} \tag{6.2}$$

where λ_{min} and λ_{max} are the minimal and the maximal eigenvalues of $\mathbf{M}$ respectively. Next, we will evaluate λ_{min} and λ_{max} of the matrix of our linear system based on the Rayleigh quotient [195], the inequality is:

$$\lambda_{min} \leqslant \frac{\mathbf{x}^\intercal \mathbf{M} \mathbf{x}}{\|\mathbf{x}\|_2^2}. \leqslant \lambda_{max} \tag{6.3}$$

According the derivation in appendix B, one can get:

$$\mathbf{v}^\intercal \mathbf{M}\mathbf{v} = 6\pi\mu \sum_{i=1}^{N} R_i \|\mathbf{v}_i\|_2^2 + \eta \sum_{((i,j)\in E)\wedge(i<j)} A_{ij} \|(\mathbb{1}_{3\times 3} - \mathbf{P}_\perp^{ij}) \cdot (\mathbf{v}_i - \mathbf{v}_j)\|_2^2. \tag{6.4}$$

Supposing that R_{min} is the minimal radius of tumor cells, one can get:

$$\mathbf{v}^\intercal \mathbf{M}\mathbf{v} \geqslant 6\pi\mu R_{min} \sum_{i=1}^{N} \|\mathbf{v}_i\|_2^2 = 6\pi\mu R_{min} \|\mathbf{v}\|_2^2 \tag{6.5}$$

when

$$\mathbf{v}_i = \mathbf{v}_j \ (\forall i, j \in N) \wedge (i \neq j) \tag{6.6}$$

where N is the total number of cells.

Similarly, one can get:

$$\mathbf{v}^\intercal \mathbf{M}\mathbf{v} \leqslant 6\pi\mu R_{max} \|\mathbf{v}\|_2^2 + \eta \sum_{((i,j)\in E)\wedge(i<j)} A_{ij} \|(\mathbb{1}_{3\times 3} - \mathbf{P}_\perp^{ij}) \cdot (\mathbf{v}_i - \mathbf{v}_j)\|_2^2 \tag{6.7}$$

where R_{max} is the maximal cell radius and

$$E = \{(i,j) : (1 \leq i \leq N) \wedge (j \in N_i)\} \subset I \times I, I = \{1,\ 2,\ \ldots,\ N\}. \tag{6.8}$$

Supposing that A_{max} is the maximal contact area between cells and λ_{max}^P is the maximal eigenvalue of the projection matrix $\mathbb{1}_{3\times 3} - \mathbf{P}_\perp^{ij}$ ($\lambda_{max}^P = 1$), one can get:

$$\mathbf{v}^\intercal \mathbf{M}\mathbf{v} \leqslant 6\pi\mu R_{max} \|\mathbf{v}\|_2^2 + \eta A_{max} \sum_{((i,j)\in E)\wedge(i<j)} \|(\mathbf{v}_i - \mathbf{v}_j)\|_2^2. \tag{6.9}$$

Applying the triangle inequality twice to $\|(\mathbf{v}_i - \mathbf{v}_j)\|_2^2$, The inequality 6.9 can be reformulated into:

$$\mathbf{v}^\intercal \mathbf{M}\mathbf{v} \leqslant 6\pi\mu R_{max} \|\mathbf{v}\|_2^2 + \eta A_{max} \sum_{((i,j)\in E)\wedge(i<j)} 2(\|\mathbf{v}_i\|_2^2 + \|\mathbf{v}_j\|_2^2). \tag{6.10}$$

Since the neighboring relationship is symmetric, the above inequality can be reformulated as:

$$\mathbf{v}^\intercal \mathbf{M}\mathbf{v} \leqslant 6\pi\mu R_{max} \|\mathbf{v}\|_2^2 + \eta A_{max} \sum_{((i,j)\in E)} (\|\mathbf{v}_i\|_2^2 + \|\mathbf{v}_j\|_2^2). \tag{6.11}$$

Supposing that the maximal number of neighboring cells is N_{max}, which means that each term $\|\mathbf{v}_i\|_2^2$ ($1 \leqslant i \leqslant N$) can repeatedly show up at most $2N_{max}$ times in the

summation $\sum_{((i,j)\in E)}(\|\mathbf{v}_i\|_2^2 + \|\mathbf{v}_j\|_2^2)$. Hence, the inequality 6.11 can be reformulated into:

$$\mathbf{v}^\intercal \mathbf{M}\mathbf{v} \leqslant 6\pi\mu R_{max}\|\mathbf{v}\|_2^2 + \eta A_{max} 2N_{max} \sum_{i=1}^{N} \|\mathbf{v}_i\|_2^2 \tag{6.12}$$

which equals to:

$$\mathbf{v}^\intercal \mathbf{M}\mathbf{v} \leqslant (6\pi\mu R_{max} + 2\eta A_{max} N_{max})\|\mathbf{v}\|_2^2. \tag{6.13}$$

According to the inequality 6.3, 6.5 and 6.13, one can get:

$$\begin{aligned} \lambda_{min} &= 6\pi\mu R_{min}, \\ \lambda_{max} &= 6\pi\mu R_{max} + 2\eta A_{max} N_{max}. \end{aligned} \tag{6.14}$$

Hence, the condition number of the matrix of the linear system of cell motion is:

$$\kappa(\mathbf{M}) = \frac{6\pi\mu R_{max} + 2\eta A_{max} N_{max}}{6\pi\mu R_{min}}. \tag{6.15}$$

From this equation, one can directly see that the condition number of the matrix of our problem is bounded since all the parameters except N_{max} in this equation are constant. As we have mentioned early that the averaged number of neighboring cells is about 15 with a standard deviation about 4, N_{max} is also a finite integer.

More importantly, the condition number is independent on the system size (the number of cells), which indicates that the number of iterations needed by the numerical solvers of linear systems should has an upper limit. In order testify this inference, we look at the number of iterations over increasing the system size. The result is shown in figure 6.4. From this figure, one can see clearly that the number of iterations stops increasing at the system size of roughly 10^4 cells.

6.3.3 Solving RDEs Using FEM

Although the RDE model of biochemicals like oxygen and glucose is continuous, they have to be solved by problem discretization over space and time, since our simulations are advanced in a time-step manner. We use FEM (the Finite Element Method) to solve the RDEs of both oxygen and glucose. DUNE, used to solve the linear system of cell motion, is actually designed to do this. We call its FEM interfaces directly.

Here, we discuss the spatial-temporal scale that is reasonable for cancer modeling, since FEM is a grid- or mash-based coarse-grained method. For normal reaction-diffusion processes with well-mixed molecules and a domain free for molecules to react and diffuse, a finer grid normally gives rises to a better solution of FEM. However, a multicellular tumor tissue is obviously not a free space for oxygen or glucose molecules

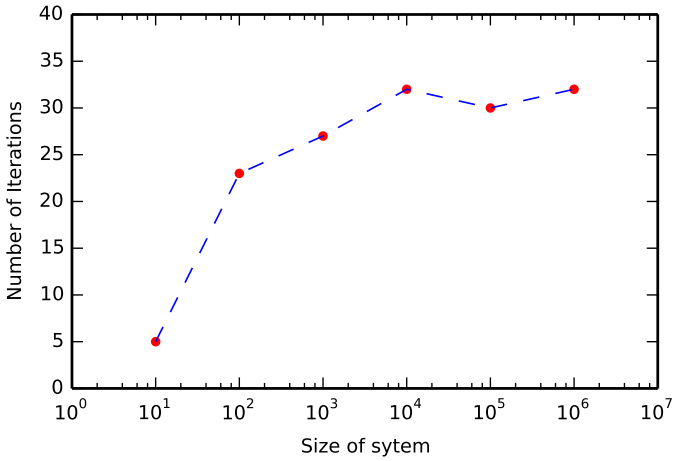


FIGURE 6.4: Number of iterations (averaged over 10 samples) needed by the numerical solver (BiCGStab) to converge with increasing the system size (the number of tumor cells).

to react and diffuse. The grid size of FEM is highly relevant to the spatial-scale at which the RDE model works, since various substances, such as the ECM, the cellular membranes and the organelles within cells, may significantly affect the diffusion of biochemical molecules and most biochemical reactions only take place at some specific organelles of cells, for example, cellular respiration is basically performed within the cell mitochondria. The experimentally measured diffusion coefficients of both oxygen and glucose are smaller than that in water [173, 187]. Obviously, these experimentally measured diffusion coefficients are the approximation at the macro-scale. Hence, it is isotropic in all directions in three dimensions. However, if the size of the FEM grid cell is smaller than that of single tumor cells, which means the spatial-scale of the model is smaller than single tumor cells, the diffusion coefficient should not be isotropic anymore, as it obviously differs in the directions of parallel and perpendicular to the cellular membrane. This micro-scale heterogeneity of the diffusion coefficient is actually hard to measure in experiments. Furthermore, we believe that anisotropic diffusion coefficients do not make too much sense for the overall accuracy of the coarse-grained cancer models like ours. Hence, we strongly recommend to use a FEM grid with its cell (the grid element) size larger than that of single tumor cells. Of course, the FEM grid should not be too coarse either considering the accuracy of the solutions. We use a grid whose grid cell volume is about 8 times of that of a single tumor cell in average in our serious simulations.

6.3.4 Performing Cell Proliferation

The CPU time needed for performing cell proliferation depends directly on the number of cells and the complexity of the cell cycle model. Our experiences of this is that performing cell cycle to all cells currently does not take too much of the overall CPU time compared with the other modular parts of our tumor model. One can see this from the test to be discussed in the next section.

6.3.5 Computational Bottlenecks

To find out the potential computational bottleneck of our tumor models, we look at the CPU time needed for solving each module of our models. Specifically, the modules include: calculating the 3D radical Voronoi tessellation, solving the linear system of cell motion, solving the RDEs of both oxygen and glucose and conducting cell proliferation.

The numerical solver and preconditioner for solving the linear system of cell motion are BiCGStab and ILU(1). The computational domain size is $1000 \times 1000 \times 1000$ (μm) with the initial tumor cell number $3.5 * 10^5$, which corresponds to an actual tumor of about 1 mm in diameter. The number of tumor cells is $4.62577 * 10^5$ at the end of the test. The FEM grid is a $30 \times 30 \times 30$ lattice. The result of this test is averaged over 100 simulation steps and shown in figure 6.5. This figure shows that solving the linear system of cell motion actually takes about 81% of the overall CPU time of each simulation step. Obviously, it is the computational bottleneck. Calculating the Voronoi tessellation and solving the RDEs of both oxygen and glucose take averagely about 11% and 7% of the entire CPU time of each simulation step separately. And cell cycle takes only about than 1% of the CPU time.

If our tumor models are used to simulate the growth of tumors with more than 10^6 cells, the simulation time step size has to be much larger than the total CPU time (about 110 seconds) needed for model solving in each simulation step in order to achieve faster-than-real-time simulations. Real-time tumor simulations may be also useful to predict the growth and invasion of tumor tissues in order to anticipate the survival time of persons with a cancer. Otherwise, our tumor models can only be used for cancer system analysis, for example, analyzing some of the working mechanisms within tumor cells.

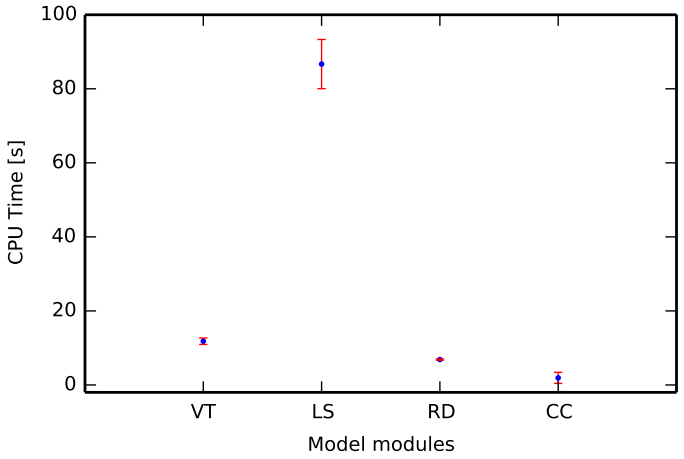


FIGURE 6.5: CPU time required by model solving. VT stands for calculating the 3D radical Voronoi tessellation. LS stands for solving the linear system of cell motion. RD is solving the RDEs of oxygen and glucose. CC represents performing cell proliferation. The number of tumor cells is $4 * 10^5$. The FEM grid applied to solve the RDEs is a $30 \times 30 \times 30$ lattice. The condition for numerical solvers to converge is theoretically described by $\|\mathbf{b}_{i+1} - \mathbf{b}\|_2 < \|\mathbf{b}\|_2 \cdot 10^{-10}$.

6.4 Summary

This chapter generally shows the possible computational complexity, which agent-based multi-scale cancer modeling may involve. If faster-than-real-time simulations are expected, the tumor model complexity has to be properly controlled, otherwise, innovative approaches are required to speed up simulations, like making use of the computational powder of high-performance computers whose computational potential has not been well explored in cancer modeling by now.

Our simulations have been focused on the growth of avascular tumors, whose size is much smaller than that of a clinically detectable tumors (with more than 10^8 cells). It is not difficult to anticipate that the computational challenges of modeling tumors of that size according to the tests presented in this paper when single-cell-oriented tumor models are applied. In general, these is no doubt that current computational challenges will become even severe for large tumor systems. More importantly, new challenges will show up gradually with increasing tumor size, for example, the memory cost of models may easily exceed the memory limit of a single computer.

Chapter 7

Summary and Future Work

7.1 Summary

A tumor is multi-scale and multi-factorial by nature. Understanding its working mechanisms at multiple spatial-temporal scales is of great help to the innovation of cancer therapies. Inspired by the wide application of the modeling and simulation techniques in investigating complex systems, we aim at attacking the tumor development problem, which is very hard to study by traditional biological means, by building cancer models and carrying out simulations on computers.

7.1.1 Development of a Tumor Growth Modeling and Simulation Software Environment

Some biological, biochemical and biophysical properties differ for distinct types of tumors, which requires to change the tumor model accordingly. Model modifications may also be required for testing different hypotheses. Model reuse can help reduce the costs of developing new models and avoids repeatedly creating similar ones. Aiming at facilitating model reuse, we have developed an agent-based multi-scale tumor growth modeling and simulation software environment TUGME. Models in TUGME use discrete and continuous approaches to couple the biological, biochemical and biophysical processes of multicellular tumor systems. The spatial scale of models is restricted from individual cells, which are modeled by single agents, to multicellular tumor tissues, which have the size (the diameter) of several centimeters. The temporal scale of models is limited from seconds to months. TUGME is implemented using the C++ programming language. It is basically built on an existing software DUNE, which is designed for numerically solving PDEs using grid-based methods. TUGME is highly

modularized, which makes it easy to maintain and extend. In chapter 4, the modules of TUGME and how to use it to build a tumor model and to carry out a simulation have been introduced in details.

7.1.2 Modeling the Growth of the EMT6/Ro MTSs

To show that TUGME is actually applicable, we have developed a tumor model by assembling the modular parts provided by TUGME to simulate the growth of MTSs of a tumor cell line, EMT6/Ro in chapter 5. In our model, each cell interacts biochemically and biomechanically with its surroundings, such as other cells and the extracellular materials. The morphological deformation of individual cells and the neighboring relationship among cells are represented by a 3D Voronoi tessellation. Driven by the mechanical forces, cells move in an over-damped manner, which is described by Newton' second law, where the linear equations of cell motion of all cells are coupled by cell velocity, which results in a linear system. Oxygen and glucose are taken as the nutrients that directly affect cell proliferation. The transport and metabolism of them are realized by RDEs, where the transport and cell metabolism of nutrient molecules are modeled as molecular diffusion and reaction respectively. Based on the model, a series of simulations have been carried out and simulation results are compared with experimental results. A good agreement of the tumor cell population growth between our simulations and experiments shows both the validity of our model and the practical applicability of TUGME. We also look at the dynamics of oxygen and glucose over simulation space and time, which is only partially compared with related experimental data, since no one-to-one experiment has been found. Besides, we discuss the influences of nutrient conditions on the invasive morphology of tumors. Our general conclusion is that a less efficient nutrient supply gives rise to a rougher tumor surface.

7.1.3 Evaluating the Computational Cost of Agent-based Multi-scale Tumor Modeling

One of the main challenges of conducting agent-based multi-scale tumor simulations is the growth of the computational cost with increasing the size and the complexity of models. The computational challenge significantly restricts the application of current tumor models. In chapter 6, we have investigated this problem by evaluating the computational bottleneck of our tumor model developed in chapter 5. For our model, the bottleneck consists in solving the linear system of cell motion, which takes about 81% of the total CPU time of each simulation step. This result can be important reference to people working on single-cell-oriented cancer modeling. In addition, we

have tested several linear system numerical solvers in order to find the best one in terms of the performance and the stability. According to our tests, the BiCGStab method preconditioned by the ILU(1) preconditioner outperforms the other numerical solvers we have tested. We have looked at some important properties of the matrix of our linear system of cell motion. In general, it is extremely sparse, symmetric and positive-definite. This result may help find more efficient numerical solvers for the linear system of cell motion.

7.2 Future Work

7.2.1 Testing Different Hypotheses About Cell Dynamics

Multi-factors directly or indirectly influence the development of a tumor. The models used in our simulations are relatively simple compared with the complexity of actual tumor systems. Hence, there is a very large space to vary our tumor models, for example, developing more realistic cell cycle rules by introducing more controllers of relevant. Our nearest future work will generally focus in this direction.

7.2.2 Modeling Dynamics of Advanced Tumors

Angiogenesis after the avascular tumor growth stage is very interesting to many cancer researchers, since it is critical for tumor malignant transformation and blocking it would theoretically stop the incidence of a cancer in reality. Since angiogenesis is believed driven by various GFs released by tumor cells, the production and transport of these GFs can be easily modeled using the RDE interfaces of TUGME. The challenges of doing so may be defining a proper distribution for the blood vessel network within the normal tissues surrounding a tumor.

7.2.3 Modeling Drug Delivery

Some anti-cancer drugs, which have shown promising curative effects in laboratory experiments, failed in clinical tests for many reasons, like its inefficient delivery and/or toxicity. Valid cancer models would be of great practical value to testing the curative effects, the toxicity, the delivery etc of these drugs. The RDE interface of TUGME can be adopted to model drug delivery. More importantly, drug delivery can be modeled coupled with tumor growth, which would generally enable more persuasive simulation results than solely modeling the the drug delivery process.

7.2.4 Extending TUGME for Parallel Simulations

Modeling the growth of a tumor starting from one or several cells to a mature tumor of millions of cells would be very meaningful to better understanding its whole developing process. However, according to the tests presented in chapter 6, it would be very time-consuming to model a tumor system with more than millions of cells using current models similar to that provided by TUGME on a single computer. One way to address this problem is to parallelize the model to take advantage of the high performance computers. The most important direction of our future work is to extend TUGME with parallel simulation support. As a matter of fact, the current implementation of TUGME has already partially considered the requirements of model parallelization, for example, the `UUID` is only used in the parallel simulation case. Model parallelization usually requires system partition, which further leads to the so-called boundary problems of sub-systems. Take tumor model as an example, one cell needs to know the information of its direct neighbors that may be assigned to another sub-system. Besides, load-balance closely related to system partition is a key problem in parallel simulations. In a word, model parallelization is much more complex to implement compared with the sequential one.

Appendix A

Derivation of the Linear System of Cell Motion

This work is initially done by Jörg Eisele and then reformulated by the author of the thesis.

In 3D space, Stokes friction coefficient of cell i can be formulated in the tensor form,

$$\boldsymbol{\gamma}_i = 6\pi\mu R_i \mathbb{1}_{3\times 3},$$

where μ is the dynamic viscosity of the ECM. R is cell radius, and $\mathbb{1}_{3\times 3}$ is the identity matrix in three dimensions. The Stokes friction force on cell i exerted by the ECM is

$$\mathbf{F}_i^f = -\boldsymbol{\gamma}_i \cdot \mathbf{v}_i.$$

The friction $\mathbf{F}_{ij}^f$ on cell i exerted by the neighboring cell j is proportional to the relative velocity between them:

$$\mathbf{F}_{ij}^f = -\boldsymbol{\gamma}_{ij} \cdot (\mathbf{v}_i - \mathbf{v}_j) \ ,$$

where $\mathbf{v}_i$ and $\mathbf{v}_j$ stand for their velocities. The friction is characterized by the friction coefficient

$$\boldsymbol{\gamma}_{ij} = c_{\parallel}\mathbf{P}_{\parallel} + c_{\perp}\mathbf{P}_{\perp} \ ,$$

where $c_{\parallel}$ and $c_{\perp}$ are the friction coefficients parallel and orthogonal to contact face between cell i and cell j, and

$$\mathbf{P}_{\perp} = \mathbf{n} \cdot \mathbf{n}^{\intercal}$$

projects vectors on the vector space spanned by a vector

$$\mathbf{n} = (n_1,\ n_2,\ n_3)^\intercal$$

normal to the interface between cell i and cell j, and

$$\mathbf{P}_\parallel = \mathbb{1}_{3\times 3} - \mathbf{P}_\perp$$

projects vectors on the vector space spanned by the contact face between cell i and cell j. It is assumed that $c_\perp = 0$ and $c_\parallel = \eta A_{ij}$, where η is the contact friction coefficient, and A_{ij} is the contact area. The friction coefficient between cells can be reformulated into:

$$\boldsymbol{\gamma}_{ij} = \eta A_{ij}(\mathbb{1}_{3\times 3} - \mathbf{P}_\perp).$$

In summary, the total force on cell i is

$$\mathbf{F}_i^t = \sum_{j\in N_i} \mathbf{F}_{ij}^c - \sum_{j\in N_i} \boldsymbol{\gamma}_{ij}\ (\mathbf{v}_i - \mathbf{v}_j) - \boldsymbol{\gamma}_i \mathbf{v}_i\ ,$$

where $\mathbf{F}_{ij}^c$ is the contact force on cell i exerted by cell j, and N_i is the set of neighbors of cell i.

Assuming that cell motion is over-damped, and that is $\mathbf{F}_i^t = 0$. According to the Newton's second law, one gets

$$\mathbf{0} = \mathbf{F}_i^t = \sum_{j\in N_i} \mathbf{F}_{ij}^c - \sum_{j\in N_i} \boldsymbol{\gamma}_{ij}\ (\mathbf{v}_i - \mathbf{v}_j) - \boldsymbol{\gamma}_i \mathbf{v}_i\ .$$

The equation can be reformulated as

$$\sum_{j\in N_i} \mathbf{F}_{ij}^c = \boldsymbol{\gamma}_i \mathbf{v}_i + \sum_{j\in N_i} \boldsymbol{\gamma}_{ij}\ (\mathbf{v}_i - \mathbf{v}_j)\ ,$$

Combining like terms, it is finally reformulated as

$$\sum_{j\in N_i} \mathbf{F}_{ij}^c = (\boldsymbol{\gamma}_i + \sum_{j\in N_i} \boldsymbol{\gamma}_{ij}\ )\mathbf{v}_i - \sum_{j\in N_i} (\boldsymbol{\gamma}_{ij}\ \mathbf{v}_j)\ .$$

Equations of motion of all cells together result in the linear system:

$$\mathbf{F}^c = \mathbf{M}\mathbf{v}\ ,$$

where the block vector

$$\mathbf{F}^c = \begin{pmatrix} \sum_{k\in N_i} \mathbf{F}^c_{1k} \\ \sum_{k\in N_2} \mathbf{F}^c_{2k} \\ \vdots \\ \sum_{k\in N_N} \mathbf{F}^c_{Nk} \end{pmatrix} \in \left(\mathbb{R}^3\right)^N$$

(N is the total number of cells) and the block matrix

$$\mathbf{M} \in \left(\mathbb{R}^3\right)^N \times \left(\mathbb{R}^3\right)^N$$

with

$$\mathbf{M}_{ij} = \begin{cases} \boldsymbol{\gamma}_i + \sum_{k\in N_i} \boldsymbol{\gamma}_{ik} & \text{if } i = j \\ -\boldsymbol{\gamma}_{ij} & \text{if } i \neq j \text{ and if } j \in N_i \\ (0)_{3\times 3} & \text{otherwise.} \end{cases}$$

and

$$\mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_N \end{pmatrix} \in \left(\mathbb{R}^3\right)^N$$

with

$$\mathbf{v}_i = (n_{i1},\ n_{i2},\ n_{i3})^\intercal$$

Thus, vectors $\mathbf{F}^c$ and $\mathbf{v}$ consist of N blocks, each representing a cell. Each block has 3 entries, which represent x, y and z coordinates associated with the corresponding cell. $\mathbf{M}$ is a matrix of $N \times N$ blocks. And each of these blocks is a 3×3 matrix.

The time-step size is dynamically adjusted according to the velocity of cells. It is chosen such that the maximal displacement of cells is controlled to ensure both accuracy and efficiency. The idea is that the maximal timestep displacement MTD (see table 5.1) is fixed for all cells during each time step. Hence, the time-step size Δt is calculated by dividing MTD using the maximal cell velocity V_{max}: $\Delta t = MTD/V_{max}$.

Theoretically, the matrix of the linear system of cell motion is symmetric and very sparse. There are efficient algorithms specifically optimized for solving large sparse linear systems. DUNE, the Distributed and Unified Numerics Environment is a modular toolbox for solving partial differential equations (PDEs) with grid-based methods [143]. Many linear system solvers have been implemented or integrated in this toolbox, such as the preconditioned conjugate gradient (CG) method, the preconditioned biconjugate gradient stabilized (BiCGStab) method and superLU. We have tested the performance of some these solvers under different preconditioners for our linear system of cell motion,

and the results have been presented in chapter 6.3.2. Considering both of the efficiency and the stability of them, we have actually used the preconditioned BiCGStab solver with the ILU(1) preconditioner.

Appendix B

Positive-definiteness of the Matrix of the Linear System

A matrix $\mathbf{M}$ is positive-definite if and only if the inequality

$$\mathbf{x}^\intercal \mathbf{M}\mathbf{x} > 0,\ \forall\ \mathbf{x} \neq 0 \tag{B.1}$$

is fulfilled. For the linear system of cell motion, $\mathbf{x}$ is the velocity $\mathbf{v}$ of cells.

According to the derivation of the linear system of cell motion introduced in appendix A, the target inequality can be reformulated as:

$$\mathbf{v}^\intercal \mathbf{M}\mathbf{v} = \sum_{i=1}^{N} \mathbf{v}_i^\intercal \cdot \left(\boldsymbol{\gamma}_i \cdot \mathbf{v}_i + \sum_{j \in N_i} \boldsymbol{\gamma}_{ij} \cdot (\mathbf{v}_i - \mathbf{v}_j) \right) > 0,\ \forall\ \mathbf{v} \neq 0 \tag{B.2}$$

where $\mathbf{v}$ is the block vector of cell velocity and $\mathbf{v}_i$ is the velocity of cell i (`TID`). N_i is the set of the `TID` of neighboring cells. Expending the middle part of inequality B.2, one gets:

$$\sum_{i=1}^{N} \mathbf{v}_i^\intercal \cdot \boldsymbol{\gamma}_i \cdot \mathbf{v}_i + \sum_{i=1}^{N} \sum_{j \in N_i} \mathbf{v}_i^\intercal \cdot \boldsymbol{\gamma}_{ij} \cdot (\mathbf{v}_i - \mathbf{v}_j). \tag{B.3}$$

To facilitate the derivation, we define a set

$$E = \{(i,j) : (1 \leq i \leq N) \wedge (j \in N_i)\} \subset I \times I, I = \{1,\ 2,\ \ldots\ ,\ N\} \tag{B.4}$$

based on the neighboring relationship of cells, where N is the number of cells. Based on this definition, formula B.3 can be reformulated into:

$$\sum_{i=1}^{N} \mathbf{v}_i^{\intercal} \cdot \boldsymbol{\gamma}_i \cdot \mathbf{v}_i + \sum_{((i,j)\in E)\wedge(i<j)} [\mathbf{v}_i^{\intercal} \cdot \boldsymbol{\gamma}_{ij} \cdot (\mathbf{v}_i - \mathbf{v}_j) + \mathbf{v}_j^{\intercal} \cdot \boldsymbol{\gamma}_{ji} \cdot (\mathbf{v}_j - \mathbf{v}_i)]. \tag{B.5}$$

as the neighboring relationship is symmetric.

According the definition of $\mathbf{P}_{\perp}^{ij}$, one can get:

$$\mathbf{P}_{\perp}^{ji} = (-\mathbf{n}_{ij}) \cdot (-\mathbf{n}_{ij}^{\intercal}) = (\mathbf{n}_{ij}) \cdot (\mathbf{n}_{ij}^{\intercal}) = \mathbf{P}_{\perp}^{ij} \tag{B.6}$$

where $\mathbf{n}_{ij}$ is the normalized normal vector of the contact face between cell i and cell j. Hence, one gets:

$$\boldsymbol{\gamma}_{ji} = \eta A_{ji}(\mathbb{1}_{3\times3} - \mathbf{P}_{\perp}^{ji}) = \eta A_{ij}(\mathbb{1}_{3\times3} - \mathbf{P}_{\perp}^{ij}) = \boldsymbol{\gamma}_{ij} \tag{B.7}$$

since equation $A_{ji} = A_{ij}$ holds as A_{ij} is the contact area between cells.

Based on equation B.7, formula B.5 can be reformulated into:

$$\sum_{i=1}^{N} \mathbf{v}_i^{\intercal} \cdot \boldsymbol{\gamma}_i \cdot \mathbf{v}_i + \sum_{((i,j)\in E)\wedge(i<j)} [\mathbf{v}_i^{\intercal} \cdot \boldsymbol{\gamma}_{ij} \cdot (\mathbf{v}_i - \mathbf{v}_j) - \mathbf{v}_j^{\intercal} \cdot \boldsymbol{\gamma}_{ij} \cdot (\mathbf{v}_i - \mathbf{v}_j)]. \tag{B.8}$$

Combining like terms in formula B.8, one gets:

$$\sum_{i=1}^{N} \mathbf{v}_i^{\intercal} \cdot \boldsymbol{\gamma}_i \cdot \mathbf{v}_i + \sum_{((i,j)\in E)\wedge(i<j)} [(\mathbf{v}_i - \mathbf{v}_j)^{\intercal} \cdot \boldsymbol{\gamma}_{ij} \cdot (\mathbf{v}_i - \mathbf{v}_j)]. \tag{B.9}$$

Replacing $\boldsymbol{\gamma}_i$ and $\boldsymbol{\gamma}_{ij}$ with their definitions in formula B.9 results in:

$$\sum_{i=1}^{N} \mathbf{v}_i^{\intercal} \cdot 6\pi\mu R_i \mathbb{1}_{3\times3} \cdot \mathbf{v}_i + \sum_{((i,j)\in E)\wedge(i<j)} [(\mathbf{v}_i - \mathbf{v}_j)^{\intercal} \cdot \eta A_{ij}(\mathbb{1}_{3\times3} - \mathbf{P}_{\perp}^{ij}) \cdot (\mathbf{v}_i - \mathbf{v}_j)] \tag{B.10}$$

which can be further reformulated into:

$$\sum_{i=1}^{N} \mathbf{v}_i^{\intercal} \cdot 6\pi\mu R_i \mathbb{1}_{3\times3} \cdot \mathbf{v}_i + \sum_{((i,j)\in E)\wedge(i<j)} [(\mathbf{v}_i - \mathbf{v}_j)^{\intercal} \cdot \eta A_{ij} \cdot (\mathbb{1}_{3\times3} - \mathbf{P}_{\perp}^{ij})^{\intercal} \cdot (\mathbb{1}_{3\times3} - \mathbf{P}_{\perp}^{ji}) \cdot (\mathbf{v}_i - \mathbf{v}_j)], \tag{B.11}$$

since equation

$$(\mathbb{1}_{3\times3} - \mathbf{P}_{\perp}^{ij})^{\intercal} \cdot (\mathbb{1}_{3\times3} - \mathbf{P}_{\perp}^{ij}) = (\mathbb{1}_{3\times3} - \mathbf{P}_{\perp}^{ij})^2 = \mathbb{1}_{3\times3} - \mathbf{P}_{\perp}^{ij} \tag{B.12}$$

holds for the symmetric projection matrix $\mathbb{1}_{3\times3} - \mathbf{P}_{\perp}^{ij}$.

Formula B.11 now can be finally reformulated into:

$$6\pi\mu \sum_{i=1}^{N} R_i \|\mathbf{v}_i\|_2^2 + \eta \sum_{((i,j)\in E)\wedge(i<j)} A_{ij} \|(\mathbb{1}_{3\times3} - \mathbf{P}_{\perp}^{ij}) \cdot (\mathbf{v}_i - \mathbf{v}_j)\|_2^2. \tag{B.13}$$

The value of the term $6\pi\mu \sum_{i\in N} R_i \|\mathbf{v}_i\|_2^2$ ($\|\cdot\|_2$ stands for the l^2 norm of a vector) of formula B.13 is always positive for non-zero velocity $\mathbf{v}$, as π is a positive constant, μ is a positive parameter and R_i is positive. The value of the rest term of this formula is non-negative since η and A_{ij} are positive. Hence, the value of the entire formula B.13 is positive, which is that the inequality

$$\mathbf{v}^{\intercal}\mathbf{M}\mathbf{v} > 0, \ \forall \ \mathbf{v} \neq 0 \tag{B.14}$$

holds.

Appendix C

Brief Introduction to Finite Element Method

Most partial differential equation (PDE) models of real systems are hard or even impossible to solve analytically, while an approximated solution using numerical methods is more practical to get. Though a lot of computations can be involved for the numerical methods to solve PDEs, it can be easily done (implemented) on computers. The Finite Element Method (FEM) is one of those numerical methods commonly adopted.

In order to explain the basic idea of FEM, a very simple PDE in one dimension (1D) is used here. The PDE reads:

$$\begin{aligned} &-u''(x) = f(x),\ in\ I = (a,\ b),\ a,\ b \in R \\ &u(a) = u(b) = 0 \end{aligned} \tag{C.1}$$

where $u(x)$ denotes the variable of interest evolving over space. $f(x)$ represents a term that can influence the values of $u(x)$. $I \in R^1$ is the spatial interval that the equation works within.

The basic idea of FEM is to construct a piecewise function $u_h(x)$ to approximate $u(x)$, that is $u_h(x) \approx u(x)$, using a set of piecewise functions $\varphi_i(x)$ that are defined on the non-overlapping sub-intervals (elements) I_i ($i \in (1, 2..., N)$, N is the number of elements) which is a partition of the computational interval I. Normally, the set of piecewise functions are chosen such that they actually define a functional space V_h, and they are the basis of the functional space. The way to construct $u_h(x)$ is to linear combine of the basis functions, which is the famous ansatz equation:

$$u_h(x) = \sum_{i=0}^{i=N} \alpha_i \varphi_i(x) \tag{C.2}$$

where $\varphi_i(x)$ is the basis function i, and N is the dimension of V_h. Once the basis functions are chosen. The problem of finding $u(x)$ now changes to finding the coefficients α_i $(i \in (0, 1, ..., N))$. Since $u(x)$ are zero at the end points a, b of the effective computational interval I, the ansatz equation can be simplified into:

$$u_h(x) = \sum_{i=1}^{i=N-1} \alpha_i \varphi_i(x). \tag{C.3}$$

After this simplification, the number of unknowns is reduced from $N+1$ to $N-1$.

For $\varphi_i(x)$ $(i \in (0, 1, ..., N))$, being the basis of the functional space V_h is only one important characteristic. Some other characteristics are important for reducing the calculations necessary for solving α_i $(i \in (0, 1, ..., N))$. The other characteristic is described as:

$$\varphi_i(x) = \begin{cases} 1, \ x = x_i, \ i \in (0, N+1) \\ 0, \ otherwise \end{cases} \tag{C.4}$$

where x_i is a point within the computational domain. The reason why this characteristic can help reduce calculations will be explained next.

The so-called hat functions are typical piecewise basis functions commonly used in implementing FEM. In 1D case, one typical expression of the hat functions is given by

$$\varphi_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}, \ i \in I_i \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, \ i \in I_{i+1} \\ 0, \ otherwise. \end{cases} \tag{C.5}$$

The graph of hat functions look like hats as it is shown in figure C.1, where $\varphi_i(x)$ is nonzero only in the interval I_i $(x_i - x_{i-1})$ and I_{i+1} $(x_{i+1} - x_i)$.

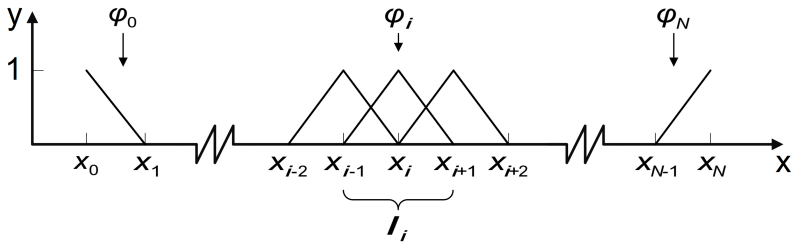


FIGURE C.1: Hat functions $\varphi(x)$ in 1D case. I_i is the sub-interval i.

To carry out FEM, the fist step is to reformulate the original PDE to the so-called weak form. In this example, it is multiplied by a so-called test function $v(x)$ and integrated

on its both sides. This yields:

$$\int_I -u''(x) \cdot v(x) = \int_I f(x) \cdot v(x). \tag{C.6}$$

When test functions are strongly forced to vanish at the end points $a,\ b$ of the effective computational interval I, the equation C.6 can be further reformulated into its weak form by applying the rule of integration by parts on the left hand side. This yields:

$$\int_I u'(x) \cdot v'(x) = \int_I f(x) \cdot v(x). \tag{C.7}$$

From figure C.1, one can find that all the basis functions except $\varphi_0(x)$ and $\varphi_N(x)$ actually vanish at $a,\ b$ of I. Hence, the test functions can be the basis functions beside the two exceptions, where the rest basis function defines a new functional space $V_{h,0}$. This method is also called Galerkin method.

For each test function $\varphi_j(x)$ $(j \in (1, 2, ..., N-1))$, replacing $u_h(x)$ by the ansatz C.3 and reordering the items, the weak form of the PDE now is changed into:

$$\sum_{i=1}^{i=N-1} \alpha_i \int_I \varphi_i'(x) \cdot \varphi_j'(x) = \int_I f(x) \cdot \varphi_j(x). \tag{C.8}$$

Denoting $\int_I \varphi_i'(x) \cdot \varphi_j'(x) = a_{ij}$ and $\int_I f(x) \cdot \varphi_j(x) = b_j$ separately. Obviously, a_{ij} is equivalent to a_{ji}. Hence, one can get the linear system of equation

$$\mathbf{M}^\intercal \boldsymbol{\alpha} = \mathbf{M}\boldsymbol{\alpha} = \mathbf{b} \tag{C.9}$$

where $\mathbf{M} = (a_{ij})_{(N-1)\times(N-1)}$, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_{(N-1)})^\intercal$ and $\mathbf{b} = (b_1, b_2, ..., b_{(N-1)})^\intercal$.

Now, one can see why the property expressed by C.4 can reduce the calculations. According to the definition given by C.5, one can get the results easily (supposing $j \geqslant i$):

$$\varphi_i(x) = \begin{cases} \int_I \varphi_i'(x) \cdot \varphi_j'(x) = \int_{I_{i-1}+I_i} \varphi_i'(x) \cdot \varphi_j'(x),\ j = i \\ \int_I \varphi_i'(x) \cdot \varphi_j'(x) = \int_{I_i} \varphi_i'(x) \cdot \varphi_j'(x),\ j = i+1 \\ 0,\ \textit{otherwise} \end{cases} \tag{C.10}$$

and

$$b_j = \int_I f(x) \cdot \varphi_j(x) = \int_{I_{i-1}+I_i} f(x) \cdot \varphi_j(x). \tag{C.11}$$

It is obvious that more partitions (larger N) will produce better approximation of $u(x)$, however, the computational costs will increase accordingly. In reality, one could try different number of partitions and compare the solutions in order to get an accepted

one. Besides, the matrix $\mathbf{M}$ is symmetric and very sparse, which can be seen easily from equation C.10. These two characteristics can contribute significantly to reduce the computational cost of solving large linear systems.

Using the simple 1D case example, we have explained the basic ideas of FEM. For PDEs, for example, RDEs (time-dependent PDEs), the ideas developed in the above section have to be extended. Next, the derivation of FEM for solving RDEs is introduced.

A general form of the RDE can be described as:

$$\begin{aligned} &\frac{\partial u}{\partial t} = \Delta u + f,\ in\ \Omega \in R^d \times (0,T)\ (d=3) \\ &u = g_D,\ on\ \partial\Omega_D \\ &\frac{\partial u}{\partial n} = g_N,\ on\ \partial\Omega_N \\ &and\ u = u_0\ in\ \Omega\ at\ t = 0 \end{aligned} \tag{C.12}$$

where Ω is the space, within which the equation works. $\partial\Omega_D$ and $\partial\Omega_D$ represent the Dirichlet and the Neumann (natural) boundary condition separately.

The weak form of this RDE reads:

$$\int_\Omega \frac{\partial u}{\partial t} v + \int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v + \int_{\partial\Omega} \nabla u \cdot v. \tag{C.13}$$

If the test functions v are also forced to vanish at the system boundaries $\partial\Omega$. This weak form can be finally simplified into,

$$\int_\Omega \frac{\partial u}{\partial t} v + \int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v. \tag{C.14}$$

For this PDE, since ansatz takes the form:

$$u_h = \sum_{j=1}^{j=N_I} \alpha_j(t)\varphi_j + \sum_{j=N_I}^{j=N_I+N_B} \alpha_j(t)\varphi_j \tag{C.15}$$

where N_I and N_B are the number of partitioning points for problem discretization within the and on the boundary of the system space respectively, the derivative on time t of u_h reads:

$$\frac{\partial u_h}{\partial t} = \sum_{j=1}^{j=N_I} \frac{\partial \alpha_j(t)}{\partial t}\varphi_j + \sum_{j=N_I}^{j=N_I+N_B} \frac{\partial \alpha_j(t)}{\partial t}\varphi_j. \tag{C.16}$$

Replacing u and $\frac{\partial u}{\partial t}$ in equation C.14 accordingly using equations C.15 and C.16 and reordering the items, one can get:

$$\begin{aligned}\sum_{j=1}^{j=N_I} \frac{\partial \alpha_j(t)}{\partial t} \int_\Omega \varphi_j \varphi_i + \sum_{j=1}^{j=N_I} \alpha_j(t) \int_\Omega \varphi_j' \varphi_i' = \int_\Omega f \varphi_i \\ - \sum_{j=N_I}^{j=N_I+N_B} \frac{\partial \alpha_j(t)}{\partial t} \int_\Omega \varphi_j \varphi_i - \sum_{j=N_I}^{j=N_I+N_B} \alpha_j(t) \int_\Omega \varphi_j' \varphi_i'.\end{aligned} \tag{C.17}$$

This equation can be reformulated into a linear system of semi-discrete ordinary differential equations (ODEs), which takes the form:

$$\mathbf{A} \frac{\partial \boldsymbol{\alpha}}{\partial t} + \mathbf{M} \boldsymbol{\alpha} = \mathbf{b} \tag{C.18}$$

where $\mathbf{A} = (a_{ij})_{N_I \times N_I}$ with $a_{ij} = \int_\Omega \varphi_j \varphi_i$, $\mathbf{M} = (a_{ij})_{N_I \times N_I}$ with $m_{ij} = \int_\Omega \varphi_j' \varphi_i'$ and $\mathbf{b} = (b_1, b_2, ..., b_{N_I})^\intercal$ with $b_i = \int_\Omega f \varphi_i - \sum_{j=N_I}^{j=N_I+N_B} \frac{\partial \alpha_j(t)}{\partial t} \int_\Omega \varphi_j \varphi_i - \sum_{j=N_I}^{j=N_I+N_B} \alpha_j(t) \int_\Omega \varphi_j' \varphi_i'$.

Since time t is continuous, it has to be discretized over time to solve the linear system C.18. The point is to eliminate the derivative of α_i over time based on the temporal discretization. There are lots methods for doing so, such as the Runge–Kutta method family. Details of them are out of this brief introduction to FEM.

Appendix D

3D Radical Voronoi Cells Versus Realistic Tumor Cells

The 3D radical Voronoi tessellation is physically quantified as the Voronoi information listed in table 4.2. Compared to the lattice-based models, this model enables much more flexibility and accuracy of cell morphology and cell-cell topology as it has been discussed in sections 2.2.3. However, it does mean that the 3D radical Voronoi tessellation is perfect for representing realistic tumor tissues.

Cell-cell topology can be precisely represented by the topology of Voronoi cells, since two physically neighboring tumor cells will definitely be neighbors in the Voronoi tessellation. However, some Voronoi information is only a approximation. First, the volume of a Voronoi cell is a rough approximation of the volume of a tumor cell. Second, the area of contact surface between cells is not accurate either. According the algorithm of the Voronoi tessellation, the entire computational domain is fully divided by Voronoi cells, in other words, each point in the domain belongs to at least one and only one Voronoi cell. However, the intra-cell space (extracellular space) exits in realistic tumor tissues. Hence, the Voronoi cell volume takes part of the extracellular space as the volume of tumor cells. As a result, the contact area between Voronoi cells is also larger than that of realistic tumor cells.

Figure D.1 shows the possibility of the inaccuracy of representing a tumor tissue using a 3D radical Voronoi tessellation. According to the algorithm of the radical Voronoi tessellation, the polyhedra of these three groups of spheroids are exactly the same in the three cases. In other words, the polyhedral volume and contact area between polyhdra correspondingly equal in these cases. However, the realistic volume of these spheroids is quite different, which can be comparatively seen from this figure. For the area of contact surfaces, the two spheroids do not contact each other at all in the case of small

radii (see the right most panel of this figure), while their corresponding Voronoi cells are obvious in contact with each other. The possible inaccuracy of the Voronoi tessellation indicates that one has to choose the proper Voronoi information in serious simulations.

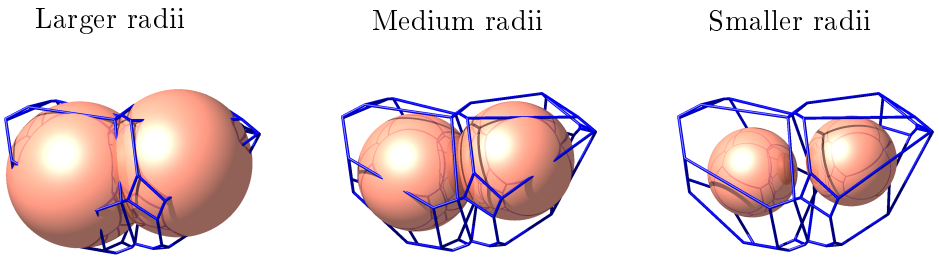


FIGURE D.1: Radical Voronoi tessellations of two spheroids of different radii. These spheroids in different groups have the same coordinates but different radii.

Appendix E

Data Representation in TUGME

As it has been introduced early in section 4.3.2 in chapter 4 of this thesis, agent objects of alive cells are stored in a C++ `vector` (`states`, data structure type: `vector<cell>`) in TUGME. When randomly accessing `states`, we notice that it is very inefficient to do so using the `UUID` or the `UID` of cells, since one has to search over it by checking the `UUID` or the `UID` when each cell is accessed until finding the targeted one. To solve this problem, the position, where the cell object is stored in `states`, is used as the temporary identifier (`TID`) of cells. It is temporary because the system is dynamically changing, during which the agent objects of dead cells are removed from it and the agent objects of newly born cells are added into it. However, within each simulation time step, the `TID` does change. One can access the agent object efficiently with the `TID` (`states[TID]`).

For each cell, its Voronoi information (see table 4.2) is accessed using the similar idea. The neighbors (`neighbors`, data structure type: `vector<TID>`), the area of the contact faces (`areas`, data structure type: `vector<double>`) with neighbors, the normalized normal vector of the contact faces (`nnvectors`, data structure type: `vector<coordiante3D-Type>`, `coordiante3DType` represents the coordinate in three dimensions) are stored in C++ `vectors` separately too. The same position within these `vectors` indicates the same neighbor. Taking the i^{th} element of them for example, `neighbors[i]`, `areas[i]`, `nnvectors[i]` return the `TID`, the area and the normalized normal vector of the contact face between current cell and its i^{th} neighbor. One can see immediately that `states[neighbors[i]]` returns the agent object of the i^{th} neighbor. Accessing the vertex of the Voronoi polyhedra is a little bit more complicated. A matrix like the data structure (`vertices`, data structure type: `vector<vector<VID»`, `VID` is the identifier of the coordinate of a vertex) is used to store this information via the surfaces of each cell. All vertices of all Voronoi cell polyhedra are stored in one C++ `vector`

(`coordinates`, data structure type: `vector<coordiante3DType>`). `coordinates[i]` returns the coordinate of the i^{th} vertex, where i is the vertex identifier (`VID`). For a cell, `vertices[i]` returns the `vector` that holds the `VID`s of vertices of the contact face with the i^{th} neighbor (please keep in mind that i is not the `TID` of the cell but the position in these vectors, `neighbors`, `areas`, `nnvectors` and `vertices`, and it is the `neighbors[i]` that actually returns the `TID` of the i^{th} neighbor of the current cell). `vertices[i][j]` returns the `VID` of the j^{th} vertex of the contact face with the i^{th} neighbor. Hence, `coordinates[vertices[i][j]]` actually returns the coordinate (data structure type: `coordiante3DType`) of the vertex.

The design using identifiers instead of the real data makes the access of them indirect, while it saves the memory cost significantly, especially for the data that are shared among several cells, hence, has to be repeatedly stored by all the holders, for example each vertex of the Voronoi polyhedra is shared by at least three Voronoi cells. In addition, it is obviously that this design improves the access efficiency to these data structures too.

Bibliography

[1] Dale K Pace. Modeling and simulation verification and validation challenges. *Johns Hopkins APL Technical Digest*, 25(2):163–172, 2004.

[2] Robert Weinberg. *The biology of cancer*. Garland Science, 2013.

[3] Thomas S Deisboeck, Zhihui Wang, Paul Macklin, and Vittorio Cristini. Multiscale cancer modeling. *Annual review of biomedical engineering*, 13, 2011.

[4] Douglas Hanahan and Robert A Weinberg. Hallmarks of cancer: the next generation. *Cell*, 144(5):646–674, 2011.

[5] Caterina Guiot, Piero Giorgio Degiorgis, Pier Paolo Delsanto, Pietro Gabriele, and Thomas S Deisboeck. Does tumor growth follow a "universal law"? *Journal of theoretical biology*, 225(2):147–151, 2003.

[6] Mickey Pentecost, Glen Otto, Julie A Theriot, and Manuel R Amieva. Listeria monocytogenes invades the epithelial junctions at sites of cell extrusion. *PLoS pathogens*, 2(1):e3, 2006.

[7] James P Freyer and Robert M Sutherland. Regulation of growth saturation and development of necrosis in emt6/ro multicellular spheroids by the glucose and oxygen supply. *Cancer research*, 46(7):3504–3512, 1986.

[8] Vincent T DeVita Jr and Steven A Rosenberg. Two hundred years of cancer research. *New England Journal of Medicine*, 366(23):2207–2214, 2012.

[9] Peyton Rous. A transmissible avian neoplasm.(sarcoma of the common fowl.). *The journal of experimental medicine*, 12(5):696–705, 1910.

[10] Subra Suresh. Biomechanics and biophysics of cancer cells. *Acta Materialia*, 55 (12):3989–4014, 2007.

[11] Jorrit J Hornberg, Frank J Bruggeman, Hans V Westerhoff, and Jan Lankelma. Cancer: a systems biology disease. *Biosystems*, 83(2):81–90, 2006.

[12] Hiroaki Kitano. Computational systems biology. *Nature*, 420(6912):206–210, 2002.

[13] Leroy Hood, James R Heath, Michael E Phelps, and Biaoyang Lin. Systems biology and new technologies enable predictive and preventative medicine. *Science*, 306(5696):640–643, 2004.

[14] IG Khalil and C Hill. Systems biology for cancer. *Current opinion in oncology*, 17(1):44–48, 2005.

[15] Lilia Alberghina and Hans V Westerhoff. *Systems biology: definitions and perspectives*, volume 13. Springer, 2007.

[16] Randall H Kramer and Garth L Nicolson. Interactions of tumor cells with vascular endothelial cell monolayers: a model for metastatic invasion. *Proceedings of the National Academy of Sciences*, 76(11):5704–5708, 1979.

[17] GARTH L Nicolson. Metastatic tumor cell attachment and invasion assay utilizing vascular endothelial cell monolayers. *Journal of Histochemistry & Cytochemistry*, 30(3):214–220, 1982.

[18] Maximilian Diehn and Michael F Clarke. Cancer stem cells and radiotherapy: new insights into tumor radioresistance. *Journal of the National Cancer Institute*, 98(24):1755–1757, 2006.

[19] Nadja Dornhöfer, Suzanne Spong, Kevin Bennewith, Ali Salim, Stephen Klaus, Neeraja Kambham, Carol Wong, Fiona Kaper, Patrick Sutphin, Rendall Nacalumi, et al. Connective tissue growth factor–specific monoclonal antibody therapy inhibits pancreatic tumor growth and metastasis. *Cancer research*, 66(11):5816–5827, 2006.

[20] Roberto Chignola, ALESSIO DEL FABBRO, Marcello Farina, and Edoardo Milotti. Computational challenges of tumor spheroid modeling. *Journal of bioinformatics and computational biology*, 9(04):559–577, 2011.

[21] Robert M Sutherland, John A McCredie, and W Rodger Inch. Growth of multicell spheroids in tissue culture as a model of nodular carcinomas. *Journal of the National Cancer Institute*, 46(1):113–120, 1971.

[22] Takanori Sonoda, Hiroaki Kobayashi, Tsunehisa Kaku, Toshio Hirakawa, and Hitoo Nakano. Expression of angiogenesis factors in monolayer culture, multicellular spheroid and in vivo transplanted tumor by human ovarian cancer cell lines. *Cancer letters*, 196(2):229–237, 2003.

[23] BENGT GLIMELIUS, BÖRJE NORLING, THORE NEDERMAN, and JÖRGEN CARLSSON. Extracellular matrices in multicellular spheroids of human glioma origin: Increased incorporation of proteoglycans and fibronectin as compared to monolayer cultures. *Apmis*, 96(1-6):433–444, 1988.

[24] Rolf Bjerkvig, Ole Didrik Laerum, and Garry J Rucklidge. Immunocytochemical characterization of extracellular matrix proteins expressed by cultured glioma cells. *Cancer research*, 49(19):5424–5428, 1989.

[25] Harry Olson, Graham Betton, Denise Robinson, Karluss Thomas, Alastair Monro, Gerald Kolaja, Patrick Lilly, James Sanders, Glenn Sipes, William Bracken, et al. Concordance of the toxicity of pharmaceuticals in humans and in animals. *Regulatory Toxicology and Pharmacology*, 32(1):56–67, 2000.

[26] T Luke Simmons, Eric Andrianasolo, Kerry McPhail, Patricia Flatt, and William H Gerwick. Marine natural products as anticancer drugs. *Molecular Cancer Therapeutics*, 4(2):333–342, 2005.

[27] Dror Meirow, Hannah Biederman, Richard A Anderson, and W Hamish B Wallace. Toxicity of chemotherapy and radiation on female reproduction. *Clinical obstetrics and gynecology*, 53(4):727–739, 2010.

[28] Carl-Henrik Heldin, Kristofer Rubin, Kristian Pietras, and Arne Östman. High interstitial fluid pressure—an obstacle in cancer therapy. *Nature Reviews Cancer*, 4(10):806–813, 2004.

[29] Vladimir P Torchilin and Anatoly N Lukyanov. Peptide and protein drug delivery to and into tumors: challenges and solutions. *Drug discovery today*, 8(6):259–266, 2003.

[30] Mauro Ferrari. Cancer nanotechnology: opportunities and challenges. *Nature Reviews Cancer*, 5(3):161–171, 2005.

[31] Trey Ideker, Timothy Galitski, and Leroy Hood. A new approach to decoding life: systems biology. *Annual review of genomics and human genetics*, 2(1):343–372, 2001.

[32] Hiroaki Kitano. Systems biology: a brief overview. *Science*, 295(5560):1662–1664, 2002.

[33] Bernard P Zeigler, Herbert Praehofer, and Tag Gon Kim. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic press, 2000.

[34] J Michael Harrison. *Brownian motion and stochastic flow systems*. Wiley New York, 1985.

[35] Kurt Binder and Dieter Heermann. *Monte Carlo simulation in statistical physics: an introduction*. Springer, 2010.

[36] Ole-Johan Dahl and Kristen Nygaard. Simula: an algol-based simulation language. *Communications of the ACM*, 9(9):671–678, 1966.

[37] Thomas J Schriber. Simulation using gpss. Technical report, DTIC Document, 1974.

[38] Richard E Nance. *A history of discrete event simulation programming languages*. ACM, 1996.

[39] Volker Springel, Simon DM White, Adrian Jenkins, Carlos S Frenk, Naoki Yoshida, Liang Gao, Julio Navarro, Robert Thacker, Darren Croton, John Helly, et al. Simulations of the formation, evolution and clustering of galaxies and quasars. *nature*, 435(7042):629–636, 2005.

[40] Seunghwa Kang, Simon Kahan, Jason McDermott, Nicholas Flann, and Ilya Shmulevich. Biocellion: accelerating computer simulation of multicellular biological system models. *Bioinformatics*, 30(21):3101–3108, 2014.

[41] Bernard P Zeigler and Hessam S Sarjoughian. Introduction to devs modeling and simulation with java: Developing component-based simulation models. *Technical Document, University of Arizona*, 2003.

[42] Volker Grimm, Uta Berger, Finn Bastiansen, Sigrunn Eliassen, Vincent Ginot, Jarl Giske, John Goss-Custard, Tamara Grand, Simone K Heinz, Geir Huse, et al. A standard protocol for describing individual-based and agent-based models. *Ecological modelling*, 198(1):115–126, 2006.

[43] Nigel Gilbert. *Agent-based models*. Number 153. Sage, 2008.

[44] Le Zhang, Zhihui Wang, Jonathan A Sagotsky, and Thomas S Deisboeck. Multiscale agent-based cancer modeling. *Journal of mathematical biology*, 58(4-5): 545–559, 2009.

[45] Zhihui Wang and Thomas S Deisboeck. Computational modeling of brain tumors: discrete, continuum or hybrid? *Scientific Modeling and Simulation SMNS*, 15(1-3):381–393, 2008.

[46] Katarzyna A Rejniak and Alexander RA Anderson. Hybrid models of tumor growth. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 3(1): 115–125, 2011.

[47] JM Osborne, A Walter, SK Kershaw, GR Mirams, AG Fletcher, P Pathmanathan, D Gavaghan, OE Jensen, PK Maini, and HM Byrne. A hybrid approach to multi-scale modelling of cancer. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1930): 5013–5028, 2010.

[48] Vinay G Vaidya and Frank J Alexandro Jr. Evaluation of some mathematical models for tumor growth. *International journal of bio-medical computing*, 13(1): 19–35, 1982.

[49] Cosmina Hogea, Christos Davatzikos, and George Biros. An image-driven parameter estimation problem for a reaction–diffusion glioma growth model with mass effects. *Journal of mathematical biology*, 56(6):793–825, 2008.

[50] SC Ferreira Jr, ML Martins, and MJ Vilela. Reaction-diffusion model for the growth of avascular tumor. *Physical Review E*, 65(2):021907, 2002.

[51] Yi Jiang, Jelena Pjesivac-Grbovic, Charles Cantrell, and James P Freyer. A multiscale model for avascular tumor growth. *Biophysical journal*, 89(6):3884–3894, 2005.

[52] Gernot Schaller and Michael Meyer-Hermann. Continuum versus discrete model: a comparison for multicellular tumour spheroids. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 364(1843): 1443–1464, 2006.

[53] Tiina Roose, S Jonathan Chapman, and Philip K Maini. Mathematical models of avascular tumor growth. *Siam Review*, 49(2):179–208, 2007.

[54] Douglas Hanahan and Robert A Weinberg. The hallmarks of cancer. *cell*, 100 (1):57–70, 2000.

[55] Alexander RA Anderson. A hybrid mathematical model of solid tumour invasion: the importance of cell adhesion. *Mathematical Medicine and Biology*, 22(2):163–186, 2005.

[56] AR Kansal, S Torquato, GR Harsh Iv, EA Chiocca, and TS Deisboeck. Cellular automaton of idealized brain tumor growth dynamics. *Biosystems*, 55(1):119–127, 2000.

[57] Joana Moreira and Andreas Deutsch. Cellular automaton models of tumor development: a critical review. *Advances in Complex Systems*, 5(02n03):247–267, 2002.

[58] Daniel G Mallet and Lisette G De Pillis. A cellular automata model of tumor–immune system interactions. *Journal of Theoretical Biology*, 239(3):334–350, 2006.

[59] Sanjay Kumar and Valerie M Weaver. Mechanics, malignancy, and metastasis: the force journey of a tumor cell. *Cancer and Metastasis Reviews*, 28(1-2):113–127, 2009.

[60] Dirk Drasdo, Stefan Hoehme, and Michael Block. On the role of physics in the growth and pattern formation of multi-cellular systems: What can we learn from individual-cell based models? *Journal of Statistical Physics*, 128(1-2):287–345, 2007.

[61] Dirk Drasdo and Stefan Hoehme. Modeling the impact of granular embedding media, and pulling versus pushing cells on growing cell clones. *New Journal of Physics*, 14(5):055025, 2012.

[62] Adelinde M Uhrmacher and Danny Weyns. *Multi-Agent systems: Simulation and applications*. CRC press, 2010.

[63] Iyad Abu Doush. Multi-agent systems modeling, control, programming, simulations and applications. 2011.

[64] Muaz Niazi and Amir Hussain. Agent-based computing from multi-agent systems to agent-based models: a visual survey. *Scientometrics*, 89(2):479–499, 2011.

[65] Charles M Macal and Michael J North. Tutorial on agent-based modeling and simulation. In *Proceedings of the 37th conference on Winter simulation*, pages 2–15. Winter Simulation Conference, 2005.

[66] Stefan Hoehme and Dirk Drasdo. A cell-based simulation software for multi-cellular systems. *Bioinformatics*, 26(20):2641–2642, 2010.

[67] LS Chin, DJ Worth, C Greenough, S Coakley, M Holcombe, and M Gheorghe. *FLAME-II: A Redesign of the Flexible Large-scale Agent-based Modelling Environment*. STFC, 2012.

[68] Barry L Nelson, John S Carson, and Jerry Banks. *Discrete event system simulation*. Prentice hall, 2001.

[69] Nelson Minar, Roger Burkhart, Chris Langton, and Manor Askenazi. The swarm simulation system: A toolkit for building multi-agent simulations. Santa Fe Institute Santa Fe, 1996.

[70] Bert Vogelstein and Kenneth W Kinzler. Cancer genes and the pathways they control. *Nature medicine*, 10(8):789–799, 2004.

[71] Christoph Lengauer, Kenneth W Kinzler, and Bert Vogelstein. Genetic instabilities in human cancers. *Nature*, 396(6712):643–649, 1998.

[72] Pierre Golstein and Guido Kroemer. Cell death by necrosis: towards a molecular definition. *Trends in biochemical sciences*, 32(1):37–43, 2007.

[73] Darja Kanduc, Abraham Mittelman, ROSARIO Serpico, EBERTA Sinigaglia, Animesh A Sinha, Costanzo Natale, Raffaella Santacroce, M GRAZIA Di Corcia, ALBERTA Lucchese, LUCIANA Dini, et al. Cell death: apoptosis versus necrosis (review). *International journal of oncology*, 21(1):165–170, 2002.

[74] Yutaka Eguchi, Shigeomi Shimizu, and Yoshihide Tsujimoto. Intracellular atp levels determine cell death fate by apoptosis or necrosis. *Cancer research*, 57(10): 1835–1840, 1997.

[75] Robert M Sutherland. Cell and environment interactions in tumor microregions: the multicell spheroid model. *Science*, 240(4849):177–184, 1988.

[76] M Tubiana. The growth and progression of human tumors: implications for management strategy. *Radiotherapy and Oncology*, 6(3):167–184, 1986.

[77] Marian Valko, Mario Izakovic, Milan Mazur, Christopher J Rhodes, and Joshua Telser. Role of oxygen radicals in dna damage and cancer incidence. *Molecular and cellular biochemistry*, 266(1-2):37–56, 2004.

[78] Dominique Stehelin, Harold E Varmus, J Michael Bishop, and Peter K Vogt. Dna related to the transforming gene (s) of avian sarcoma viruses is present in normal avian dna. 1976.

[79] Franziska Michor, Yoh Iwasa, and Martin A Nowak. Dynamics of cancer progression. *Nature Reviews Cancer*, 4(3):197–205, 2004.

[80] Michael W Nachman and Susan L Crowell. Estimate of the mutation rate per nucleotide in humans. *Genetics*, 156(1):297–304, 2000.

[81] Laura D Wood, D Williams Parsons, Siân Jones, Jimmy Lin, Tobias Sjöblom, Rebecca J Leary, Dong Shen, Simina M Boca, Thomas Barber, Janine Ptak, et al. The genomic landscapes of human breast and colorectal cancers. *Science*, 318(5853):1108–1113, 2007.

[82] Karin E de Visser, Alexandra Eichten, and Lisa M Coussens. Paradoxical roles of the immune system during cancer development. *Nature reviews cancer*, 6(1): 24–37, 2006.

[83] Muhammad Al-Hajj and Michael F Clarke. Self-renewal and solid tumor stem cells. *Oncogene*, 23(43):7274–7282, 2004.

[84] Tannishtha Reya, Sean J Morrison, Michael F Clarke, and Irving L Weissman. Stem cells, cancer, and cancer stem cells. *nature*, 414(6859):105–111, 2001.

[85] Mark Shackleton, Elsa Quintana, Eric R Fearon, and Sean J Morrison. Heterogeneity in cancer: cancer stem cells versus clonal evolution. *Cell*, 138(5):822–829, 2009.

[86] Zhu A Wang, Antonina Mitrofanova, Sarah K Bergren, Cory Abate-Shen, Robert D Cardiff, Andrea Califano, and Michael M Shen. Lineage analysis of basal epithelial cells reveals their unexpected plasticity and supports a cell-of-origin model for prostate cancer heterogeneity. *Nature cell biology*, 15(3):274–283, 2013.

[87] Leonard Nunney. Lineage selection and the evolution of multistage carcinogenesis. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 266 (1418):493–498, 1999.

[88] Andrew P Feinberg, Rolf Ohlsson, and Steven Henikoff. The epigenetic progenitor origin of human cancer. *Nature reviews genetics*, 7(1):21–33, 2006.

[89] Gloria H Heppner. Tumor heterogeneity. *Cancer research*, 44(6):2259–2265, 1984.

[90] Michael Dean, Tito Fojo, and Susan Bates. Tumour stem cells and drug resistance. *Nature Reviews Cancer*, 5(4):275–284, 2005.

[91] Rolf Bjerkvig, Berit B Tysnes, Karen S Aboody, Joseph Najbauer, and AJA Terzis. The origin of the cancer stem cell: current controversies and new insights. *Nature Reviews Cancer*, 5(11):899–904, 2005.

[92] Michael F Clarke and Margaret Fuller. Stem cells and cancer: two faces of eve. *Cell*, 124(6):1111–1115, 2006.

[93] Hans Clevers. The cancer stem cell: premises, promises and challenges. *Nature medicine*, pages 313–319, 2011.

[94] Priscilla N Kelly, Aleksandar Dakic, Jerry M Adams, Stephen L Nutt, and Andreas Strasser. Tumor growth need not be driven by rare cancer stem cells. *Science*, 317(5836):337–337, 2007.

[95] Louis Vermeulen, Felipe de Sousa E Melo, Dick J Richel, and Jan Paul Medema. The developing cancer stem-cell model: clinical challenges and opportunities. *The lancet oncology*, 13(2):e83–e89, 2012.

[96] Peter Valent, Dominique Bonnet, Ruggero De Maria, Tsvee Lapidot, Mhairi Copland, Junia V Melo, Christine Chomienne, Fumihiko Ishikawa, Jan Jacob Schuringa, Giorgio Stassi, et al. Cancer stem cell definitions and terminology: the devil is in the details. *Nature Reviews Cancer*, 12(11):767–775, 2012.

[97] Elsa Quintana, Mark Shackleton, Michael S Sabel, Douglas R Fullen, Timothy M Johnson, and Sean J Morrison. Efficient tumour formation by single human melanoma cells. *Nature*, 456(7222):593–598, 2008.

[98] Anna Golebiewska, Nicolaas HC Brons, Rolf Bjerkvig, and Simone P Niclou. Critical appraisal of the side population assay in stem cell and cancer stem cell research. *Cell Stem Cell*, 8(2):136–147, 2011.

[99] Piyush B Gupta, Christine L Chaffer, and Robert A Weinberg. Cancer stem cells: mirage or reality? *Nature medicine*, 15(9):1010–1012, 2009.

[100] Jun Miki, Bungo Furusato, Hongzhen Li, Yongpeng Gu, Hiroyuki Takahashi, Shin Egawa, Isabell A Sesterhenn, David G McLeod, Shiv Srivastava, and Johng S Rhim. Identification of putative stem cell markers, cd133 and cxcr4, in htert–immortalized primary nonmalignant and malignant tumor-derived human prostate epithelial cell lines and in prostate cancer specimens. *Cancer research*, 67(7):3153–3161, 2007.

[101] Felix Zeppernick, Rezvan Ahmadi, Benito Campos, Christine Dictus, Burkhard M Helmke, Natalia Becker, Peter Lichter, Andreas Unterberg, Bernhard Radlwimmer, and Christel C Herold-Mende. Stem cell marker cd133 affects clinical outcome in glioma patients. *Clinical Cancer Research*, 14(1):123–129, 2008.

[102] Sheila K Singh, Ian D Clarke, Mizuhiko Terasaki, Victoria E Bonn, Cynthia Hawkins, Jeremy Squire, and Peter B Dirks. Identification of a cancer stem cell in human brain tumors. *Cancer research*, 63(18):5821–5828, 2003.

[103] Muhammad Al-Hajj, Max S Wicha, Adalberto Benito-Hernandez, Sean J Morrison, and Michael F Clarke. Prospective identification of tumorigenic breast cancer cells. *Proceedings of the National Academy of Sciences*, 100(7):3983–3988, 2003.

[104] ME Prince, R Sivanandan, A Kaczorowski, GT Wolf, MJ Kaplan, P Dalerba, IL Weissman, MF Clarke, and LE Ailles. Identification of a subpopulation of cells with cancer stem cell properties in head and neck squamous cell carcinoma. *Proceedings of the National Academy of Sciences*, 104(3):973–978, 2007.

[105] A Eramo, F Lotti, G Sette, E Pilozzi, M Biffoni, Aet Di Virgilio, C Conticello, L Ruco, C Peschle, and R De Maria. Identification and expansion of the tumorigenic lung cancer stem cell population. *Cell Death & Differentiation*, 15(3): 504–514, 2007.

[106] Anna Merlos-Suárez, Francisco M Barriga, Peter Jung, Mar Iglesias, María Virtudes Céspedes, David Rossell, Marta Sevillano, Xavier Hernando-Momblona, Victoria da Silva-Diz, Purificación Muñoz, et al. The intestinal stem cell signature identifies colorectal cancer stem cells and predicts disease relapse. *Cell stem cell*, 8(5):511–524, 2011.

[107] Lucia Ricci-Vitiani, Dario G Lombardi, Emanuela Pilozzi, Mauro Biffoni, Matilde Todaro, Cesare Peschle, and Ruggero De Maria. Identification and expansion of human colon-cancer-initiating cells. *Nature*, 445(7123):111–115, 2006.

[108] Chenwei Li, David G Heidt, Piero Dalerba, Charles F Burant, Lanjing Zhang, Volkan Adsay, Max Wicha, Michael F Clarke, and Diane M Simeone. Identification of pancreatic cancer stem cells. *Cancer research*, 67(3):1030–1037, 2007.

[109] Tetsuhiro Chiba, Kaoru Kita, Yun-Wen Zheng, Osamu Yokosuka, Hiromitsu Saisho, Atsushi Iwama, Hiromitsu Nakauchi, and Hideki Taniguchi. Side population purified from hepatocellular carcinoma cells harbors cancer stem cell–like properties. *Hepatology*, 44(1):240–251, 2006.

[110] Dong Fang, Thiennga K Nguyen, Kim Leishear, Rena Finko, Angela N Kulp, Susan Hotz, Patricia A Van Belle, Xiaowei Xu, David E Elder, and Meenhard Herlyn. A tumorigenic subpopulation with stem cell properties in melanomas. *Cancer research*, 65(20):9328–9337, 2005.

[111] Jing Wang, Li-Ping Guo, Li-Zhen Chen, Yi-Xin Zeng, and Shih Hsin Lu. Identification of cancer stem cell–like side population cells in human nasopharyngeal carcinoma cell line. *Cancer research*, 67(8):3716–3724, 2007.

[112] Masaru Katoh. Wnt signaling in stem cell biology and regenerative medicine. *Current drug targets*, 9(7):565–570, 2008.

[113] Zhaoyu Jin and Wafik S El-Deiry. Review overview of cell death signaling pathways. *Cancer biology & therapy*, 4(2):139–163, 2005.

[114] Elena Sancho, Eduard Batlle, and Hans Clevers. Signaling pathways in intestinal development and cancer. *Annu. Rev. Cell Dev. Biol.*, 20:695–723, 2004.

[115] Jonna Frasor, Jeanne M Danes, Barry Komm, Ken CN Chang, C Richard Lyttle, and Benita S Katzenellenbogen. Profiling of estrogen up-and down-regulated gene

expression in human breast cancer cells: insights into gene networks and pathways underlying estrogenic control of proliferation and cell phenotype. *Endocrinology*, 144(10):4562–4574, 2003.

[116] Jun Yao, Shunbin Xiong, Kristine Klos, Nina Nguyen, Rebecca Grijalva, Ping Li, and Dihua Yu. Multiple signaling pathways involved in activation of matrix metalloproteinase-9 (mmp-9) by heregulin-beta1 in human breast cancer cells. *Oncogene*, 20(56):8066–8074, 2001.

[117] Julian Downward. Targeting ras signalling pathways in cancer therapy. *Nature Reviews Cancer*, 3(1):11–22, 2003.

[118] James E Trosko and Randall J Ruch. Cell-cell communication in carcinogenesis. *Front Biosci*, 3(3):d208–236, 1998.

[119] Eric C Lai. Notch signaling: control of cell communication and cell fate. *Development*, 131(5):965–973, 2004.

[120] Leland H Hartwell and Michael B Kastan. Cell cycle control and cancer. *Science*, 266(5192):1821–1828, 1994.

[121] Joan Massagué. G1 cell-cycle control and cancer. *Nature*, 432(7015):298–306, 2004.

[122] Michael B Kastan and Jiri Bartek. Cell-cycle checkpoints and cancer. *Nature*, 432(7015):316–323, 2004.

[123] Calvin B Harley, A Bruce Futcher, and Carol W Greider. Telomeres shorten during ageing of human fibroblasts. 1990.

[124] Nam W Kim, Mieczyslaw A Piatyszek, Karen R Prowse, Calvin B Harley, Michael D West, P d L Ho, Gina M Coviello, Woodring E Wright, Scott L Weinrich, and Jerry W Shay. Specific association of human telomerase activity with immortal cells and cancer. *Science*, 266(5193):2011–2015, 1994.

[125] Eiso Hiyama, Lauren Gollahon, Tsuyoshi Kataoka, Katsumasa Kuroi, Takashi Yokoyama, Adi F Gazdar, Keiko Hiyama, Mieczyslaw A Piatyszek, and Jerry W Shay. Telomerase activity in human breast tumors. *Journal of the National Cancer Institute*, 88(2):116–122, 1996.

[126] Eiso Hiyama, Takashi Yokoyama, Naokuni Tatsumoto, Keiko Hiyama, Yuji Imamura, Yoshiaki Murakami, Takashi Kodama, Mieczyslaw A Piatyszek, Jerry W Shay, and Yuichiro Matsuura. Telomerase activity in gastric cancer. *Cancer research*, 55(15):3258–3262, 1995.

[127] Christopher M Counter, Ariel A Avilion, Catherine E LeFeuvre, Nancy G Stewart, Carol W Greider, Calvin B Harley, and Sylvia Bacchetti. Telomere shortening associated with chromosome instability is arrested in immortal cells which express telomerase activity. *The EMBO journal*, 11(5):1921, 1992.

[128] AM Petit, J Rak, MC Hung, P Rockwell, N Goldstein, B Fendly, and RS Kerbel. Neutralizing antibodies against epidermal growth factor and erbb-2/neu receptor tyrosine kinases down-regulate vascular endothelial growth factor production by tumor cells in vitro and in vivo: angiogenic implications for signal transduction therapy of solid tumors. *The American journal of pathology*, 151(6):1523, 1997.

[129] Otto Warburg et al. On the origin of cancer cells. *Science*, 123(3191):309–314, 1956.

[130] Jung-whan Kim and Chi V Dang. Cancer's molecular sweet tooth and the warburg effect. *Cancer research*, 66(18):8927–8930, 2006.

[131] Robert A Gatenby and Robert J Gillies. Why do cancers have high aerobic glycolysis? *Nature Reviews Cancer*, 4(11):891–899, 2004.

[132] Sui Huang and Donald E Ingber. Cell tension, matrix mechanics, and cancer development. *Cancer cell*, 8(3):175–176, 2005.

[133] AL Willis, F Sabeh, X-Y LI, and SJ Weiss. Extracellular matrix determinants and the regulation of cancer cell invasion stratagems. *Journal of microscopy*, 251 (3):250–260, 2013.

[134] Avri Ben-Ze'ev. The cytoskeleton in cancer cells. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, 780(3):197–212, 1985.

[135] Yeh-Shiu Chu, Sylvie Dufour, Jean Paul Thiery, Eric Perez, and Frederic Pincet. Johnson-kendall-roberts theory applied to living cells. *Physical review letters*, 94 (2):028102, 2005.

[136] José R García and Andrés J García. Cellular mechanotransduction: Sensing rigidity. *Nature materials*, 13(6):539–540, 2014.

[137] Nahid S Waleh, John Gallo, Tandora D Grant, Brian J Murphy, Randall H Kramer, and Robert M Sutherland. Selective down-regulation of integrin receptors in spheroids of squamous cell carcinoma. *Cancer research*, 54(3):838–843, 1994.

[138] Jochen Guck, Stefan Schinkinger, Bryan Lincoln, Falk Wottawah, Susanne Ebert, Maren Romeyke, Dominik Lenz, Harold M Erickson, Revathi Ananthakrishnan,

Daniel Mitchell, et al. Optical deformability as an inherent cell marker for testing malignant transformation and metastatic competence. *Biophysical journal*, 88(5): 3689–3698, 2005.

[139] Gabriel Helmlinger, Paolo A Netti, Hera C Lichtenbeld, Robert J Melder, and Rakesh K Jain. Solid stress inhibits the growth of multicellular tumor spheroids. *Nature biotechnology*, 15(8):778–783, 1997.

[140] PK Srivastava and RG Maki. Stress-induced proteins in immune response to cancer. *Current topics in microbiology and immunology*, 167:109–123, 1990.

[141] Ming Wu, José Carlos Pastor-Pareja, and Tian Xu. Interaction between rasv12 and scribbled clones induces tumour growth and invasion. *Nature*, 463(7280): 545–548, 2010.

[142] Thomas Lecuit and Pierre-François Lenne. Cell surface mechanics and the control of cell shape, tissue patterns and morphogenesis. *Nature Reviews Molecular Cell Biology*, 8(8):633–644, 2007.

[143] Peter Bastian, Markus Blatt, Christian Engwer, Andreas Dedner, Robert Klöfkorn, S Kuttanikkad, Mario Ohlberger, and Oliver Sander. The distributed and unified numerics environment (dune). In *Proc. of the 19th Symposium on Simulation Technique in Hannover*, 2006.

[144] Amy Henderson Squillacote and James Ahrens. *The paraview guide*, volume 366. Kitware, 2007.

[145] Andreas Dedner, Robert Klöfkorn, Martin Nolte, and Mario Ohlberger. A generic interface for parallel and adaptive discretization schemes: abstraction principles and the dune-fem module. *Computing*, 90(3-4):165–196, 2010.

[146] Dirk Drasdo and Stefan Höhme. A single-cell-based model of tumor growth in vitro: monolayers and spheroids. *Physical biology*, 2(3):133, 2005.

[147] Junhwan Jeon, Vito Quaranta, and Peter T Cummings. An off-lattice hybrid discrete-continuum model of tumor growth and invasion. *Biophysical journal*, 98 (1):37–47, 2010.

[148] Gernot Schaller and Michael Meyer-Hermann. Multicellular tumor spheroid in an off-lattice voronoi-delaunay cell model. *Physical Review E*, 71(5):051910, 2005.

[149] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. *Spatial tessellations: concepts and applications of Voronoi diagrams*, volume 501. John Wiley & Sons, 2009.

[150] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: applications and algorithms. *SIAM review*, 41(4):637–676, 1999.

[151] Deok-Soo Kim, Youngsong Cho, Donguk Kim, Sangsoo Kim, Jonghwa Bhak, and Sung-Hoon Lee. Euclidean voronoi diagrams of 3d spheres and applications to protein structure analysis. *Japan Journal of Industrial and Applied Mathematics*, 22(2):251–265, 2005.

[152] Ingeborg MM Van Leeuwen, GR Mirams, A Walter, A Fletcher, P Murray, J Osborne, S Varma, SJ Young, J Cooper, B Doyle, et al. An integrative computational model for intestinal tissue renewal. *Cell proliferation*, 42(5):617–636, 2009.

[153] FA Meineke, Christopher S Potten, and Markus Loeffler. Cell migration and organization in the intestinal crypt using a lattice-free model. *Cell proliferation*, 34(4):253–266, 2001.

[154] Tilo Beyer and Michael Meyer-Hermann. Multiscale modeling of cell mechanics and tissue organization. *Engineering in Medicine and Biology Magazine, IEEE*, 28(2):38–45, 2009.

[155] Steven Fortune. Voronoi diagrams and delaunay triangulations. *Computing in Euclidean geometry*, 1:193–233, 1992.

[156] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[157] Jonathan Richard Shewchuk. Triangle: A two-dimensional quality mesh generator and delaunay triangulator. *Computer Science Division, University of California at Berkeley, Berkeley, California*, pages 94720–1776, 2005.

[158] Chris Rycroft. Voro++: A three-dimensional voronoi cell library in c++. 2009.

[159] Wolfgang H Goldmann, Vera Auernheimer, Ingo Thievessen, and Ben Fabry. Vinculin, cell mechanics and tumour cell invasion. *Cell biology international*, 37 (5):397–405, 2013.

[160] Martin Benoit, Daniela Gabriel, Günther Gerisch, and Hermann E Gaub. Discrete interactions in cell adhesion measured by single-molecule force spectroscopy. *Nature cell biology*, 2(6):313–317, 2000.

[161] Thomas R Weikl, Mesfin Asfaw, Heinrich Krobath, Bartosz Różycki, and Reinhard Lipowsky. Adhesion of membranes via receptor–ligand complexes: Domain

formation, binding cooperativity, and active processes. *Soft Matter*, 5(17):3213–3224, 2009.

[162] José CM Mombach and James A Glazier. Single cell motion in aggregates of embryonic cells. *Physical review letters*, 76(16):3032, 1996.

[163] David Selmeczi, Stephan Mosler, Peter H Hagedorn, Niels B Larsen, and Henrik Flyvbjerg. Cell motility as persistent random motion: theories from experiments. *Biophysical journal*, 89(2):912–931, 2005.

[164] Joshua Zimmerberg and Michael M Kozlov. How proteins produce cellular membrane curvature. *Nature Reviews Molecular Cell Biology*, 7(1):9–19, 2006.

[165] Narla Mohandas, Joel Anne Chasis, and Stephen B Shohet. The influence of membrane skeleton on red cell deformability, membrane material properties, and shape. In *Seminars in hematology*, volume 20, pages 225–242, 1983.

[166] GG Adams and M Nosonovsky. Contact modeling—forces. *Tribology International*, 33(5):431–442, 2000.

[167] Jörg Galle, Markus Loeffler, and Dirk Drasdo. Modeling the effect of deregulated proliferation and apoptosis on the growth dynamics of epithelial cell populations in vitro. *Biophysical journal*, 88(1):62–75, 2005.

[168] H. Hertz. über die berührung fester elastischer körper. *J. Reine Angewandte Mathematik*, 21:156–171, 1882.

[169] KL Johnson, K Kendall, and AD Roberts. Surface energy and the contact of elastic solids. *Proceedings of the royal society of London. A. mathematical and physical sciences*, 324(1558):301–313, 1971.

[170] Tilo Beyer and Michael Meyer-Hermann. Delauny object dynamics for tissues involving highly motile cells. *Cell mechanics: From single scale-based models to multiscale modeling, A. Chauviere, L. Preziosi, and C. Verdier, Eds. CRC Press*, pages 417–42, 2010.

[171] Tomás Alarcón, Helen M Byrne, and Philip K Maini. A multiple scale model for tumor growth. *Multiscale Modeling & Simulation*, 3(2):440–475, 2005.

[172] LL Bonilla, V Capasso, M Alvaro, and M Carretero. Hybrid modeling of tumor-induced angiogenesis. *Physical Review E*, 90(6):062716, 2014.

[173] J Grote, R Süsskind, and P Vaupel. Oxygen diffusivity in tumor tissue (ds-carcinosarcoma) under temperature conditions within the range of 20–40 c. *Pflügers Archiv*, 372(1):37–42, 1977.

[174] John R Farver. Oxygen self-diffusion in calcite: dependence on temperature and water fugacity. *Earth and Planetary Science Letters*, 121(3):575–587, 1994.

[175] JP Freyer and RM Sutherland. A reduction in the in situ rates of oxygen and glucose consumption of cells in emt6/ro spheroids during growth. *Journal of cellular physiology*, 124(3):516–524, 1985.

[176] Joseph J Casciari, Stratis V Sotirchos, and Robert M Sutherland. Variations in tumor cell growth rates and metabolism with oxygen concentration, glucose concentration, and extracellular ph. *Journal of cellular physiology*, 151(2):386–394, 1992.

[177] Paul K Davis. Generalizing concepts and methods of verification, validation, and accreditation (vv&a) for military simulations. Technical report, DTIC Document, 1992.

[178] Osman Balci. Verification, validation, and accreditation. In *Proceedings of the 30th conference on Winter simulation*, pages 41–4. IEEE Computer Society Press, 1998.

[179] James P Freyer and Robert M Sutherland. Selective dissociation and characterization of cells from different regions of multicell tumor spheroids. *Cancer research*, 40(11):3956–3965, 1980.

[180] James P Freyer. Role of necrosis in regulating the growth saturation of multicellular spheroids. *Cancer Research*, 48(9):2432–2439, 1988.

[181] Markus W Gross, Ulrich Karbach, Karlfried Groebe, Allan J Franko, and Wolfgang Mueller-Klieser. Calibration of misonidazole labeling by simultaneous measurement of oxygen tension and labeling density in multicellular spheroids. *International journal of cancer*, 61(4):567–573, 1995.

[182] Alessandro Bertuzzi, Antonio Fasano, Alberto Gandolfi, and Carmela Sinisgalli. Necrotic core in emt6/ro tumour spheroids: Is it caused by an atp deficit? *Journal of theoretical biology*, 262(1):142–150, 2010.

[183] WF Mueller-Klieser and RM Sutherland. Oxygen tensions in multicell spheroids of two cell lines. *Br. J. Cancer*, 45:256–264, 1982.

[184] W Mueller-Klieser, JP Freyer, and RM Sutherland. Influence of glucose and oxygen supply conditions on the oxygenation of multicellular spheroids. *British journal of cancer*, 53(3):345, 1986.

[185] Janna P Wehrle, Cheng E Ng, Kathy A McGovern, Nanci R Aiken, Dikoma C Shungu, Edwin M Chance, and Jerry D Glickson. Metabolism of alternative substrates and the bioenergetic status of emt6 tumor cell spheroids. *NMR in Biomedicine*, 13(6):349–360, 2000.

[186] Leoni A Kunz-Schughart, Joerg Doetsch, Wolfgang Mueller-Klieser, and Karlfried Groebe. Proliferative activity and tumorigenic conversion: impact on cellular metabolism in 3-d culture. *American Journal of Physiology-Cell Physiology*, 278 (4):C765–C780, 2000.

[187] Joseph J Casciari, Stratis V Sotirchos, and Robert M Sutherland. Glucose diffusivity in multicellular tumor spheroids. *Cancer Research*, 48(14):3905–3909, 1988.

[188] Wolfgang F Mueller-Klieser and Robert M Sutherland. Influence of convection in the growth medium on oxygen tensions in multicellular tumor spheroids. *Cancer research*, 42(1):237–242, 1982.

[189] Allan J Franko and Robert M Sutherland. Oxygen diffusion distance and development of necrosis in multicell spheroids. *Radiation research*, 79(3):439–453, 1979.

[190] Antonio Brú, Juan Manuel Pastor, Isabel Fernaud, Isabel Brú, Sonia Melle, and Carolina Berenguer. Super-rough dynamics on tumor growth. *Physical Review Letters*, 81(18):4008, 1998.

[191] Pietro Ghezzi, Charles A Dinarello, Marina Bianchi, Mary E Rosandich, John E Repine, and Carl W White. Hypoxia increases production of interleukin-1 and tumor necrosis factor by human mononuclear cells. *Cytokine*, 3(3):189–194, 1991.

[192] Adrian L Harris. Hypoxia—a key regulatory factor in tumour growth. *Nature Reviews Cancer*, 2(1):38–47, 2002.

[193] Reza Fazel, Harlan M Krumholz, Yongfei Wang, Joseph S Ross, Jersey Chen, Henry H Ting, Nilay D Shah, Khurram Nasir, Andrew J Einstein, and Brahmajee K Nallamothu. Exposure to low-dose ionizing radiation from medical imaging procedures. *New England Journal of Medicine*, 361(9):849–857, 2009.

[194] D Fuentes, JT Oden, KR Diller, JD Hazle, A Elliott, A Shetty, and RJ Stafford. Computational modeling and real-time control of patient-specific laser treatment of cancer. *Annals of biomedical engineering*, 37(4):763–782, 2009.

[195] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Selected Publications

[1] **Fei Xing**, Jörg Eisele, Peter Bastian and Dieter W. Heermann. Computational Complexity of Agent-based Multi-scale Cancer Modeling. 47^th Summer Computer Simulation Conference-SCSC 2015, 2015, under peer review.

[2] **Fei Xing**, Jörg Eisele, Peter Bastian, Felix Heimann, Jürgen Hesser and Dieter W. Heermann. A Computational Multiscale Model for Avascular Tumor Tissue Growth, to be submitted soon.

[3] **Fei Xing**, Jörg Eisele, Peter Bastian and Dieter W. Heermann. TUGME: A Tumor Growth Multi-scale Modeling and Simulation Environment, in preparation.