POLITECNICO DI TORINO

Master's Degree in Mathematical Engineering

Master's Degree Thesis

A Cellular Potts Model to descrive tumor infiltration in confined in vivo tissues

Supervisor

Prof. Marco SCIANNA

Candidate

Sofia GIORDANO

March 2025

Summary

Tumor invasion is one of the most complex and critical processes in cancer progression. This thesis explores the ability of tumor cells to overcome biological barriers, specifically muscle fibers. This process is driven by the activity of matrix metalloproteinases (MMPs), which degrade the extracellular matrix (ECM) and facilitate cellular movement.

To investigate this phenomenon, the Cellular Potts Model (CPM) was employed as a mathematical framework particularly well-suited for modeling tumor invasion. The CPM enables the simulation of collective cell behavior and the analysis of interactions between tumor cells, ECM, and muscle fibers, providing both quantitative and qualitative insights into the invasion process.

Numerical simulations were conducted using CompuCell3D. First, a realistic tumor invasion scenario was recreated, simulating the migration of tumor cells through muscle fibers without the ECM and then the ECM was introduced. Subsequently, the effects of key model parameters such as cell elasticity, motility, and MMPs secretion were analyzed to identify conditions that could slow or stop tumor progression.

The findings offer a deeper understanding of tumor invasion dynamics and suggest potential strategies to limit tumor spread, providing valuable insights for future clinical and therapeutic developments.

Table of Contents

List of Tables

List of Figures

Acronyms

IFS

Interstitial fluid spaces

CPM

Cellular Potts Model

GGH

Glazier-Graner-Hogeweg

MCS

Monte Carlo Step

PDEs

partial differential equations

ODEs

ordinary differential equations

CC3D

CompuCell3D

PIFF

Pixel Initialization Files

ECM

extracellular matrix

MMPs

matrix metalloproteinases

Chapter 1

Introduction

Connective tissue is a complex and multi-scale biomaterial made up of proteins, glycosaminoglycans, and water. It plays a crucial role in connecting tissues and organs, both mechanically and functionally. This tissue serves as the main environment where cells migrate during processes like inflammation and cancer development. Therefore, to accurately study cell behavior in biological models, it is essential to create connective tissue models that closely mimic the real in vivo environment. These models should be able to predict the molecular and physical mechanisms of cell movement with high precision.

One of the main challenges in developing in vitro models, such as reconstituted basement membrane, fibrillar collagen, and cell-derived extracellular matrix (ECM), is to ensure their physiological relevance. To improve these models, it is necessary to create a realistic representation of connective tissue, considering aspects such as different scales, spatial organization, and structural patterns. Achieving detailed mapping of living connective tissue and its interactions with cells is key to improving tissue models for in vitro studies.

Cell migration within the ECM is highly influenced by the design of extracellular confinement. If the spaces between collagen fibers are too narrow for the cell nucleus to pass through, movement relies on matrix metalloproteinases (MMPs) to break down the ECM and create a path. On the other hand, models that lack sufficient collagen fibers may not provide the necessary viscoelastic properties, which are crucial for generating traction forces needed for mesenchymal movement. Instead, such models may only support cell growth through pushing mechanisms. In microfluidic channels with irregular spatial patterns, non-deformable surfaces allow friction-free leukocyte migration driven by cortical actin flow. However, it is still unclear which specific tissue conditions and interstitial environments support these migration mechanisms in living organisms. Connective tissue consists of an

ECM with a heterogeneous texture, which can act as either a guide or a barrier for moving cells, such as leukocytes, stromal cells, and cancer cells. Recent studies suggest that the ECM and its integrated cells are not arranged randomly; instead, they exhibit a highly organized structure. Interstitial fluid spaces (IFS), which are present in various organs such as the liver, play an important role in nutrient and fluid exchange. However, it remains unclear how these spaces are organized within different tissue types, such as collagen-rich or adipocyte-rich connective tissue, and whether they facilitate or hinder cell migration under normal and disease conditions.

The structure and function of the IFS have been investigated in previous research [1], which serves as the starting point for this thesis. In that study they performed spatial mapping of the deep mouse dermis using intravital microscopy in models of inflammation and cancer progression. This technique allows for the visualization of tissue architecture while minimizing artifacts caused by tissue processing, such as dehydration or shrinkage. The results indicate that connective tissue is organized as a deformable network of confined spaces, similar to microfluidic channels, with varying shapes and sizes depending on surrounding structures like muscle fibers, fat cells, nerves, or blood vessels. The structural organization of these conduits can be either symmetrical or asymmetrical, bordered by bundled collagen or basement membrane layers. These structures provide a rich environment for cells to migrate efficiently without the need for MMPs activity, allowing for nearly barrier-free directional movement.

This thesis develops computational models using the Cellular Potts Model (CPM) to simulate tumor cell migration through muscle fibers. The proposed simulations aim to replicate the interactions between tumor cells and their microenvironment, focusing on key factors such as cell elasticity, adhesion energies between different components, and the enzymatic degradation of ECM by MMPs. The initial simulation setting consists of tumor cells near muscle fibers. The second simulation setting consists of tumor cells near muscle fibers held together by ECM .The objective of the model is to recreate these two environments and analyze how the cell properties affect migration. Through this approach, the study provides a quantitative and qualitative analysis of the factors that most significantly influence tumor cell movement, offering potential insights for improving therapeutic interventions.

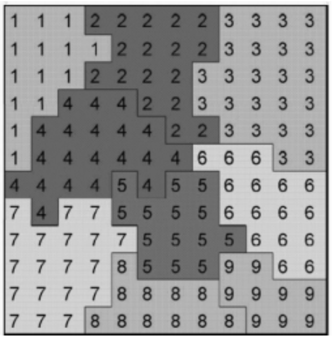
Chapter 2

Cellular Potts Model (CPM)

The Cellular Potts Model (CPM), also known as the Glazier-Graner-Hogeweg model (GGH), is a computational framework used to study the spatiotemporal behavior of cells and tissues [2]. François Graner and James A. Glazier introduced the first CPM by simulating cell sorting as a modification of a large-Q Potts model. Initially, it was developed to model biological cells, but the CPM has since been expanded to simulate individual parts of cells or even fluid regions.

The CPM uses a framework derived from statistical mechanics to describe cellular behaviors, a choice that is not immediately intuitive. Unlike continuum models, which overlook individual cells and treat tissues as homogeneous materials, the CPM focuses on cells as the fundamental units. This approach contrasts with pointillistic models, which simplify biological tissues into point-like cells, ignoring critical aspects such as cell shape and adhesion at cellular membranes. Both continuum and pointillistic models offer convenience and have contributed significantly to understanding tissue development and physiology. However, many biological structures, like capillaries and pancreatic islets, operate on scales close to the size of individual cells, requiring a more detailed spatial representation. Therefore, the CPM provides a more accurate analysis of these systems by focusing on the spatial organization of cells [3].

The CPM, also, excels in capturing dynamic behaviors such as cell detachment, reattachment, and diverse forms of cell-cell adhesion. This capability highlights its strength in representing complex biological processes at the cellular level, including inherent stochasticity in movements and deformations [4]. The CPM is often characterized as a time-discrete Markov chain, simplifying the complexity of millions of molecular interactions by focusing on key cell behaviors.



1	1	1	2	2	2	2	2	3	3	3	3
1	1	1	1	2	2	2	2	3	3	3	3
1	1	1	2	2	2	2	3	3	3	3	3
1	1	4	4	4	2	2	3	3	3	3	3
1	4	4	4	4	4	2	2	3	3	3	3
1	4	4	4	4	4	4	6	6	6	3	3
4	4	4	4	5	4	5	5	6	6	6	6
7	4	7	7	5	5	5	5	6	6	6	6
7	7	7	7	7	5	5	5	5	6	6	6
7	7	7	7	8	5	5	5	9	9	6	6
7	7	7	7	8	8	8	8	9	9	9	9
7	7	7	8	8	8	8	8	8	9	9	9

Figure 2.1: Representative 2D CPM domain formed by 144 square sites. The numbers indicate cell index values σ. The levels of gray indicate cell types $\tau(\sigma)$. A CPM object is a collection of lattice sites with the same spin value [3].

2.1 CPM domain and cell representation

In the CPM framework, the system is represented on a grid domain $D \subset R^n$, being $n \in \{1,2,3\}$. The grid domain is composed of identical square or hexagonal sites, each identified by a vector $\vec{i} \in R^n$ which indicates its center and labeled by an integer number $\sigma(\vec{i}) \in N$, which can be interpreted as a degenerate spin. A group of domain sites with the same identification number form a generalized cell, i.e., a CPM object that may represent an entire biological cell, a subcellular compartment, a cluster of cells, or even non-cellular material or a component of the surrounding environment. Of course, a biological cell conceptualized as a cluster of subunits, can be characterized by more complex and detailed morphologies.

Each CPM object σ, has an associated type, $\tau(\sigma)$. The boundaries between lattice sites with different indices represent the contact regions between objects [3]. In Figure 2.1, a typical configuration of a bidimensional CPM domain is shown.
Each generalized cell in the CPM has further defining attributes, including surface area, volume, and more intricate properties, e.g. its biochemical state and internal regulatory networks, that may drive its behavior and dynamics in various biological contexts.
CPM frameworks can include also molecular elements, described in terms of continuous variables whose evolution is established by appropriate partial differential equations (PDEs) that possibly account for their diffusion, decay, absorption, secretion. The genetic scale can be included as well by the introduction of boolean/dichotomic variables. The connection between the different levels gives the CPM an intrinsic multiscale characteristic, coupled with a hybrid approach, since the cell scale is reproduced by an IBM whereas the subcellular scale by a continuous approximation.

2.2 Modified Metropolis algorithm

In the CPM, the system evolution is determined by a stochastic process that implements a Metropolis algorithm for Monte Carlo dynamics. It iteratively reduces the system free energy defined by a Hamiltonian functional H (to be discussed in more detail later).
To represent the temporal evolution of the model, the concept of a Monte Carlo Step (MCS) is introduced. A single MCS is defined as 1 index-copy attempt, it is the natural time unit for the model and in biologically realistic scenarios can be considered approximately proportional to experimental time.
Each MCS consists of the following sequence of actions:

1. Randomly select a source lattice site, $\vec{i}$, and an unlike neighboring target site, $\vec{i'}$.

2. If these two sites belong to different generalized cells, i.e, if $\sigma(\vec{i}) \neq \sigma(\vec{i'})$ calculate the energy difference (ΔH) between the current configuration of the domain and the trial one obtained by assigning to i' the same index of i $(\sigma(\vec{i'}) = \sigma(\vec{i}))$.

3. Accept the configuration update with a probability $P(\sigma(\vec{i}) \to \sigma(\vec{i'}))$ based on a Boltzmann like acceptance function:

$$P(\sigma(\vec{i}) \to \sigma(\vec{i'})) = \begin{cases} 1 & \text{if } \Delta H \leq h = 0 \\ e^{-\frac{\Delta H}{T_{\tau(\sigma(\vec{i}))}}} & \text{if } \Delta H > h = 0 \end{cases} \tag{2.1}$$

In Eq.(2.1), $T_{\tau(\sigma(\vec{i}))}$ is "Boltzmann temperature" which affects the likelihood of changes and a "transition threshold" h which controls whether a new configuration is accepted. Initially, the transition threshold was set to a fixed value in the original model, but subsequent studies have shown that adjusting it in specific applications can improve results.
ΔH represents the change in effective energy if the copy occurs, and T_τ is a parameter controlling the magnitude of cell-membrane fluctuations. The value of T_τ can be set globally for the entire system or tailored for specific cells or cell types, as in this case. The ratio $\Delta H/T_\tau$ for a given object of type τ determines how much its boundaries fluctuate. If $\Delta H/T_\tau$ is large, cells become more rigid and exhibit minimal or no motility, resulting in little cell rearrangement. Conversely, a low $\Delta H/T_\tau$ allows for more significant fluctuations, promoting higher cell motility and rearrangement. When $\Delta H/T_\tau$ is extremely small, cells may fragment if no constraint maintains their boundary integrity.

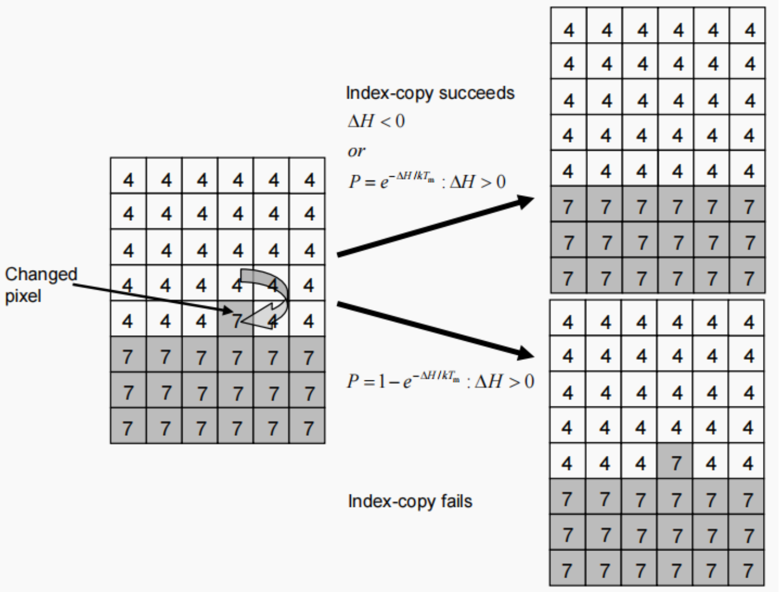

Changed pixel
Index-copy succeeds
$\Delta H < 0$
or
$P = e^{-\Delta H / kT_m} : \Delta H > 0$
$P = 1 - e^{-\Delta H / kT_m} : \Delta H > 0$
Index-copy fails

Figure 2.2: CPM representation of an index-copy attempt for two cells on a 2D square lattice. The "white" pixel (source) attempts to replace the "grey" pixel (target). The probability of accepting the index copy is given by Eq.(2.1). This image is adapted from [5]

4. If the new configuration is accepted, update the lattice site. Otherwise, keep the current configuration unchanged.

The basic step of the algorithm has to be repeated until the system reaches an energy global minimum or for a given number of iterations. To illustrate an index-copy attempt in this algorithm, Figure 2.2 provides a graphical representation of two cells on a 2D square lattice, where the "white" pixel (source) attempts to replace the "grey" pixel (target). The probability of accepting the index copy is given by Eq.(2.1).

These dynamics are thought to mimic membrane fluctuations, where one cell reduces its volume by releasing a lattice site while a neighboring cell gains volume by occupying that same site.

An index copy that increases the effective energy is unlikely to occur. As a result, cell configurations evolve following biologically relevant principles encoded in the effective energy. The Metropolis algorithm updates the configuration of the cell lattice in a way that satisfies the constraints to the greatest extent possible, with perfect damping (i.e., average velocities are proportional to the forces applied). Consequently, the time evolution of the cell lattice mirrors what could be achieved deterministically using methods like finite-element models or center-model approaches with perfect damping [6].

To finalize the model, suitable boundary conditions must be established. When the influence of domain boundaries is irrelevant, periodic boundary conditions are applied, effectively treating the space as if it were wrapped onto a torus. This approach ensures that cells can move continuously across boundaries. Alternatively, fixed boundary conditions can be imposed, explicitly modeling the interactions between cell surfaces and the surrounding environment within the framework, offering more precise control over cell behavior near boundaries [7].

2.3 Energy Function: Hamiltonian

The energy function, called the Hamiltonian, controls the behavior of the system in the CPM. The Hamiltonian reflects the energetic cost associated with different cellular configurations and depends on several factors, such as cell adhesion, volume constraints, and surface tension. The Hamiltonian captures most of the cell characteristics, behaviors, and interactions through constraint terms.
To avoid confusion, it's important to clarify that the term "energy function" simulates specific cellular behaviors and doesn't correspond to the actual physical energy of the cells.
In the time-discrete Markov chain, the Hamilton function is associated with the transition probabilities, as shown in Eq.(2.1).
Generally, the Hamiltonian consists of several terms, which control distinct aspects of cell behaviors. The typical structure of a CPM-Hamiltonian is:

$$H = H_{adhesion} + H_{attribute} + H_{force} \tag{2.2}$$

Most studies incorporate an adhesive interaction term, characterized by a symmetric matrix of contact energy coefficients:

$$H_{adhesion} = \sum_{(\vec{i},\vec{i'})-\text{neighbors}} J(\tau(\sigma(\vec{i})), \tau(\sigma(\vec{i'})))(1 - \delta(\sigma(\vec{i}), \sigma(\vec{i'}))) \tag{2.3}$$

Eq.(2.3) takes over all pairs of neighboring lattice sites $\vec{i}$ and $\vec{i'}$, that belongs to different elements, computes the boundary or contact energy between them, thereby modeling adhesion interactions. The coefficients $J(\tau(\sigma(\vec{i})), \tau(\sigma(\vec{i'})))$ represent a boundary energy per unit of contact area for a pair of cells, where $\sigma(\vec{i})$ of type $\tau(\sigma(\vec{i}))$ occupies the lattice site $\vec{i}$ and $\sigma(\vec{i'})$ of type $\tau(\sigma(\vec{i'}))$ occupies the neighboring site $\vec{i'}$. $J(\tau(\sigma(\vec{i})), \tau(\sigma(\vec{i'})))$ is typically specified as a matrix indexed by the types of cells. Higher boundary energies between cells lead to increased repulsion, while lower boundary energies correspond to greater adhesion. The Kronecker delta term $1 - \delta(\sigma(\vec{i}), \sigma(\vec{i'}))$ ensures that neighboring lattice sites of the same state do not contribute to the total energy of the system [8].

Depending on the specific biological scenario being investigated, other types of terms can be integrated into the model. They describe selected attributes of the set of modeled elements (such as their geometrical characteristics) and have the structures of elastic potentials:

$$H_{attribute} = \sum_{attr} \sum_{\sigma} [\lambda_{attr}(\sigma)(a(\sigma, t) - A(\sigma))^2] \tag{2.4}$$

The summation in Eq. (2.4) calculates the effective energy arising from attribute constraints on each generalized cell, where $a(\sigma, t)$ represents the value of the attribute for element σ at time t, and A denotes the target value. Deviations of the cell's attribute from its target value $A(\sigma)$ increase the effective energy, thus penalizing such deviations. The parameter λ_{attr} functions like a Young's modulus, with larger values reducing fluctuations of the cell's attribute around its target value.

The last term accounts for the energetic contributions of the forces (both effective and generalized) that act on the simulated individuals. All of these are modeled :

$$H_{force} = \sum_{\sigma} \sum_{\vec{i} \in \sigma} \sum_{k\text{-}force} \mu_{\sigma}^{k}(t) F^{k}(t) r_{\vec{i}} \tag{2.5}$$

where $r_{\vec{i}} = (x_{\vec{i}}, y_{\vec{i}}, z_{\vec{i}})^T$ is the position vector of lattice site $\vec{i}$, which represents the application point of force F^k, and μ_{σ}^{k} is the relative parameter, which measures the effective strength of the force on object σ. Common examples in CPM simulations include forces exerted by extracellular chemical gradients, which are continuous CPM objects, on a discrete population of cells:

$$H_{force}^{chemical} = \text{-} \sum_{\sigma} \sum_{\vec{i} \in \sigma} \mu_{\sigma}(t) c(\vec{i}, t) \tag{2.6}$$

where $c(\vec{i}, t)$ is the concentration of the chemical sensed by cell site $\vec{i}$ and the Potts coefficient μ_{σ} is, in this case, interpreted as an effective chemical potential of cell σ. Moreover, the net energy difference caused by such a chemical force is:

$$\Delta H_{force}^{chemical}|_{\sigma(\vec{i}) \to \sigma(\vec{i'})} = \mu_{\sigma}[c(\vec{i}, t)\text{-}c(\vec{i'}, t)] \tag{2.7}$$

where $\vec{i}$, which belongs to the boundary of σ, and $\vec{i'}$ are the two neighboring lattice sites randomly selected during the trial update at time t. If μ_{σ} is a constant, σ has a linear chemical sensitivity. In particular, $\mu_{\sigma} > 0$ results in movement up the chemical gradient (making c a chemoattractant, and the relative force is called chemotaxis), while $\mu_{\sigma} < 0$ causes motion down the gradient (and c is a chemorepellent).

Chapter 3

CPM Application to Tumor Infiltration

Given in the previous chapter the general structure of the CPM, this part of the thesis will be focus on its application to describe tumor infiltration within confined tissues.

More specifically, we will deal with two different scenarios: in the first one (S1), a single malignant cell is assumed to migrate through a pair of muscle fibers. The second one (S2) is similar but introduces a key difference: the muscle fibers are encased in and connected by a layer of extracellular matrix filaments that ensure their structural cohesion. In both cases, we will employ a two-dimensional approximation.

In this respect, the first part of this chapter will be focused on the model, and the relative simulations, of setting S1, whereas in the second part of the chapter we will add the model ingredients necessary to implement the setting S2 and present the relative numerical results.

3.1 Scenario S1

3.1.1 S1: Model

In the first scenario S1, a single tumor cell, i.e., a CPM object of type $\tau = T$, is set to migrate through a pair of muscle fibers. Each of them is composed of muscle elements which are CPM objects of type $\tau = F$. Domain sites not occupied by the previously mentioned elements are assigned a generalized spin value of $\sigma = 0$ and a type $\tau = M$, indicating the medium, i.e., a sort of interstitial fluid characteristic of biological environments.

In more details, cell movement is established by a directional potential that biases the locomotion of the malignant individual in between the pair of muscle elements. Effective tumor infiltration is determined by the elastic properties of both the cell and the muscle fibers. In this respect, the tumor cell is set to maintain an almost constant surface area during movement, with temporary variations of its perimeter. This adaptive mechanism enables it to deform and adjust its shape to navigate through such narrow spaces. Muscle fibers can also display elastic properties, though to a lesser extent than tumor cells, as they have to retain a relatively stable structure.

Building upon this biological framework, the Hamiltonian of the system is given by

$$H(t) = H_{\text{sur}}(t) + H_{\text{per}}(t) + H_{\text{pot}}(t) + H_{\text{adh}}(t), \tag{3.1}$$

where

- $H_{\text{sur}}(t)$ and $H_{\text{per}}(t)$ account for surface and perimeter constraints, the former ensuring variations in the area of the cell the latter dealing with its deformability. These terms are derived from a specific formulation of the attribute energy term (H_{attr}), as introduced earlier in Eq.(2.4);
- $H_{\text{pot}}(t)$ models the directional movement of the tumor cell along the y-axis. This is a particular instance of the force energy term (H_{force}), described in Eq. (2.5);
- $H_{\text{adh}}(t)$ describes the possible adhesive interactions between the tumor cell and muscle fibers, as detailed in Eq.(2.3). This term reflects the energy associated with contact dynamics between different types of elements.

Expanding the Hamiltonian, it can be explicitly written as:

$$\begin{aligned} H(t) = {} & \sum_{\sigma} \lambda_{\text{sur}}(\sigma) \left(s(\sigma,t) - S(\sigma)\right)^2 + \sum_{\sigma} \lambda_{\text{per}}(\sigma) \left(p(\sigma,t) - P(\sigma)\right)^2 \\ & + \sum_{i \in \sigma} v(t) r_{\vec{i}} + \sum_{(\vec{i},\vec{i'})\text{-neighbors}} J\left(\tau(\sigma(\vec{i})), \tau(\sigma(\vec{i'}))\right) \left(1 - \delta(\sigma(\vec{i}), \sigma(\vec{i'}))\right) \end{aligned} \tag{3.2}$$

We recall that the system evolves according to the Boltzmann-like transition law introduced in Eq. (2.1).

Simulation details and parameter setting. The initial conditions of the simulation are designed to replicate a biologically relevant scenario while ensuring computational feasibility. The simulation domain $D \in R^2$, consists of a 230 × 1000 pixels regular grid. The temporal resolution of the model is defined in terms of Monte Carlo Steps (MCS).

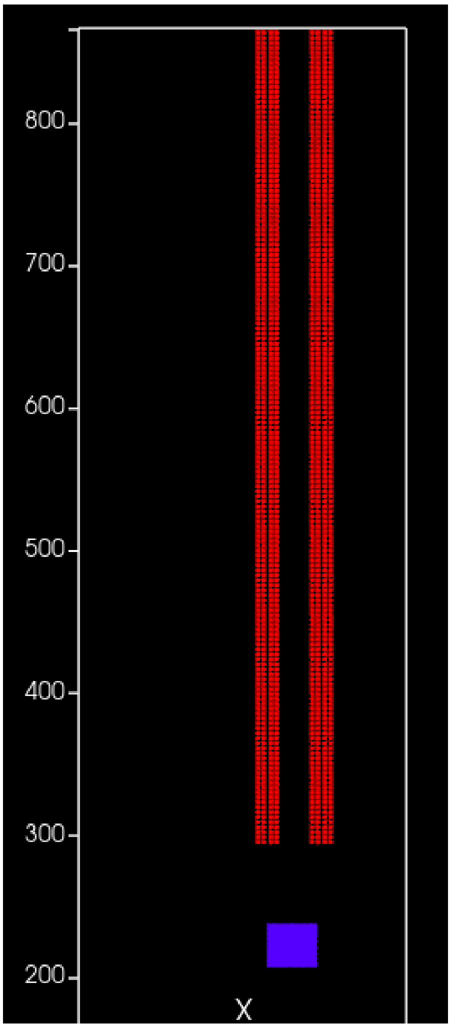


Figure 3.1: Initial configuration for setting S1: Tumor cells migrating through muscle fibers without ECM.

The tumor cell is initially represented as a circular object with an area of 1200 pixels. The domain also includes two parallel myo-fiber structures, separated by 21 pixels. Each muscle fiber consists of two closely spaced parallel elements, each with a thickness of 8 pixels and separated by a single pixel. Each fiber has a length of 660 pixels and is composed of small elements with an individual area of 16 pixels. Cell and fibers initial measures also correspond to the target values present in the Hamiltonian. Such a structural choice that ensures an accurate representation of the tumor infiltration through the constrained environment.

The parameters required for the Hamiltonian calculation, including elasticity coefficients, target values, and interaction energies, are summarized in Table 3.1.

In particular, T_T is a measure of the intrinsic motility of the overall individual, as it gives the frequency of the ruffles of its cytosol (which, on a molecular level, are determined by polarization/depolarization processes of the actin cytoskeleton). It is set, after some preliminary simulations, equal to an high 20. T_F instead determines the vibration degree of the components of the muscle fibers, which are not allowed to substantially move from their original position. For this reason, we set a lower $T_F = 4$.

Assuming that the tumor cell does not significantly grow during migration, the fluctuations of its area are kept negligible with a high constant value $\lambda_{sur} \gg 1$. Moreover, malignant cells moving in confined environments are typically deformable: therefore, we set a low $\lambda_{per} = 5$. Muscle components are finally assumed to maintain their extension but to be characterized by a high degree of deformability. For them, we indeed fix $\lambda_{sur} = 30$ and $\lambda_{per} = 2$.

$J_{T,M}$ and $J_{T,F}$ evaluate the heterophilic contact interactions between the tumor cell and extracellular components: specifically, the former coefficient is a measure of the affinity between integrins complexes on cell surfaces and soluble ligands present in the medium, whereas the latter one quantifies possible adhesive interaction between the malignant individuals and muscle elements. Specifically, we set high values (i.e., >> 1) for both parameters: the aim of the thesis in fact to analyze the direct influence of cell and fiber deformability on tumor infiltration and therefore we prefer to avoid that adhesive interactions affect cell movement. This choice is also consistent with the experimental literature, which widely demonstrates that most cell lines display sustained ameboid motility in confined environments in a poorly adhesive mode

This mathematical framework provides a robust basis for analyzing the interplay between the physical and biological factors that drive cell migration in the simulated environment.

3.1.2 S1: Numerical results

Reference simulation. With the terminology reference simulation, we hereafter indicates the numerical setting given the parameter values listed in Table 3.1.

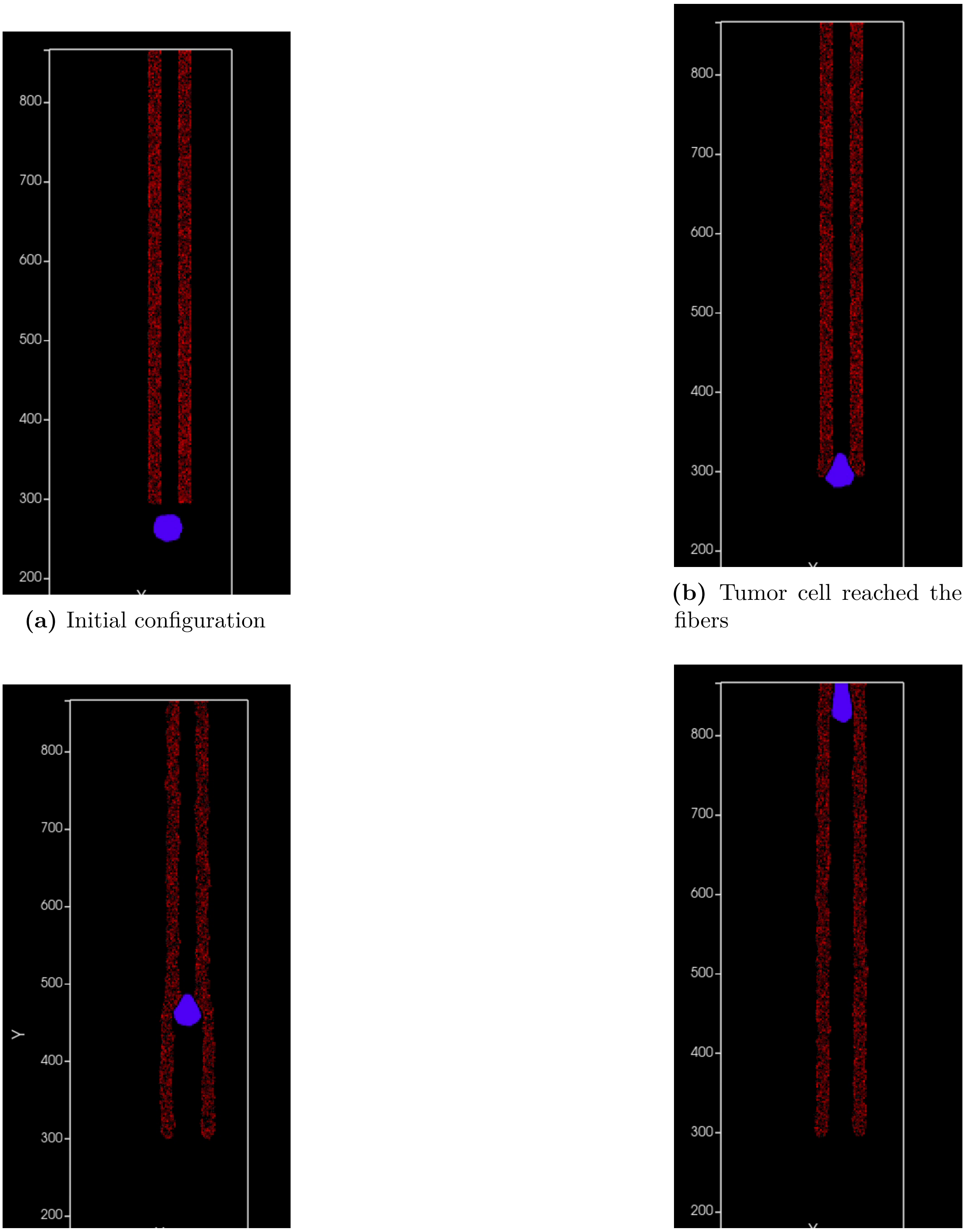
Initially, the tumor cell is seeded in the close proximity to the muscular structure and display an unpolarized morphology.

It then starts to squeeze between the pair of fibers. In particular, it remodels towards an elongated shape. The transition from a stationary cell morphology to a polarized shape, which is completely self generating and due to the geometry of the environment, is necessary but not sufficient to determine the infiltration capacity of the individual. The muscle elements, subjected by mechanical stresses exerted by the tumor cell, in fact widen allowing the malignant individual to penetrare in between them. The figure 3.2 shows this kind of behavior.

Table 3.1: Model Parameters (I)

Parameter	**Description**	**Value**	**Reference(s)**
Tumor cell			
λ_{sur}	elasticity coefficient for surface deformation	40	estimated
λ_{per}	elasticity coefficient for perimeter deformation	5	estimated
S	initial/target surface area	1200	Friedl et Al [1]
P	initial/target perimeter	120	Friedl et Al [1]
v	potential driving directional migration.	-6	estimated
$J_{T,M}$	adhesion energy between medium and tumor cell	15	estimated
$J_{T,F}$	adhesion energy between tumor cell and muscle fibers	40	estimated
T_T	Motility of tumor cell	20	estimated
Muscle fibers			
λ_{sur}	elasticity coefficient	30	estimated
λ_{per}	elasticity coefficient	2	estimated
S	initial/target surface area of muscle fibers	16	Friedl et Al [1]
P	initial/target perimeter of muscle fibers	16	Friedl et Al [1]
$J_{F,M}$	adhesion energy between medium and muscle fibers	2	estimated
T_F	Motility of muscle fibers.	4	estimated

The same mechanism, i.e., inter-fiber space enlargement upon tumor cell passage, is repeated until the malignant individual reached the top border of the domain. The time needed by the cell to cross the entire muscle structure is ≈ 30000 MCS.



(a) Initial configuration

(b) Tumor cell reached the fibers

(c) Tumor cell in the middle of the fibers

(d) The tumor cell reached the top border of the domain

Figure 3.2: Tumor cell infiltration through muscle fibers in S1

Varying cell elasticity. As previously pointed out, to migrate within the pair of muscle fibers, the tumor cell needs to deform its body. Thus, the elastic properties of the cell represent a critical determinant for its migration efficacy. In this respect, we run a series of simulations where the malignant individual is assumed to have different degrees of deformability, given by different values of λ_{per}. The muscle tissue is instead fixed, i.e., non-deformable, by setting $T_F = 0$. All the other model parameters are kept the same as in the reference simulation.

The migration time, expressed in Monte Carlo Steps (MCS), was recorded for different values of λ_{per}, as summarized in Table 3.2.

λ_{per}	Migration Time (MCS)
5	76500
20	99000
30	116000
45	190000
60	206000
80	144000*
90	110700*

Table 3.2: Migration time for different values of λ_{per}. *Values for may be affected by numerical instabilities.

As shown in Table 3.2 and in the corresponding plot in Fig. 3.3, the results indicate a clear trend: as the cell becomes more rigid (i.e., for higher values λ_{per}), the migration time increases, being the migration time evaluated as the number of MCS needed by the cell to reach the top border of the domain.

Such numerical outcomes suggest that cell excessive rigidity hinders movement. Conversely, greater deformability facilitates a more efficient infiltration through confined environments.

The above-described behavior is qualitatively illustrated in Fig. 3.4 where we can observe the migration distance of the cell for different values of λ_{per}.

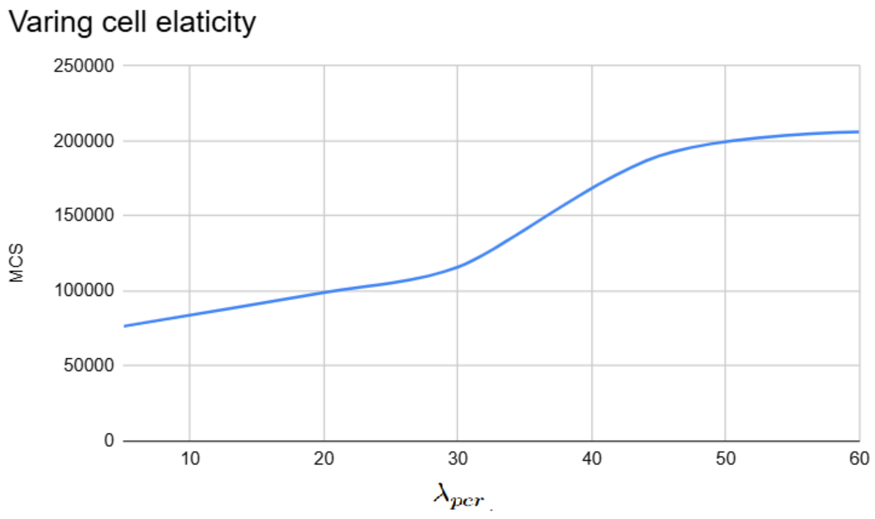
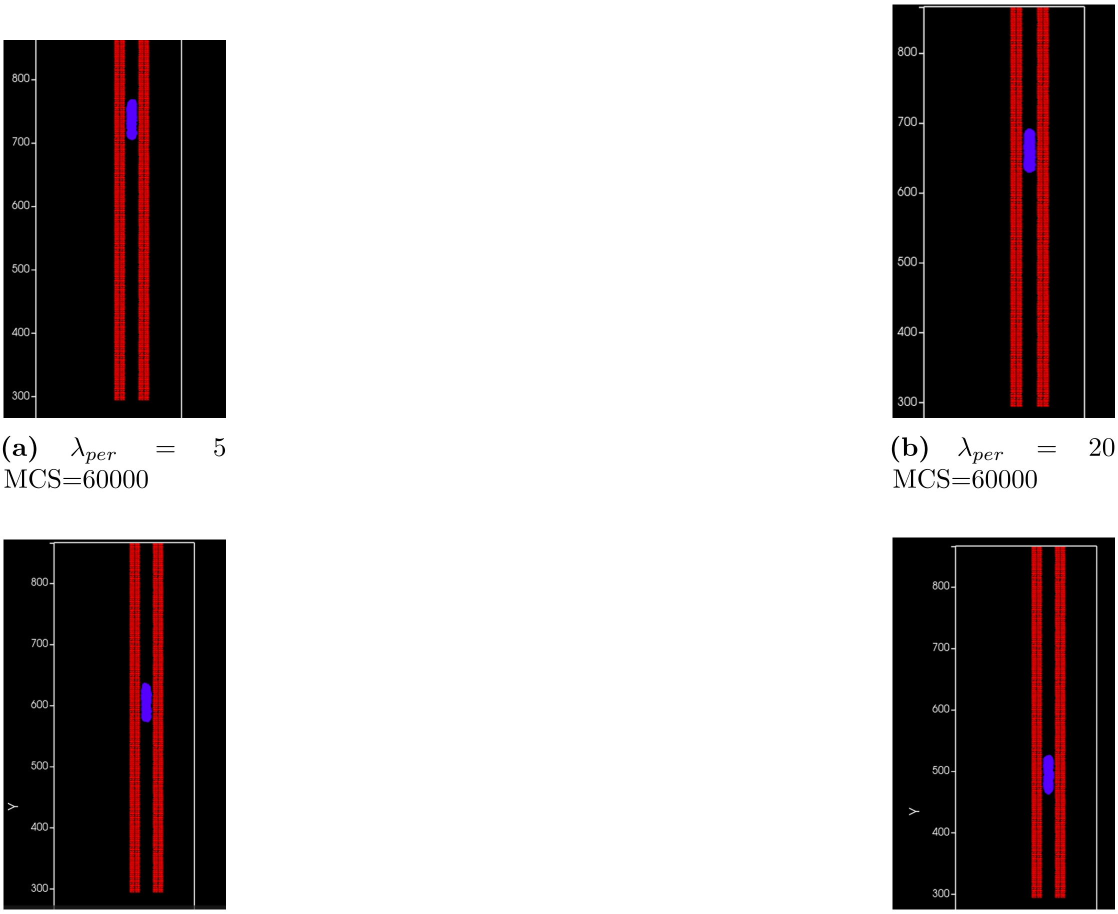


Figure 3.3: the impact of cellular elasticity on migration time



(a) $\lambda_{per} = 5$ MCS=60000

(b) $\lambda_{per} = 20$ MCS=60000

(c) $\lambda_{per} = 30$ MCS=60000

(d) $\lambda_{per} = 45$ MCS=60000

Figure 3.4: Tumor cell position at a fixed MCS equal to 60000 for different values of λ_{per}

We remark that the results with λ_{per} greater than 60 appear to deviate from the expected trend. These anomalies are likely due to numerical instabilities, and therefore they can be omitted from the graphical representation.

This finding is consistent with biological literature, which suggests that more rigid cells experience greater difficulty in infiltrating confined environments. For instance, Boyden chamber assays are typically used to correlate an increment in the ability of cancer cells to squeeze and migrate through microporous membranes to a drop in their elastic modulus, measured by a micro-plate based single-cell stretcher. Furthermore, glioma cell lines have been shown to squeeze through narrow locations in a brain model *in vivo*, thereby increasing their metastatic potential, by significantly compressing their body upon recruitment of nonmuscle myosin II (NMMII). Moreover, other authors have been provided that the directional persistence of cancer cells in microsized structures is mainly regulated by their steric hindrance. Finally, the number of acute promyelocytic leukemia (APL) cells able to migrate through filters smaller than cell diameters have been shown to be significantly reduced upon exposition to paclitaxel, which stabilizes the intracellular microtubule network.

Varying fiber elasticity. In the body, extracellular tissues display a range of elastic characteristics that affect the efficacy of tumor infiltration. To highlight this aspect, we now use the reference value for cell deformability $\lambda_{per} = 6$ while varying the rigidity of muscle elements. The results of the simulations are summarized in Table 3.3 and illustrated in Figure 3.5.

Muscle Rigidity	**Migration time (MCS)**
0.5	130000
1	148000
2	199000
3	230000
5	250000

Table 3.3: Migration time for the tumor cell to pass through muscle fibers as a function of muscle rigidity.

The results presented in Table 3.3 indicate a clear trend: as the muscle rigidity increases, the number of MCS required by the tumor cell to pass through the structure increases too.

The underlying explanation is that the more the muscle elements are rigid, the more the cell needs time to deform and penetrate between the muscular elements, given that it can not adapt its elastic properties.

Entering in more details, from the plot in Fig. 3.5, we observe a sort of saturation behavior for substantially high stiffness of muscle elements (i.e., for

$\lambda_{per} > 3$), suggesting that in this range of values muscle fibers form an effective physical barrier to infiltration.

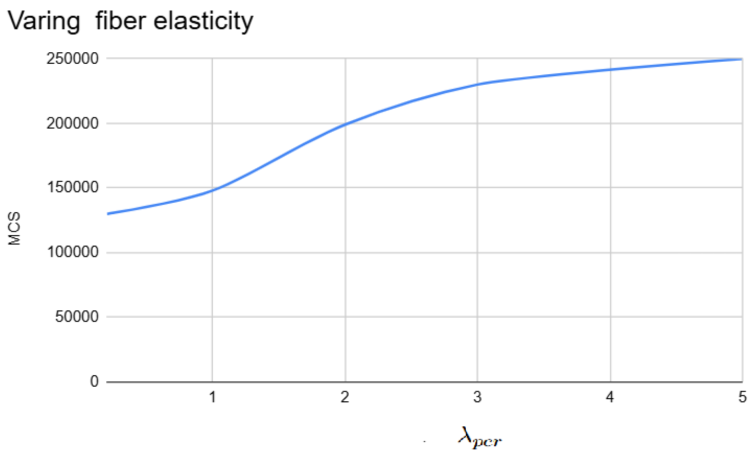
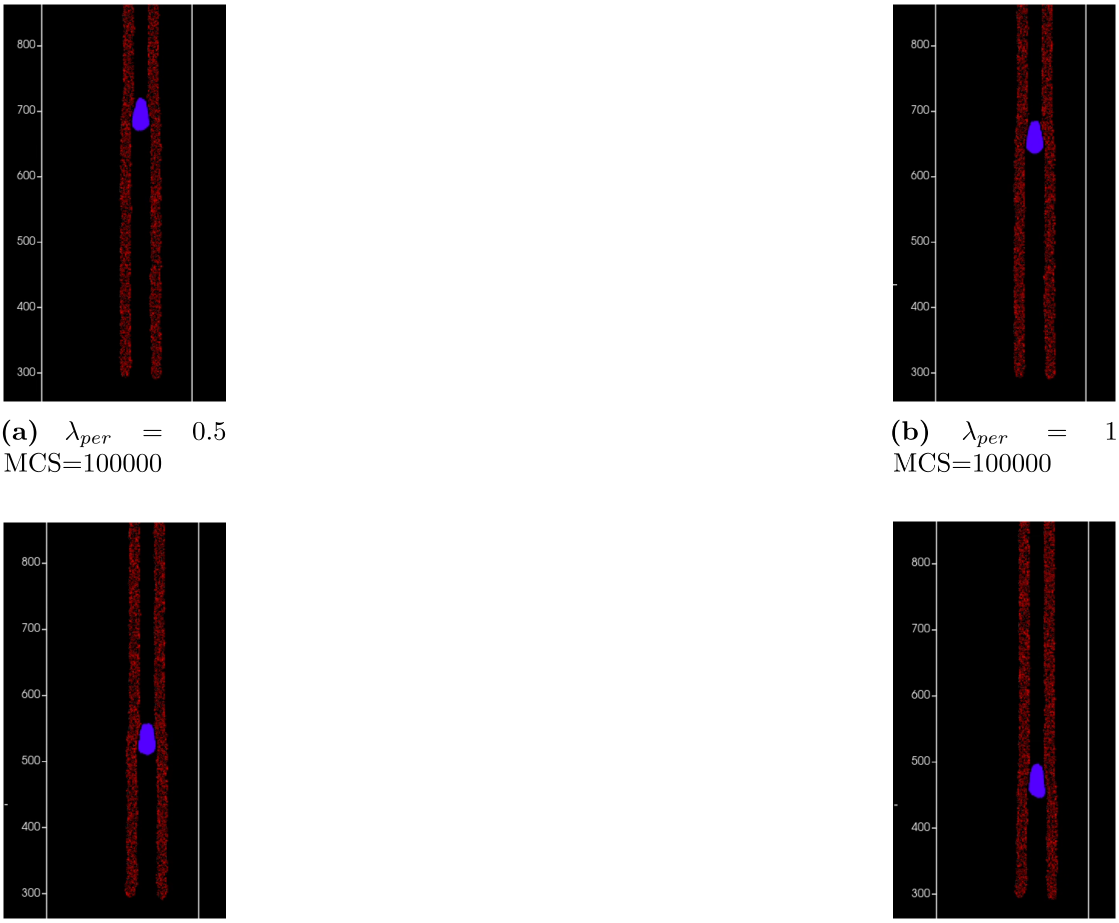


Figure 3.5: Number of MCS required for tumor cell passage as a function of muscle rigidity.

The above-described behavior is qualitatively illustrated in Fig. 3.6 where we can observe the migration distance of the cell for different values of λ_{per}.



(a) $\lambda_{per} = 0.5$ MCS=100000

(b) $\lambda_{per} = 1$ MCS=100000

(c) $\lambda_{per} = 3$ MCS=100000

(d) $\lambda_{per} = 5$ MCS=100000

Figure 3.6: Tumor cell position at a fixed MCS equal to 100000 for different values of λ_{per}

3.2 Scenario S2

3.2.1 S2: Model

In the second setting S2, ECM elements, i.e., CPM objects of type $\tau = E$, surround and link the pair muscle fibers. The primary function of the extracellular matrix is to provide mechanical and structural support, acting also as a barrier that the tumor cell has to overcome to infiltrate the tissue. ECM elements are assumed to be fixed over time, i.e., they are not allowed to move or change their dimensions.

With respect to the previous scenario, the tumor cell indeed exhibits an additional dynamic: it secretes enzymes, such as matrix metalloproteinases (MMPs), that selectively degrade extracellular matrix components, thereby facilitating their passage through the muscle tissue.

According to these considerations, the Hamiltonian of the system is analogous to the case of the first scenario, as given in Eq. (3.2) except to the fact that in the adhesive term we have to account for the contact interactions between ECM components and both the cell and the muscle fibers. iIt is useful to remark that geometrical constrained are not needed for matrix elements since they are set to be fixed objects.

The kinetics of the tumor-secreted matrix metalloproteinases (MMPs) are then modeled using a reaction-diffusion equation. The current concentration of MMPs at a given domain site i, denoted as $h_{MMP}(\vec{i}, t)$, evolves according to the following equation:

$$\frac{\partial h_{MMP}(\vec{i}, t)}{\partial t} = D_{MMP} \nabla^2 h_{MMP} - k_{MMP} h_{MMP} + \theta_{MMP}(\vec{i}), \tag{3.3}$$

where D_{MMP} and k_{MMP} are, respectively, the diffusion and degradation rate, and θ_{MMP} is the production rate:

$$\theta_{MMP}(\vec{i}) = \begin{cases} \theta_{MMP} & \text{if } \tau(\sigma(\vec{i})) = T \text{ and } \tau(\sigma(\vec{i'})) = E, \\ 0 & \text{otherwise,} \end{cases} \tag{3.4}$$

where $\vec{i}$ and $\vec{i'}$ are neighboring pixels. We are indeed assuming that the MMPs are secreted by the malignant individuals upon contact with an ECM component.

The mechanism of matrix degradation by tumor-produced proteolytic enzymes is implemented as follows: a lattice site $\vec{i}$ within an ECM element becomes a generalized pixel of medium when the local level of MMPs ($h_{MMP}(\vec{i})$) is sufficiently high, i.e., reaching a given threshold h_{deg}. This change is implemented by changing ($\tau(\sigma(\vec{i}))$ from $\tau(E)$ to $\tau(M)$, corresponding to the disruption of the ECM element.

From a numerical point of view, Eq. (3.3) is solved with a finite element method employed on the CPM mesh underlying the spatial domain D, with 10 diffusion step for MCS to ensure stability.

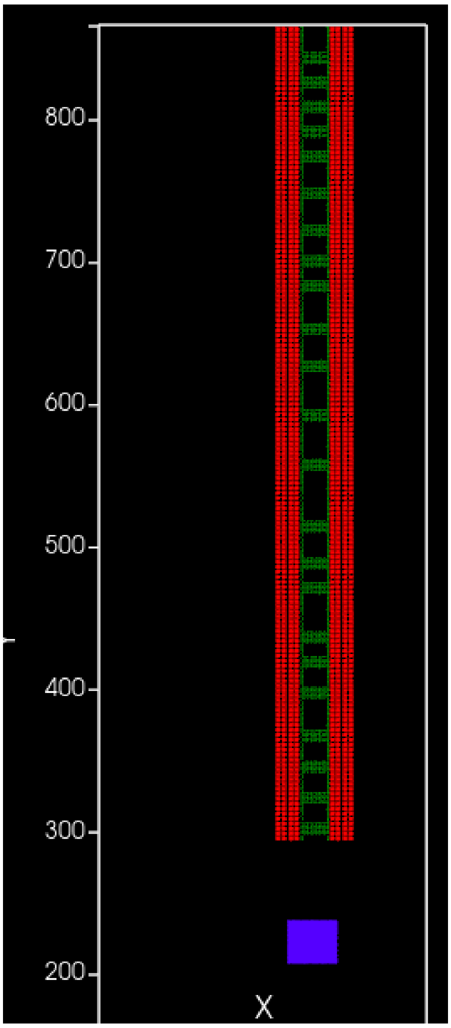


Figure 3.7: Initial configuration for setting 2: Tumor cells migrating through muscle fibers with ECM.

Simulation details and parameter setting. In the second setting S2, the simulation domain D and temporal resolution remain unchanged, but the structural composition of the extratumoral environment is modified to include an additional ECM component. A layer of ECM, with a thickness of 1 pixel, surrounds the inner side of each fiber along its entire length of 660 pixels. Randomly distributed ECM filaments connect the two fibers as well, maintaining their cohesion. These filaments have a thickness of 10 pixels and span the gap between the two myo-fiber structures. Initially there are no proteolytic enzymes throughout the domain, i.e., $h_{MMP}(\vec{i},0) = 0$ for any $\vec{i} \in D$.

Table 3.4 lists the parameters that are specific for this second scenario S2: the remaining model coefficients are exactly the same as those included in Table 3.1. In particular, we neglect the adhesive interactions between the structural ECM and the tumor individuals by setting $J_{T,E} = 0$. The analysis of the effect on cell migration of adhesive dynamics is in fact beyond the scope of this thesis. We then use a negative contact energy between muscle elements and ECM components to ensure that they remain attached over time.

The parameters relative to MMP kinetics are instead taken from the literature, see [9] and references therein.

Table 3.4: Model Parameters (II)

Parameter	Description	Value	Reference(s)
Tumor cell			
$J_{T,E}$	adhesion energy between ECM and tumor cell	0	estimated
Muscle fibers			
$J_{F,E}$	adhesion energy between ECM and muscle fibers	-2	estimated
MMPs			
D_{MMP}	diffusion rate of MMPs	0.2	[9]
k_{MMP}	degradation rate of MMPs	0.1	[9]
θ_{MMP}	MMPs production rate	0.3	[9]
h_{deg}	MMP threshold for ECM degradation	2.5	[9]

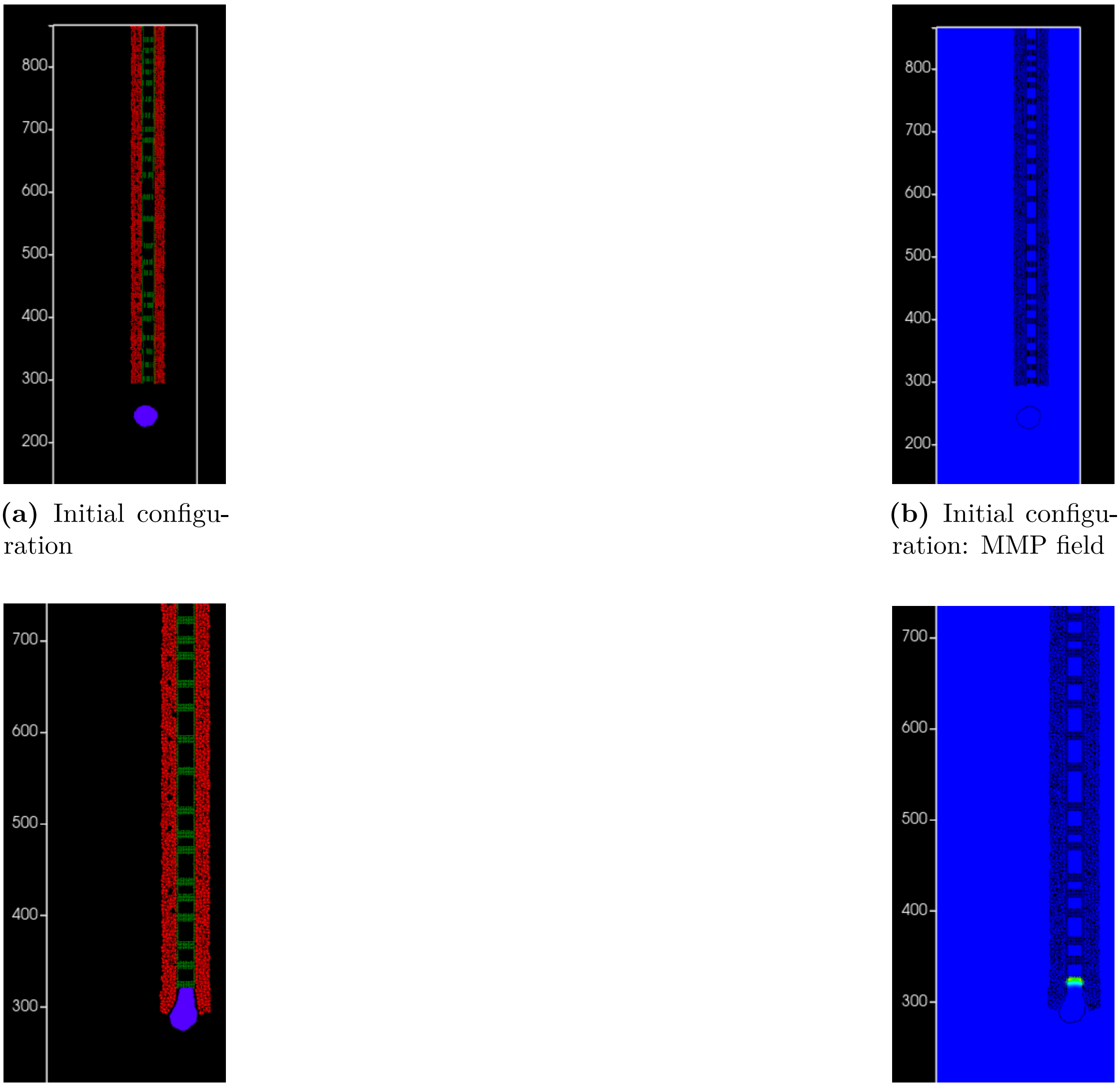
3.2.2 S2: Numerical results

Reference simulation. As shown in the time-lapse images in Fig.3.8 and 3.9, the cell approaches the entrance of the muscle structure whose overall rigidity is due to the presence of the ECM. The structural matrix then starts to be degraded by the tumor-derived MMPs: as a consequence, muscle elements are free to fluctuate around their position.

The malignant individual indeed remodels in a polarized elliptic morphology and initiates to penetrate the inter-fiber space, that gradually enlarges due ECM digestion.

Once the tumor cell has infiltrated its body within the muscle structure, it starts a sustained locomotion which is permitted by the above-described coordinated mechanisms: matrix degradation and fiber widening.

The overall migrations takes to the cell approximately 47 000 MCS, a larger value w.r.t. the reference simulation of S1 since the proteolytic enzymes need time to digest elements of the structural matrix.



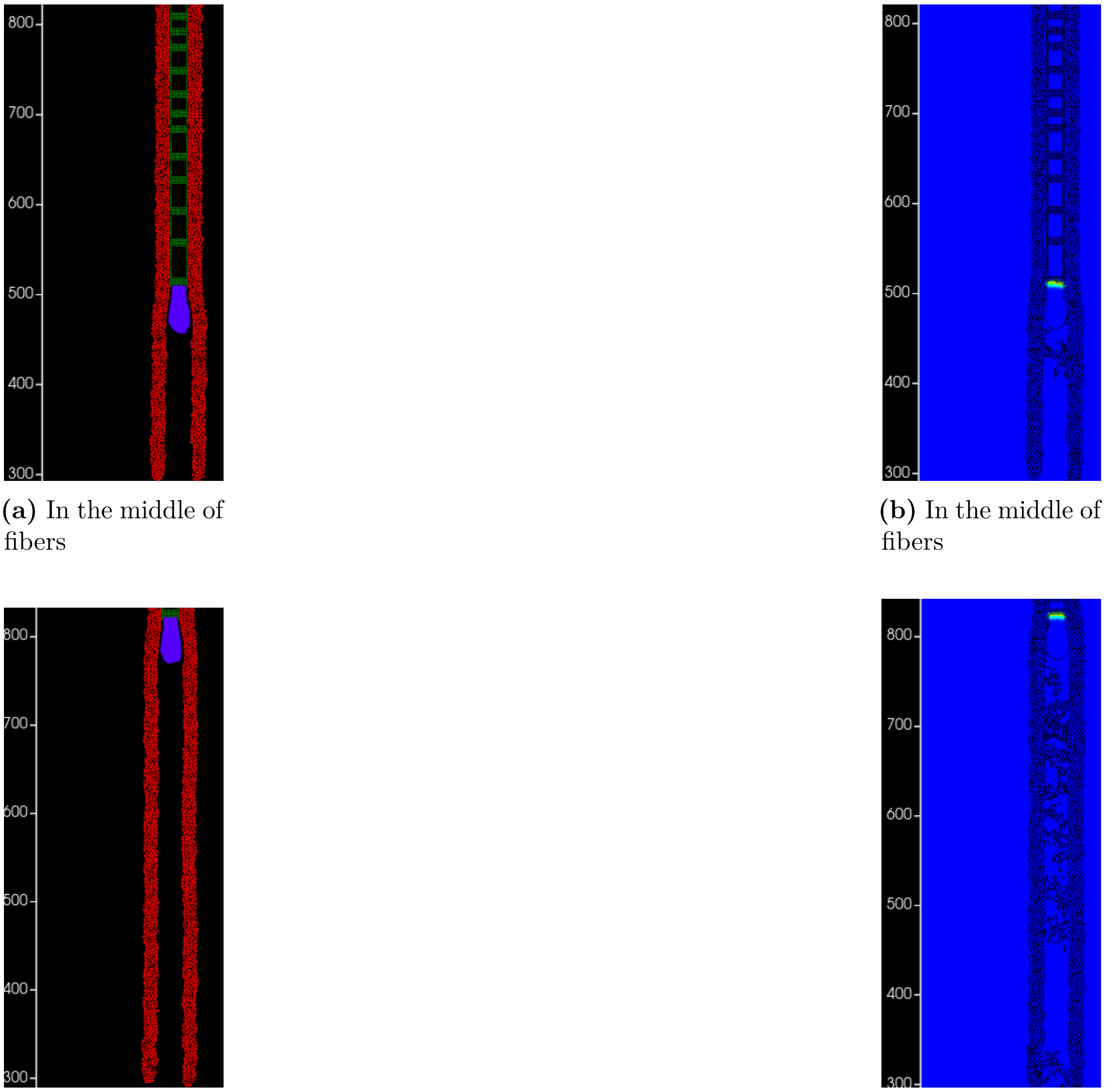
(a) Initial configuration

(b) Initial configuration: MMP field

(c) First contact between the tumor cell and the fibers

(d) The tumor cell begins producing MMPs

Figure 3.8: Time evolution of tumor cell migration and the MMPs field (Part 1)



(a) In the middle of fibers

(b) In the middle of fibers

(c) The tumor cell reached the top border of the domain

(d) The tumor cell reached the top border of the domain

Figure 3.9: Time evolution of tumor cell migration and the MMPs field (Part 2)

Varying MMP secretion rate. As seen, the tumor cell is able to move within the muscle tissue deforming its overall body and activating proteolytic enzymes able to degrade the structural ECM. In this respect, we now evaluate the relationship between the amount of MMPs secreted by the malignant individual and the extent of its directional movement.

In this respect, Table 3.5 summarizes the time required for the tumor cell to reach the bottom of the domain for different MMPs secretion rates. The corresponding trend is also displayed in the plot in Figure 3.10.

MMP Secretion Rate	migration time (MCS)
0.30	47000
0.25	50000
0.20	60000
0.17	100000
0.15	600000
0.10	no infiltration

Table 3.5: Time (in MCS) required for the tumor cell to migrate through the ECM for different MMPs secretion rates.

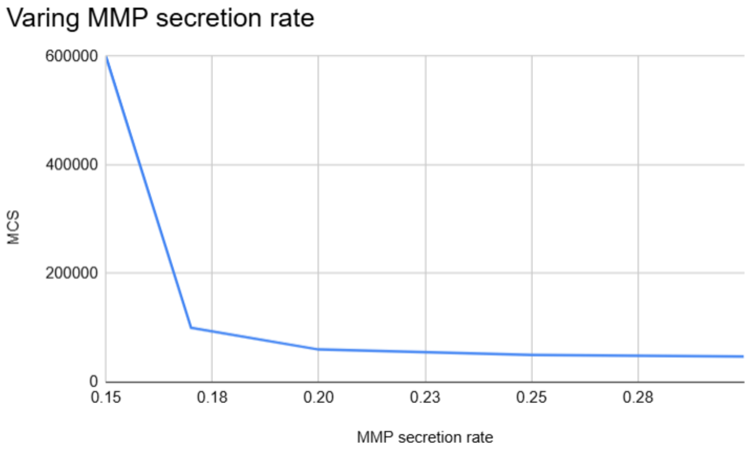


Figure 3.10: Trend of tumor cell migration time as a function of MMPs secretion rate. The increase in time suggests a nonlinear relationship, with a critical threshold below which migration is blocked.

The results appear consistent with the hypothesis that MMPs secretion significantly influences tumor cell migration speed. Several key observations can be made: for high enough secretion rates (i.e., > 0.20) the time required by the cell to reach the end of the muscle is almost constant. It possibly depends on the time needed by both the cell and the muscle elements to deform and adapt their shapes.

Lower values of MMPs secretion rate result in higher values of the cell migration time: they in fact imply a slow down of ECM degradation and therefore a delay in tumor cell infiltration and subsequent locomotion.

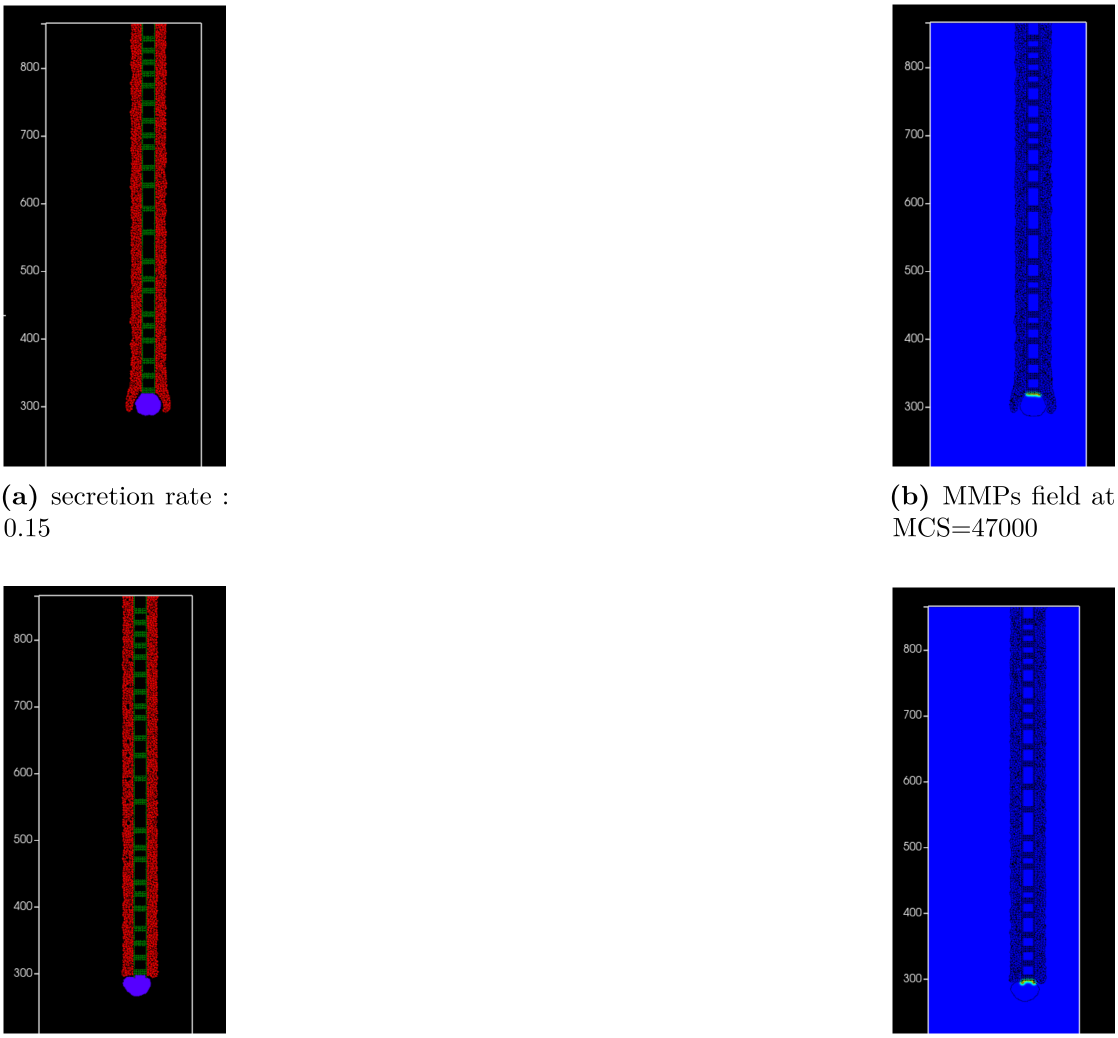
At substantially low values of MMPs secretion rates (i.e., < 0.10) the tumor cell is unable to penetrate and move through the muscle structure within the simulated timeframe. The underlying rationale is that in such a range of values of MMPs production rate ECM degradation is too slow to allow cell migration within biologically relevant timescales.

The above-findings provide insight into the role of ECM degradation in tumor invasion, reinforcing the importance of proteolytic enzyme activity in facilitating metastatic progression.

To better illustrate the impact of MMPs secretion on tumor cell migration, we initially considered including simulation snapshots for a range of secretion rates. However, our analysis revealed that for secretion rates above 0.15, the tumor cell attains similar positions at the same simulation time (MCS), remaining close to its initial configuration S2. As a result, including these cases would be redundant.

Therefore, we have chosen to present snapshots for secretion rates of 0.15 and 0.10, captured at the moment when the tumor cell, in S2, encounters the upper boundary of the domain at MCS = 47,000. This selection effectively highlights the differences in migratory behavior: with a secretion rate of 0.15, the cell continues its infiltration with a noticeably delayed progression, whereas at 0.10, the cell is unable to penetrate the muscle tissue at all.

Analogous outcomes would be obtained by keeping fixed MMPs secretion rates, while increasing the MMPs concentration threshold needed to degrade a matrix component.



(a) secretion rate : 0.15

(b) MMPs field at MCS=47000

(c) secretion rate : 0.1

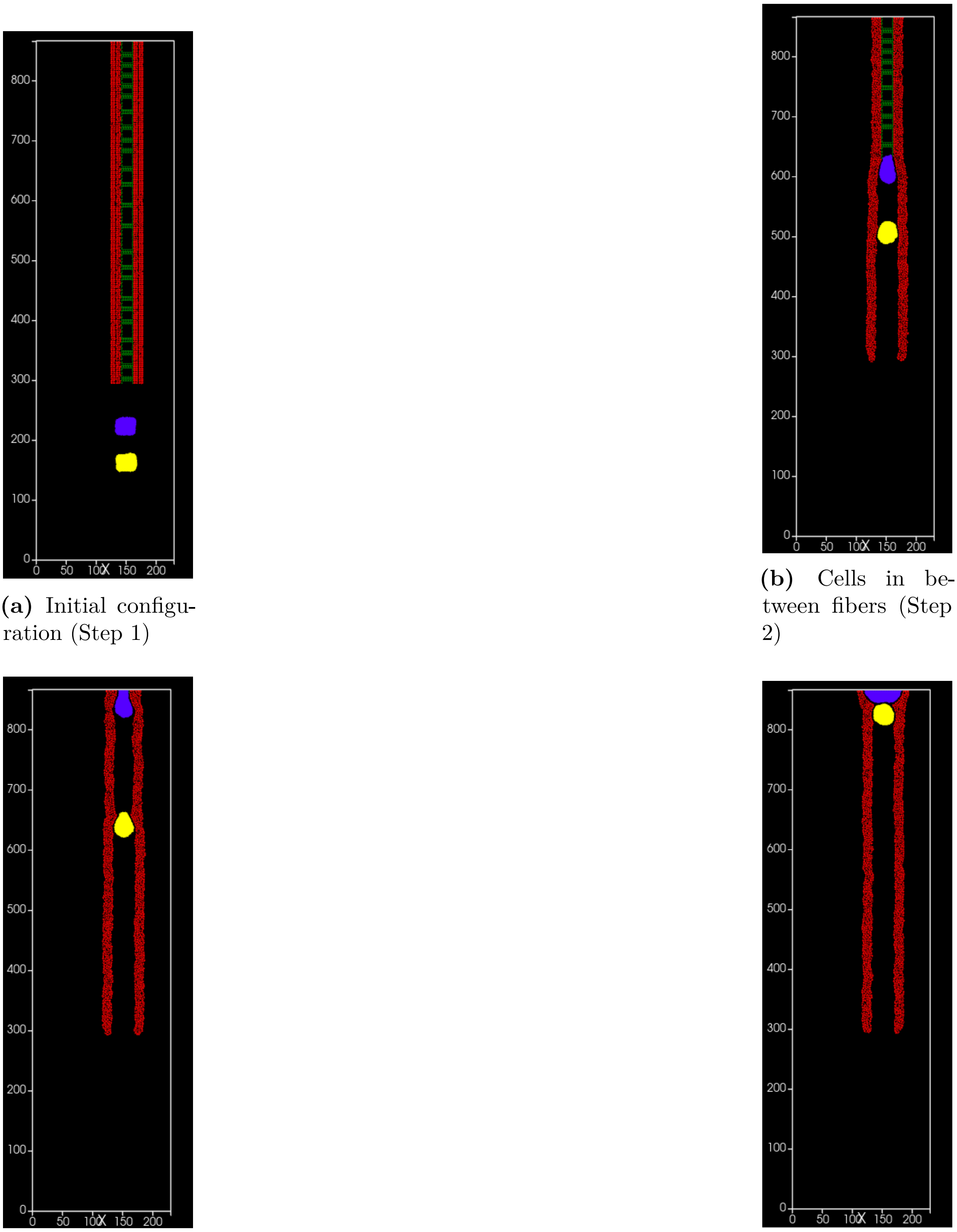
(d) MMPs field at MCS=47000

Figure 3.11: Time evolution of tumor cell migration and the MMPs field varing secretion rate

Modeling collective and coordinated migration. We finally keep the numerical setting of the reference simulation while adding a second tumor cell. It is characterized by the same biophysical properties of the other malignant individual except from the fact that it has an inhibited proteolytic activity, i.e., it is not allowed to produce MMPs.

In the simulation, the cell producing MMPs is represented in blue, while the one lacking proteolytic activity is colored yellow.

The different steps of the simulation have been documented through images, allowing a visual analysis of the collective migration dynamics. As it can be seen, the leading blue cell, capable of degrading the ECM, is able to permit the movement of the following cell, which instead lacks proteolytic activity. This visual representation helps assess the interaction between the two tumor cells and the efficiency of their coordinated invasion.



(a) Initial configuration (Step 1)

(b) Cells in between fibers (Step 2)

(c) Leading cell reached the bottom (Step 3)

(d) Both cells reached the bottom (Step 4)

Figure 3.12: Time evolution of tumor cells migration. The first cell actively degrades the ECM, facilitating the movement of the second one.

Chapter 4

Conclusions and Future Perspectives

This thesis explored the migration of a tumor cell through two muscle fibers using the Cellular Potts Model. The primary objective was to analyze how different parameters affect the cell's ability to traverse the fibers, identifying the key factors that influence the migration process. Two main simulation settings were considered: one without an extracellular matrix (ECM) and one with ECM.

In the absence of ECM, two types of simulations were performed. The first examined the role of cell elasticity by keeping the muscle fibers fixed while varying the cell's surface elasticity parameter across a range of values. The results indicated that higher values of the parameter, i.e., the cell was more rigid, led to longer migration times. The second simulation focused on fiber rigidity, maintaining a constant cell elasticity while varying the stiffness of the muscle fibers. The findings revealed that increased fiber rigidity significantly prolonged migration time, confirming that stiffer fibers act as stronger physical barriers to movement.

In the ECM-inclusive setting, two additional simulations were conducted. The third simulation investigated the effect of proteolytic enzyme secretion by varying the secretion rate of matrix metalloproteinases (MMPs). The results showed that lower MMPs secretion rates drastically increased migration time, with complete blockage occurring at the lowest tested rate. The fourth simulation explored collective migration by introducing a second cell, incapable of secreting MMPs, behind the leading cell. The results highlight the crucial role of MMPs in facilitating invasion. This simulation was designed to illustrate tumor cell behavior in the presence of multiple cells.

Overall, the study identified cell elasticity, fiber rigidity, and ECM degradation as key determinants of migration dynamics. These findings contribute to a deeper understanding of cell movement in constrained environments, with potential implications for tissue remodeling, cancer cell invasion, and related biological processes.

Future research could expand on this work in several directions:

- **Alternative Migration Scenarios**: Investigating tumor cell migration through adipocytes to compare movement dynamics in different tissue environments.
- **Collective Migration Mechanisms**: Expanding the model to simulate larger groups of tumor cells, allowing for a more comprehensive study of cooperative dynamics and collective invasion strategies.
- **Heterogeneous Cellular Properties**: Incorporating biomechanical differentiation between nuclear (more rigid) and cytoplasmic (more elastic) regions of tumor cells for increased model accuracy.
- **3D Simulations**: Extending the model to three dimensions in order to capture more realistic spatial constraints and cell-cell interactions.
- **Integration of Blood Vessels**: Introducing vascular structures to simulate tumor cell intravasation and explore the transition from tissue invasion to metastatic dissemination.

Appendix A

CompuCell3D

CompuCell3D (CC3D) is an open-source simulation environment, which is built on the CPM. CC3D enables fast and intuitive modeling and simulation of both individual cellular behaviors and multi-cellular dynamics, particularly in the context of tissue formation and development.

Specifically, it captures various cell behaviors such as growth, division, adhesion, and migration, making it suited for studying complex biological phenomena like morphogenesis, tumor growth, and tissue regeneration.

The CC3D environment includes several important tools that facilitate model development, execution, and analysis:

- **Twedit++-CC3D**: A dedicated model editor and code generator. Twedit++ includes a Simulation Wizard that generates draft CC3D model code based on high-level specifications, such as cell types, fields, and interactions. Users currently need to adjust the default parameters of the generated code. Twedit++ also offers a Python code-snippet generator, simplifying the process of writing custom Python scripts for CC3D models.

- **CellDraw**: A graphical tool that helps users define the initial configuration of the cell lattice. This tool is particularly useful for setting up simulations that require precise initial conditions, such as tissue structures or patterns of cell distributions.

- **CC3D Player**: The graphical interface used to run, replay, and analyze CC3D simulations. It supports steering, allowing users to adjust model parameters in real-time as the simulation is running. CC3D Player also provides multiple visualization options and supports batch-mode execution for high-throughput simulations.

CC3D uses a modular architecture, which means that only the necessary components for a particular model are loaded, optimizing performance.The structure of a CC3D model typically consists of:

- **CC3DML scripts**: these are XML-based files that define the core parameters of the simulation, such as the lattice dimensions, cell types, biological mechanisms, and file paths for configuration. CC3DML are static, which means that, throughout the simulation, those parameters remain unchanged.
- **Python scripts**: Python scripting allows building complex simulations wherein the behaviors of individual cells change (according to user specification) as the simulation progresses. For instance, in a tumor growth model, Python scripts can adjust the type of a cell based on the oxygen partial pressure.
- **Initialization files:** These files define the initial configuration of the cell lattice, they are called Pixel Initialization Files (PIFFs). A PIFF is a text file that allows users to assign multiple rectangular (parallelepiped in 3D) pixel regions or single pixels to particular cells.

The system includes several key modules:

- **Plugins:** These modules calculate the effective energy terms and monitoring events on the cell lattice. They are invoked frequently during each pixel copy attempt. Due to their frequent use, most plugins are written in C++ for improved computational speed.
- **Steppables:** These modules perform operations at the cellular level rather than the pixel level, and are called at fixed intervals (measured in Monte Carlo steps). Steppables have a variety of uses, including adjusting cell parameters based on simulation events, solving PDEs, and loading or saving simulation results. Most steppables are implemented in Python, allowing for significant customization by users.

In addition to Python and CC3DML scripts, CompuCell3D also uses C++ for modules that require high computational efficiency. While Python provides flexibility for customizing cell behaviors and handling dynamic aspects of the simulation, C++ is employed for operations that are computationally intensive, such as calculating energy terms or handling frequent pixel-copy events. Most plugins, which are invoked repeatedly during the simulation, are written in C++ to ensure optimal performance, making the system capable of handling large-scale and complex models efficiently [10].
CC3D also supports parallel computation through OpenMP, enabling efficient execution on multi-core machines and providing substantial speed improvements in large-scale simulations [6] [5].

Appendix B

S1 code

In this simulation scenario, only the XML file is reported, as it is the key component for the setup and configuration of the simulation environment. The XML file contains the specifications for the Potts model, cell types, energy parameters, and regions of interest for the simulation.

File xml

Listing B.1: XML Configuration for S1 Simulation

```
 <CompuCell3D>
 <!— Python script controlling the simulation —>
 <PythonScript>C:\CompuCell3D\MyDemos\sim_fineNov\sim_gen\
Simulation\provaSteppables.py</PythonScript>

 <Potts>
     <!— Lattice and simulation settings —>
     <Dimensions x="230" y="1000" z="1"/> <!— 2D simulation with
large y—dimension —>
     <LatticeType>Hexagonal</LatticeType> <!— Using a hexagonal
lattice —>
     <Steps>100000000</Steps>

     <!— Cell motility parameters —>
     <CellMotility>
         <MotilityParameters CellType="cell" Motility="20"/> <!—
High motility for tumor cells —>
         <MotilityParameters CellType="muscle" Motility="4"/> <!—
 Lower motility for muscle fibers —>
     </CellMotility>

     <NeighborOrder>2</NeighborOrder> <!— Considering second—
order neighbors —>
 </Potts>

```

```
    <!-- Tracks neighboring cells -->
    <Plugin Name="NeighborTracker"/>

    <!-- Defining cell types -->
    <Plugin Name="CellType">
        <CellType TypeName="Medium" TypeId="0"/> <!-- Background -->
        <CellType TypeName="cell" TypeId="1"/> <!-- Tumor cell -->
        <CellType TypeName="muscle" TypeId="2"/> <!-- Muscle fiber -->
    </Plugin>

    <!-- Volume constraints to maintain realistic cell sizes -->
    <Plugin Name="Volume">
        <VolumeEnergyParameters CellType="cell" TargetVolume="1200" LambdaVolume="40"/>
        <VolumeEnergyParameters CellType="muscle" TargetVolume="16" LambdaVolume="30"/>
    </Plugin>

    <!-- Surface constraints to control cell morphology -->
    <Plugin Name="Surface">
        <VolumeEnergyParameters CellType="cell" TargetSurface="120" LambdaSurface="5"/> <!-- Surface tension for tumor cells -->
        <VolumeEnergyParameters CellType="muscle" TargetSurface="16" LambdaSurface="2"/> <!-- Lower surface constraint for muscle fibers -->
    </Plugin>

    <!-- Contact energy between different cell types -->
    <Plugin Name="Contact">
        <Energy Type1="Medium" Type2="Medium">0</Energy>
        <Energy Type1="cell" Type2="Medium">15</Energy> <!-- Tumor cells weakly interact with medium -->
        <Energy Type1="cell" Type2="cell">250</Energy> <!-- High adhesion between tumor cells -->
        <Energy Type1="muscle" Type2="muscle">0</Energy>
        <Energy Type1="Medium" Type2="muscle">2</Energy>
        <Energy Type1="cell" Type2="muscle">40</Energy> <!-- Moderate interaction between tumor and muscle -->
        <NeighborOrder>3</NeighborOrder> <!-- Extends interaction range -->
    </Plugin>

    <!-- External force applied to tumor cells -->
    <Plugin Name="ExternalPotential">
        <ExternalPotentialParameters CellType="cell" x="0" y="-6" z="0"/> <!-- Downward force applied to tumor cells -->
    </Plugin>

    <!-- Initializing muscle fibers in structured vertical regions -->
    <Steppable Type="UniformInitializer">
        <Region>
            <BoxMin x="124" y="340" z="0"/>
            <BoxMax x="132" y="1000" z="1"/>
            <Gap>0</Gap>
            <Width>4</Width>
            <Types>muscle</Types>
        </Region>
        <Region>
            <BoxMin x="133" y="340" z="0"/>
            <BoxMax x="141" y="1000" z="1"/>
            <Gap>0</Gap>
            <Width>4</Width>
            <Types>muscle</Types>
        </Region>
        <Region>
            <BoxMin x="162" y="340" z="0"/>
            <BoxMax x="170" y="1000" z="1"/>
            <Gap>0</Gap>
            <Width>4</Width>
            <Types>muscle</Types>
        </Region>
        <Region>
            <BoxMin x="171" y="340" z="0"/>
            <BoxMax x="179" y="1000" z="1"/>
            <Gap>0</Gap>
            <Width>4</Width>
            <Types>muscle</Types>
        </Region>

        <!-- Initializing tumor cells in a compact region at the bottom -->
        <Region>
            <BoxMin x="132" y="240" z="0"/>
            <BoxMax x="168" y="280" z="1"/>
            <Gap>4</Gap>
            <Width>36</Width>
            <Types>cell</Types>
        </Region>
    </Steppable>
</CompuCell3D>
```

Appendix C

S2 code

This appendix presents the key components of the code used for simulation S2. The simulation is performed using the CompuCell3D software, and the following files are essential for its execution:
XML File: This file contains the initial configuration of the simulation, defining the parameters related to cells, the extracellular matrix (ECM), and other simulated objects. Specifically, the XML sets the initial conditions for tumor cells, muscle fibers, and the ECM.
Steppable File: The Steppable file describes the dynamic behavior of the simulation. It contains the rules governing the interaction between tumor cells and the ECM, including the ability to modify the ECM as the simulation progresses.
Python File: The Python file manages the execution of the simulation, interacting with the XML and Steppable files. It controls the simulation's life cycle, the execution of time steps, and the eventual collection and analysis of data.
Below are the details of each file:

Listing C.1: XML Configuration for S2 Simulation

```xml
<CompuCell3D>

  <!-- Path to the Python script responsible for controlling the simulation -->
  <PythonScript>C:\CompuCell3D\MyDemos\sim_fineNov\sim_gen\Simulation\provaSteppables.py</PythonScript>

  <!-- Define simulation lattice dimensions and properties -->
  <Potts>
    <Dimensions x="230" y="1000" z="1"/> <!-- Lattice dimensions -->
    <LatticeType>Hexagonal</LatticeType> <!-- Hexagonal lattice type -->
    <Steps>70000</Steps> <!-- Number of simulation steps -->

    <!-- Define cell motility parameters -->
    <CellMotility>
        <MotilityParameters CellType="cell" Motility="20"/>  <!-- Tumor cell motility -->
        <MotilityParameters CellType="cell1" Motility="20"/> <!-- Tumor cell motility , Modeling collective and coordinated migration -->
        <MotilityParameters CellType="muscle" Motility="4"/> <!-- Muscle cell motility -->
    </CellMotility>

    <NeighborOrder>2</NeighborOrder> <!-- Neighbor interaction order -->
  </Potts>

  <!-- Plugin for neighbor tracking -->
  <Plugin Name="NeighborTracker"/>

  <!-- Define cell types in the simulation -->
  <Plugin Name="CellType">
    <CellType TypeName="Medium" TypeId="0"/>
    <CellType TypeName="cell" TypeId="1"/> <!-- Tumor cell -->
    <CellType TypeName="muscle" TypeId="2"/> <!-- Muscle fiber -->
    <CellType Freeze="" TypeName="ecm" TypeId="3"/> <!-- Extracellular Matrix -->
    <CellType TypeName="cell1" TypeId="4"/> <!-- Another type of tumor cell , Modeling collective and coordinated migration -->
  </Plugin>

  <!-- Define volume energy parameters for cells -->
  <Plugin Name="Volume">
    <VolumeEnergyParameters CellType="cell" TargetVolume="1200" LambdaVolume="40"/>
    <!--Modeling collective and coordinated migration-->
    <VolumeEnergyParameters CellType="cell1" TargetVolume="1200" LambdaVolume="40"/>
    <VolumeEnergyParameters CellType="muscle" TargetVolume="16" LambdaVolume="30"/>
  </Plugin>

  <!-- Define surface energy parameters for cells -->
  <Plugin Name="Surface">
    <VolumeEnergyParameters CellType="cell" TargetSurface="120" LambdaSurface="30"/>
    <!--Modeling collective and coordinated migration-->
    <VolumeEnergyParameters CellType="cell1" TargetSurface="120" LambdaSurface="30"/>
    <VolumeEnergyParameters CellType="muscle" TargetSurface="16" LambdaSurface="2"/>
  </Plugin>
```

```

<!-- Define contact energy between different cell types -->
<Plugin Name="Contact">
  <Energy Type1="Medium" Type2="Medium">0</Energy>
  <Energy Type1="cell" Type2="Medium">15</Energy>
  <Energy Type1="cell1" Type2="Medium">15</Energy>
  <Energy Type1="cell" Type2="cell1">250</Energy>
  <Energy Type1="muscle" Type2="muscle">1</Energy>
  <Energy Type1="Medium" Type2="muscle">2</Energy>
  <Energy Type1="cell" Type2="muscle">40</Energy>
  <Energy Type1="cell1" Type2="muscle">40</Energy>
  <Energy Type1="muscle" Type2="ecm">-2</Energy>
  <NeighborOrder>3</NeighborOrder> <!-- Neighbor interaction order for contact energy -->
</Plugin>

<!-- Define external potential applied to tumor cells -->
<Plugin Name="ExternalPotential">
  <ExternalPotentialParameters CellType="cell" x="0" y="-6" z="0"/>
  <!--Modeling collective and coordinated migration-->
  <ExternalPotentialParameters CellType="cell1" x="0" y="-6" z="0"/>
</Plugin>

<!-- Steppable configuration for simulation -->
<Steppable Type="FlexibleDiffusionSolverFE">
  <!-- Specification of PDE solvers for MMP diffusion -->
  <DiffusionField Name="MMP">
    <DiffusionData>
      <FieldName>MMP</FieldName>
      <DiffusionConstant>0.02</DiffusionConstant> <!-- Diffusion constant for MMP -->
      <DecayConstant>0.01</DecayConstant> <!-- Decay constant for MMP -->
    </DiffusionData>
    <SecretionData>
      <SecretionOnContact Type="cell" SecreteOnContactWith="ecm">0.1</SecretionOnContact> <!-- Tumor cell secretion on ECM -->
    </SecretionData>
  </DiffusionField>
</Steppable>

<!-- Initial configuration of regions and cell types in the simulation -->
<Steppable Type="UniformInitializer">
  <!-- Define muscle cell regions -->
  <Region>
    <BoxMin x="124" y="340" z="0"/>
    <BoxMax x="132" y="1000" z="1"/>
```

```
      <Gap>0</Gap>
      <Width>4</Width>
      <Types>muscle</Types>
    </Region>
    <!-- Additional muscle and cell regions -->
    <!-- (more region definitions here) -->

    <!-- Define ECM regions -->
    <Region>
      <BoxMin x="142" y="340" z="0"/>
      <BoxMax x="161" y="350" z="1"/>
      <Gap>0</Gap>
      <Width>2</Width>
      <Types>ecm</Types>
    </Region>
    <!-- Additional ECM regions -->
    <!-- (more region definitions here) -->

  </Steppable>

</CompuCell3D>
```

Listing C.2: Python Steppable for S2 Simulation

```
from cc3d.core.PySteppables import *
from cc3d.core.XMLUtils import CC3DXMLListPy
import numpy as np
import sys

# Define the SwapCellSteppable class, which extends SteppableBasePy
class SwapCellSteppable(SteppableBasePy):

    # Initialize the frequency at which the steppable executes
    def __init__(self, frequency=1):
        super().__init__(frequency)

    def start(self):
        # Initialize variables that depend on the simulator
        self.cell_inventory = self.simulator.getPotts().getCellInventory()  # Get the cell inventory
        self.cell_list = CellList(self.cell_inventory)  # Create a cell list
        self.cell_field = self.simulator.getPotts().getCellFieldG()  # Get the cell field
        self.dimensions = self.cell_field.getDim()  # Get the dimensions of the cell field

    def setFieldName(self, _fieldName):
        # Set the name of the concentration field to use in the steppable
        self.fieldName = _fieldName

    def step(self, mcs):
        # Function that is called at each step of the simulation (MCS)
        fileName = self.fieldName + "_" + str(mcs) + ".txt"  # Define the output file name
        self.outputField(self.fieldName, fileName)  # Call the output function

    def outputField(self, _fieldName, _fileName):
        # Function that writes the concentration field values to a file
        field = CompuCell.getConcentrationField(self.simulator, _fieldName)  # Get the concentration field
        pt = CompuCell.Point3D()  # 3D point to iterate over the field
        if field:
            try:
                # Open the file for writing
                with open(_fileName, "w") as fileHandle:
                    # Iterate over all the cells in the field
                    for i in range(self.dimensions.x):
                        for j in range(self.dimensions.y):
                            for k in range(self.dimensions.z):
                                pt.x = i
                                pt.y = j
                                pt.z = k
                                value = field.get(pt)  # Get the concentration value of the cell
                                # Check the adjacent cell (if it's ECM)
                                neighborCell = self.cell_field.get(pt)
                                if neighborCell and neighborCell.type == 3:  # Only ECM
                                    # If the concentration of MMP is above a threshold, degrade ECM
                                    if value > 2.5:
                                        neighborCell.type = 0  # Change ECM to Medium
            except IOError:
                # Handle error if the file can't be opened
                print("Could not open file for writing. Check if you have necessary permissions.")
```

Listing C.3: Python Configuration for S2 Simulation

```python
import sys
import os
from os import environ
from os import getcwd
import string

from provaSteppables import SwapCellSteppable  # Import the defined Steppable

# Create an instance of the Steppable with a frequency of 10 steps
swap_cell_steppable = SwapCellSteppable(frequency=10)

# Set the name of the concentration field (in this case MMP)
swap_cell_steppable.setFieldName("MMP")

# Register the steppable with CompuCellSetup
CompuCellSetup.register_steppable(steppable=swap_cell_steppable)

# Run the simulation
CompuCellSetup.run()
```

Bibliography

[1] Peter Friedl and Katarina Wolf. «Title of the Unpublished Article». Experiments conducted by the group of Peter Friedl and Katarina Wolf at Radboud University Medical Center and MD Anderson Cancer Center. 2025 (cit. on pp. 2, 13).

[2] Tsuyoshi Hirashima, Elisabeth G. Rens, and Roeland M. H. Merks. «Cellular Potts modeling of complex multicellular behaviors in tissue morphogenesis». In: *Development, Growth & Differentiation* 59.5 (2017), pp. 329–339. DOI: `https://doi.org/10.1111/dgd.12358` (cit. on p. 3).

[3] James A. Glazier, Ariel Balter, and Nikodem J. Pop. «Magnetization to Morphogenesis: A Brief History of the Glazier–Graner–Hogeweg Model». In: *The Cellular Potts Model*. Elsevier, 2021, pp. 7–40 (cit. on pp. 3, 4).

[4] Elisabeth G. Rens and Leah Edelstein-Keshet. «From energy to cellular forces in the Cellular Potts Model: An algorithmic approach». In: *PLoS Computational Biology* 15.12 (2019). DOI: `10.1371/journal.pcbi.1007459` (cit. on p. 3).

[5] Maciej H. Swat, Julio Belmonte, Randy W. Heiland, Benjamin L. Zaitlen, James A. Glazier, and Abbas Shirinifard. *Introduction to CompuCell3D*. Version 3.6.2. Biocomplexity Institute and Department of Physics, Indiana University. Bloomington, IN, USA, 2015 (cit. on pp. 6, 34).

[6] Maciej H Swat, Gilberto L Thomas, Julio M Belmonte, Abbas Shirinifard, Dimitrij Hmeljak, and James A Glazier. «Multi-Scale Modeling of Tissues Using CompuCell3D». In: *Methods in Cell Biology* 110 (2012), pp. 325–366. DOI: `10.1016/B978-0-12-388403-9.00013-8` (cit. on pp. 6, 34).

[7] A. Voss-Böhme. «Multi-Scale Modeling in Morphogenesis: A Critical Analysis of the Cellular Potts Model». In: *PLoS ONE* 7.9 (2012). DOI: `10.1371/journal.pone.0042852`. URL: `https://doi.org/10.1371/journal.pone.0042852` (cit. on p. 7).

[8] Athanasius F.M. Marée, Veronica S. Grieneisen, and Paulien Hogeweg. «The Cellular Potts Model and Biophysical Properties of Cells, Tissues, and Morphogenesis». In: *Single-cell-based models in biology and medicine*. Ed. by Andreas Deutsch and Sabine Dormann. Birkhäuser, 2011, pp. 107–136 (cit. on p. 7).

[9] C. Giverso, M. Scianna, L. Preziosi, N. Lo Buono, and A. Funaro. «Individual Cell-Based Model for In-Vitro Mesothelial Invasion of Ovarian Cancer». In: *Mathematical Modelling of Natural Phenomena* 3.1 (2008), pp. 1–3 (cit. on p. 23).

[10] CompuCell3D Team. *CompuCell3D 4.5.0 Python Scripting Manual for (Python 3) Online Manual*. Retrieved from `https://compucell3d.org/Documentation`. 2024 (cit. on p. 34).