

A Mathematical Approach to Tissue Development

Nargess Khalilgharibi, CoMPLEX

Supervisors: Prof Mark Miodownik & Dr Jeremy Green

Total Word count: 8049

Main Body: 7806

(Excluding figure captions, displayed equations, acknowledgement, references and appendix)

August 23, 2012

In the past few years, an increasing number of models for tissue development have been published in the literature. While many of these models studied the features of a tissue with a restricted size, only a few focused on the properties of the growing tissue.

Directional growth of the tissue plays an important role in the development of the embryo and its organs. The model tissue in this research is mouse palatal epithelium. Recent findings have revealed that non-proliferative mechanisms should account for its anteroposterior growth [Unpublished data from the Green lab].

This research is an attempt to identify the amount of contribution of different growth mechanisms in directional growth of the tissue. A 2D vertex model of the growing tissue will be developed for this purpose. This model will then be used to find explanations for the role of different growth mechanisms, arising from anisotropic tension, in tissue elongation.

Contents

1	Introduction	1
2	Directional Growth of the Tissue: The Underlying Mechanisms	2
2.1	Intercalation	2
2.2	Oriented Cell Division	3
2.3	Cell Shape Changes	3
3	Models of Tissue Development	4
3.1	The Cellular Potts Model	4
3.2	The Vertex Model	5
3.3	Finite Element Models	6
4	Methods	7
4.1	Implementation of the Vertex Model	7
4.1.1	Representation of the Tissue	7
4.1.2	The Energy Function	7
4.1.3	Temporal Evolution of the Tissue	8
4.1.4	Cellular Growth and Division	8
4.1.5	Topological Changes: T1 and T2 Transitions	9
4.1.6	Setting Parameters	9
4.2	Initial Configuration of the Vertex Model	10
4.3	Quantifying the Growth of the Tissue	10
4.3.1	Distribution of Neighbour Numbers	10
4.3.2	Cell and Tissue Deformation	13
4.3.3	Growth Rates	13
5	Results	14
5.1	Growth Under Isotropic Tension	14
5.2	Growth Under Anisotropic Tension	15
5.3	Contribution of Cell Divisions to Tissue Elongation	20
6	Discussion and Conclusion	23
7	Acknowledgements	24
A	Appendix	25
A.1	The Code	25
A.1.1	The Vertex Model Code - Main Function	25
A.1.2	The Code for Changing the Configuration of the Tissue	26
A.1.3	The Code for Cell Divisions	28
A.1.4	The Code for T1 and T2 Transitions	31
A.1.5	The Code for Plotting Histograms of Cell Neighbour Numbers	36
A.1.6	The Code for Determining Cells' Ellipses of Inertia	39
A.1.7	The Code for Measuring Growth Rates of the Tissue	41
A.2	Snapshots of Simulations	43

1 Introduction

Epithelial tissue, also known as epithelium, is a structured layer of cells where cells are joined together, side to side, to form multicellular sheets [2]. It is one of the four major types of animal tissues, which appears in the lining of all body structures [1, 38]. Acting as a selective barrier, epithelial tissue protects the underlying cells from injury and damage, exchanges chemicals between them and the body cavity, and synthesises and releases secretion [26]. During embryonic development, the first tissue to emerge is epithelial tissue [29], which contributes to the shaping of the embryo and its organs [23].

Considering its various functions and responsibilities, the proper formation and development of epithelial tissue plays a crucial role throughout animal's life. For example, it has been proposed that cleft palate, one of the most popular craniofacial defects in newborns [32], could be the result of improper development of the palate medial-edge epithelium [31, 32]. In adults, about %90 of all cancers is *carcinomas*, which arises from abnormal growth of epithelial cells, leading to the formation of tumour tissue [1, 29]. Thus, understanding the mechanisms that direct the tissue development is of high importance, as it will help us to identify the causes of its abnormal growth.

The model tissue used in this study is mouse palatal epithelium, a flat tissue marked with 8 regularly spaced ridges, known as "rugae" [11] (figure 1). There are several reasons to choose this as the model tissue. First, the rugae can be used as landmarks on the tissue, making quantitative analysis of its growth easier. Second, flatness of this tissue enables us to use two-dimensional models to simulate its growth and development.

The studies have shown that rugae formation is a sequential process [32]. In fact, the first ruga to appear during embryonic development is ruga 8. Ruga 2 is the second ruga to appear. Except ruga 1 which appears out of order, all other rugae are formed between ruga 8 and their predecessor [11].

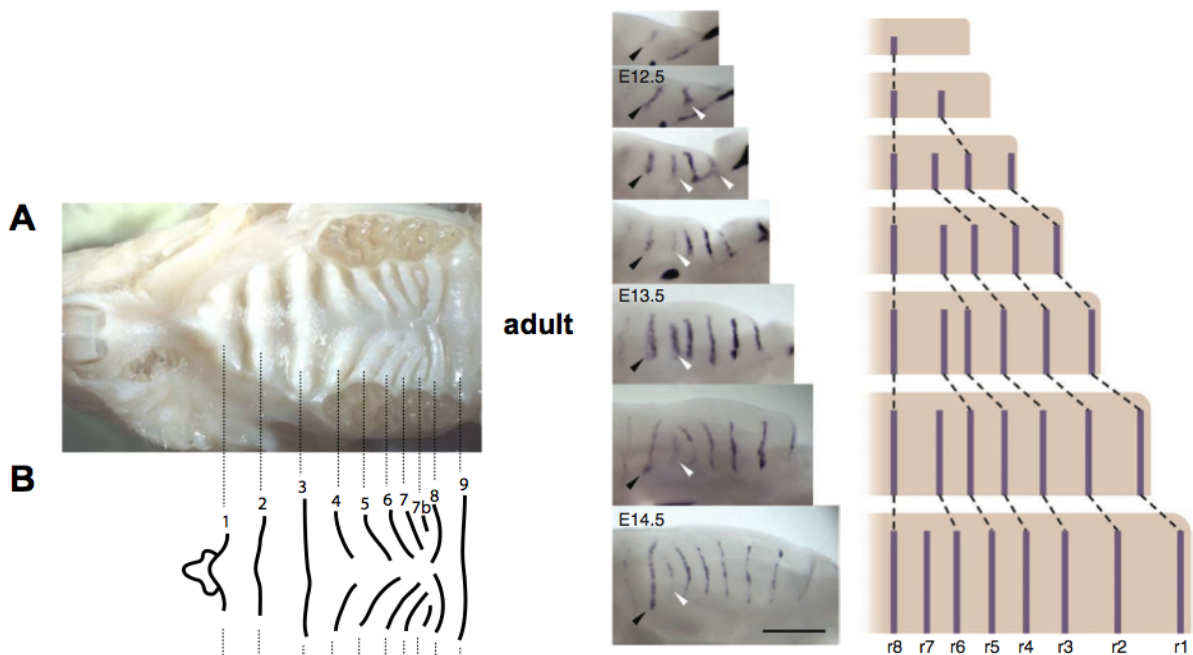


Figure 1: **(Left)** A. The palatal rugae on the roof of the oral cavity of an adult mouse. The pattern of these rugae is shown in B. Figure taken from [32] **(Right)** Sequential appearance of palatal rugae. The mid-palatal Rugae Growth Zone (RGZ) is the region anterior to ruga 8 (r8) where new rugae appear. Figure taken from [11].

A specific domain of the palate anterior to ruga 8, where sequential formation of rugae occurs, is known as mid-palatal Rugae Growth Zone (RGZ) [39]. The anteroposterior growth in RGZ moves the rugae

in this region apart from each other, while the spacing between more anterior rugae remains relatively constant [11]. Studies have been done to identify the relation between rugae formation and anteroposterior growth in this region (for more information, see [11, 32, 39]). However, less is known about the mechanisms which lead to the elevated growth in RGZ.

Different mechanisms can contribute to the growth in RGZ. The first possible explanation of the elevated growth in RGZ could be an increased rate of cell proliferation in this region. However, detailed analysis of cell proliferation shows that there is no difference between the rate of cell division in RGZ and other regions of the palate [32, 39] [Also unpublished data from the Green lab]. This suggests that non-proliferative mechanisms should be the main contributors to the non-uniform directional growth of the palate [Unpublished data from the Green lab].

In the following essay, the major mechanisms underlying directional growth of the tissue will be briefly explained (Section 2). Several analytical and computational models have been proposed to simulate tissue development. These models will be described in Section 3. A model will be developed in this study to simulate the growth of the tissue from a small patch of cells. Section 4 contains a detailed description of this model, as well as a description of the measures used to study the tissue growth. The developed model will then be used to study how different growth mechanisms can contribute to the non-uniform directional growth observed in the palatal tissue. The results of this study will be presented in section 5. The essay ends with a discussion section (Section 6), where the findings will be discussed and compared with previous works. Moreover, the limitations of the developed model will be explored and suggestions will be made for future work.

2 Directional Growth of the Tissue: The Underlying Mechanisms

Tissue elongation is an important step during embryogenesis and organogenesis [22]. Below the major mechanisms driving directional extension of the tissue will be explained. Recent findings of the role each of these mechanisms play in the elongation of mouse palatal tissue will also be represented.

2.1 Intercalation

Intercalation is one of the major mechanisms that gives rise to tissue elongation, by which cells are rearranged as they reorganise their junctions [24]. Cell intercalation has been extensively studied in *Drosophila* embryo, where it drives germband elongation [24, 35]. Recent findings from the Green lab suggest that cell intercalation (or more specifically, mediolateral cell intercalation) plays an important role in the anteroposterior growth observed in mouse palatal epithelium.

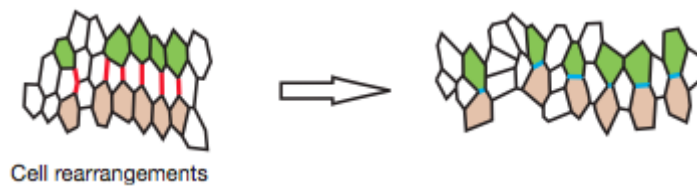


Figure 2: Cell rearrangements (intercalation) in horizontal direction caused by the shrinkage of vertical junctions (red) and formation of new horizontal junctions (blue). Figure taken from [24].

Figure 2 shows how cell intercalation can lead to extension of the tissue in a specific direction. In fact, the junctions oriented along the vertical axis shrink. This is followed by formation of new junctions perpendicular to the previous junctions, along the horizontal axis. It has been shown that shrinkage of junctions with a direction close to the horizontal axis (in the case of *Drosophila*, close to the dorsoventral axis) is not caused by "external forces exerted at the tissue boundaries" [24], but by local cortical tension arising from the enrichment of Myosin-II at these junctions [7, 35].

2.2 Oriented Cell Division

Oriented cell division is another major mechanism participating in tissue elongation [24]. Although being overlooked for many years, recent studies have revealed the major role of oriented cell divisions in directional growth of both plant and animal tissues. Examples of this include morphogenesis of *Antirrhinum* petal [3], zebrafish gastrula and neurula [8] and *Drosophila* embryonic epithelia [9].

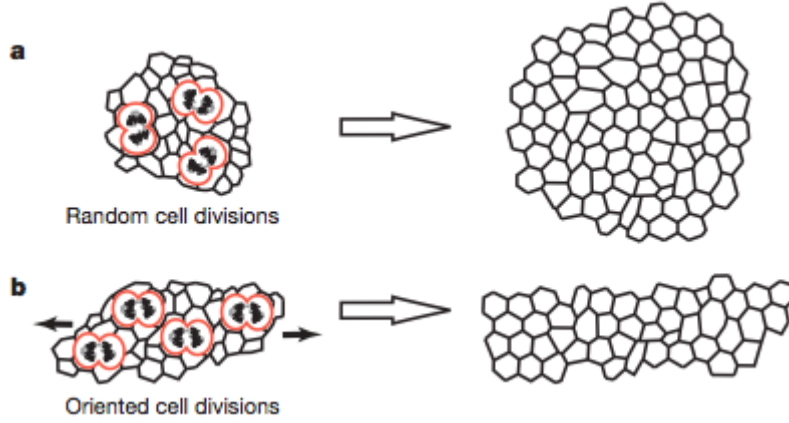


Figure 3: **(a)** Random cell divisions (outlined in red) lead to uniform growth of the tissue. **(b)** Oriented cell divisions lead to tissue elongation. Figure taken from [24].

Figure 3 shows how oriented cell division can give rise to tissue elongation. In fact, it is known that orientation of cell division is directed by mitotic spindles. In the absence of any bias in the orientation of cell divisions, the cell divisions will be in random directions, leading to uniform growth of the tissue. However, studies have shown that "systematic alignment of mitotic spindles in dividing populations could ... lead to significant tissue elongation" [40].

In mouse palatal epithelium, a strong bias has been observed in the orientation of mitotic spindles, suggesting the presence of oriented cell divisions in this tissue. However, as this specific orientation of mitotic spindles has been observed throughout the tissue and not just RGZ, it is suggested that this mechanism cannot account for the promoted directional growth of the tissue in RGZ [Unpublished data from the Green lab].

2.3 Cell Shape Changes

The shape of the cells in the tissue can be determined by a competition between adhesion and cortical tension [25]. How the interplay between these mechanical forces can lead to different topologies has been extensively studied in the literature [15, 13]. Although it has been proposed that changes in cell shapes contribute to tissue elongation [24], less is known about the details of this contribution. A computational model in [27] has explored the indirect effect of cell shape changes on tissue elongation. In fact, it been shown that polarised tension can lead to changes in cell shapes, which can then promote oriented cell divisions and lead to tissue extension. The direct contribution of cell shape changes to tissue elongation is still unclear and yet to be studied.

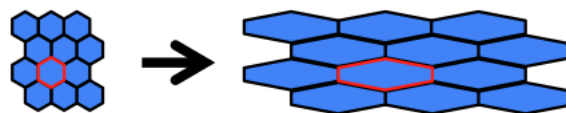


Figure 4: Effect of cell shape changes on tissue elongation. As it can be seen, lengthening of cells in a specific direction can give rise to effective tissue elongation. Figure courtesy of Andrew Economou.

Recent findings from the Green lab suggest that the changes in the shape of the cells partly contribute to the directional growth of the RGZ in mouse palate. In fact, it has been observed that cells in RGZ elongate in the anteroposterior direction, while cells in other regions do not. It has also been observed that the growth arising from cell shape changes cannot account for the significant tissue elongation, suggesting the involvement of other growth mechanisms in the growth of the RGZ [Unpublished data from the Green lab].

3 Models of Tissue Development

Modelling tissue development, both in animals and plants, has been the topic of many studies in the past 30 years. These models can be divided into two main groups: analytical and computational.

Many of the analytical models are focused on finding the transformation tensor that could describe the growth of the studied organ [18, 19, 10, 36]. In these studies, the transformation tensor is derived from measurements of the growth of the organ. For example, in studying the development of dorsal petal lobes of *Antirrhinum majus*, Lagan and colleagues [36] divide the growing structure into several segments and calculate the elements of the transformation tensor T_k^r from the experimental data. Given the shape of a segment at time $k - \Delta$, its shape at the next time step k can be calculated by T_k^r , where r represents the position of the segment and k represents the time interval $[k - \Delta, k]$. Thus, the changes in the shape of the organ can be calculated by continuously applying transformation T to its initial shape. Since analysing and measuring growth is easier in 2D flat tissues, such as leaves or petals, analytical models have been widely used for studying development of plant tissues [36].

Several 2D and 3D computational models have been proposed to study the development of animal tissues. Below, the main groups of these models are presented.

3.1 The Cellular Potts Model

The cellular Potts model is an extension to the large-Q Potts model, a model in foam physics that can successfully describe the dynamics of the grain growth driven by the surface energy (e.g. in soap froth) [17]. The large-Q Potts model is a lattice model, in which each lattice site, $\sigma_{(i,j)}$, has a spin value between 1 and N. Lattice sites with the same spin belong to the same cell. Therefore, the system is comprised of N cells. The Hamiltonian of the system is defined as:

$$H = \sum_{\text{neighbours}} 1 - \delta_{\sigma(i,j),\sigma(i',j')} \quad (1)$$

Given the initial state of the system, its temporal evolution can be computed as follows. At each time step, the state of a randomly chosen lattice site is changed from σ to σ' with Monte Carlo probability. To be more explicit, for temperatures above 0, if the change is energetically favourable, i.e. if $\Delta H < 0$, the change is accepted. Otherwise, it is accepted with probability $e^{-\frac{\Delta H}{k_B T}}$, where k_B is the Boltzmann constant.

In order to be applied to biological systems, the large-Q Potts model has been modified by Graner and Glazier [17]. In fact, they argued that biological systems differ from soap bubbles, as they have a constraint on their size (i.e. volume), whereas bubbles are free to expand [17]. This is implemented in the model by adding an "elastic-area constraint" to the Hamiltonian. In addition, a second "quantum number", τ , is introduced, which represents differential adhesion between different cell types. The Hamiltonian of the cellular Potts model is as below:

$$H = \sum_{\text{neighbours}} J(\tau(\sigma(i,j)), \tau(\sigma(i',j')))(1 - \delta_{\sigma(i,j),\sigma(i',j')}) + \lambda \sum_{\sigma} [a(\sigma) - A_{\tau(\sigma)}]^2 \theta(A_{\tau(\sigma)}) \quad (2)$$

In the first term, J describes the surface energy between two different cell types τ and τ' . The elastic-area constraint is implemented in the second term. In fact, λ is a Lagrange multiplier defining the strength of the elastic area constraint, a is the actual area of the cell and A_{τ} is the target area of the cells of type τ .

Adjustments have been made to the Hamiltonian of the cellular Potts model, in order to enable it to simulate the dynamics of tissue development [21, 33]. For example, Kafer and colleagues [21] study the shape of the cells in a *Drosophila* retinal ommatidium. They argue that the cell packing observed in the epithelium cannot be modelled only by adhesion-related tension, as in the original cellular Potts model. Therefore, they add a term to the Hamiltonian as below:

$$\lambda_P(P_i - P_{0i})^2 \quad (3)$$

where λ_P is the perimeter modulus, and P_i and P_{0i} are the actual and target perimeters of the cell i , respectively. This term accounts for the cell contractility, arising from contraction of actin cytoskeleton [12].

In another study, Poplawski et al [33] used a 3D cellular Potts model to simulate growth of the chick *Gallus gallus* limb bud. Similar to [21], they used an adjusted version of the Hamiltonian that included a term for cell contractility. In addition, diffusion of morphogen Fibroblast Growth Factor 8 (FGF8) was implemented in the model. FGF8 is a gene that encodes for a protein involved in various biological processes, including embryonic development and cellular growth [14]. Their research demonstrated that differential cell adhesion and diffusion of morphogens are required to simulate growth of the paddle shaped chick limb bud.

3.2 The Vertex Model

The vertex model is a model in statistical mechanics that has extensively been used to simulate epithelial packing and development [13, 27, 28, 35].

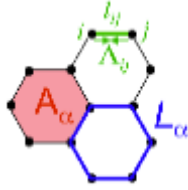


Figure 5: In the vertex model, cells are represented as polygons, with their sides being defined by vertices. Each cell has an area A_α and perimeter L_α . The length of the junction between vertex i and j is described by l_{ij} . Figure taken from [27].

It is comprised of a network of N_c cells, where the cell edges are shown by N_v vertices (figure 5). The general energy function proposed in [13] is as follows:

$$E(R_i) = \sum_{\alpha} \frac{K_{\alpha}}{2} (A_{\alpha} - A_{\alpha}^{(0)})^2 + \sum_{\langle i,j \rangle} \Lambda_{ij} l_{ij} + \sum_{\alpha} \frac{\Gamma_{\alpha}}{2} L_{\alpha}^2 \quad (4)$$

The first term accounts for the elastic-area constraint, also known as *area elasticity*. Here K_{α} is the elastic coefficient, A_{α} is the actual area of the cell and $A_{\alpha}^{(0)}$ is its target area. The second term represents the line tension between vertices, with Λ_{ij} being the line tension and l_{ij} the length of the junction between vertices i and j . This line tension is the sum of the adhesive forces between cells and contractile forces arising from the cell cortex [13, 27]. The mechanics of the actomyosin cytoskeleton is embedded in the third term, where cortical contractility is described by the cell perimeter L_{α} and a coefficient Γ_{α} .

The cells in the vertex model are able to grow, divide, exchange neighbours and die. The growth is implemented by gradually increasing the target area of the cells to twice their initial area. The cells reaching this threshold size enter the division phase [13, 27, 28].

Different approaches were taken to implement division in the model. For example, in [13], the cells are divided in a random direction through their centroid, while in [27], the cells are divided along their

short axis. In [28], a new boundary is introduced by randomly choosing a cell side and connecting its midpoint to the midpoint of the opposite side [30].

Neighbour exchange and cell death were enabled by introducing T1 and T2 transitions in the model. Figure 6 shows the steps of these processes.

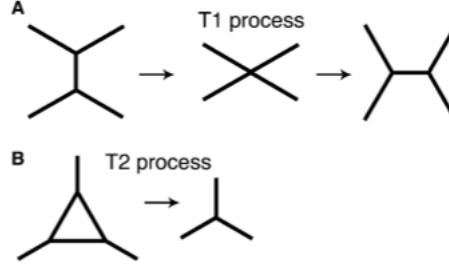


Figure 6: **(A)**T1 transition. During T1 transition, a junction shrinks and a new junction is formed perpendicular to it. **(B)** T2 transition. During this transition, triangular cells smaller than a threshold value are removed from the system. Figure taken from [13].

Different methods could be used to determine the time evolution of the system. In [13] for example, the configuration of the system at each time step is determined using the conjugate gradient method [34]. This is a numerical method which finds the local minimum of the energy function at each time step. The system is perturbed and driven out of equilibrium by cell divisions and apoptosis [13]. The Monte Carlo methods have also been used to determine the configuration of the system [28]. This is similar to finding the system configuration in the cellular Potts model, with the difference that instead of changing the state of a lattice site, the position of a vertex is changed.

Adjustments have been made to the energy function presented above in order to account for the elongation observed in some tissues, such as *Drosophila melanogaster* embryo [35] and *Drosophila* wing epithelia [27].

In [27], it is argued that appearance of an atypical myosin, Dachs, in certain locations correlates with the directional cell division and elongation. In fact, it is suggested that Dachs generates tension along the cell-cell junctions, forcing the cells to change shape. The change in the shape of the cells affects the orientation of mitotic spindle, leading to oriented cell divisions and elongation of the tissue. The presence of Dachs in certain locations has been implemented in the model by adding a term in the form $\sum_{\langle i,j \rangle} \Lambda_{ij} \Phi_{ij}^{Dachs} l_{ij}$ to the energy function. Here, $\Phi_{ij}^{Dachs} = \Phi_{MAX}^{Dachs} \times \sin(\phi)$ and $0 < \phi < 90$ is the edge angle with respect to the proximal-distal axis. It has been found that while the polarised tension due to the presence of Dachs can generate tissue elongation similar to those observed in vivo, the addition of forced polarised cell division (division along proximal-distal axis) leads to unrealistic directional growth of the tissue.

In [35], a tension anisotropy has been embedded in the line tension and cortical contractility terms of the energy function. The results of this study show that cell elongation is the result of anisotropic cortical tension at apical cell junctions. In fact, polarised tension, arising from the increased concentration of myosin II at junctions with a direction close to the dorsoventral axis, leads to their shrinkage and formation of new junctions along anterior-posterior axis. Besides the fact that an adjusted energy function is used in this study, the model differs from those presented earlier in that the cellular growth, division and apoptosis were not implemented in it.

3.3 Finite Element Models

The Finite Element Methods (FEM) are another set of computational techniques which use both cellular mechanics and geometry to simulate the evolution of biological tissue through time [16]. Unlike previ-

ous models where the energy function was used to determine the temporal evolution of the tissue, FEM uses the forces acting on the cells to compute their change in shape.

Each cell is divided into a finite number of elements. The forces acting on the cell arise from circumferential microfilament bundles, the contractility of the cell cortex, adhesion mechanisms between cells and apical mat of microtubules [5, 6]. However, all these forces can be summed into a force γ , equivalent to interfacial tension along the cell boundary [6].

The dynamics of the system could be described by the following equation:

$$\mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f} \quad (5)$$

Here \mathbf{C} and \mathbf{K} are matrices representing the damping and the stiffness of the elastic components of the system, respectively. \mathbf{u} is the vector describing the displacements of each element, $\dot{\mathbf{u}}$ is the time derivative of the displacements and \mathbf{f} is the vector of applied forces [6]. Equation 5 is a non-linear equation because \mathbf{C} , \mathbf{K} and \mathbf{f} change as the cells deform. Therefore, in order to find the time evolution of the system, the displacements of elements during short time increments are calculated. At the end of each time increment, the geometry of the system is used to find estimations of \mathbf{C} , \mathbf{K} and \mathbf{f} for the next time step.

In [5], this finite element formulation has been used to explore the role of mitosis and its orientation on cell shape and epithelia reshaping. Gibson et al [16] have also used this formulation to study how orientation of mitotic cleavage plane affects the epithelial topology. In addition to cell proliferation implemented in [5], they allow neighbour exchanges in their model. Their study shows that the orientation of mitotic cleavage plane for each cell is defined by its shape and number of neighbouring cells. This oriented cell division then determines the tissue topology (i.e. the distribution of cells with different numbers of sides).

A different formulation of FEM has been used in [4] to simulate the morphogenesis of the mouse limb bud. In fact, assuming that cells behave like liquids, the Navier Stokes equations have been used to describe the forces acting on the cells. The results from this 3D FEM show that isotropic cell divisions are not sufficient to generate the correct shape of the limb bud. Further analyses of the experimental data show that oriented cell divisions, cell shapes and Golgi orientation likely drive the directional outgrowth of the limb bud.

4 Methods

4.1 Implementation of the Vertex Model

The vertex model developed in this research is a variant to the existing models [13, 27, 28, 30]. The code is written in Matlab and is an adaption to the vertex model implemented in Matlab by Mehonic [30].

4.1.1 Representation of the Tissue

The tissue is represented by a 2D network of N_c polygonal cells, with their edges defined by N_v vertices. The model developed by Mehonic [30], similar to that described in [13], made use of periodic boundary conditions. However, this condition is true for situations where the planar area of the tissue remains constant during development. This is not the case for the mouse palatal epithelium, where the planar area is increased. Thus, this constraint was removed from the boundaries of the tissue, allowing it to grow freely. This is similar to [27], where no special conditions were put on the tissue boundaries in order to study its oriented growth [Tournier, personal communication].

4.1.2 The Energy Function

Two different energy functions were examined. For a set of simulations, the energy function described in [13] (equation 4) was used. This energy function represents isotropic tension along the cell junctions

and has been explained in detail in section 3.2. In order to analyse the effect of anisotropic tension on tissue growth, the energy function proposed in [27] was used:

$$E(R_i) = \sum_{\alpha} \frac{K_{\alpha}}{2} (A_{\alpha} - A_{\alpha}^{(0)})^2 + \sum_{\langle i,j \rangle} \Lambda_{ij} (1 + \Phi_{ij}) l_{ij} + \sum_{\alpha} \frac{\Gamma_{\alpha}}{2} L_{\alpha}^2 \quad (6)$$

$$\Phi_{ij} = \Phi_{MAX} \times \sin(\phi) \\ 0 < \phi < 90$$

Here, all parameters represent those described in section 3.2. It should also be noted that these energy functions (equation 4 and 6) have been simplified by using constant values of K , Λ and Γ for all cells and junctions of the system.

4.1.3 Temporal Evolution of the Tissue

The vertex model developed by Mehonic [28, 30] is stochastic, as it uses the Monte Carlo sampling to determine the evolution of the tissue through time. Previous vertex models [13, 20] were deterministic, in the sense that at each time step, the system would choose a configuration which minimised its energy function, using the conjugate gradient algorithm. This meant that the final configuration of the tissue was determined by its initial configuration and precise path of its temporal evolution [30]. However, the presence of noise in the system prevents it from necessarily choosing the configuration that minimises its energy function at each time step.

Since "active molecular processes" introduce some noise into biological systems [30], in the model developed in this study, I keep the stochastic feature of Mehonic's model. Thus, the tissue is evolved through time using the Monte Carlo Metropolis-Hastings algorithm:

1. A vertex is chosen at random and displaced through a small distance $\delta = (\delta_x, \delta_y)$. δ_x and δ_y are randomly chosen from a uniform distribution on the interval $[0, \delta_{max}]$, with δ_{max} being a free parameter.
2. ΔE , the change in the energy function due to this displacement is then calculated.
3. If $\Delta E < 0$, i.e. if the move is energetically favourable, it is accepted. Otherwise, the move is accepted with the Monte Carlo probability $e^{-\Delta E}$.

At each simulation time step, the system was relaxed by repeating the above steps for $t_{relax} \times N_v$ times, with t_{relax} being the relaxation time of the system. This implies that as the system grew and the number of vertices (or cells) increased, more time was given to the system to relax.

4.1.4 Cellular Growth and Division

As in previous models, cellular growth was implemented by gradually increasing the target area of the cells [13, 27, 28, 30]. More explicitly, at each simulation time step, the target area of each cell was multiplied by its growth rate g , a random number that was chosen from a uniform distribution in the interval $[1, g_{max}]$ and was assigned to each cell when it first appeared in the system. After increasing the target area of the cells, the tissue was left to relax, using the procedure described in section 4.1.3.

Next, the tissue was monitored for cells that had grown to twice their standard size. These cells were then divided by introducing a new cell boundary that went through the centroid of the cell and had a random direction. The steps of introducing this new boundary is as follows (figure 7):

1. One edge of the dividing cell is chosen at random. The midpoint of this edge is the first end of the boundary.
2. The other end of the boundary is the intersection of a line going through its first end and the centroid of the cell with the cell perimeter.

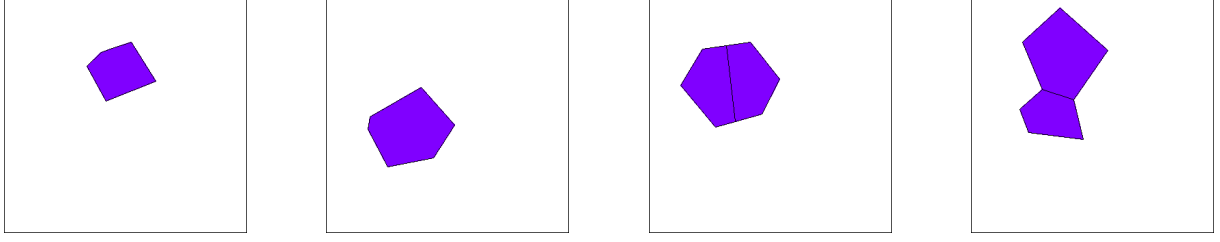


Figure 7: The steps of the cell division in the vertex model. First, the cell is grown to twice its initial size. Then the cell is divided by introducing a new boundary. The resulting cells are then left to relax.

The result of this division procedure described above is formation of two cells of approximately the same size, thus mimicking the division of the cell through mitosis [1]. It should be noted that this division procedure is similar to that described in [13] and differs from the division method originally implemented in Mehonic's model. In fact, in Mehonic's model, the new boundary was introduced by connecting the midpoint of the chosen edge to the midpoint of this opposite edge. However, in cases where the cell is asymmetric, this method does not lead to formation of two cells of the same size.

4.1.5 Topological Changes: T1 and T2 Transitions

Cell neighbour exchanges (T1 transition) and cell apoptosis (T2 transition) were included to allow for minor rearrangements of the cells [27].

To allow for T1 transitions, the tissue is monitored at every time step for junctions smaller than a threshold value δ_{T1} . If such a junction exists, it is shrunk to zero with a Monte Carlo probability and expanded in the opposite direction. T2 transitions were implemented as follows: the tissue was monitored for triangular cells with area smaller than a threshold value A_{min} . These cells were then removed from the tissue.

It is worth mentioning that while T1 transitions introduce reversible topological changes in the tissue, the changes following T2 transitions are irreversible [28, 30].

4.1.6 Setting Parameters

Similar to [13] and [27] normalised line tension $\underline{\Lambda} = \frac{\Lambda}{K(A(0))^{3/2}}$ and normalised contractility coefficient $\underline{\Gamma} = \frac{\Gamma}{KA(0)}$ were used. The values of these parameters were set to 0.1 and 0.033, as suggested in [27].

In order to allow for gradual changes in the configuration of vertices, δ_{max} should not be large compared to the length of the cell junctions. Thus, δ_{max} was set to 0.025. A range of values in the interval $[0.01, 0.025]$ were also tested. However, it was found that choosing a value smaller than 0.025 was not necessary, as it made no changes to the topology of the tissue.

The relaxation time t_{relax} was determined as follows: tissue was grown using an isotropic tension energy function with relaxation time of 1000. This was assumed to be the maximum relaxation time and I aimed to find a time smaller than this, in order to reduce the run time of the simulations. After a patch of 68 were formed, I ran the simulation for another time step and measured the total energy per cell of the tissue after N_v attempts to move the vertices. As figure 8 shows, the system reaches a minimum energy after about 500 time steps. Therefore, the relaxation time t_{relax} was set to 500 for further simulations.

g_{max} was set to 1.5. Various values have been used in literature for δ_{T1} . In this study, δ_{T1} is set to 0.1, as in [27, 30]. A_{min} is set to $\frac{A_0}{4}$, with A_0 being the standard size of the cells. This is in accordance with [30].

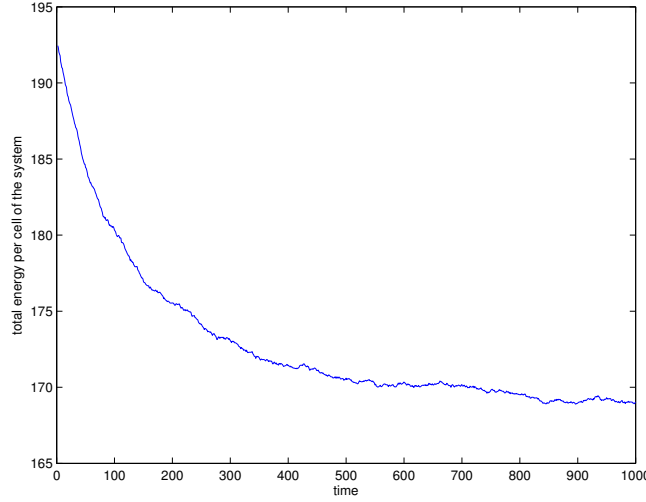


Figure 8: The total energy per cell of the tissue over time. The relaxation time is defined as the time after which the energy does not change.

4.2 Initial Configuration of the Vertex Model

In previous studies, the initial configuration was composed of a large number of cells (e.g. 305-459 cells in [35], 250 cells in [28]). In the first attempt to model the growing tissue, I decided to start the growth from a single cell. This required introducing additional conditions for the movement of vertices. For example, it was found that a vertex belonging to only one cell is free to move, because there are no other cells to be affected by its movements. Therefore, the vertices might make large displacement, sometimes leading to formation of self-intersecting polygons. Thus, conditions were embedded in the model to prevent the moves leading to generation of self-intersecting polygons.

Moves that led to cells growing into each other or crossing each other were also rejected. These conditions were not required in models with continuous boundary conditions.

Figure 9 shows a tissue grown from a single cell. As figure 10 shows, a large proportion of cells at each time step are tetragonal, which is not realistic for biological tissues. Also the shape of the cells do not resemble those observed in real tissues. To find the cause of the unrealistic shape of the cells, the initial number of cells was increased from one to seven hexagonal cells. However, it was observed that after two or three simulation time steps, the cells acquired abnormal shapes. In another attempt, all of the above conditions were removed. This led to development of a more realistic tissue. Thus, further simulations were done by starting from seven hexagonal cells, with no excess conditions on the vertex movements.

4.3 Quantifying the Growth of the Tissue

Several measures have been used to quantify the growth of the tissue and decide if it resembled the biological tissue.

4.3.1 Distribution of Neighbour Numbers

Distribution of cell neighbour numbers is a useful way to characterise tissue topology [37]. It has been used in the literature to explore the parameter space of the vertex model and choose the best parameters for the energy function [13, 27, 30]. In this study, I do not try to reproduce the exact histogram of cell neighbour numbers observed in mouse palatal tissue, as this requires an in depth study of the parameter space which is beyond the scope of this project. However, a recent study has shown that the histogram



Figure 9: Snapshots of the growth of the tissue from a single cell.

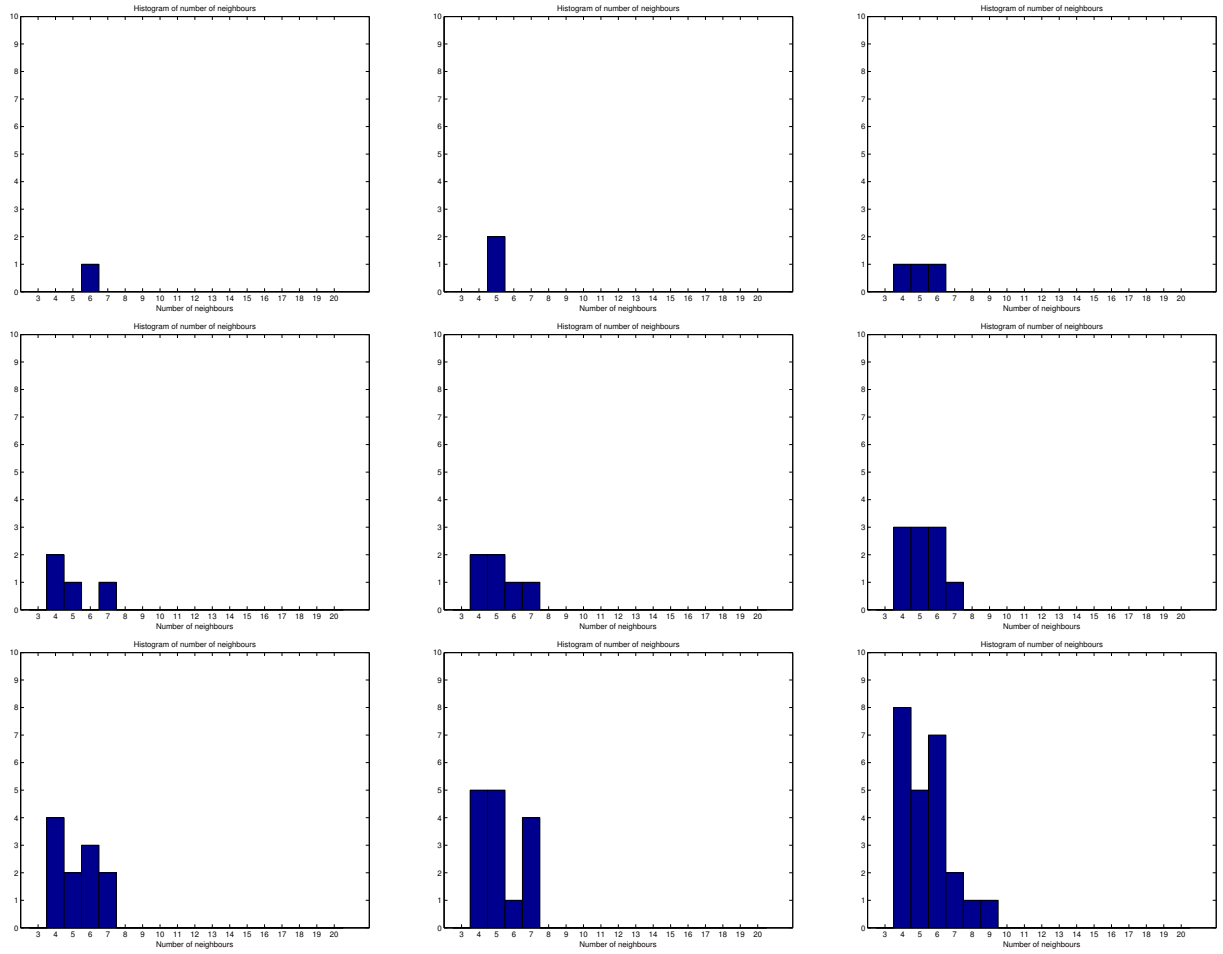


Figure 10: Histograms of the cell neighbour numbers at each simulation time step, corresponding to the simulation represented in figure 9.

of the cell neighbour numbers for growing epithelial tissues in several species fall onto a “universal” curve [15]. Thus, it might not be far from reality to expect that if the model is working properly, predominance of pentagonal and hexagonal cells should be observed in the tissue (as in the experimental data provided in [13]).

Two types of plots will be used in this report to represent the topology of the tissue. The first type of plot represents the distribution of cell neighbour numbers. To calculate this, the distribution of cell neighbour numbers has been derived for every time step of each simulation. This has then been averaged over the last 10 time steps of every simulation. The reason for averaging over the last 10 and not all time steps is that in the first time steps, the tissue has not evolved far from its initial state, which was deterministic. Thus, considering these time steps in the average topology of the tissue will affect the stochasticity of the model. The other type of plot illustrates the change in the average cell neighbour number as the tissue evolves. This has been calculated by measuring the average neighbour numbers at each time step of each simulation and then averaging over all simulations.

4.3.2 Cell and Tissue Deformation

In order to study the changes in the shape of the cells, the relative dimensions of the tissue cells are extracted. This is done by finding the ellipse of inertia for each cell and calculating the ratio of its long axis to short axis. In fact, the ratio of the long axis to short axis of a cell’s ellipse of inertia, together with the orientation of the ellipse can be a good measure of elongation of the cell in a certain direction. This can be seen in figure 11.

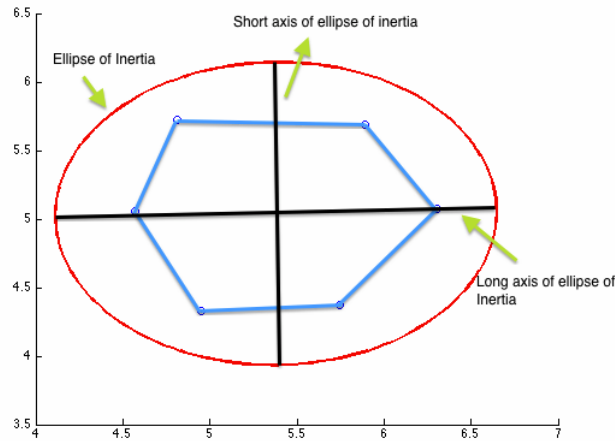


Figure 11: Ellipse of inertia of a hexagonal cell. The ratio of the long to short axis gives a measure for the relative dimensions of the cell. The angle between the long axis and horizontal axis gives a measure for the cell orientation.

The deformation of the whole tissue was also quantified by finding its ellipse of inertia and measuring the ratio of its long axis to short axis, as well as its orientation. It is expected that if the growth of the tissue is isotropic, this ratio should be close to one, whereas if the tissue is elongating in a certain direction, this ratio should be larger than one. The direction of elongation can be given by the angle the major axis of the ellipse makes with the horizontal direction.

4.3.3 Growth Rates

The growth rates in horizontal and vertical directions were also used to quantify the growth of the tissue in different directions. These were defined as below:

$$g_{horizontal}(t) = \frac{[x_{max}(t) - x_{min}(t)] - [x_{max}(t-1) - x_{min}(t-1)]}{\Delta t} \quad (7)$$

$$g_{vertical}(t) = \frac{[y_{max}(t) - y_{min}(t)] - [y_{max}(t-1) - y_{min}(t-1)]}{\Delta t} \quad (8)$$

5 Results

5.1 Growth Under Isotropic Tension

In order to find the effect of anisotropic tension on the development of the tissue, I first had to quantify its growth in the absence of such tension. Thus, 4 simulations were run, each comprising of 15 simulation time steps, with the number of tissue cells increasing from 7 to an average of 53.7. Figure 12 shows the snapshots of one of these simulations (full simulation can be found in supplementary movie 1). The energy function in equation 4 was used for evolving the tissue through time.

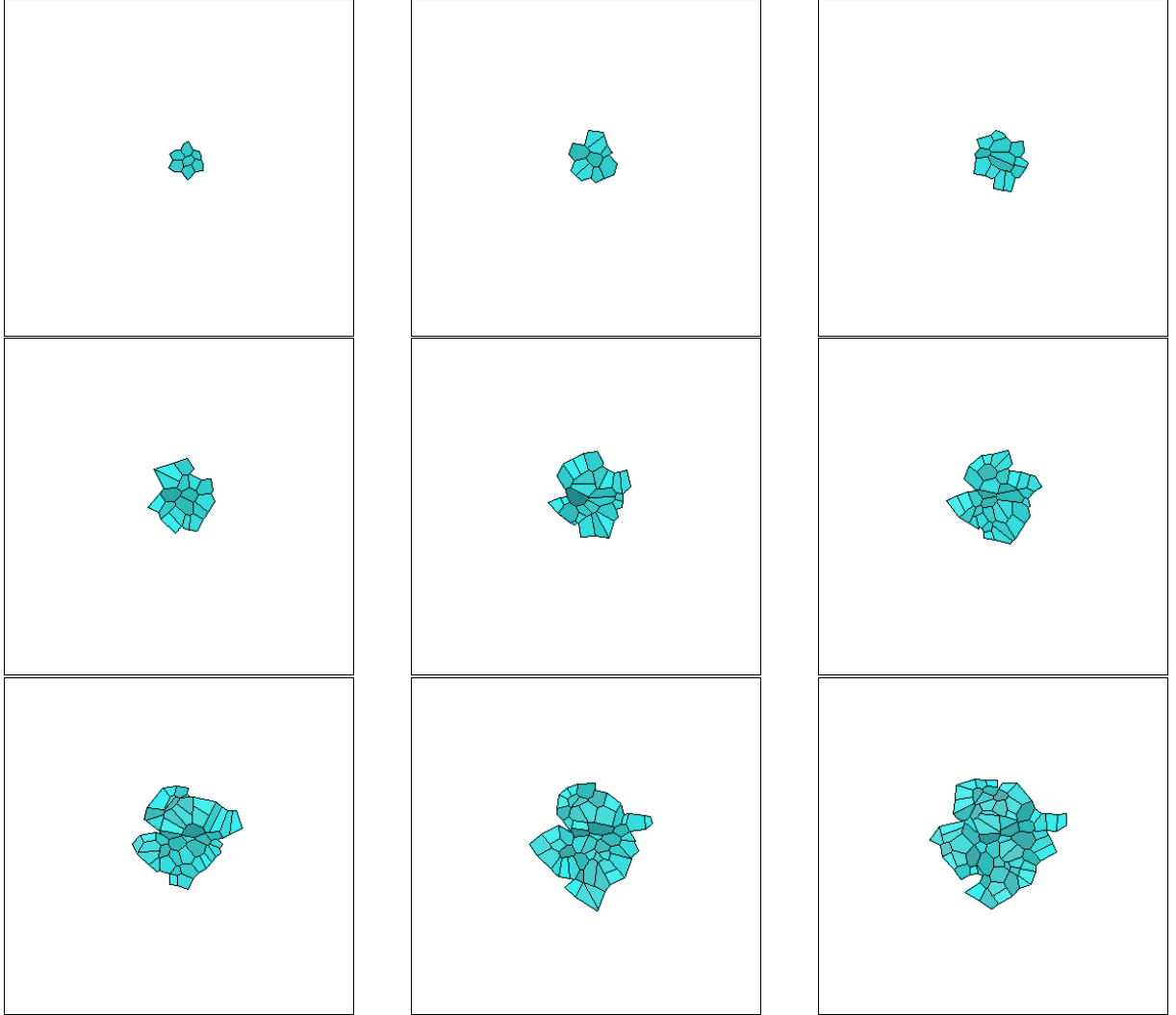


Figure 12: Snapshots of the growth of the tissue under isotropic tension. The growth has been started from a patch of 7 hexagonal cells.

The histogram of cell neighbour numbers is presented in figure 13. As it can be seen, the majority of cells are pentagons, in accordance with [13]. This is followed by hexagonal and tetragonal cells. These results are also consistent with the case of growing tissue with uniform rate of cell divisions in [30]. However, it should be noted that exact comparison of statistics is not possible, as the model itself and the parameters used are different from those in [13] and [30]. It can be inferred from the histogram that

under isotropic tension, the model developed in this research can produce a tissue topology similar to those in previous models.

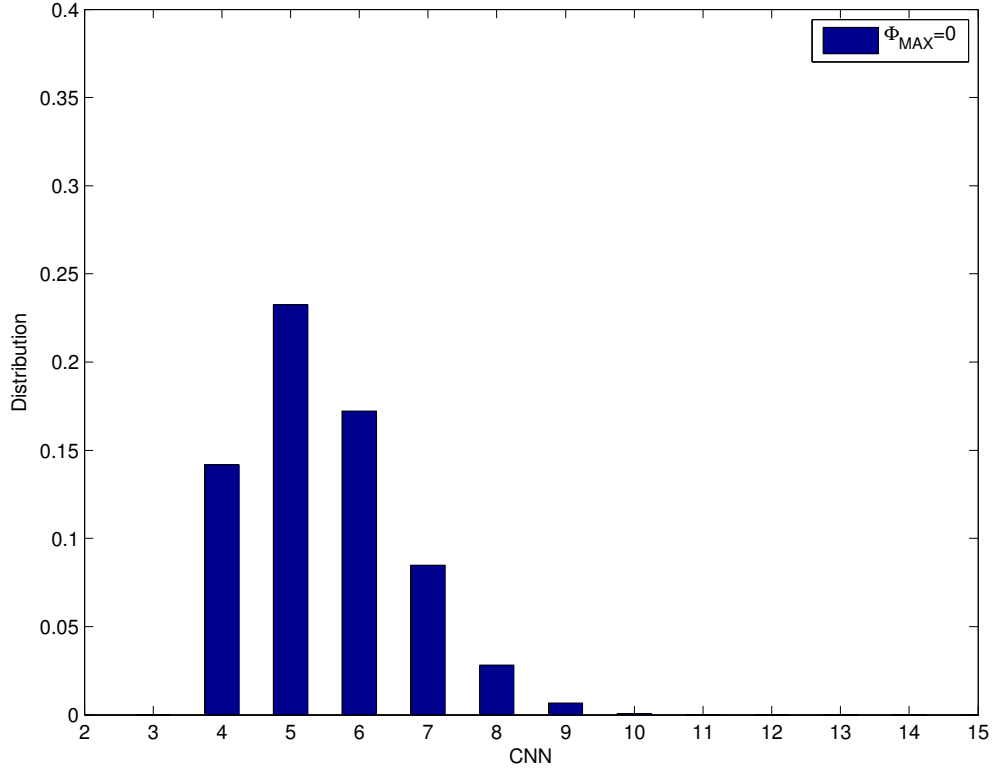


Figure 13: Histogram of Cell Neighbour Numbers (CNN) for tissue development under isotropic tension

Figure 17 (red line) shows that the relative dimensions of the cells vary in the interval between 1 and 2. This suggests that the shape of the cells are roughly equally proportioned. Considering the whole tissue, its dimensions in different directions remain almost equal throughout its development. This is illustrated in figure 19, where the relative dimensions of the tissue remains in the interval between 1 and 1.3.

The above results confirm that in the absence of any kind of anisotropy in the implementation of the model, the growth of the tissue will be uniform in all directions. Although this result is trivial, it was essential to analyse the growth of the tissue in the presence of isotropic tension in order to make comparisons with the results of the next section.

5.2 Growth Under Anisotropic Tension

A total of 16 simulations were run, with 4 simulations for each value of Φ_{MAX} varying from 1 to 4, incremented by 1. In these simulations, the energy function in equation 6 was used for the Metropolis-Hastings algorithm. The snapshots of one of the simulations with $\Phi_{MAX} = 2$ has been presented (figure 14, also supplementary movie 2). Snapshots of simulations for other values of Φ_{MAX} have been provided in the appendix A.2.

Histograms of cell neighbour numbers for each value of Φ_{MAX} are represented (figure 15). Values larger than this led to rapid deformation of the cells. In principle, it should be possible to run simulations with these parameter values. However, in practice, the cells grow into each other, because the conditions that prevented the vertices to move into neighbouring cells were removed from the model. Therefore, it was not possible to try larger values of Φ_{MAX} with the current implementation of the vertex model. As it can be seen, for all cases except $\Phi_{MAX} = 3$, the majority of cells are pentagonal, followed by hexagonal and



Figure 14: Snapshots of the growth of the tissue under anisotropic tension $\Phi_{MAX} = 2$. The growth has been started from a patch of 7 hexagonal cells.

tetragonal cells. For $\Phi_{MAX} = 3$, the population of tetragonal cells is larger than hexagonals. Figure 16 shows the average cell neighbour numbers during tissue development. It can be seen that as the tissue develops, the average number of neighbours decreases. For the case of isotropic tension and all cases of anisotropic tension except $\Phi_{MAX} = 3$, this number decreases to a value in the interval $[5.35, 5.45]$, whereas for $\Phi_{MAX} = 3$, it decreases to 5.56. As no major difference can be observed between the tissue cell neighbour numbers under isotropic and anisotropic tension, it can be concluded that addition of isotropic tension does not affect the tissue topology.

However, anisotropic tension does affect the shape of the cells. This is evident from figures 17 and 18. In fact, increasing Φ_{MAX} leads to a rise in the ratio of the long axis to short axis of cells' ellipses of inertia (figure 17). Figure 18 shows that the increasing anisotropic tension does not affect the average orientation of the ellipses of inertia, as their angle with horizontal axis remains within the range $[4.6, 5.7]$. Thus, it can be concluded that introducing anisotropic tension into the model leads to elongation of cells in a specific direction. In our simulations, this elongation is in the horizontal direction, as the tension is maximum in vertical junctions.

Figures 19 and 20 illustrate the overall changes in the shape of the tissue due to introducing and increasing an anisotropic tension. It can be seen that as Φ_{MAX} is increased, the ratio of the long axis to short axis of tissue's ellipse of inertia is increased, while its orientation remains close to the horizontal axis. This suggests that the tissue is also elongating in the horizontal direction, although its elongation is minor compared to that of individual cells. In fact, it can be seen that while there is %78 increase (on average) in the average ratio of the long axis to short axis of the cells' ellipses of inertia, this value is %8 for the overall tissue.

One point that requires explanation is the variations in the angle of long axis of tissue's ellipse of inertia with horizontal axis under isotropic tension, θ . The fact that under isotropic tension, the ratio of the long axis to short axis of tissues' ellipses of inertia remained close to 1 (changing between 1 and 1.3) suggests that these ellipses are close to circles and thus, θ does not really define their orientation.

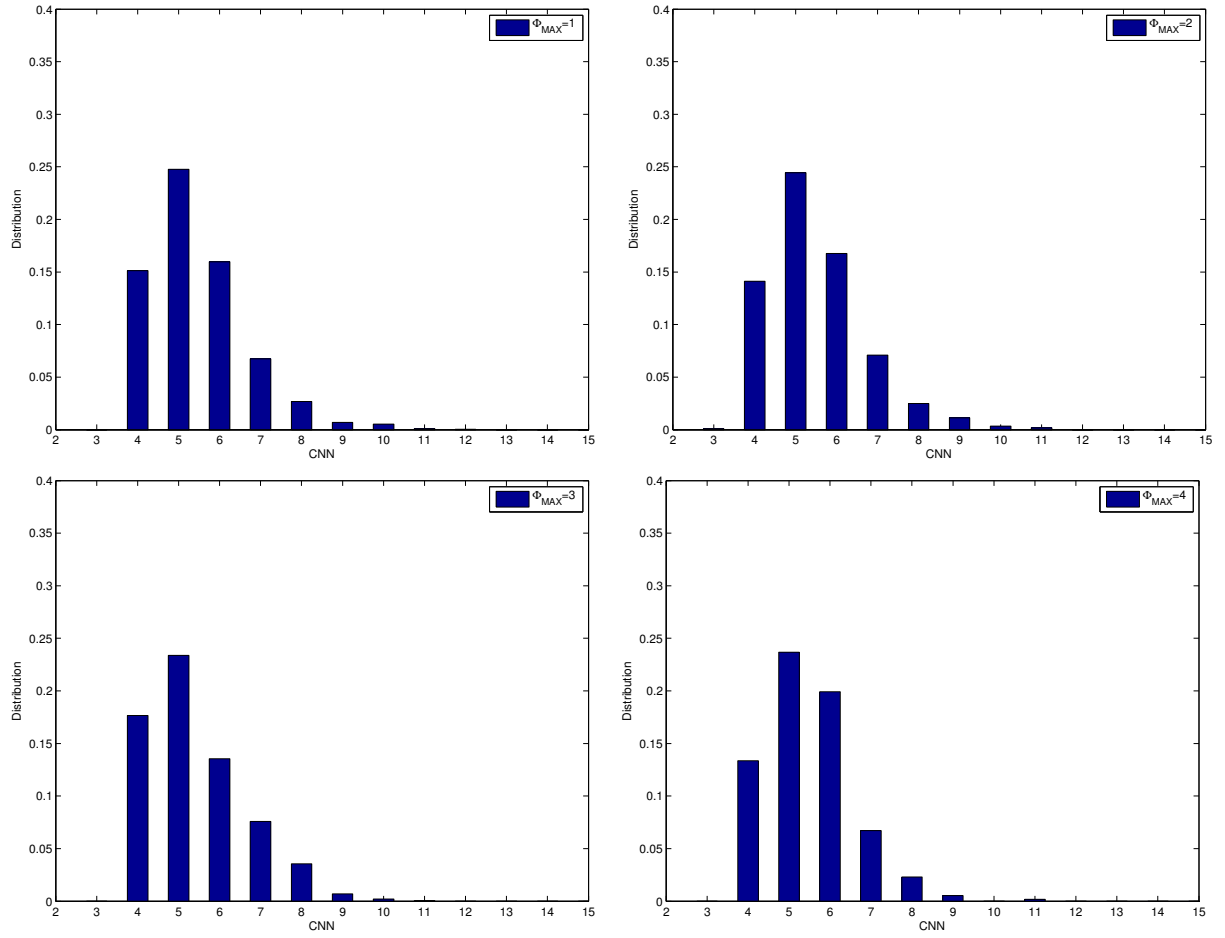


Figure 15: Histogram of the average cell neighbour numbers during tissue development under anisotropic tension.

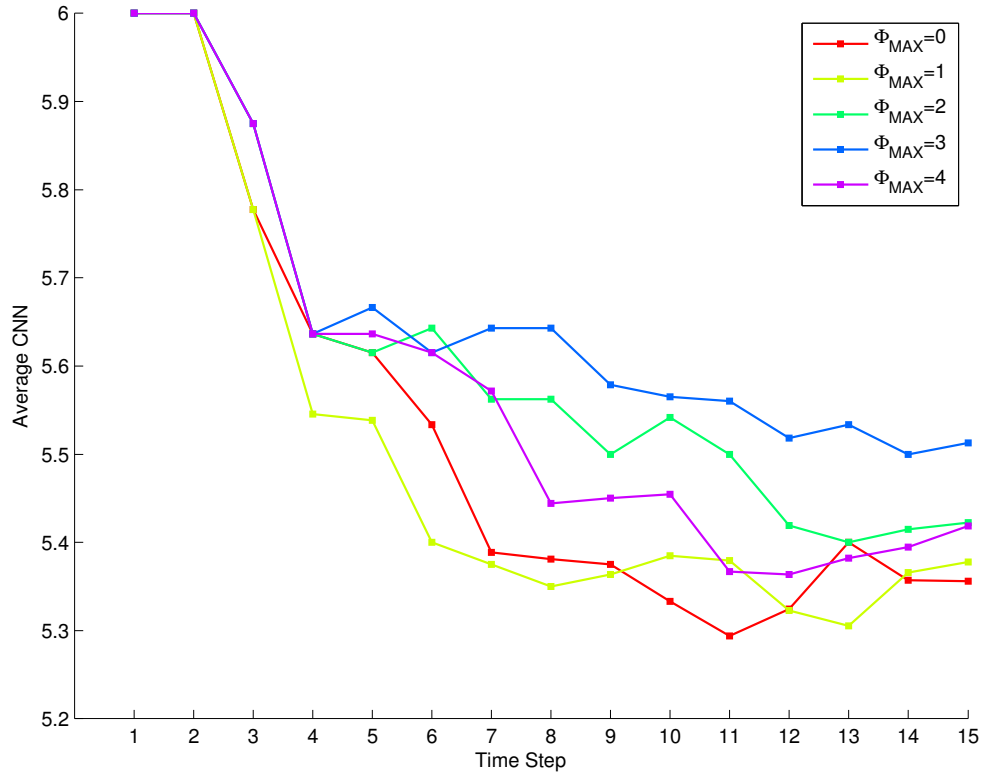


Figure 16: Average cell neighbour numbers of the whole tissue versus simulation time steps

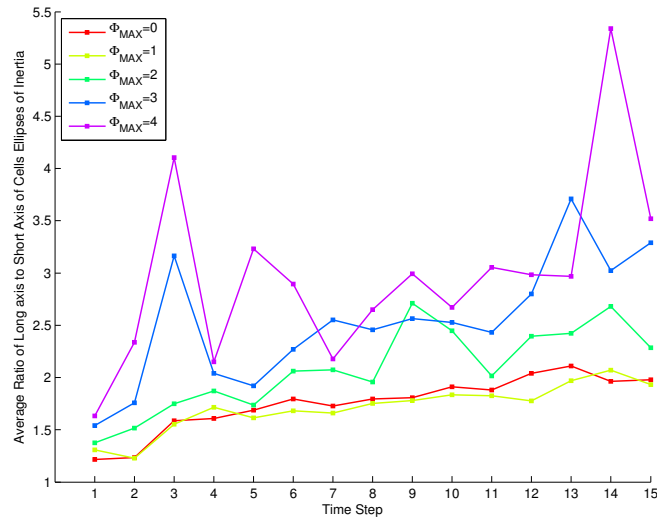


Figure 17: Average ratio of long axis to short axis of the cells' ellipses of inertia at each simulation time step. $\Phi_{MAX} = 0$ corresponds to the case of isotropic tension, while its non-zero values represent the cases of anisotropic tension. Greater values show increased anisotropy.

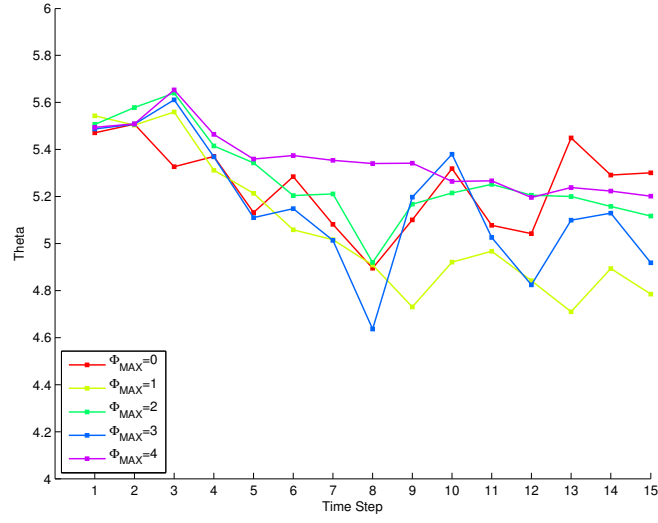


Figure 18: θ is the angle of the long axis of the cells' ellipses of inertia with the horizontal axis. The average value of this angle has been shown for every simulation time step. Again, $\Phi_{MAX} = 0$ corresponds to the case of isotropic tension and its non-zero values represent the cases of anisotropic tension.

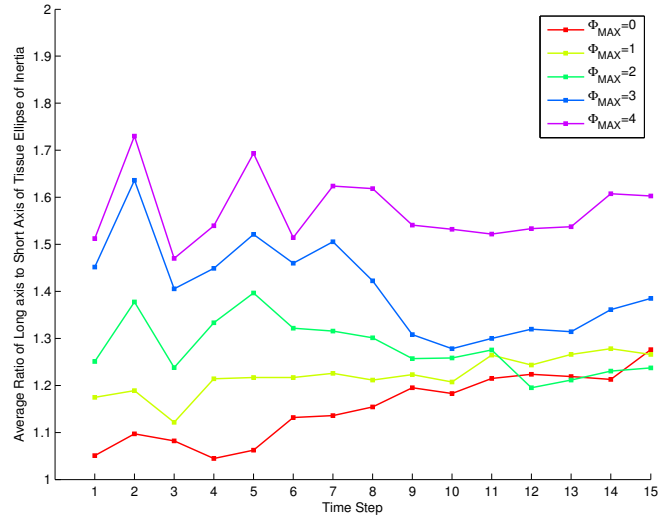


Figure 19: Temporal changes in the average ratio of the long axis to short axis of the ellipse of inertia of the whole tissue. $\Phi_{MAX} = 0$ corresponds to the case of isotropic tension, while its non-zero values represent the cases of anisotropic tension. Greater values show increased anisotropy.

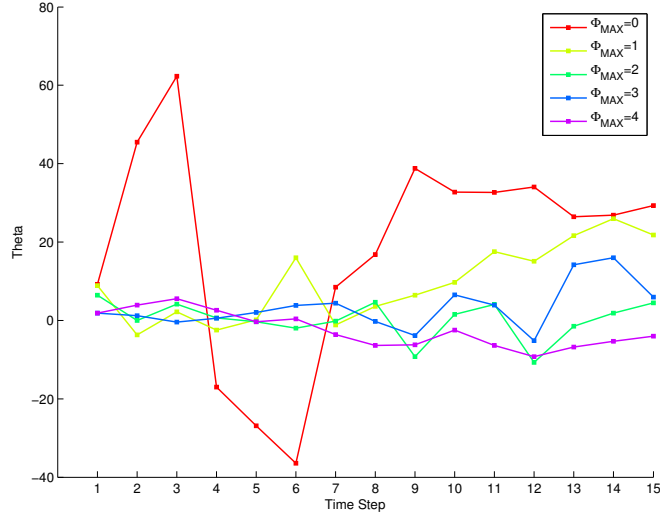


Figure 20: θ is the angle of the long axis of the ellipse of inertia of the whole tissue with the horizontal axis. The average value of this angle has been shown for every simulation time step. Again, $\Phi_{MAX} = 0$ corresponds to the case of isotropic tension and its non-zero values represent the cases of anisotropic tension.

The growth rates of the tissue in horizontal and vertical directions are illustrated in figure 21. It can be seen that introduction of anisotropic tension into the model does not have major effects on tissue's growth rates. This supports the idea that although anisotropic tension leads to lengthening of individual cells, its effect on the overall shape of the tissue, when the cell divisions are in random directions, might not be considerable.

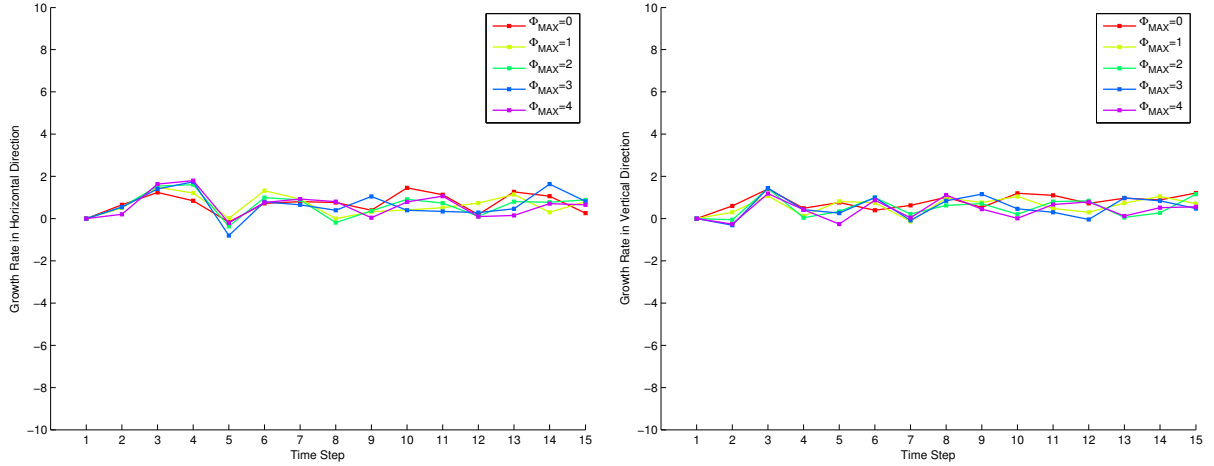


Figure 21: Growth rates of the tissue in horizontal and vertical directions. The cases of isotropic and anisotropic tension have been shown.

5.3 Contribution of Cell Divisions to Tissue Elongation

How much of the elongation observed in the presence of anisotropic tension is due to cell proliferation and how much of it arises from the change in the shape of the cells? In order to answer this question, I decided to study the contribution of random cell divisions to tissue elongation. For this purpose, tissues

developed in the previous section under anisotropic tension ($\Phi_{MAX} = 2$) were used. In fact, the configuration of these tissues at time step 10 was used as the initial state for the new set of simulations, where the tension was still anisotropic but cell divisions were stopped. The results were then compared with the original simulations.

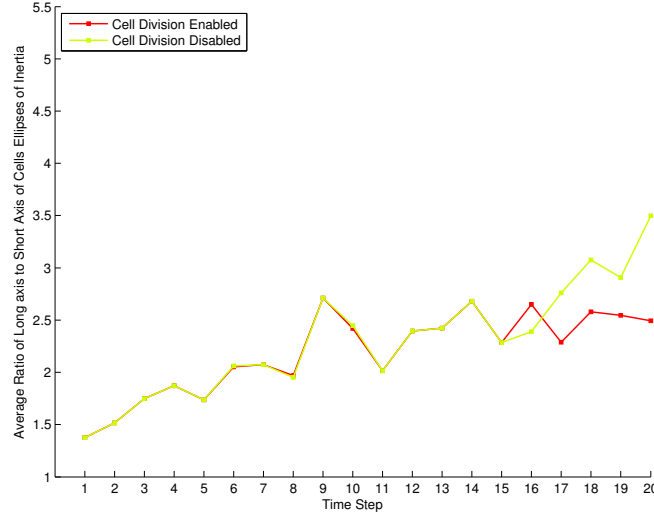


Figure 22: Average ratio of long axis to short axis of the cells' ellipses of inertia at each simulation time step. For the green line, cell divisions were disabled from time step 15.

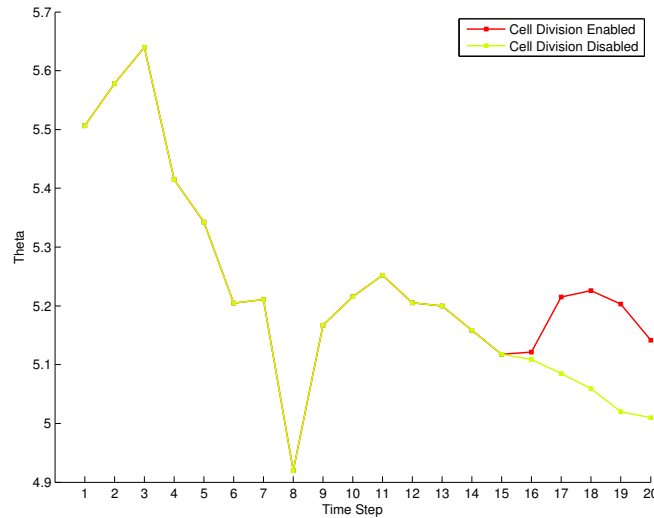


Figure 23: θ is the angle of the long axis of the cells' ellipses of inertia with the horizontal axis. For the green line, cell divisions were disabled from time step 15.

The changes in the cell shapes are compared in figures 22 and 23. It can be seen that even when the cell divisions are stopped, anisotropic tension promotes cell elongation in the horizontal direction. Figures 24 and 25 show the changes in the overall shape of the tissue. These figures show no change in the degree of tissue elongation when cell divisions were stopped. Thus, it can be concluded that cell shape changes are the main contributors to the minor elongation observed when the tissue is under anisotropic tension.

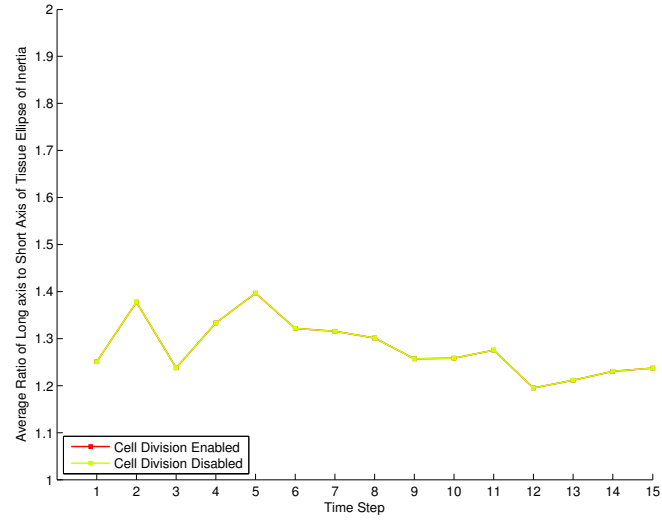


Figure 24: Temporal changes in the average ratio of the long axis to short axis of the ellipse of inertia of the whole tissue. For the green line, cell divisions were disabled from time step 15.

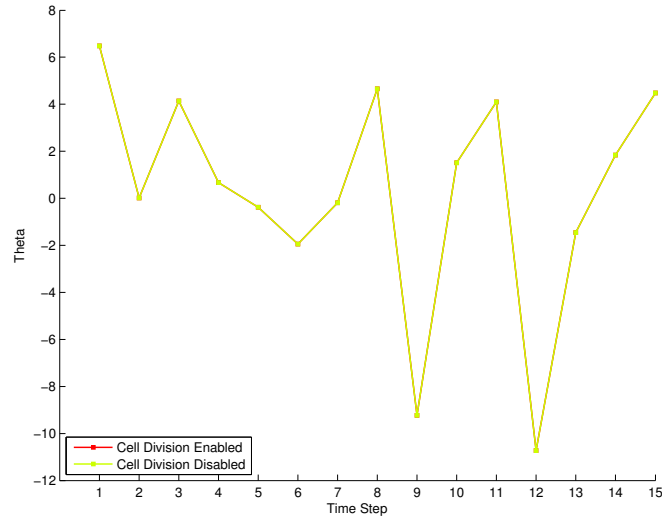


Figure 25: θ is the angle of the long axis of the ellipse of inertia of the whole tissue with the horizontal axis. The average value of this angle has been shown for every simulation time step. For the green line, cell divisions were disabled from time step 15.

6 Discussion and Conclusion

The effects of polarised tension on tissue development was studied. It was found that under such tension, the cells elongate in the direction of minimum tension. This could be explained as follows: when the tension is polarised, the cells tend to orient their junctions in a specific direction along which tension is minimum. This leads to lengthening of cells in the direction of minimum tension [27]. Minor tissue elongation was also observed under anisotropic tension. The simulations show that cell shape changes (elongations) give rise to this minor elongation and cell divisions either do not contribute to it or have negligible effects.

To be more specific, in the model used in this study, cells divided in random directions. Results from another study [27] show that anisotropic tension, together with division along the long axis of the cells lead to directional growth of the tissue. The results of the current study show that in order for an anisotropic tension to lead to major directional growth, this type of cell division is essential and divisions in a random direction do not contribute a lot to tissue elongation.

The model developed in this study has qualitatively shown how cell shape changes, as a result of anisotropic tension, can lead to minor elongation of the tissue. This is consistent with experimental data from the growth in the RGZ of the mouse palate. These data revealed that cell shape changes may be one of, but not the only, mechanism involved in tissue elongation [Unpublished data from the Green lab]. It is worth noting that while in the model, cell shape changes were due to anisotropic tension on the cell junctions, the source of cell elongations in mouse palatal epithelium is still unclear. One possibility is that these cell elongations, similar to the model, are caused by a polarised tension due to the non-uniform presence of cell adhesion molecules (such as Dachs [24, 27]) at cell junctions. However, currently there is no evidence for the presence of such molecules at the cell junctions in RGZ and further study should be done to explore this.

Studies in [35] suggest that as a result of anisotropic tension, cell neighbour exchanges (T1 transitions) will be promoted in the direction of minimum tension, thus elongating the tissue in that direction. The exact rates and directions of T1 transitions have not been studied in this report, as they were not many T1 transitions occurring in the developed tissues. However, it was observed that as the tissue grows, the number of T1 transitions increases. Thus, it might be that if the tissues were left to grow larger, enough T1 transitions would occur to produce effective elongation.

Despite the fact that the developed model has been used to find qualitative explanations of the directional growth of the tissue, its results cannot be used for quantitative analysis of the growth as it still has some limitations and requires to be improved.

One limitation of the model is that some cells grow too small. This is especially promoted for the cells on the edges of the tissue. This might be due to the fact that the vertices on the edge of the tissue are shared between either one or two cells. This makes them more free to move compared to 3-way vertices. Having continuous boundary conditions in previous models [13, 30] resulted in all vertices being shared between three cells, thus more limited to move. It was also observed that small cells appear more often in the tissue under anisotropic tension. One reason for this might be that under isotropic tension, the area elasticity term plays an important role in the energy function by promoting moves that increase the area of the cell and help them to reach their target area. By adding anisotropic term to the line tension term, this term might become the dominant term in the energy function, thus reducing the effect of the cell elasticity. Therefore, through development, the moves that reduce the tension on the junctions would be preferred to those that increase the cell area and keep it close to its target value, leading to formation of small cells. No conditions were embedded in the model to prevent the cells from getting smaller than a threshold. The only process that removed the cells smaller than a certain threshold from the tissue was T2 transition and that only removed 3-sided cells.

Another limitation of the model is that in some cases, the cells intersect with each other. As mentioned earlier, conditions were implemented in the model to prevent such moves. However, these were removed from the model for final simulations, as they affected the shape of the cells. It should be noted

that this growth of the cells into each other only occurred on the edge of the tissue, as the vertices there were free to move. However, as the tissue grows, the ratio of the number of cells on the edge to the number of cells inside the tissue decreases, thus reducing the effect of these cells on the overall statistics of the tissue. Hence, it is believed that although these errors were present in the model, they did not affect the results deduced in the previous section. It should also be noted that the simulations used for the analysis of growth were carefully chosen from a number of simulations, in order to minimise the degree of error.

These are believed to be the major issues of the current model. It is also believed that these issues may partly explain why the modelled tissues do not fully resemble real tissues. Better choice of parameters may also improve the simulations. In fact, the parameter space of this model is comprised of several parameters. Some of the parameters have been explored. The main parameters which are yet to be explored are K , Λ and Γ in the energy function. It has been shown that the value of these parameters affect the topology of the tissue [13]. Since the model developed here is an adaption to previous models, the parameter values used in those models cannot necessarily be the best choices for this model. Thus, a parameter study should be carried out on the parameters mentioned above.

In the developed vertex model, continuous boundary conditions were not used for the growing tissue. This is similar to [27] and differs from [13, 28]. [30] has modelled growing tissue with continuous boundary conditions by increasing the boundaries of the tissue in every 5th time step. Using continuous boundary conditions has some advantages and disadvantages. The advantage is that under these conditions, all vertices will be 3-way, thus the issues raised above, which were due to the vertices being free to move, would be solved. However, since putting continuous boundary conditions on a physical system implies its infinite size, it is not certain how a tissue under these conditions can represent growth. Thus, a more in depth study should be done to decide whether any conditions (in particular, continuous boundary conditions) are needed to be implemented in the model in order to increase the resemblance between simulated and real tissues.

To conclude, the aim of this study was to develop a model of growing tissue and use it to explore the contribution of different mechanisms to its directional growth. Outcomes of the simulations suggest that minor tissue elongation might be caused by cell shape changes. In addition, anisotropic tension has been proposed as a possible explanation for the change in the cell shapes. Although the model has been used to make qualitative analysis of the tissue growth, a more quantitative study which could be compared with experimental data cannot be done before solving the model's issues. These limitations have been discussed in detail and suggestions have been made to direct the future work. It is hoped that by applying the suggested changes, the model could be employed as a useful tool to increase insight into the mechanisms involved in directional growth of the tissue.

7 Acknowledgements

I am grateful to both my supervisors, Prof Mark Miodownik and Dr Jeremy Green for their continuing help, support and encouragement throughout the project. It has been through their guidance that I learnt that the key element in modelling biological systems is *deciding the amount of detail to be put into the model, so that it holds the key features of the real biological systems while being kept simple.*

A Appendix

The code for the vertex model developed in this research, as well as two movies showing the tissue growth under isotropic and anisotropic tension are available as supplementary materials. These have been handed in to CoMPLEX and will be provided upon request.

A.1 The Code

Below are the main functions of the vertex model code. This is an adaption to the code developed by Aida Mehonic [30] and provided to me by Charlotte Strandkvist. The codes for plotting histograms of cell neighbour numbers, determining ellipses of inertia and calculating growth rates are also presented. All codes are written in Matlab.

A.1.1 The Vertex Model Code - Main Function

```
tStart=tic;
tm_newsetup

global ncell nvertex tmax areamax;
global vertex_xy vertex_cell;
global cell_vertex cell_nvertex cell_cell cell_area0 cell_generation cell_energy cell_growthrate;

% Preallocation of output arrays
noCells=zeros(1,tmax);
totalenergy=zeros(1,tmax);
totalarea=zeros(1,tmax);
cellvertex=zeros(tmax,500,15);
cellnvertex=zeros(tmax,500);
cellcell=zeros(tmax,500,15);
vertexcell=zeros(tmax,3000,3);
vertexxy=zeros(tmax,3000,2);
cellgeneration=zeros(tmax,1,500);
tenergy=zeros(tmax,500);

for t = 1:tmax

    % T1 and T2 transitions
    tm_t1
    tm_t2

    % Growth and Relaxation of the system
    for jcell=1:ncell
        cell_area0(jcell)=cell_area0(jcell)*cell_growthrate(jcell);
    end

    for i=1:500
        for kvertex=1:nvertex
            tm_move(kvertex);
        end
    end

    % Cell Division
    for jcell=1:ncell
        if (tm_area(jcell)>areamax)
            tm_newgrowanddivide(jcell)
        end
    end
end
```

```

end

% Plot the Tissue Configuration
tm_plotcells
Video(t) = getframe;
[totalenergy(t) totalarea(t) noCells(t) cellvertex(t,,:) cellnvertex(t,,:) ...
cellcell(t,,:) vertexcell(t,,:) vertexxy(t,,:) cellgeneration(t,,:)]=tm_totalenergy;

end

save
% movie(Video)
movie2avi(Video, 'tm-movie-70')
toc(tStart)

```

A.1.2 The Code for Changing the Configuration of the Tissue

```

function tm_move(kvertex)
% move vertices

global ncell
global vertex_xy vertex_cell cell_nvertex cell_vertex
global delta

% calc energy of the 3 cells associated with the vertex
energy0 = 0;
for j = 1:3
    jcell = vertex_cell(kvertex,j);
    if (jcell~=0)
        energy0 = energy0 + tm_energy(jcell);
    end
end

% move vertex through random displacement
xy0 = vertex_xy(kvertex,:);
dx = (rand-0.5)*delta;
dy = (rand-0.5)*delta;
x1=xy0(1)+dx;
y1=xy0(2)+dy;
IN=0;
for jcell=1:ncell
    if ~ismember(jcell,vertex_cell(kvertex,:))
        [xv, yv] = tm_cellcoords(jcell);
        if (inpolygon(x1,y1,xv,yv)==1)
            IN = 1;
        end
    end
end
end
if (IN==0)
    vertex_xy(kvertex,1) = x1;
    vertex_xy(kvertex,2) = y1;
end

energy = 0;
for j = 1:3

```

```

    jcell = vertex_cell(kvertex,j);
    if (jcell~=0)
        energy = energy + tm_energy(jcell);
    end
end

% % reject move if the new cells intersects others or themselves
% isect=0;
% selfintersect=1;
% vertex_list=zeros(6,1);
% for i=1:3
%     cell=vertex_cell(kvertex,i);
%     if (cell~=0)
%         poly=[];
%         [poly(:,1) poly(:,2)]=tm_cellcoords(cell);
%         if (selfintersect==1)
%             selfintersect=isempty(polygonSelfIntersections(poly));
%         end
%     end
% end
%
% if (selfintersect==0)
%     vertex_xy(kvertex,:) = xy0;
% elseif (selfintersect==1)
%     for i=1:3
%         cell=vertex_cell(kvertex,i);
%         if (cell~=0) && (isect==0)
%             P1=[];
%             [P1(1,:) P1(2,:)]=tm_cellcoords(cell);
%             for jcell=1:ncell
%                 P2=[];
%                 [P2(1,:) P2(2,:)]=tm_cellcoords(jcell);
%                 c=poly2poly(P1,P2);
%                 if ismember(0,unique(ismember(c,P1)))
%                     isect=1;
%                     break
%                 end
%             end
%         end
%     end
%     if (isect==1)
%         vertex_xy(kvertex,:) = xy0;
%     end
% end
% %
% % if (isect~=0) || (selfintersect==0)
% % % if (nnz(isect1)~=0) || (selfintersect~=0)
% %     vertex_xy(kvertex,:) = xy0;
% % end

% reject move according to Boltzmann probability, if energy increases
if (energy > energy0) && (rand > exp(energy0-energy))
    vertex_xy(kvertex,:) = xy0;
end

```

```
end
```

```
end
```

A.1.3 The Code for Cell Divisions

```
function tm_newgrowanddivide(jcell)
% divide cell jcell

global ncell nvertex area0
global vertex_xy vertex_cell
global cell_vertex cell_nvertex cell_cell cell_area0 cell_generation cell_growthrate

%-----
% Divide
%-----
% choose vertex at random
k1 = floor(rand*cell_nvertex(jcell)) + 1;

% find the centroid of the cell
poly=[];
[poly(:,1) poly(:,2)]=tm_cellcoords(jcell);
centroid=polygonCentroid(poly);

% find other vertex, on the line going through cenroid
point1=tm_midedge(jcell,k1);
point2=zeros(1,2);
line=createLine(point1,centroid);
intersects=intersectLinePolygon(line,poly);
for i=1:size(intersects,1)
    if (abs(intersects(i,1)-point1(1,1))>0.0001) && (abs(intersects(i,2)-point1(1,2))>0.0001)
        point2(1,1)=intersects(i,1);
        point2(1,2)=intersects(i,2);
    end
end

for i=1:cell_nvertex(jcell)
    i1=i;
    i2=i+1;
    if i1==cell_nvertex(jcell)
        i2=1;
    end
    EDGE=[vertex_xy(cell_vertex(jcell,i1),1) vertex_xy(cell_vertex(jcell,i1),2)...
        vertex_xy(cell_vertex(jcell,i2),1) vertex_xy(cell_vertex(jcell,i2),2)];
    if (isPointOnEdge(point2,EDGE)==1)
        k2=i1;
    end
end

if (k1==k2)
    'YES'
end

if (k2<k1)
    k2=k2+cell_nvertex(jcell);
end
```

```

end

% number of vertices of old cell (nv0), new old cell (nv), and new cell (nv3)
nv0 = cell_nvertex(jcell);
nv = k2-k1+2;
nv3 = nv0-nv+4;

if (k2 > nv0);
    k2 = k2 - nv0;
end

% indices of new vertices (kvertex1, kvertex2), neighbouring cells (jcell1, jcell2) ...
and new cell (jcell3)
kvertex1 = nvertex + 1;
kvertex2 = nvertex + 2;
nvertex = nvertex + 2;
jcell1 = cell_cell(jcell,k1);
jcell2 = cell_cell(jcell,k2);
jcell3 = ncell+1;
ncell = ncell + 1;

% assign new coordinates
vertex_xy(kvertex1,:) = [point1(1,1) point1(1,2)];
vertex_xy(kvertex2,:) = [point2(1,1) point2(1,2)];

% assign adjacent cells to new vertices
vertex_cell(kvertex1,:) = [jcell, jcell3, jcell1];
vertex_cell(kvertex2,:) = [jcell3, jcell, jcell2];

% copy list of vertices and neighbouring cells
vertex_list = cell_vertex(jcell,:);
cell_list = cell_cell(jcell,:);

% new old cell
for k = 1:nv-2
    kk = k1 + k;
    if (kk > nv0)
        kk = kk - nv0;
    end
    cell_vertex(jcell,k) = vertex_list(kk);
    cell_cell(jcell,k) = cell_list(kk);
end
cell_vertex(jcell,nv-1) = kvertex2;
cell_cell(jcell,nv-1) = jcell3;
cell_vertex(jcell,nv) = kvertex1;
cell_cell(jcell,nv) = jcell1;
cell_vertex(jcell,nv+1) = 0;    % not strictly necessary
cell_cell(jcell,nv+1) = 0;
cell_nvertex(jcell) = nv;

% new cell
for k = 1:nv3-2
    kk = k2 + k;
    if (kk > nv0)
        kk = kk - nv0;
    end
end

```

```

        cell_vertex(jcell3,k) = vertex_list(kk);
        cell_cell(jcell3,k) = cell_list(kk);
    end
    cell_vertex(jcell3,nv3-1) = kvertex1;
    cell_cell(jcell3,nv3-1) = jcell;
    cell_vertex(jcell3,nv3) = kvertex2;
    cell_cell(jcell3,nv3) = jcell2;
    cell_vertex(jcell3,nv3+1) = 0;    % not strictly necessary
    cell_cell(jcell3,nv3+1) = 0;
    cell_nvertex(jcell3) = nv3;

% replace jcell by jcell3 in neighbour-list of cells adjacent to new cell
for k = 1:nv3-2
    kk = k2 + k;
    if (kk > nv0)
        kk = kk - nv0;
    end
    j0 = cell_list(kk);
    if (j0~=0)
        for k0 = 1:cell_nvertex(j0)
            if (cell_cell(j0,k0) == jcell)
                cell_cell(j0,k0) = jcell3;
            end
        end
    end
end

% add vertex kvertex1 to cell jcell1 before vertex_list(k1)
% assign jcell3 as corresponding new neighbour of jcell1
if (jcell1~=0)
    index = find(cell_cell(jcell1,:)==jcell3);    % switch jcell for jcell3
    cell_cell(jcell1,index) = jcell;
    incr = 1;
    for kk = cell_nvertex(jcell1):-1:1
        cell_vertex(jcell1,kk+incr) = cell_vertex(jcell1,kk);
        cell_cell(jcell1,kk+incr) = cell_cell(jcell1,kk);
        if (cell_vertex(jcell1,kk) == vertex_list(k1))
            incr = 0;
            cell_vertex(jcell1,kk+incr) = kvertex1;
            cell_cell(jcell1,kk+incr) = jcell3;
        end
    end

    cell_nvertex(jcell1) = cell_nvertex(jcell1) + 1;
end

% add vertex kvertex2 to cell jcell2 before vertex_list(k2)
% assign jcell as corresponding new neighbour of jcell2
if (jcell2~=0)
    index = find(cell_cell(jcell2,:)==jcell);    % switch jcell3 for jcell
    cell_cell(jcell2,index) = jcell3;
    incr = 1;
    for kk = cell_nvertex(jcell2):-1:1
        cell_vertex(jcell2,kk+incr) = cell_vertex(jcell2,kk);
        cell_cell(jcell2,kk+incr) = cell_cell(jcell2,kk);
        if (cell_vertex(jcell2,kk) == vertex_list(k2))

```



```

        incr = 0;
        cell_vertex(jcell2, kk+incr) = kvertex2;
        cell_cell(jcell2, kk+incr) = jcell;
    end
end
cell_nvertex(jcell2) = cell_nvertex(jcell2) + 1;
end

% update generation and set target area and growth rate of new cells
cell_generation(jcell) = cell_generation(jcell) + 1;
cell_generation(jcell3) = cell_generation(jcell);
cell_growthrate(jcell)=1+(rand/2);
cell_growthrate(jcell3)=1+(rand/2);
% cell_growthrate(jcell)=2;
% cell_growthrate(jcell3)=2;

nvertices=max(max(cell_vertex(:,:)));
for i=1:nvertices
    k=1;
    for j=1:ncell
        if (k<4) && (ismember(i, cell_vertex(j,:)))
            vertex_cell(i,k)=j;
            k=k+1;
        end
    end
end
end

% Assign initial target area to cells
cell_area0(jcell) = tm_area(jcell);
cell_area0(jcell3) = tm_area(jcell3);

```

A.1.4 The Code for T1 and T2 Transitions

```

function tm_t1
% attempt T01 transitions

global xmax ymax
global ncell nvertex
global vertex_xy vertex_cell
global cell_vertex cell_nvertex cell_cell cell_area0 cell_generation
global kappa2 lambda gamma2 area0 delta delta_t1 area_t2

% total number of attempts per time step equal to number of cells
for j = 1:ncell;

    %choose a random vertex of a random cell

    jcell=j;
%    jcell = floor(rand*ncell) + 1;
    k = floor(rand*cell_nvertex(jcell)) + 1;
    kvertex = cell_vertex(jcell,k);

    %attempt T1 if edge is short
    if tm_edge(jcell,k) < delta_t1
        k1 = k + 1;
    end
end

```

```

if k1 > cell_nvertex(jcell)
    k1 = 1;
end
kvertex1 = cell_vertex(jcell,k1);

if (nnz(vertex_cell(kvertex,:))==3) && (nnz(vertex_cell(kvertex1,:))==3)
    % calc initial energy of the 3 cells associated with the vertex
    energy0 = 0;
    for j = 1:3
        energy0 = energy0 + tm_energy(vertex_cell(kvertex,j));
    end

    % save original coords and move vertex to position of neighbour
    xy0 = vertex_xy(kvertex,:);
    vertex_xy(kvertex,:) = vertex_xy(kvertex1,:);

    % calc final energy of the 3 cells associated with the vertex
    energy = 0;
    for j = 1:3
        energy = energy + tm_energy(vertex_cell(kvertex,j));
    end

    % reject move according to Boltzmann probability, if energy increases
    if (energy > energy0) && (rand > exp(energy0-energy))
        vertex_xy(kvertex,:) = xy0;

    % otherwise do T1
    else
        'T1'
        beep

        % identify cells that will receive a new vertex
        k2 = k - 1;
        if k2 < 1
            k2 = cell_nvertex(jcell);
        end
        jcell1 = cell_cell(jcell,k1);
        jcell2 = cell_cell(jcell,k2);
        % and cell that will lose a vertex (in addition to jcell)
        jcell3 = cell_cell(jcell,k);

        % add vertex kvertex to jcell1 before kvertex1
        % assign jcell2 as new neighbour of jcell1
        incr = 1;
        for kk = cell_nvertex(jcell1):-1:1
            cell_vertex(jcell1,kk+incr) = cell_vertex(jcell1,kk);
            cell_cell(jcell1,kk+incr) = cell_cell(jcell1,kk);
            if cell_vertex(jcell1,kk) == kvertex1
                incr = 0;
                cell_vertex(jcell1,kk+incr) = kvertex;
                cell_cell(jcell1,kk+incr) = jcell2;
            end
        end
        end
        cell_nvertex(jcell1) = cell_nvertex(jcell1) + 1;

        % add vertex kvertex1 to jcell2 before kvertex

```

```

% assign jcell1 as new neighbour of jcell2
incr = 1;
for kk = cell_nvertex(jcell2):-1:1
    cell_vertex(jcell2, kk+incr) = cell_vertex(jcell2, kk);
    cell_cell(jcell2, kk+incr) = cell_cell(jcell2, kk);
    if cell_vertex(jcell2, kk) == kvertex
        incr = 0;
        cell_vertex(jcell2, kk+incr) = kvertex1;
        cell_cell(jcell2, kk+incr) = jcell1;
    end
end
cell_nvertex(jcell2) = cell_nvertex(jcell2) + 1;

% remove vertex kvertex from jcell3
if cell_vertex(jcell3, 1) == kvertex
    cell_vertex(jcell3, 1) = cell_vertex(jcell3, cell_nvertex(jcell3));
    % don't shift, to match cell_cell
else
    incr = 0;
    for kk = 1:cell_nvertex(jcell3)
        cell_vertex(jcell3, kk+incr) = cell_vertex(jcell3, kk);
        if cell_vertex(jcell3, kk) == kvertex
            incr = -1;
        end
    end
end
cell_vertex(jcell3, cell_nvertex(jcell3)) = 0;
% remove cell jcell from neighbours of jcell3
incr = 0;
for kk = 1:cell_nvertex(jcell3)
    cell_cell(jcell3, kk+incr) = cell_cell(jcell3, kk);
    if cell_cell(jcell3, kk) == jcell
        incr = -1;
    end
end
cell_cell(jcell3, cell_nvertex(jcell3)) = 0;
cell_nvertex(jcell3) = cell_nvertex(jcell3) - 1;

% remove vertex kvertex1 from jcell
if cell_vertex(jcell, 1) == kvertex1
    cell_vertex(jcell, 1) = cell_vertex(jcell, cell_nvertex(jcell));
    % don't shift, to match cell_cell
else
    incr = 0;
    for kk = 1:cell_nvertex(jcell)
        cell_vertex(jcell, kk+incr) = cell_vertex(jcell, kk);
        if cell_vertex(jcell, kk) == kvertex1
            incr = -1;
        end
    end
end
cell_vertex(jcell, cell_nvertex(jcell)) = 0;
% remove cell jcell3 from neighbours of jcell
incr = 0;
for kk = 1:cell_nvertex(jcell)
    cell_cell(jcell, kk+incr) = cell_cell(jcell, kk);

```

```

        if cell_cell(jcell,kk) == jcell3
            incr = -1;
        end
    end
    cell_cell(jcell,cell_nvertex(jcell)) = 0;
    cell_nvertex(jcell) = cell_nvertex(jcell) - 1;

    % reassign adjacent cells to vertices
    vertex_cell(kvertex,1) = jcell;
    vertex_cell(kvertex,2) = jcell2;
    vertex_cell(kvertex,3) = jcell1;
    vertex_cell(kvertex1,1) = jcell1;
    vertex_cell(kvertex1,2) = jcell2;
    vertex_cell(kvertex1,3) = jcell3;

end
end
end
end

function tm_t2
% perform T2 transition for all small triangular cells

global vertex_xy vertex_cell
global ncell nvertex
global cell_vertex cell_nvertex cell_cell cell_area0 cell_generation cell_energy
global kappa2 lambda gamma2 delta delta_t1 area_t2 t2_counter

%total number of attempts per time step equal to number of cells
for jcell=1:ncell

    %do T2 if cell is triangular and smaller than the threshold area
    if cell_nvertex(jcell)==3 && tm_area(jcell) < area_t2
        'T2', jcell
        vertex_list=sort(cell_vertex(jcell,1:3));

        %create new vertex at centroid of cell, and index as vertex_list(1)
        [xcm, ycm] = tm_centroid(jcell);
        vertex_xy(vertex_list(1),1) = xcm;
        vertex_xy(vertex_list(1),2) = ycm;

        %identify neighbouring cells and update vertex_cell for the new vertex
        cell_list=cell_cell(jcell,1:3)
        vertex_cell(vertex_list(1),:) = cell_list;

        %remove the vertex corresponding to jcell from each of the neighbouring cells
        %and relabel the following vertex as vertex_list(1)
        %also remove jcell from the neighbour-list of neighbouring cells
        for jj = 1:3
            incr = 0;
            if cell_list(jj)~=0
                for k = 1:cell_nvertex(cell_list(jj))-1
                    if cell_cell(cell_list(jj),k) == jcell
                        cell_vertex(cell_list(jj),k) = vertex_list(1);
                        incr = 1;
                    end
                end
            end
        end
    end
end
end

```

```

        cell_cell(cell_list(jj),k) = cell_cell(cell_list(jj),k+incr);
    else
        cell_vertex(cell_list(jj),k) = cell_vertex(cell_list(jj),k+incr);
        cell_cell(cell_list(jj),k) = cell_cell(cell_list(jj),k+incr);
    end
end
end
%wrap around if vertex corresponding to jcell was at end of array
if cell_cell(cell_list(jj),cell_nvertex(cell_list(jj))) == jcell
    cell_vertex(cell_list(jj),1) = vertex_list(1);
end
cell_vertex(cell_list(jj),cell_nvertex(cell_list(jj))) = 0;
cell_cell(cell_list(jj),cell_nvertex(cell_list(jj))) = 0;
cell_nvertex(cell_list(jj)) = cell_nvertex(cell_list(jj)) - 1;
end
end

%update total number of vertices
nvertex = nvertex - 2;

%left with two spaces for vertices at vertex_list(2:3) to fill in from top of old list
%(if they are not above top of new list)
if vertex_list(2) <= nvertex
    if vertex_list(3) ~= nvertex+2
        vertex_sub = nvertex+2;
    else
        vertex_sub = nvertex+1;
    end
    %substitute vertex in list
    vertex_xy(vertex_list(2),:) = vertex_xy(vertex_sub,:);
    vertex_cell(vertex_list(2),:) = vertex_cell(vertex_sub,:);
    %adjust label in cell_vertex
    for j = 1:ncell
        for k = 1:cell_nvertex(j)
            if cell_vertex(j,k) == vertex_sub
                cell_vertex(j,k) = vertex_list(2);
            end
        end
    end
end
if vertex_list(3) <= nvertex
    vertex_sub = nvertex+1;
    %substitute vertex in list
    vertex_xy(vertex_list(3),:) = vertex_xy(vertex_sub,:);
    vertex_cell(vertex_list(3),:) = vertex_cell(vertex_sub,:);
    %adjust label in cell_vertex
    for j = 1:ncell
        for k = 1:cell_nvertex(j)
            if cell_vertex(j,k) == vertex_sub
                cell_vertex(j,k) = vertex_list(3);
            end
        end
    end
end
%clear the two vertices removed from top of list (not strictly necessary)
vertex_xy(nvertex+1,:) = 0;
vertex_xy(nvertex+2,:) = 0;

```

```

vertex_cell(nvertex+1,:) = 0;
vertex_cell(nvertex+2,:) = 0;

%update total number of cells and energies of neighbours of vanished cell
ncell = ncell - 1;
for jj = 1:3
    if cell_list(jj)~=0
        cell_energy(cell_list(jj)) = tm_energy(cell_list(jj));
    end
end

%left with a space for a cell at jcell to fill in from top of old list
%(if it is not above top of new list)
if jcell <= ncell
    %substitute cell in list
    cell_vertex(jcell,:)=cell_vertex(ncell+1,:);
    cell_nvertex(jcell)=cell_nvertex(ncell+1);
    cell_cell(jcell,:) = cell_cell(ncell+1,:);
    cell_area0(jcell)=cell_area0(ncell+1);
    cell_generation(jcell)=cell_generation(ncell+1);
    cell_energy(jcell)=cell_energy(ncell+1);
    %adjust label in vertex_cell and cell_cell
    for k=1:nvertex
        for j = 1:3
            if vertex_cell(k,j) == ncell+1
                vertex_cell(k,j) = jcell;
            end
        end
    end
    for j = 1:ncell
        for k = 1:cell_nvertex(j)
            if cell_cell(j,k) == ncell+1
                cell_cell(j,k) = jcell;
            end
        end
    end
end
%clear the cell removed from top of list (not strictly necessary)
cell_vertex(ncell+1,:)=0;
cell_nvertex(ncell+1)=0;
cell_cell(ncell+1,:)=0;
cell_area0(ncell+1)=0;
cell_generation(ncell+1)=0;
cell_energy(ncell+1)=0;

t2_counter=t2_counter+1;

end
end

```

A.1.5 The Code for Plotting Histograms of Cell Neighbour Numbers

```

% Create matrix to store CNN
Averagecnn=zeros(5,4,10);
Avgcnn=zeros(5,10);
Normalizedcnn=zeros(5,4,10,18);

```

```

nccCounts=zeros(5,4,18);
FinalnccCounts=zeros(5,18);
nccCounts2=zeros(5,10,18);

for n=1:5
    nphi=79+n;
    for nsimulation=1:4
        % load data file
        name=['tm-results-',num2str(nphi),'-',num2str(nsimulation)];
        load (name);

        % Merge cnn from different simulations
        for t=6:15
            nvertex=cellnvertex(t,:);
            nvertex=nvertex(find(nvertex));
            Averagecnn(n,nsimulation,t)=mean(nvertex);
            nbins=3:1:20;
            [Normalizedcnn(n,nsimulation,t,:),xout] = hist(nvertex,nbins);
            Normalizedcnn(n,nsimulation,t,:)=Normalizedcnn(n,nsimulation,t,:)/noCells(t);
        end
    end
end

%Average normalized cnn over all time steps
for n=1:5
    for nsimulation=1:4
        for cc=1:18
            nccCounts(n,nsimulation,cc)=mean(Normalizedcnn(n,nsimulation,:,cc));
        end
    end
end

%Average normalized cnn over simulations
for n=1:5
    for cc=1:18
        FinalnccCounts(n,cc)=mean(nccCounts(n,:,cc));
    end
end

%Average normalized cnn over simulations
for n=1:5
    for t=1:10
        for cc=1:18
            nccCounts2(n,t,cc)=mean(Normalizedcnn(n,:,t,cc));
        end
    end
end

% %Average mean cnn over simulations
% for n=1:5
%     for t=1:10
%         Avgcnn(n,t)=mean(Averagecnn(n,:,t));
%     end
% end
%%
%=====

```

```

% Plot histogram
%-----
for n=1:5
    x=[3:1:20];
    m=zeros(1,18);
    m(1,:)=FinalnccCounts(n,:);
    cmap = hsv(5);
    figure;bar(x,m,0.5);
%     bar(m,...
%         'bar_color',cmap(n,:));
    clear title xlabel ylabel
    set(gca,'XTick',2:1:15)
    set(gca,'XLim',[2 15])
    set(gca,'YLim',[0 0.5])
    xlabel('CNN')
    ylabel('Distribution')
    name=['tm_CNN_',num2str(n+79)];
    legendname=['\Phi_{MAX}=',num2str(n-1)];
    legend(legendname,'Location','NE');
    print ('-depsc','-r300',name);
end

%%
%=====
% Plot Average cnn over timesteps
%-----
for n=1:5
    hold on;
    m=zeros(1,10);
    m(1,:)=Averagecnn(n,1,:);
    cmap = hsv(5);
    plot(m,'-rs','LineWidth',1.1,...
        'Color',cmap(n,:),...
        'MarkerEdgeColor',cmap(n,:),...
        'MarkerFaceColor',cmap(n,:),...
        'MarkerSize',3);figure(gcf)
    clear title xlabel ylabel
    set(gca,'XTick',1:1:10)
    xlabel('Time Step')
    ylabel('Average CNN')
    name=['tm_average_CNN'];
    legend(['\Phi_{MAX}=0','\Phi_{MAX}=1','\Phi_{MAX}=2','\Phi_{MAX}=3',...
        '\Phi_{MAX}=4','Location','NE']);
    print ('-depsc','-r300',name);
end

%%
%=====
% Plot time evolution of cnn
%-----
for n=1:5
    cmap = hsv(8);
    hold off;
    for cc=2:9
        m=zeros(1,10);
        m(1,:)=nccCounts2(n,:,cc);

```



```

plot(m,'-rs','LineWidth',1.1,...
      'Color',cmap(cc-1,:),...
      'MarkerEdgeColor',cmap(cc-1,:),...
      'MarkerFaceColor',cmap(cc-1,:),...
      'MarkerSize',3);figure(gcf)

clear title xlabel ylabel
set(gca,'XTick',1:1:10)
xlabel('Time Step')
ylabel('Average CNN')
name=['tm_CNN_time_evolution_',num2str(n+79)];
legend('CNN=4','CNN=5','CNN=6','CNN=7','CNN=8','CNN=9','CNN=10',...
      'CNN=11','Location','NE');
print ('-depsc','-r300',name);
hold on
end
end

```

A.1.6 The Code for Determining Cells' Ellipses of Inertia

```

%=====
% Quantify the shape of the cells by measuring their ellipse of inertia
%-----
%%
tmax=15;
% Define an array to store the parameters of ellipse of inertia
EllipseofInertia=zeros(5,4,15,100,5);
AxisRatio=zeros(5,4,15,100);
meanratio=zeros(5,4,15);
mratio=zeros(5,1,15);
meantheta=zeros(5,4,15);
mtheta=zeros(5,1,15);
%%
% Determine Cells' ellipses of inertia
for n=1:1
    nphi=79+n;
    for nsimulation=1:1
        % load data file
        name=['tm-results-',num2str(nphi),'-',num2str(nsimulation)];
        load (name);

        % Find the ellipse of inertia
        for t=1:1
            for jcell=1:noCells(t)
                % Calculate the coordination of the edges of the cells
                xy=zeros(cellInvertex(t,jcell),2);
                for k = 1:cellInvertex(t,jcell)
                    xy(k,1) = vertexxy(t,cellvertex(t,jcell,k),1);
                    xy(k,2) = vertexxy(t,cellvertex(t,jcell,k),2);
                end
                EllipseofInertia(n,nsimulation,t,jcell,:) = inertiaEllipse(xy);
                ell = EllipseofInertia(n,nsimulation,t,jcell,:);
                figure(1); clf; hold on;
                drawPoint(xy);
                drawEllipse(ell, 'linewidth', 2, 'color', 'r');
                pause(0.5)
            end
        end
    end
end

```

```

        end
    end

    % Calculate ratio of long to short axis
    for nsimulation=1:4
        for t=1:tmax
            for jcell=1:noCells(t)
                ax1=EllipseofInertia(n,nsimulation,t,jcell,3);
                ax2=EllipseofInertia(n,nsimulation,t,jcell,4);
                AxisRatio(n,nsimulation,t,jcell)=ax1/ax2;
            end
        end
    end

    % Calculate average of ratios at each time step of simulation
    for nsimulation=1:4
        for t=1:15
            ratio=0;
            theta=0;
            ntotal1=0;
            ntotal2=0;
            for jcell=1:noCells(t)
                if AxisRatio(n,nsimulation,t,jcell)>0
                    ratio=ratio+AxisRatio(n,nsimulation,t,jcell);
                    ntotal1=ntotal1+1;
                end
                if EllipseofInertia(n,nsimulation,t,jcell,5)>-100
                    theta=theta+EllipseofInertia(n,nsimulation,t,jcell);
                    ntotal2=ntotal2+1;
                end
            end
            if ntotal1~=0
                meanratio(n,nsimulation,t)=ratio/ntotal1;
                meantheta(n,nsimulation,t)=theta/ntotal2;
            end
        end
    end

    % Average ratios over 4 simulations
    for t=1:15
        mratio(n,1,t)=mean(meanratio(n,:,t));
        mtheta(n,1,t)=mean(meantheta(n,:,t));
    end
end

%%
%=====
% Plot mean ratio of long to short axis
%-----
for n=1:5
    hold on;
    m=zeros(1,15);
    m(1,:)=mratio(n,1,:);
    cmap = hsv(5);
    plot(m,'-rs','LineWidth',1.1,...
        'Color',cmap(n,:),...
        'MarkerEdgeColor',cmap(n,:),...)
end

```

```

        'MarkerFaceColor',cmap(n,:),...
        'MarkerSize',3);figure(gcf)
clear title xlabel ylabel
set(gca,'XTick',1:1:15)
xlabel('Time Step')
ylabel('Average Ratio of Long axis to Short Axis of Cells Ellipses of Inertia')
name=['tm_axis_ratio_cellEoI'];
legend('\Phi_{MAX}=0','\Phi_{MAX}=1','\Phi_{MAX}=2','\Phi_{MAX}=3',...
'\Phi_{MAX}=4','Location','NW');
print ('-depsc','-r300',name);
end
%%
%=====
% Plot mean angle of ellipse of inertia with horizontal axis
%-----
for n=1:5
    hold on;
    m=zeros(1,15);
    m(1,:)=mtheta(n,1,:);
    cmap = hsv(5);
    plot(m,'-rs','LineWidth',1.1,...
        'Color',cmap(n,:),...
        'MarkerEdgeColor',cmap(n,:),...
        'MarkerFaceColor',cmap(n,:),...
        'MarkerSize',3);figure(gcf)
    clear title xlabel ylabel
    ylim([4 6])
    set(gca,'XTick',1:1:15)
    xlabel('Time Step')
    ylabel('Theta')
    name=['tm_theta_cellEoI'];
    legend('\Phi_{MAX}=0','\Phi_{MAX}=1','\Phi_{MAX}=2','\Phi_{MAX}=3',...
'\Phi_{MAX}=4','Location','SW');
    print ('-depsc','-r300',name);
end

```

A.1.7 The Code for Measuring Growth Rates of the Tissue

```

%=====
% Measure growth rates in different directions
%-----
%%
tmax=15;
% Define an array to store the parameters of ellipse of inertia
gx=zeros(5,4,15);
gy=zeros(5,4,15);
mgx=zeros(5,1,15);
mgy=zeros(5,1,15);

for n=1:5
    nphi=79+n;
    for nsimulation=1:4
        % load data file
        name=['tm-results-',num2str(nphi),'-',num2str(nsimulation)];
        load (name);
    end
end

```

```

% Find the coordinates of vertices
for t=2:tmax
    x2=vertexxy(t,:,1);
    x2=x2(find(x2));

    x1=vertexxy(t-1,:,1);
    x1=x1(find(x1));

    y2=vertexxy(t,:,2);
    y2=y2(find(y2));

    y1=vertexxy(t-1,:,2);
    y1=y1(find(y1));

    gx(n,nsimulation,t)=(max(x2)-min(x2))-(max(x1)-min(x1));
    gy(n,nsimulation,t)=(max(y2)-min(y2))-(max(y1)-min(y1));
end
end

% Average growth rates
for t=1:tmax
    mgx(n,1,t)=mean(gx(n,:,t));
    mgy(n,1,t)=mean(gy(n,:,t));
end
end

%%
%=====
% Plot growth rate in horizontal direction
%-----
for n=1:5
    hold on;
    m=zeros(1,15);
    m(1,:)=mgx(n,1,:);
    cmap = hsv(5);
    plot(m,'-rs','LineWidth',1.1,...
        'Color',cmap(n,:),...
        'MarkerEdgeColor',cmap(n,:),...
        'MarkerFaceColor',cmap(n,:),...
        'MarkerSize',3);figure(gcf)
    clear title xlabel ylabel
    set(gca,'XTick',1:1:15)
    ylim([-10 10])
    xlabel('Time Step')
    ylabel('Growth Rate in Horizontal Direction')
    name=['tm_mgx_tissueEoI'];
    legend('\Phi_{MAX}=0','\Phi_{MAX}=1','\Phi_{MAX}=2','\Phi_{MAX}=3',...
        '\Phi_{MAX}=4','Location','NE');
    print ('-depsc','-r300',name);
end
hold off;
%%
%=====
% Plot growth rate in vertical direction
%-----
for n=1:5

```

```

hold on;
m=zeros(1,15);
m(1,:)=mgy(n,1,:);
cmap = hsv(5);
plot(m,'-rs','LineWidth',1.1,...
      'Color',cmap(n,:),...
      'MarkerEdgeColor',cmap(n,:),...
      'MarkerFaceColor',cmap(n,:),...
      'MarkerSize',3);figure(gcf)

clear title xlabel ylabel
set(gca,'XTick',1:1:15)
xlabel('Time Step')
ylabel('Growth Rate in Vertical Direction')
name=['tm_mgy_tissueEoI'];
legend('\Phi_{MAX}=0','\Phi_{MAX}=1','\Phi_{MAX}=2','\Phi_{MAX}=3',...
      '\Phi_{MAX}=4','Location','NE');
print ('-depsc','-r300',name);
end

```

A.2 Snapshots of Simulations

Below, the snapshots of the simulations of growth under anisotropic tension for three values of Φ_{MAX} are presented.

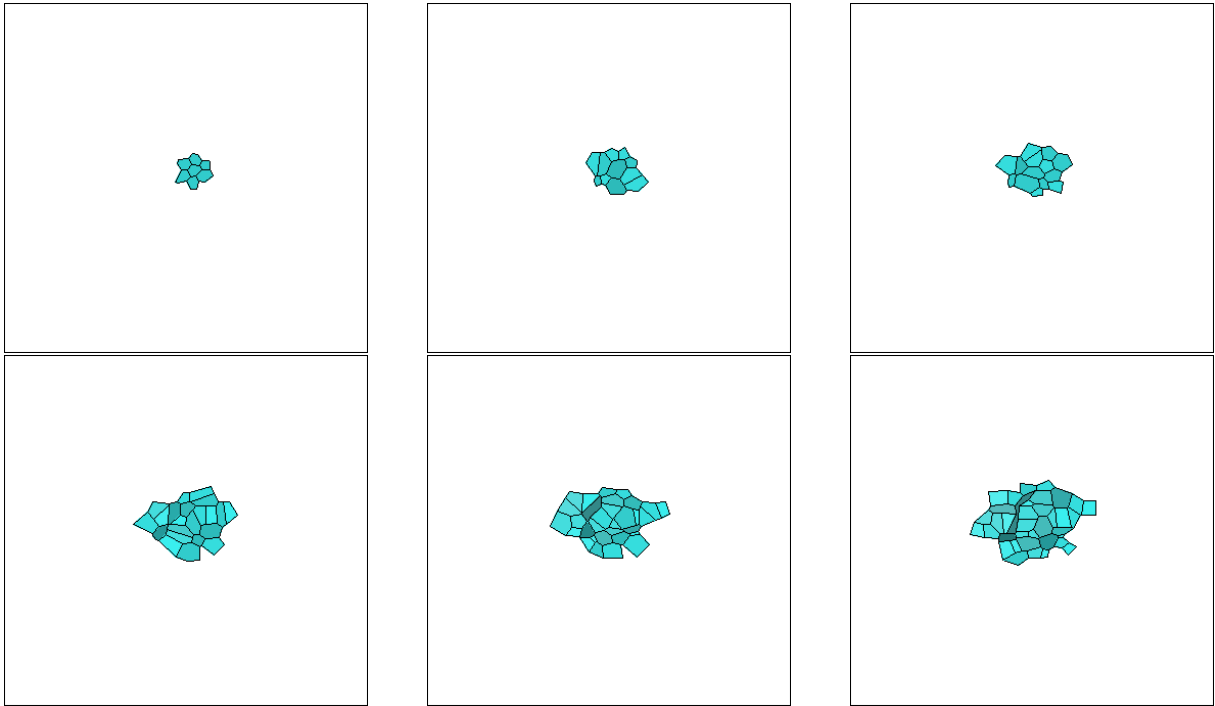


Figure 26: Snapshots of the growth of the tissue under anisotropic tension $\Phi_{MAX} = 1$.

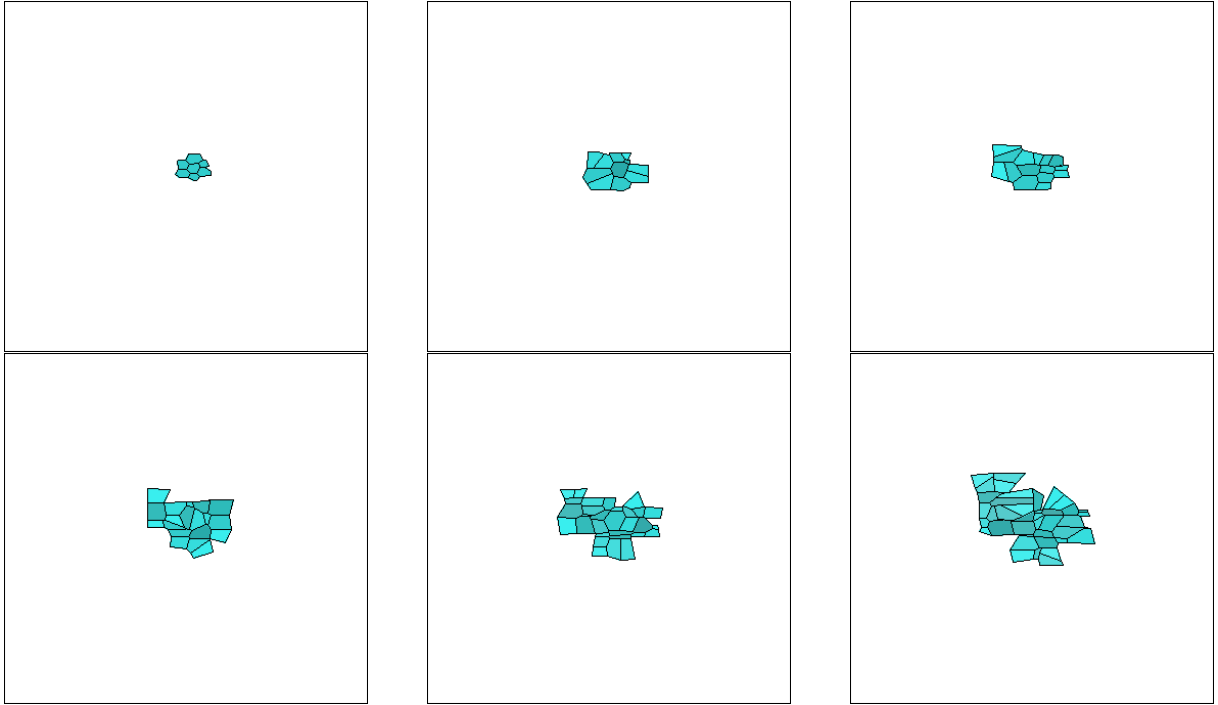


Figure 27: Snapshots of the growth of the tissue under anisotropic tension $\Phi_{MAX} = 3$.

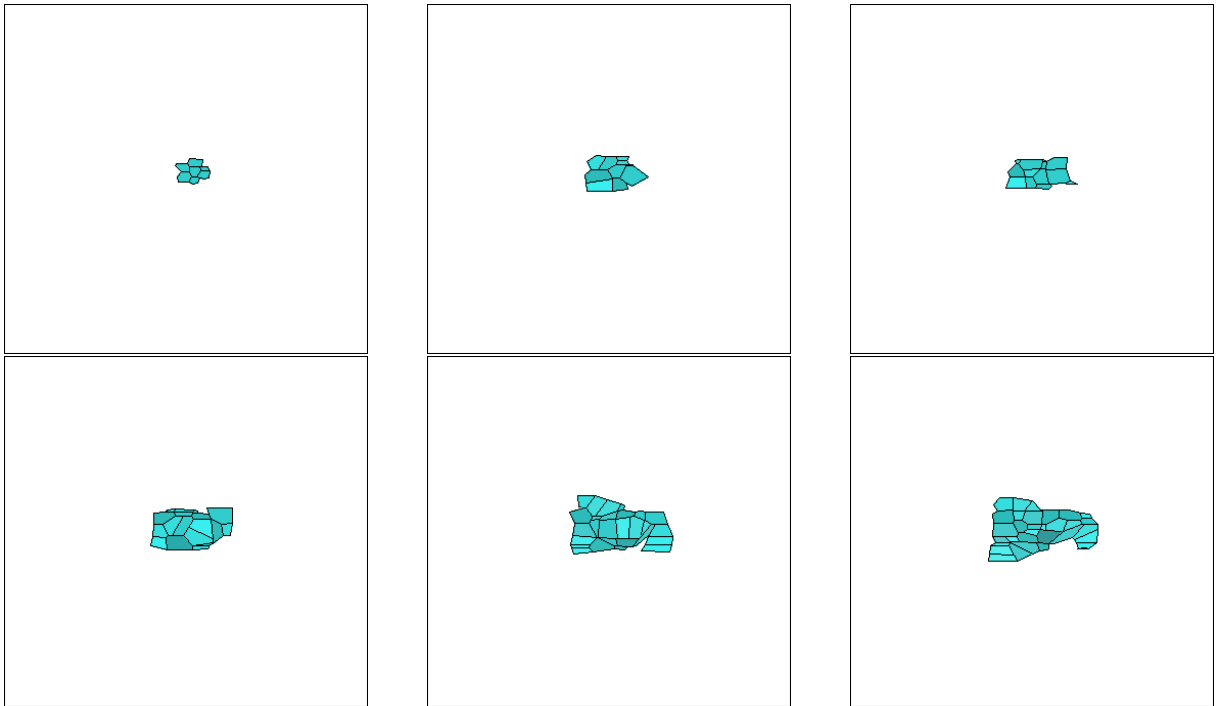


Figure 28: Snapshots of the growth of the tissue under anisotropic tension $\Phi_{MAX} = 4$.

References

- [1] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. Garland Science, 5th edition, 2008.
- [2] Bruce Alberts, Dennis Bray, Karen Hopkin, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Essential Cell Biology*. Garland Science, 3rd edition, 2010.
- [3] J Andrew Bangham Anne Gaele Rolland-Lagan and Enrico Coen. Growth dynamics underlying petal shape and asymmetry. *Nature*, 422:161–163, 2003. doi: 10.1038/nature01443.
- [4] Bernd Boehm, Henrik Westerberg, Gaja Lesnicar-Pucko, Sahdia Raja, Michael Rautschka, James Cotterell, Jim Swoger, and James Sharpe. The role of spatially controlled cell proliferation in limb bud morphogenesis. *PLoS Biology*, 8(7), 2010. doi: 10.1371/journal.pbio.1000420.
- [5] G Wayne Brodland and Jim H Veldhuis. Computer simulations of mitosis and interdependencies between mitosis orientation, cell shape and epithelia reshaping. *Journal of Biomechanics*, 35:673–681, 2002.
- [6] Helen H Chen and G Wayne Brodland. Cell-level finite element studies of viscous cells in planar aggregates. *ASME Journal of Biomechanical Engineering*, 122:394–401, 2000. doi: 10.1115/1.1286563.
- [7] Lawrence Sulak Claire Bertet and Thomas Lecuit. Myosin-dependent junction remodelling controls planar cell intercalation and axis elongation. *Nature*, 429:667–671, 2004. doi: 10.1038/nature02590.
- [8] M.L. Concha and R.J. Adams. Oriented cell divisions and cellular morphogenesis in the zebrafish gastrula and neurula: a time-lapse analysis. *Development*, 125(983-994), 1998.
- [9] Sara Morais da Silva and Jean-Paul Vincent. Oriented cell divisions in the extending germband of drosophila. *Development*, 134:3049–3054, 2007. doi: 10.1242/dev.004911.
- [10] Jacques Dumais and Dorota Kwiatkowska. Analysis of surface growth in shoot apices. *The Plant Journal*, 31(2):229–241, 2002.
- [11] Andrew D Economou, Atsushi Ohazama, Thantrira Porntaveetus, Paul T Sharpe, Shigeru Kondo, M Albert Basson, Amel Gritli-Linde, Martyn T Cobourne, and Jeremy B A Green. Periodic stripe formation by a turing mechanism operating at growth zones in the mammalian palate. *Nature Genetics*, 44(3):348–352, 2012. doi: 10.1038/ng.1090.
- [12] E Evans and A Yeung. Apparent viscosity and cortical tension of blood granulocytes determined by micropipet aspiration. *Biophysical Journal*, 56:151–160, 1989. doi: 10.1016/S0006-3495(89)82659-1.
- [13] Reza Farhadifar, Jens-Christian Roper, Benoit Aigouy, Suzanne Eaton, and Frank Juliche. The influence of cell mechanics, cell-cell interactions, and proliferation on epithelial packing. *Current Biology*, 17:2095–2104, 2007. doi: 10.1016/j.cub.2007.11.049.
- [14] The National Center for Biotechnology Information, 16 August 2012 . URL <http://www.ncbi.nlm.nih.gov/sites/entrez?Db=gene&Cmd=ShowDetailView&TermToSearch=2253>.
- [15] Matthew C. Gibson, Ankit B. Patel, Radhika Nagpal, and Norbert Perrimon. The emergence of geometric order in proliferatin metazoan epithelia. *Nature*, 442:1038–1041, 2006. doi: 10.1038/nature05014.
- [16] William T. Gibson, James H. Veldhius, Boris Rubinstein, Heather N. Cartwright, Norbert Perrimon, G. Wayne Brodland, Radhika Nagpal, and Matthew C. Gibson. Control of the mitotic cleavage plane by local epithelial topology. *Cell*, 144:427–438, 2011. doi: 10.1016/j.cell.2010.12.035.
- [17] Francois Graner and James A Glazier. Simulation of biological cell sorting using a two-dimensional extended potts model. *Physical Review Letters*, 69(13):2013–2017, 1992. doi: 10.1103/PhysRevLett.69.2013.
- [18] Z Hejnowicz. Vector and scalar fields in modeling of spatial variations of growth rates within plant organs. *Journal of Theoretical Biology*, 96:161–173, 1982.

- [19] Z Hejnowicz. Growth tensor of plant organs. *Journal of Theoretical Biology*, 110:93–114, 1984.
- [20] Lars Hufnagel, Aurelio A. Teleman, Herve Rouault, Stephen M. Cohen, and Boris I. Shraiman. On the mechanism of wing size determination in fly development. *PNAS*, 104(10):3835–3840, 2007.
- [21] Jos Kafer, Takashi Hayashi, Athanasius F M Maree, Richard W Carthew, and Francois Graner. Cell adhesion and cortex contractility determine cell patterning in the drosophila retina. *PNAS*, 104(47):18549–18554, 2007. doi: 10.1073/pnas.0704235104.
- [22] Ray Keller. Mechanisms of elongation in embryogenesis. *Development*, 133:2291–2302, 2006. doi: 10.1242/dev.02406.
- [23] Thomas Lecuit. Adhesion remodeling underlying tissue morphogenesis. *TRENDS in Cell Biology*, 15(1):34–42, 2005. doi: 10.1016/j.tcb.2004.11.007.
- [24] Thomas Lecuit and Loic Le Goff. Orchestrating size and shape during morphogenesis. *Nature*, 450:189–192, 2007. doi: 10.1038/nature06304.
- [25] Thomas Lecuit and Pierre-Francois Lenne. Cell surface mechanics and the control of cell shape, tissue patterns and morphogenesis. *Nature Reviews Molecular Cell Biology*, 8:633–644, 2007. doi: 10.1038/nrm2222.
- [26] Alfons T.L. Van Lommel. *From cells to organs: A histology textbook and atlas*. Luwer Academic Publishers, Boston, 2002.
- [27] Yanlan Mao, Alexander L Tournier, Paul A ates, Jonathan E Gale, Nicolas Tapon, and Barry J Thompson. Planar polarization of the atypical myosin dachs orients cell divisions in drosophila. *Genes Development*, 25:131–136, 2011. doi: 10.1101/gad.610511.
- [28] Eliana Marinari, Aida Mehonic, Scott Curran, Jonathan Gale, Thomas Duke, and Buzz Baum. Live-cell delamination counterbalances epithelial growth to limit tissue overcrowding. *Nature*, 484:542–546, 2012. doi: 10.1038/nature10984.
- [29] Fernando Martin-Belmonte and colleagues. How epithelial tissues make single-hole tubes, 2012. URL <http://www.hfsp.org/frontier-science/awardees-articles/how-epithelial-tissues-make-single-hole-tubes>.
- [30] Aida Mehonic. *The role of physics in epithelial homeostasis and development*. PhD thesis, Department of Physics and Astronomy, University College London, August 2011.
- [31] Ali Nawshad, Damian Medici, Chang-Chi Liu, and Elizabeth D Hay. Tgf3 inhibits e-cadherin gene expression in palate medial-edge epithelial cells through a smad2-smad4-lef1 transcription complex. *Journal of Cell Science*, 120:1646–1653, 2007.
- [32] Sophie Pantalacci, Jan Prochazka, Arnaud Martin, Michaela Rothova, Anne Lambert, Laure Bernard, Cyril Charles, Laurent Viriot, Renata Peterkova, and Vincent Laudet. Patterning of palatal rugae through sequential addition reveals an anterior/posterior boundary in palatal development. *BMC Developmental Biology*, 8:116, 2008. doi: 10.1186/1471-213X-8-116.
- [33] Nikodem J Poplawski, Maciej Swas, J Scott Gens, and James A Glazier. Adhesion between cells, diffusion of growth factors, and elasticity of the aer produce the paddle shape of the chick limb. *Physica A*, 373:521–532, 2007. doi: 10.1016/j.physa.2006.05.028.
- [34] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: the Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.
- [35] Matteo Rauzi, Pascale Verant, Thomas Lecuit, and Pierre-Francois Lenne. Nature and anisotropy of cortical forces orienting drosophila tissue morphogenesis. *Nature Cell Biology*, 10(12):1401–1410, 2008. doi: 10.1038/ncb1798.
- [36] Anne Gaelle Rolland-Lagan, Enrico Coen, Stephen J. Impey, and J. Andrew Bangham. A computational method for inferring growth parameters and shape changes during development based on clonal analysis. *Journal of Theoretical Biology*, 232:157–177, 2005.

- [37] Sebastian A. Sandersius, Manli Chuai, Cornelis J. Weijer, and Timothy J. Newman. Correlating cell behavior with tissue topology in embryonic epithelia. *PLoS ONE*, 6(4):e18081, 2011. doi: 10.1371/journal.pone.0018081.
- [38] Georgios Trichas, Aaron M Smith, Natalia White, Vivienne Wilkins, Tomoko Watanabe, Abigail Moore, Bradley Joyce, Jacintha Sugnaseelan, Tristan A Rodriguez, David Kay, Ruth E Baker, Philip K Maini, and Shankar Srinivas. Multi-cellular rosettes in the mouse visceral endoderm facilitate the ordered migration of anterior visceral endoderm cells. *PLoS Biology*, 10(2):e1001256, 2012. doi: 10.1371/journal.pbio.1001256.
- [39] Ian C Welsh and Timothy P O'Brien. Signaling intergration in the rugae growth zone directs sequential shh signaling center formation during the rostral outgrowth of the palate. *Developmental Biology*, 336:53–67, 2009. doi: 10.1016/j.ydbio.2009.09.028.
- [40] Jennifer A Zallen. Planar polarity and tissue morphogenesis. *Cell*, 129:1051–1063, 2007. doi: 10.1016/j.cell.2007.05.050.