AN EXTENSION FOR VIRTUALLEAF TO EXPLORE THE MECHANICAL DIVISION RULE IN PLANT MORPHOGENESIS

MSc Thesis - PCC80424

Aimée R. Kok

Student ID: 1031593

Supervisors:
prof. Jasper van der Gucht
prof. Roeland Merks
prof. Joris Sprakel

Physical Chemistry and Soft Matter
Wageningen University and Research Centre
Stippeneng 4, 6708 WE, Wageningen, The Netherlands

August 2023

Abstract

Contents

1 Introduction

Plant cells are tightly adhered to each other by cell walls, a fibrous network structure that envelops each cell[1]. While these cell walls provide structural support to the plant tissue, they also severely restrict the mobility of the cells they encase. This restriction prevents changes in the tissue topology through cell rearrangement, a process commonly observed during the morphogenesis of animal tissues[2–4]. As a result, the shape and cell patterning of plant tissues solely emerge from the direction of cell division and the subsequent (an)isotropic growth of daughter cells[5,6]. While plant hormones[7], and cell polarity[8,9] play coordinating roles in regulating these morphogenic processes, recent attention has also been directed to the impact of mechanical cues[10–14].

A notable example of an intracellular structure affected by mechanical signals are cortical microtubules, a cytoskeletal filament. These microtubules serve as markers for the position of the preprophase band during cell division, which in turn indicates the site where the new cell wall will be inserted[15,16]. Although the exact mechanism by which microtubules sense tension remains unclear, they strikingly align with the maximal tensile stress experienced by cell walls[17–23]. In line with this observation, Louveaux et al. recently proposed that the orientation of the new cell wall might naturally follow the path of highest tensile stress experienced by the cell's walls[24,25]. This hypothesis brings a fresh perspective to the classical division rule, which relies exclusively on the geometry of the cell[26,27]. Specifically, this classical rule states that, in the absence of external stimuli, the new cell wall is formed along the shortest symmetric path through the cell, leading to two daughter cells of equal volume and minimizing the surface area of the inserted cell wall. While this geometric rule accurately predicts cell division in symmetric, proliferating tissues, it falls short when applied to tissues experiencing anisotropic stress, such as in folded boundaries[24,28,29]. The 'mechanical' division rule addresses this limitation, as it not only accounts for deviations from the shortest path caused by anisotropic stresses but also emulates the shortest path rule in situations where these tissue-wide stresses are absent.

Although a clear relation between mechanical stress and cell division orientation is evident, so far no systemic studies have been performed on, for example, the magnitude of tissue stress required to override the self-generated stresses governing division and growth, and how this interplay between local and global stresses changes as the tissue grows. Previous studies on plant cell division have always been performed within the context of a growing plant[24,28–30]. While this context is crucial for understanding in situ division patterns, with stress fields caused by relevant tissue geometries, it leaves very little control over physical parameters. Additionally, complex bio(mechano)chemical networks established within these tissues could bias division responses and make direct interpretation of mechanical cues more complicated. Therefore, to better understand the connection between mechanical stress and plant morphogenesis, especially in regards to cell division, we propose an experimental set-up in which we can study tissue growth in a highly controlled mechanical environment. Based on methods used for investigating cellular mechanoresponses in animal tissues[31–34], this set-up involves embedding plant protoplasts in a hydrogel matrix that is subjected to well-defined tensions. The Biochemistry group at Wageningen University and Research will perform these experiments.

In this report, we focus on developing a simulation method to support the experiments. Specifically, we introduce a software extension for VirtualLeaf, an open-source modeling framework for plant morphogenesis[35]. Our extension implements two key features: 1) a means for division based on mechanical stress, and 2) a mechanism for applying an anisotropic tensile stress to the tissue during its growth. Once complete, the simulations combined with the experiments would allow us to thoroughly investigate whether mechanical stress is indeed a universal predictor of the cell division plane, and how tissue morphologies evolve under increasing external mechanical stress.

We first introduce the general model and simulation workflow of the original VirtualLeaf framework. Next, we provide a detailed walkthrough of our extension VirtualLeaf, detailing the incorporation of the main goals and revision of some aspects of the original framework to improve the simulations for mechanical modeling. Finally, we outline future improvements for the extension. For those interested in using or further developing the extension, an appendix containing code-related details is included.

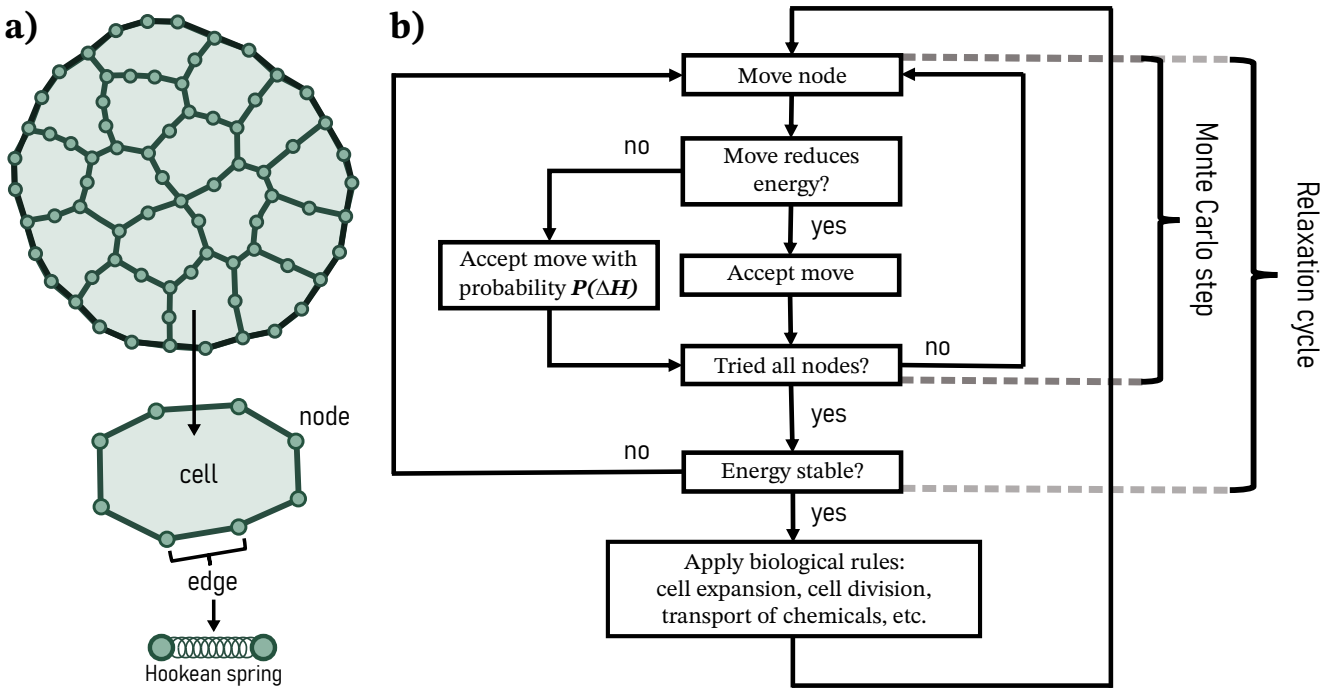

Figure 1: The VirtualLeaf framework. a) The plant tissue is described by a vertex model, where each polygon represents a cell, with edges acting as segments of the cell wall and nodes connecting them. b) Simulation workflow schematic based on Ref. [35]. During the relaxation cycle, the energy of the system is minimized through successive Monte Carlo steps. Each step consists of iteratively applying a trial displacement to every node in the tissue, of which the acceptance depends on the change in energy caused by the move. The relaxation cycle concludes when the energy stabilizes, after which biological rules are applied. These rules can include cell behaviors like expansion and division, as well as the simulation of biochemical networks. After completing these processes, a new relaxation cycle is initiated.

2 VirtualLeaf Framework

2.1 Model

We use VirtualLeaf[35], an open-source framework for modeling plant development, to simulate the growth of a 2D monolayer of plant tissue. The framework employs a vertex model, where a plant tissue is expressed as a set of interconnected polygons. Each polygon represents a cell, of which the edges act as cell wall segments and are connected via the cell's nodes (Fig. 1a). The balance of passive mechanical forces in the tissue is governed by the turgor pressure, resisting compression, and cell wall tension, resisting expansion[36]. This interplay is described by the Hamiltonian through energy constraints:

$$H = \lambda_A \sum_{i \in \text{cells}} (A^{(i)} - A_T^{(i)})^2 + \lambda_M \sum_{j \in \text{edges}} (l^{(j)} - L_T^{(j)})^2 \quad (1)$$

Here, $A^{(i)}$ is the area of cell i, and $A_T^{(i)}$ its resting area, which is the area the cell would take when the turgor pressure and ambient pressure are equal. Similarly, $l^{(j)}$ denotes the length of edge j, and $L_T^{(j)}$ its target length, which is the length the edge would take in the absence of turgor pressure. The Langrange multipliers λ define the area modulus of the cells (λ_A) and the spring constant of the edges (λ_M). Additional energy constraints, like bending or an external field, can be implemented with a similar construct as used for the constraints of the cell area and edge length in Eq. (1).

2.2 Simulation Method

The simulation is divided into two main routines: 1) a relaxation cycle to equilibrate mechanical forces, and 2) a step for applying biological rules (Fig. 1b). Here, it is assumed that the timescale of stress relaxation in the tissue is significantly faster than the rate at which the biological rules take place.

During the relaxation cycle, the potential energy H is minimized by a Metropolis Monte Carlo algorithm. A single Monte Carlo (MC) step consists of iteratively applying a trial displacement to every node k in the tissue: $\boldsymbol{r}_{\text{trial}}^{(k)} = \boldsymbol{r}^{(k)} + \xi \Delta \boldsymbol{r}_{\text{rand}}$, where $\Delta \boldsymbol{r}_{\text{rand}}$ is a random

vector sampled from a uniform distribution $[-\frac{1}{2}, \frac{1}{2}]$ independently in each direction, and ξ defines the step size. If moving the node in the random direction decreases the potential energy ($\Delta H < 0$), the move is accepted; if not, the move is accepted with a Boltzmann probability $P(\Delta H) = \exp(-\Delta H/T)$, where T represents a relative noise factor comparable to thermal fluctuations. When the iteration over all nodes in a random order is complete, two subsequent checks are performed. First, to emulate cell wall relaxation[36,37] and maintain a relatively uniform distribution of nodes, any edges exceeding a threshold length (ν_{yield}) are split in two by inserting a new node into the middle of the edge. Next, the procedure checks whether the sum of all energy differences of the accepted moves (ΔH_{MCS}) is smaller than a predefined energy threshold (θ_H). If $\Delta H_{MCS} < \theta_H$, indicating that the system is in mechanical equilibrium, the relaxation cycle terminates and the simulation proceeds with executing biological rules; else, a new MC step initiates.

Relevant biological rules to implement range from cell growth and division to the transport of chemical signals across cell walls. While VirtualLeaf excels in constructing metabolic networks and hormone patterning using differential equations for transport rules[35,38,39], we focus exclusively on cell expansion and cell division in the absence of biochemical signalling. Cell expansion is initiated by raising the turgor pressure of the cell through an increase of its target area $A_T^{(i)}$. The actual 'expansion' of the cell occurs during the relaxation cycle, as the tissue adapts to accommodate the elevated turgor pressure of the cells. When the area of the cell exceeds a preset threshold, typically twice the cell's area after a previous division, the cell divides. During division, a new cell wall is placed through the centroid of the cell, along a chosen axis. As previously stated, the conventional rule for this axis is the shortest axis; in the extension section we will explain how this axis is calculated, and discuss mechanical-based alternatives. Once all the desired biological rules are applied, a new relaxation cycle starts, allowing the system to adjust accordingly.

3 VirtualLeaf Extension

In this section, we showcase the features added in our extension for VirtualLeaf. The two primary objectives of this extension are: 1) to enable cells to divide according to a mechanical division rule, and 2) to apply an external tension on the system as it grows from a single cell to a tissue. To achieve these goals, we first introduce a method to calculate local and global stresses, and refine the energy minimization. Next, we present options for a mechanical division rule, altering the cell's equilibrium geometry, and for exerting an external tension on the system.

A detailed explanation of the extension features in terms of the code, and how to access these functionalities in VirtualLeaf, is provided in Appendix A.2, organized into corresponding subsections as discussed here.

3.1 Calculating Cell and Tissue Stress

To introduce a mechanical division rule to the system and assess its effect, both in the absence of and under external tension, we need to calculate local and global stresses in the tissue.

In our model, the local stress tensor $\sigma^{(i)}$, characterising the internal stress of cell i, is defined as[40,41]:

$$\sigma_{\alpha\beta}^{(i)} = -\Pi^{(i)}\delta_{\alpha\beta} + \frac{1}{2A^{(i)}} \sum_{j \in i} T_\alpha^{(j)} l_\beta^{(j)} \tag{2}$$

Here, $\alpha, \beta \in \{x, y\}$, and $\delta_{\alpha\beta}$ is the Kronecker delta function. The pressure of cell i is denoted as $\Pi^{(i)}$, while $\boldsymbol{T}^{(j)} = T^{(j)}\hat{\boldsymbol{l}}^{(j)}$ represents the tension of edge j, where $\boldsymbol{l}^{(j)}$ is the vector belonging to edge j, and $\hat{\boldsymbol{l}}^{(j)} = \boldsymbol{l}^{(j)}/|\boldsymbol{l}^{(j)}|$ the normalized vector. The factor $1/2$ for calculating the tension accounts for the edges being shared with neighboring cells. The contribution of boundary edges is weighted twice compared to inner edges as its tension is not shared with a neighbor. Both the scalar values of the pressure and tension are computed from the gradients of their respective energy constraints in Eq. (1):

$$\Pi^{(i)} = -\frac{\partial E}{\partial A^{(i)}} = -2\lambda_A(A^{(i)} - A_T^{(i)}) \tag{3}$$

$$T^{(j)} = \frac{\partial E}{\partial l^{(j)}} = 2\lambda_M(l^{(j)} - L_T^{(j)}) \tag{4}$$

We use the following sign convention for stress: a positive stress value indicates the cell is under tension, causing it to contract when separated from neighboring cells, while a negative stress value indicates compression, causing the cell to expand when isolated from neighboring cells.

Using the local stress tensor, we extract the maximal tensile stress acting on each cell, which serves as a prime indicator of where a new cell wall is placed during division according to the mechanical division rule. The maximal tension and compression are given by the largest and smallest principle eigenvalues λ of the local stress tensor, respectively, and align with the corresponding eigenvectors $\boldsymbol{v}$:

$$\lambda_1, \lambda_2 = \frac{1}{2}\left((\sigma_{xx}+\sigma_{yy}) \pm \sqrt{(\sigma_{xx}-\sigma_{yy})^2 + 4\,\sigma_{xy}\sigma_{yx}}\right)$$
$$\boldsymbol{v}_1 = (\sigma_{xy},\ \lambda_1 - \sigma_{xx}),\ \boldsymbol{v}_2 = (\sigma_{xy}, \lambda_2 - \sigma_{xx}) \qquad (5)$$

In addition to calculating the local stress tensor, we compute the global stress tensor σ by summing all pressure and tension contributions in the tissue:

$$\sigma_{\alpha\beta} = \sum_{i\in\text{cells}} -\Pi^{(i)}\delta_{\alpha\beta} + \frac{1}{2A^{(i)}} \sum_{j\in\text{edges}} T_\alpha^{(j)} l_\beta^{(j)} \qquad (6)$$

Alternatively, the sum over all individual cell contributions from Eq. (2) should yield the same result. In the absence of external tension, the global force density $\nabla \cdot \sigma$ should be zero when the system is in mechanical equilibrium.

3.2 Improved Energy Minimization

Proper energy minimization is essential to prevent the accumulation of a tissue-wide stress in the absence of stress applied extraneously. This prevention is necessary to distinguish between local stresses arising from self-generated tension and those resulting from externally imposed tension.

The original procedure for energy minimization checks whether the difference in energy from before and after the MC step is below a predefined energy threshold, $\Delta H_{MCS} < \theta_H$. This method has two shortcomings, which are both grounded in energy fluctuations. The first issue is that by only evaluating individual MC steps, there is a chance of ΔH_{MCS} falling below the energy threshold through an atypical energy fluctuation while the system has yet not fully equilibrated. We solve this problem by setting a minimum number of consecutive MC steps (MCS_{min}) that must satisfy the energy threshold check. The second complication is the lack of a dynamic energy threshold in response to tissue growth. As the tissue develops, its energy rises due to the increasing number of nodes and edges as well as cell expansion. Consequently, this rise in the average energy leads to an increased amplitude of energy fluctuations. The existing method requires setting a value for θ_H far above the range of energy fluctuations to prevent the simulation from getting stuck in the relaxation cycle as the tissue develops, causing the cycle to preemptively terminate in the early stages of the simulation. To maintain θ_H just slightly above the range of energy fluctuations, it needs scale with the tissue growth. A decent option could be to scale θ_H with the number of cells or nodes, for example $\theta_H \propto \ln(N_{nodes})$. We implemented the option to define a scaling factor α such that $\theta_H = \alpha\,\theta_{init}$.

We evaluate the effectiveness of the original and revised energy minimization methods by comparing simulations of tissue growth with two different MCS_{min}: $MCS_{min} = 1$ for the original approach, and $MCS_{min} = 100$ for our revised procedure. As a proxy for the quality of energy minimization, we follow the trace of the tissue stress tensor $\text{tr}(\sigma)$, which should

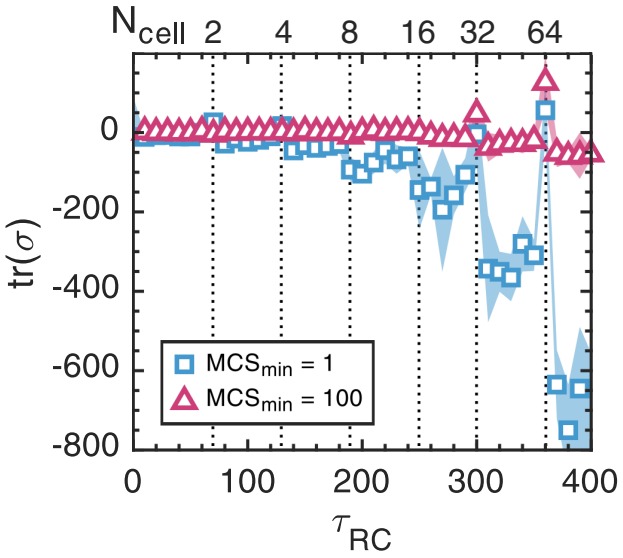

Figure 2: A comparison of the original and revised energy minimization method for simulating tissue growth with a fixed growth rate and shortest path division. Here, the trace of tissue stress tensor $\text{tr}(\sigma)$ is plotted against the number of the relaxation cycles τ_{RC}. The results are averaged over 5 independent simulations, and the colored areas indicate the standard deviation. Vertical striped lines mark significant cell division events, with the numbers above indicating the number of cells in the tissue N_{cell} after each event. The initial energy threshold is set at $\theta_{init} = 10$, and scales as $\theta_H = 0.3\ln(N_{nodes})$. In the original method (blue squares), the relaxation cycle concludes when the energy change from the Monte Carlo step ΔH_{MCS} drops below the energy threshold once, i.e., the minimum number of consecutive steps is 1: $MCS_{min} = 1$. For the revised method (pink triangles) the number of successive steps is increased to $MCS_{min} = 100$.

be zero at mechanical equilibrium, as a function of the number of relaxation cycles τ_{RC} (Fig. 2). We find that the revised method improves on the original method, with $\mathrm{tr}(\sigma)$ staying close to zero during tissue growth, whereas for the original method, $\mathrm{tr}(\sigma)$ becomes increasingly negative. However, the revised approach comes with a trade-off: the simulation time quadrupled compared the original method for these specific simulations, even with the scaling factor for the energy threshold employed, and worsens as the tissue grows. We also strikingly observe that distinctive peaks in $\mathrm{tr}(\sigma)$ emerge following major cell division events, indicating that a large number of cells dividing in close succession drastically alters the mechanical balance of the system (Fig. 2).

When performing large-scale simulations to study mechanical properties during tissue development with VirtualLeaf, we recommend investigating the energy trend and inferring a minimum number of successive iterations (MCS_{min}), initial energy threshold (θ_{init}), and an appropriate scaling factor (α) to optimize both energy minimization and simulation time.

3.3 Division Rules

3.3.1 Shortest Symmetric Path Rule

The classical division rule[26,27] is set as the default division rule in the original VirtualLeaf, and is realized by placing the new cell wall along the shortest axis of the mother cell, through its center of mass. VirtualLeaf determines the shortest axis of cell i from the cell's shape, which, given a uniform distribution of mass over the cell, is characterized by the cell's inertia tensor:

$$I^{(i)} = \iint_D \begin{bmatrix} (x-x_c)^2 & -(x-x_c)(y-y_c)^2 \\ -(x-x_c)(y-y_c)^2 & (y-y_c)^2 \end{bmatrix} \mathrm{d}x\mathrm{d}y \tag{7}$$

Here, $\iint_D$ denotes the area integral over the cell, and (x_c, y_c) the center of mass of the cell. The longest and shortest axes of the cell align with the eigenvectors corresponding to the largest and smallest principal eigenvalues of this tensor, respectively.

Using this division rule for tissue growth yields a circular, symmetric tissue with isotropic cells (Fig. 3a).

3.3.2 Mechanical Division Rule

The primary objective of this expansion to VirtualLeaf is to introduce a mechanical division rule. This rule, as defined by Louveaux et al., states: "the new cell walls orient along the local maximum of tensile stress in cell walls"[24] We interpret this description in two ways.

First, we emulate the rule by placing new cell wall along the maximal tensile stress averaged over the cell. The axis of maximal tensile stress is the eigenvector corresponding first principal eigenvalue of the stress tensor as calculated using Eq. (2) and (5). For the second option, we take the 'local' descriptor a step further by orienting the division plane along the direction of the edge in the cell with the highest tension, also employed in a recent model by Banwarth-Kuhn et al.[42].

We find that tissue growth based on the division rule of average tensile stress yields decidedly more asymmetric tissues than those formed by the shortest path rule (Fig. 3). Although using the rule of the maximal edge tension results in a more symmetric tissue, most cells still tend to adopt a more anisotropic shape. Notably, elongated cells prefer to divide along the long axis of the cell, resulting in equally anisotropic daughter cells. Especially for the rule using the averaged cell stress, this domino effect of elongated cells is prominent and leads to the formation of tissue protrusions (Fig. 3b). Keep in mind that the boundary edges having twice the weight affects the average and local tension of the cell walls. The result of using these mechanical division rules deviates from what was found by Louveaux et al.[24], where, in the absence of a tissue-wide stress, the mechanical division rule replicated the shortest path rule. This discrepancy might in part be rooted by the equilibrium shape of the cell, which we discuss in the next subsection.

3.4 The Equilibrium Geometry of Cells

To understand why our implementation of the mechanical division rule yields different results than those found by Louveaux et al.[24], we evaluate the significance of the equilibrium geometry of cells in stress alignment.

In VirtualLeaf, all edges are assigned an equal rest length L_T and spring constant λ_M. Consequently, when a cell is isolated from the influence of neighboring cells or external forces, it adopts a circular equilib-

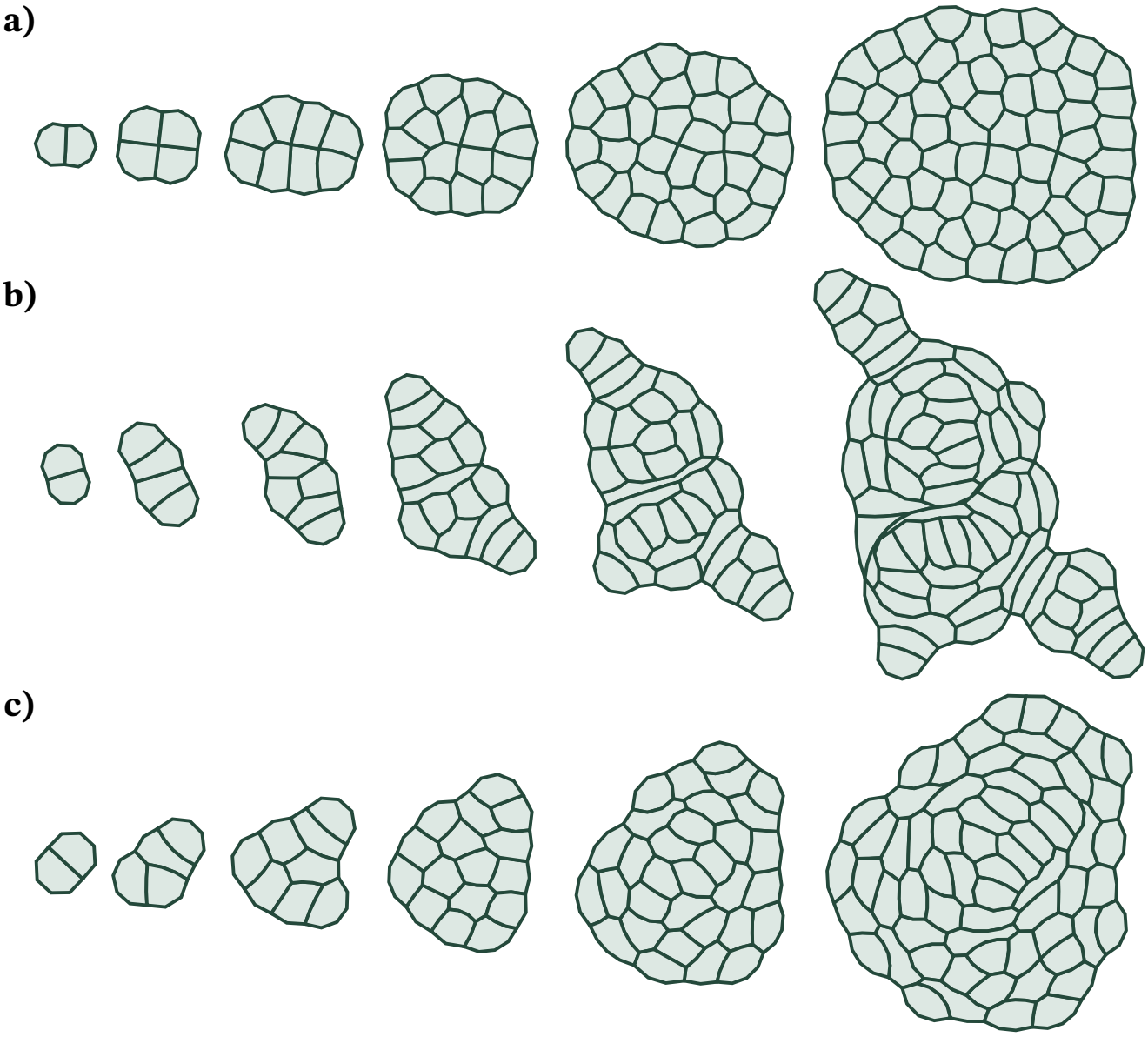

Figure 3: A showcase of tissue growth with a fixed growth rate using different division rules. Simulation snapshots are taken at 120, 180, 240, 280, 340, and 400 relaxation cycles, when the tissue has 2, 4, 8, 16, 32, and 64 cells respectively, just before the next major cell division event. a) The conventional shortest path division rule. b) Mechanical division rule where the new cell wall is placed along the axis of maximal tensile stress of the cell. c) Mechanical division rule where the new cell wall is placed parallel to the axis of the edge with the highest tension of the cell.

rium shape. The same result can be obtained by substituting the edge constraint in the Hamiltonian (Eq. 1) with a perimeter constraint. Perimeter constraints are widely applied in vertex models for modeling epithelial animal tissues[41,43,44]. Moreover, minimizing the perimeter, or analogously the surface area for a 3D cell, is also in line with the comparison of plant cells to soap bubbles[29,45], which formed the basis of the original shortest path division rule[26]. However, the equilibrium shape of cells are much more intricate due to the ongoing remodeling and anisotropic expansion of the cell wall during cell growth[46–48]. A cell's equilibrium shape dictates how wall tensions are distributed, and subsequently the alignment of the cell's principal stresses, under elastic deformation. This tension distribution in turn drives directional growth of the cell and feeds back into the wall remodeling. For a circular rest shape, the maximal

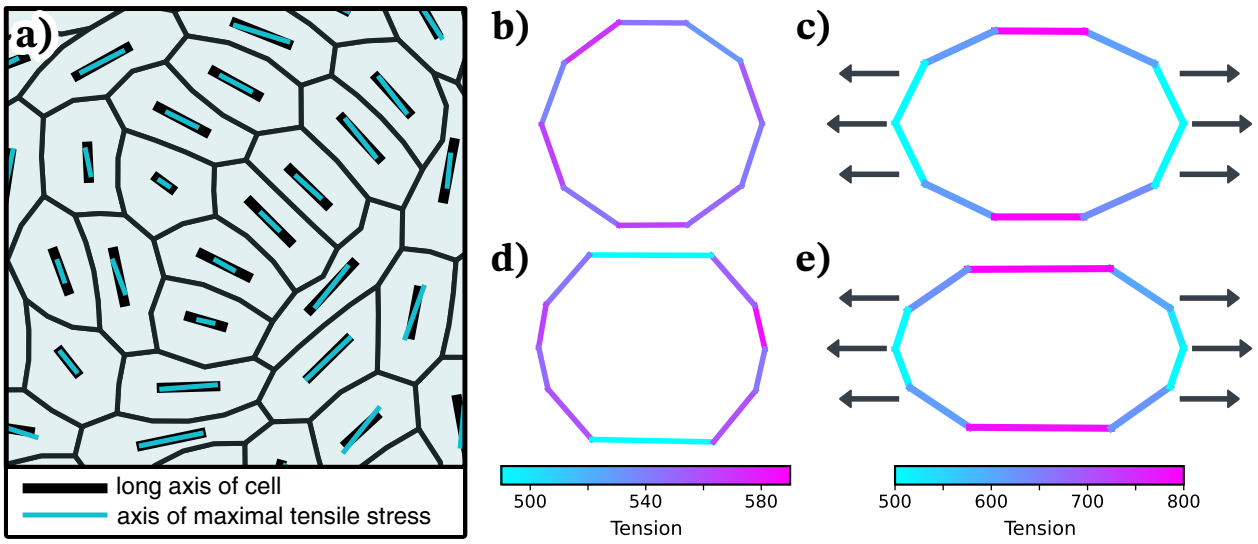

Figure 4: The equilibrium geometry of the cell. a) Comparison of cell geometry and stress in a tissue snapshot with all edges having a uniform target length. The thick black bars are oriented along the long axis of the cell, calculated from the cell's inertia tensor (Eq. 7). The thinner blue bars are directed in the orientation of maximal tensile stress, calculated from the cell's stress tensor (Eq. 2). The lengths of both bars are proportional to their respective tensor anisotropy: $(\lambda_1 - \lambda_2)/\sqrt{\lambda_1^2 + \lambda_2^2}$. The axis of maximal stress and the long axis of the cell strongly overlap in most cases. (b-e) Tension distributions in pressurized cells with two different equilibrium shapes: (b) a cell with uniform resting lengths of the edges under pressure, and (d) a cell with variable target lengths such that the cell has an anisotropic rest shape. In (c,e) these cells are put under external tension, by setting an energy constraint on the boundary nodes with $\lambda_s = 100$ and $\Delta x = 0.5$ (refer to subsection 3.5 for the method).

tensile stress predominantly aligns with the long axis of the cell when its anisotropy is caused by local, neighboring forces[49], which we also observe in our simulations (Fig. 4a).

Wall remodeling in VirtualLeaf is achieved through the yielding of edges exceeding a threshold length (ν_{yield}) into two daughter edges, which inherit the same global rest length L_T as the original edge. Consequently, the equilibrium geometry after yielding is only altered in its resting perimeter, essentially the cell's rest size, but the equilibrium shape remains unaltered. Moreover, the default threshold length is set to $\nu_{yield} = 4$, meaning edges need to extend by 400% strain before yielding occurs. This threshold seems beyond a stage where a linear elastic regime could properly describe the wall mechanics and should be reconsidered in our model. In contrast, Louveaux et al.'s model conceptualizes the cell wall as having with an evolving equilibrium shape; the wall stretches for only a few percent of strain before the cell wall is remodeled[24]. As a consequence, the maximal tensile stress aligns along the shorter axis of an anisotropic cell in their model. This orientation of maximal tensile stress along the short axis corresponds to the alignment of microtubules[17,22] and forms the foundation for understanding the shortest path rule in the absence of tissue-wide stress[27].

To better understand how changing the equilibrium geometry affects the tension distribution of the cell, we introduce an option for adjusting the individual target lengths of edges in VirtualLeaf. As a starting point, we compare a cell with uniform rest lengths (Fig. 4b) to a cell with target lengths assigned such that the rest shape is a bit elongated (Fig. 4d). Both cells are pressurized. Naturally, the cell with equal rest lengths has a completely isotropic distribution of tension in its cell wall. In case of the cell with the with anisotropic rest shape, the highest tension in the cell wall is experienced along the shortest axis of the cell. When we put the two cells under external tension (refer to the next subsection for the method), the highest cell wall tension as well as the general maximal tensile stress axis align with the direction of applied external tension (Fig. 4c,e). We will address further ideas about the equilibrium geometry in the discussion.

3.5 Applying External Tissue Tension

Here, we introduce two methods for applying an external tension to tissue, with the aim of replicating

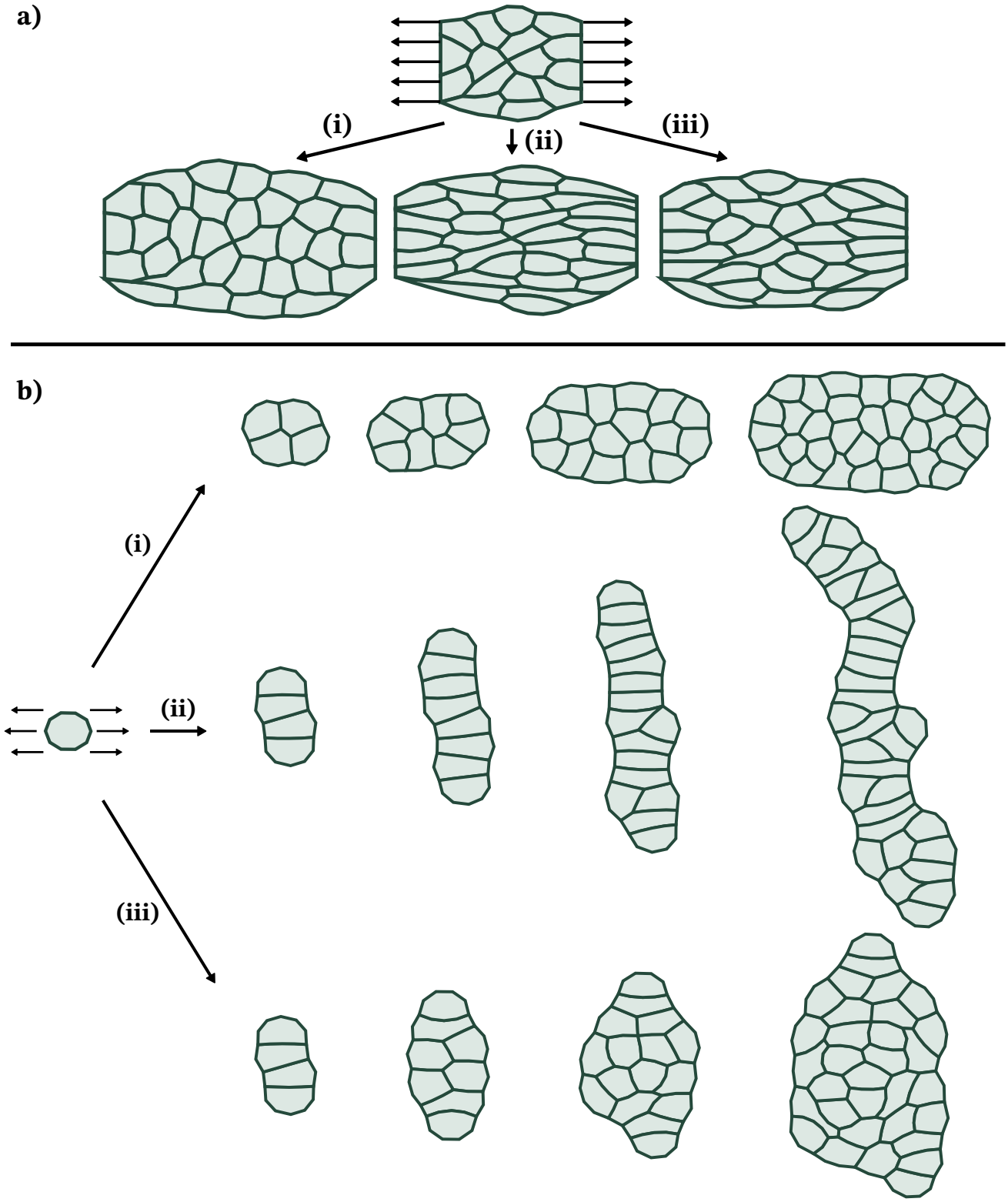

Figure 5: A showcase of two approaches for applying an external tension during tissue growth with a fixed growth rate using different division rules: (i) shortest path division rule, (ii) division along the maximal tension averaged over cell, (iii) division along the edge with maximal tension. a) A tensile test: a pre-grown tissue of 16 cells has fixed nodes on opposing sides which are displaced along the x-axis by $\Delta x = \pm 0.5$ before each relaxation cycle. Snapshots corresponding to division rules i-iii are captured after all cells in the initial tissue have divided at least once. b) Tension is introduced through an energy constraint applied to boundary nodes using Eq. 8. Here, $\lambda_S = 100$, and $\boldsymbol{r}^{*(k)}$ was set as $\boldsymbol{r}^{*(k)} = \boldsymbol{r}^{(k)} + (\pm 1, 0, 0)$. Snapshots for division rules i-iii are taken at 180, 240, 280, and 340 relaxation cycles, when the tissue has 4, 8, 16, and 32 cells, respectively.

the experiment of subjecting a hydrogel matrix in which proliferating cells are embedded to an externally applied tension.

First, we use an approach resembling a tensile test to introduce external tension. This test is accomplished by displacing a predefined group of fixed nodes, whose positions are unaffected by energy minimization, on two sides of the tissue by a set amount $\Delta \boldsymbol{r}$ in opposing directions before each relaxation cycle (Fig. 5a). To demonstrate the method, we perform it on a pre-grown tissue featuring two straight sides of which the nodes have been fixed. These sides are displaced in opposing directs by a fixed displacement $\Delta x = \pm 0.5$ per relaxation cycle. We examine the resulting tissue after all cells have divided for different division rules (Fig. 5a). For division over the shortest axis, most cells remain isotropic (Fig. 5a-i). When using the mechanical division rule based on the cell's average tensile stress, all cells become elongated, having divided along their long axes, and approximately align with the direction of applied strain (Fig. 5a-ii). Finally, for the mechanical division rule using the maximal wall tension we observe a mix of isotropic and anisotropic, with the most anisotropic cells generally being part of the fixed sides (Fig. 5a-iii). Keep in mind that since the edges between fixed nodes are relaxed, the maximal (average) tension for these cells is biased.

Rather than directly pulling on the tissue, we ideally want to embed the tissue in a matrix on which tension is applied instead. While this should be possible in VirtualLeaf, there was no time to implement this. Instead, for the second method, we tried to implicitly emulate the effect of a surrounding hydrogel matrix by applying tension to all boundary nodes, as they are the points where the forces propagated through the hydrogel matrix first arrive. Rather than fixing the boundary nodes, we assign them an energy constraint, resulting in a modified Hamiltonian:

$$H' = H + \lambda_S \sum_{k \in k_B} (\mathbf{r}^{(k)} - \mathbf{r}^{*(k)})^2 \tag{8}$$

Here, λ_S is the Lagrange multiplier for this constraint, and k_B refers to boundary nodes. $\mathbf{r}^{(k)}$ is the position of the node k before a new relaxation cycle, and $\mathbf{r}^{*(k)}$ an affine reference position of the node. The sign of $\mathbf{r}^{*(k)}$ depends on which side of the tissue a boundary node is relative to the center position of the tissue. Before each relaxation cycle, $\mathbf{r}^{*(k)}$ is updated by a set amount $\Delta \boldsymbol{r}$. We illustrate this second technique by applying it as a tissue proliferates starting from a single cell; again, using the three discussed division rules. We set $\lambda_S = 100$ and $\Delta x = \pm 0.5$. Similar to the tensile test, this second method combined with the shortest path division rule results in a stretched tissue with isotropic cells (Fig. 5b-i). In the case of the division rule of average maximal tension we observe worm-like tissue geometry (Fig. 5b-ii). This outcome is likely due to the tension along the x-axis overriding the extra boundary weight compared to the growing tissue without the applied tension (Fig. 3b). For the rule of division along the edge with maximal tension, there is an initial development of a worm-like structure, but the tissue ultimately widens. This change is probably caused by the boundary edges perpendicular to the applied tension having a higher stress, because of the extra weight, even though the average cell stress remains along the axis of applied tension.

4 Discussion

The precise regulation of cell division orientation is crucial for the morphogenesis of plants, where cell rearrangement is restricted. There is increasing evidence that mechanical stress plays a pivotal role in this regulation, as the alignment of maximal tensile stress in cell walls seems to prescribe the axis of the division plane. In this report, we presented an extension for VirtualLeaf that allows us to simulate the growth of a plant tissue placed under mechanical tension, with cell division guided by tension in the cell wall. This extension will complement upcoming experiments in the Biochemistry group at Wageningen University, with which we aim to gain a controlled understanding of the impact of mechanical stress on tissue morphogenesis. In this discussion section, we highlight results, address remaining challenges for improving the model and outline future prospects.

First, we want to address the tissue morphologies resulting from applying a mechanical division rule. Although we established that our underlying model might be lacking in a proper description for the equilibrium geometry of cells, resulting in a maximal tension along the long axis of the cell rather than the expected short axis in the absence of applied external tension, we can still infer some insights from our tissue growth results. For instance, continuous divisions over the long axis could play a notable

role in developing tissue protrusions (Fig. 3b, 5b-ii). Moreover, as we apply external tension during tissue growth, this imposed tension seems to gradually become less influential to the stress alignment in the cell wall than the stress induced by cell growth (Fig. 5b-iii). It will be interesting to see whether experiments yield similar results.

We introduced a method to modify the equilibrium shape of cells by allowing individual edges to have a specific target length rather than a shared global value. To obtain a true equilibrium geometry, we would also need to incorporate a bending in the model. Furthermore, aside from resetting target lengths and bending angles, we might also need to adjust spring and bending constants to define the equilibrium shape. Resetting only the target lengths, for example, results in elastic deformation being biased along the direction with the most individual edges, due to the squared energy constraint. As an alternative approach we could replace our edge and bending constraints in the Hamiltonian by a shape constraint, based on present cell shape and the reference equilibrium shape as defined by the inertia tensor (Eq. 7). While this approach decreases the number of parameters in the model, it does lead to the loss of information on the local edge tensions. Keeping this information provides more versatility for approaching mechanical division, including our interpretation of a division rule based on the edge with the highest tension. Moreover, if wanted, it would allow for incorporating additional wall anisotropy based on local reinforcement of the cell wall as a response to mechanical stress[48,50–52]. We need to carefully investigate how to implement the cell shape remodeling during the simulation and properly match it with the rate of other processes like cell expansion.

In terms of energy minimization, our updated approach has improved upon the old method by increasing the number of subsequent Monte Carlo steps that need to pass the energy threshold check before concluding the relaxation cycle. However, there are still two notable challenges for the minimization. Firstly, multiple cell divisions occurring in quick succession appears to significantly disrupt the system's energy balance, leading to distinct peaks in tissue stress after a relaxation cycle (Fig. 4). We assume this behavior might in part due to how cell division is implemented. In the original VirtualLeaf, the pressure of daughter cells is halved compared to the mother cell after division. For the results in this report we changed this so daughter cells have equal pressure post division (see Appendix A.2.2), yet, this did not fix the problem. Additionally, the placement of the new cell wall reduces tension in the walls perpendicular to the new wall, if new nodes are inserted. The new cell wall itself has a variable tension after division, depending on the wall's length, due to not being able to modify edge parameters. With our implementation of variable edges, we should reevaluate our approach and assess whether this aspect indeed contributes to the energy disruption. We also added an option for limiting cell division to one cell per relaxation cycle, which needs to be tested (see Appendix A.2.2). The second challenge of the new method is related to the simulation time. Despite the implementation of a scaling factor for the energy threshold, the number of Monte Carlo steps often reaches the assigned maximum of 10^5 steps after the tissue has grown to 64 cells. The scaling factor itself does not necessarily seem to be the problem, as by making it more lenient, there is an undesired built-up of compressive tissue stress. If an exact cause for this problem cannot be found, we might need to consider alternative schemes for energy minimization to improve simulation speed, such as the FIRE algorithm[53] or Position Based Dynamics scheme[54,55].

We presented two approaches for applying external tension to the tissue, each with its own challenges. The first method involves the deformation of the tissue through displacing a set of fixed nodes, which unfortunately introduces tension artifacts at the fixed boundary due to the static tension between these nodes. When focusing on smaller tissues, these boundary effects are especially unwanted. Furthermore, fixing nodes might hinder the formation of tissue shapes that would otherwise emerge. In the second method, we implicitly introduced a surrounding matrix by adding a tensional energy constraint on the boundary nodes (Eq. 8). While this approach is suitable for starting from a single cell, how the stress of the matrix translates to tension in the outer cell walls remains uncertain. Moreover, we should consider that with tissue growth, the number of boundary nodes increases, resulting in more points of connection with the implicit matrix. Yet, the tension put on the tissue by setting a constraint on the boundary

nodes, or moving fixed nodes, becomes less impactful as the cumulative cell growth inherently stretches the tissue through self-generated tensions. The interplay between the growth rate and the rate or magnitude of imposed tension or deformation will be intriguing to examine. An alternative for applying tension could be imposing a constraint on tissue stress, using the current and a reference tissue stress (see Appendix A.2.5). This approach has its own set of challenges: firstly, the accuracy of the reference stress decreases with growing deviations in tissue stress as the tissue size increases, due to larger energy fluctuations. Secondly, recalculating the full stress tensor of the tissue for each trial move would be very expensive computationally, although a clever approach should solve this. Lastly, a method closest to the actual experiments that will be performed would be to explicitly incorporate a hydrogel matrix surrounding the cell and subjecting that matrix to tension.

Finally, we want to briefly outline some future ideas as (plant-based) food for thought. Our model assumes a new cell wall is always positioned through the cell's geometric center as a straight line. This approach works well for sheets of cells at the tissue boundary where division is typically anticlinal, i.e., perpendicular to the boundary[56]. However, considering our model starts with a single cell, the 3D shape might be more relevant than we are currently assuming. We will need to evaluate whether a 2D model can adequately describe the division processes in the experiment data. If 3D analysis and modeling are necessary, MorphoGraphX could be a suitable tool[57]. Also regarding division, while we repeatedly state the significance of mechanical stress in tissue morphogenesis, it is important to recognize the intertwined nature of mechanical cues and the establishment of cell polarity[8,11,58]. Cell polarity is a primary driver of asymmetric cell division, where the new cell wall is not placed through the centroid of the cell. Studying the impact of increasing mechanical tension on cell polarity establishment and the resulting asymmetric divisions could be intriguing, as asymmetric division are often at the root of initiating developmental patterns and organoid formation.

References

[1] B. Zhang, Y. Gao, L. Zhang, and Y. Zhou, "The plant cell wall: Biosynthesis, construction, and functions", Journal of Integrative Plant Biology **63**, 251–272 (2021).

[2] C. Bertet, L. Sulak, and T. Lecuit, "Myosin-dependent junction remodelling controls planar cell intercalation and axis elongation", Nature **429**, 667–671 (2004).

[3] E. Walck-Shannon and J. Hardin, "Cell intercalation from top to bottom", Nature Reviews Molecular Cell Biology **15**, 34–48 (2014).

[4] E. Trubuil, A. D'Angelo, and J. Solon, "Tissue mechanics in morphogenesis: Active control of tissue material properties to shape living organisms", Cells and Development **168**, 203777 (2021).

[5] E. Coen, A. G. Rolland-Lagan, M. Matthews, J. A. Bangham, and P. Prusinkiewicz, "The genetics of geometry", Proceedings of the National Academy of Sciences of the United States of America **101**, 4728–4735 (2004).

[6] L. Dupuy, J. Mackenzie, and J. Haseloff, "Coordination of plant cell division and expansion in a simple morphogenetic system", Proceedings of the National Academy of Sciences of the United States of America **107**, 2711–2716 (2010).

[7] S. Yoshida, P. BarbierdeReuille, B. Lane, G. W. Bassel, P. Prusinkiewicz, R. S. Smith, and D. Weijers, "Genetic control of plant development by overriding a geometric division rule", Developmental Cell **29**, 75–87 (2014).

[8] V. Gorelova, J. Sprakel, and D. Weijers, "Plant cell polarity as the nexus of tissue mechanics and morphogenesis", Nature Plants **7**, 1548–1559 (2021).

[9] K. S. Hartman and A. Muroyama, "Polarizing to the challenge: New insights into polarity-mediated division orientation in plant development", Current Opinion in Plant Biology **74**, 102383 (2023).

[10] M. Sassi and J. Traas, "When biochemistry meets mechanics: A systems view of growth control in plants", Current Opinion in Plant Biology **28**, 137–143 (2015).

[11] M. Bringmann and D. C. Bergmann, "Tissue-wide Mechanical Forces Influence the Polarity of Stomatal Stem Cells in Arabidopsis", Current Biology **27**, 877–883 (2017).

[12] S. Robinson, "Mechanobiology of cell division in plant growth", New Phytologist **231**, 559–564 (2021).

[13] D. C. Trinh, J. Alonso-Serra, M. Asaoka, L. Colin, M. Cortes, A. Malivert, S. Takatani, F. Zhao, J. Traas, C. Trehin, and O. Hamant, "How Mechanical Forces Shape Plant Organs", Current Biology **31**, R143–R159 (2021).

[14] E. Coen and D. J. Cosgrove, "The mechanics of plant morphogenesis", Science **379**, `10.1126/science.ade8055` (2023).

[15] A. Smertenko, F. Assaad, F. Baluška, M. Bezanilla, H. Buschmann, G. Drakakaki, M. T. Hauser, M. Janson, Y. Mineyuki, I. Moore, S. Müller, T. Murata, M. S. Otegui, E. Panteris, C. Rasmussen, A. C. Schmit, J. Šamaj, L. Samuels, L. A. Staehelin, D. Van Damme, G. Wasteneys, and V. Žárský, "Plant Cytokinesis: Terminology for Structures and Processes", Trends in Cell Biology **27**, 885–894 (2017).

[16]P. Livanos and S. Müller, "Division Plane Establishment and Cytokinesis", Annual Review of Plant Biology **70**, 239–267 (2019).

[17]M. Louveaux, S. Rochette, L. Beauzamy, A. Boudaoud, and O. Hamant, "The impact of mechanical compression on cortical microtubules in Arabidopsis: a quantitative pipeline", Plant Journal **88**, 328–342 (2016).

[18]V. Mirabet, P. Krupinski, O. Hamant, E. M. Meyerowitz, H. Jönsson, and A. Boudaoud, "The self-organization of plant microtubules inside the cell volume yields their cortical localization, stable alignment, and sensitivity to external cues", PLoS Computational Biology **14**, e1006011 (2018).

[19]S. Robinson and C. Kuhlemeier, "Global Compression Reorients Cortical Microtubules in Arabidopsis Hypocotyl Epidermis and Promotes Growth", Current Biology **28**, 1794–1802.e2 (2018).

[20]S. Verger, Y. Long, A. Boudaoud, and O. Hamant, "A tension-adhesion feedback loop in plant epidermis", eLife **7**, 10.7554/eLife.34460 (2018).

[21]O. Hamant, D. Inoue, D. Bouchez, J. Dumais, and E. Mjolsness, "Are microtubules tension sensors?", Nature Communications **10**, 1–12 (2019).

[22]L. Colin, A. Chevallier, S. Tsugawa, F. Gacon, C. Godin, V. Viasnoff, T. E. Saunders, and O. Hamant, "Cortical tension overrides geometrical cues to orient microtubules in confined protoplasts", Proceedings of the National Academy of Sciences of the United States of America **117**, 32731–32738 (2020).

[23]Y. Yan, Z. Sun, P. Yan, T. Wang, and Y. Zhang, "Mechanical regulation of cortical microtubules in plant cells", New Phytologist **239**, 1609–1621 (2023).

[24]M. Louveaux, J. D. Julien, V. Mirabet, A. Boudaoud, and O. Hamant, "Cell division plane orientation based on tensile stress in Arabidopsis thaliana", Proceedings of the National Academy of Sciences of the United States of America **113**, E4294–E4303 (2016).

[25]P. M. Lintilhac and T. B. Vesecky, "Stress-induced alignment of division plane in plant tissues grown in vitro", Nature **307**, 363–364 (1984).

[26]L. Errera, "On a fundamental condition of equilibrium for living cells", CR Hebd Seances Acad Sci **103**, 822–824 (1886).

[27]S. Besson and J. Dumais, "Universal rule for the symmetric division of plant cells", Proceedings of the National Academy of Sciences of the United States of America **108**, 6294–6299 (2011).

[28]B. E. Shapiro, C. Tobin, E. Mjolsness, and E. M. Meyerowitz, "Analysis of cell division patterns in the Arabidopsis shoot apical meristem", Proceedings of the National Academy of Sciences of the United States of America **112**, 4815–4820 (2015).

[29]P. Martinez, L. A. Allsman, K. A. Brakke, C. Hoyt, J. Hayes, H. Liang, W. Neher, Y. Rui, A. M. Roberts, A. Moradifam, B. Goldstein, C. T. Anderson, and C. G. Rasmussen, "Predicting division planes of three-dimensional cells by soap-film minimization", Plant Cell **30**, 2255–2266 (2018).

[30]S. A. Belteton, W. Li, M. Yanagisawa, F. A. Hatam, M. I. Quinn, M. K. Szymanski, M. W. Marley, J. A. Turner, and D. B. Szymanski, "Real-time conversion of tissue-scale mechanical forces into an interdigitated growth pattern", Nature Plants **7**, 826–841 (2021).

[31]S. Jungbauer, H. Gao, J. P. Spatz, and R. Kemkemer, "Two characteristic regimes in frequency-dependent dynamic reorientation of fibroblasts on cyclically stretched substrates", Biophysical Journal **95**, 3470–3478 (2008).

[32]Y. Morita, S. Watanabe, Y. Ju, and S. Yamamoto, "In Vitro Experimental Study for the Determination of Cellular Axial Strain Threshold and Preferential Axial Strain from Cell Orientation Behavior in a Non-uniform Deformation Field", Cell Biochemistry and Biophysics **67**, 1249–1259 (2013).

[33]C. Giverso, N. Loy, G. Lucci, and L. Preziosi, "Cell orientation under stretch: A review of experimental findings and mathematical modelling", Journal of Theoretical Biology **572**, 111564 (2023).

[34]Y. Li, O. Kučera, D. Cuvelier, D. M. Rutkowski, M. Deygas, D. Rai, T. Pavlovič, F. N. Vicente, M. Piel, G. Giannone, D. Vavylonis, A. Akhmanova, L. Blanchoin, and M. Théry, "Compressive forces stabilize microtubules in living cells", Nature Materials **22**, 913–924 (2023).

[35]R. M. Merks, M. Guravage, D. Inzé, and G. T. Beemster, "Virtualleaf: An open-source framework for cell-based modeling of plant tissue growth and development", Plant Physiology **155**, 656–666 (2011).

[36]J. A. Lockhart, "An analysis of irreversible plant cell elongation", Journal of Theoretical Biology **8**, 264–275 (1965).

[37]D. J. Cosgrove, "Cell Wall Yield Properties of Growing Tissue", Plant Physiology **78**, 347–356 (1985).

[38]D. De Vos, E. D. Borger, J. Broeckhove, and G. T. Beemster, "Simulating leaf growth dynamics through Metropolis-Monte Carlo based energy minimization", Journal of Computational Science **9**, 107–111 (2015).

[39]I. Lebovka, B. Hay Mele, X. Liu, A. Zakieva, T. Schlamp, N. R. Gursanscky, R. M. Merks, R. Großeholz, and T. Greb, "Computational modelling of cambium activity provides a regulatory framework for simulating radial plant growth", eLife **12**, 10.7554/elife.66627 (2023).

[40]K. K. Chiou, L. Hufnagel, and B. I. Shraiman, "Mechanical stress inference for two dimensional cell arrays", PLoS Computational Biology **8**, e1002512 (2012).

[41]X. Yang, D. Bi, M. Czajkowski, M. Merkel, M. L. Manning, and M. C. Marchetti, "Correlating cell shape and cellular stress in motile confluent tissues", Proceedings of the National Academy of Sciences of the United States of America **114**, 12663–12668 (2017).

[42]M. Banwarth-Kuhn, K. Rodriguez, C. Michael, C. K. Ta, A. Plong, E. Bourgain-Chang, A. Nematbakhsh, W. Chen, A. Roy-Chowdhury, G. V. Reddy, and M. Alber, "Combined computational modeling and experimental analysis integrating chemical and mechanical signals suggests possible mechanism of shoot meristem maintenance", PLoS Computational Biology **18**, e1010199 (2022).

[43]D. L. Barton, S. Henkes, C. J. Weijer, and R. Sknepnek, "Active Vertex Model for cell-resolution description of epithelial tissue mechanics", PLoS Computational Biology **13**, e1005569 (2017).

[44]A. Nestor-Bergmann, E. Johns, S. Woolner, and O. E. Jensen, "Mechanical characterization of disordered and anisotropic cellular monolayers", Physical Review E **97**, 10 . 1103 / PhysRevE.97.052409 (2018).

[45]O. Ali, I. Cheddadi, B. Landrein, and Y. Long, "Revisiting the relationship between turgor pressure and plant cell growth", New Phytologist **238**, 62–69 (2023).

[46]D. Qiu, S. Xu, Y. Wang, M. Zhou, and L. Hong, *Primary Cell Wall Modifying Proteins Regulate Wall Mechanics to Steer Plant Morphogenesis*, 2021.

[47]D. J. Cosgrove, "Plant cell wall extensibility: Connecting plant cell growth with cell wall structure, mechanics, and the action of wall-modifying enzymes", Journal of Experimental Botany **67**, 463–476 (2016).

[48]T. Zhang, D. Vavylonis, D. M. Durachko, and D. J. Cosgrove, "Nanoscale movements of cellulose microfibrils in primary cell walls", Nature Plants **3**, 1–6 (2017).

[49]A. Nestor-Bergmann, G. Goddard, S. Woolner, and O. E. Jensen, "Relating cell shape and mechanical stress in a spatially disordered epithelium using a vertex-based model", Mathematical Medicine and Biology **35**, 1–27 (2018).

[50]A. Sampathkumar, P. Krupinski, R. Wightman, P. Milani, A. Berquand, A. Boudaoud, O. Hamant, H. Jönsson, and E. M. Meyerowitz, "Subcellular and supracellular mechanical stress prescribes cytoskeleton behavior in Arabidopsis cotyledon pavement cells", eLife **2014**, 10.7554/eLife.01967.001 (2014).

[51]T. I. Baskin, "Anisotropic expansion of the plant cell wall", Annual Review of Cell and Developmental Biology **21**, 203–222 (2005).

[52]H. Oliveri, J. Traas, C. Godin, and O. Ali, "Regulation of plant cell wall stiffness by mechanical stress: a mesoscale physical model", Journal of Mathematical Biology **78**, 625–653 (2019).

[53]E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch, "Structural relaxation made simple", Physical Review Letters **97**, 170201 (2006).

[54]M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, "Position based dynamics", Journal of Visual Communication and Image Representation **18**, 109–118 (2007).

[55]M. Marconi and K. Wabnik, "Shaping the Organ: A Biologist Guide to Quantitative Models of Plant Morphogenesis", Frontiers in Plant Science **12**, 2171 (2021).

[56]O. Hamant and J. Traas, "The mechanics behind plant development", New Phytologist **185**, 369–385 (2010).

[57]P. B. de Reuille, A. L. Routier-Kierzkowska, D. Kierzkowski, G. W. Bassel, T. Schüpbach, G. Tauriello, N. Bajpai, S. Strauss, A. Weber, A. Kiss, A. Burian, H. Hofhuis, A. Sapala, M. Lipowczan, M. B. Heimlicher, S. Robinson, E. M. Bayer, K. Basler, P. Koumoutsakos, A. H. Roeder, T. Aegerter-Wilmsen, N. Nakayama, M. Tsiantis, A. Hay, D. Kwiatkowska, I. Xenarios, C. Kuhlemeier, and R. S. Smith, "MorphoGraphX: A platform for quantifying morphogenesis in 4D", eLife **4**, 1–20 (2015).

[58]Y. Zhang and J. Dong, "Cell polarity: compassing cell division and differentiation in plants", Current Opinion in Plant Biology **45**, 127–135 (2018).

[59]C. C. Antonovici, G. Y. Peerdeman, H. B. Wolff, and R. M. Merks, "Modeling Plant Tissue Development Using VirtualLeaf", in *Methods in molecular biology*, Vol. 2395 (Humana Press Inc., 2022), pp. 165–198.

Acknowledgements

A big thank you to my supervisors for letting me work on this project! Jasper, thanks for being my supervisor (again)! It is always a pleasure to have meetings with you. Roeland, thanks for answering my many questions about VirtualLeaf at the start of the project, it's a really great program! :) Joris, thanks for your input from the plant mechanobiology side, I hope the experimental set-up will work out soon!

Code Availability

The extension for VirtualLeaf discussed in this report is available at: github.com/AimeeKok/VirtualLeaf_Extension.
The original version of VirtualLeaf is available at: github.com/rmerks/VirtualLeaf2021

Appendix

A.1 VirtualLeaf Coding Framework

The VirtualLeaf framework is written in C++ using the cross-platform development library Qt. It consists of three main components: the source code, model plugins, and a graphical user interface (GUI). The source code is where all the fundamental object classes and routines are defined, such as the relaxation cycle routine (Figure 1b), input and output procedures, and the GUI's underlying code. Typically, users are not intended to make direct changes to the source code. In contrast, the model plugin is a software module which users can develop independently from the source code. Within this module, users define their own model, where they set which biological rules are called upon and under what conditions, working within the constraints set by the source code. For example, the user can specify how morphogen concentrations relate to the cell growth. Lastly, the GUI serves as an interactive platform where users can visualize and directly interact with their defined model and run simulations.

A.1.1 VirtualLeaf Tutorial

For an introductory tutorial on installing and using VirtualLeaf we refer to the book section *"Modeling Plant Tissue Development Using VirtualLeaf"*[59].

A.1.2 Reference List of Classes

Here, we briefly explain the most relevant classes for understanding the VirtualLeaf code.

- `Mesh` class: Only a single `Mesh` object is constructed for the simulation, which represents the tissue as a whole. This object maintains references to all the cells, walls, and nodes in the system, and houses many essential functions, including the Monte Carlo step routine.

- `Cell` class (subclass of `CellBase`): A `Cell` object represents a single cell within the simulation. It not only tracks essential properties like the cell's area and target area but also maintains references to the nodes that define the cell's structure, as well as its neighboring cells and walls for managing biochemical transport. The `Cell` class also hosts the core function for cell division.

- `Node` class (subclass of `Vector`): The `Node` class is responsible for defining the nodes in the tissue. A `Node` object's main function is to keep track of a node's coordinates.

- `Neighbor` class: The `Neighbor` class is closely associated with the `Node` class. It allows nodes to maintain references to their two neighboring nodes for each cell they are part of. The `Neighbor` class provides a convenient means to keep track of all cells and nodes affected by performing a trial move in the energy minimization routine.

- `Wall` class (subclass of `WallBase`): Although not directly used in our extension, the `Wall` class serves as a blueprint for cell walls in the simulation. This class is unrelated to the edges referred to in the energy minimization. `Wall` objects are exclusively used for setting up biochemical transport between neighboring cells.

- `Parameter` class: Similar to the `Mesh` class, only a single `Parameter` object is initialized per simulation. The `Parameter` object stores all of the input parameters defined by the user beforehand, either within the input file or in the GUI. The values of its data members are unaltered during the simulation's runtime.

A.1.3 Reference List of Core Functions

Provided below is a short list of the most relevant core functions in the source code of VirtualLeaf for easy reference:

- `TIMESTEP` block: The core simulation procedure (Figure 1b). (/GUI/VirtualLeaf.cpp)

- `Mesh::DisplaceNodes`: The Monte Carlo step routine. (/GUI/mesh.cpp)

- `Mesh::InsertNode`: The cell wall yielding procedure. (/GUI/mesh.cpp)

- `Mesh::DoCellHouseKeeping`: Calls upon the model plugin function where cell behavioral rules are set by the user. (/GUI/mesh.h)

- `Cell::DivideWalls`: The cell division procedure. (/GUI/cell.cpp)

- The /GUI/qtxmlwrite.cpp file contains the body of most data input and output functions.

A.2 A Coder's Guide to the VirtualLeaf Extension

In this appendix section, we provide a code-oriented summary of the VirtualLeaf extension. For each part of the extension highlighted in the main text, we outline the most important modifications made to core functions, newly added functions, their locations in the code, and how they can be used. It is important to note that some parts of the extension are still a work in progress. As such, another aim of this section is to outline tasks still to be performed.

Throughout the code, all changes to the original VirtualLeaf program are marked with the tag: '`AIMEE`'. Modifications related to a specific subsection can be found by adding the section number to tag, e.g.: '`AIMEE A.2.1`'.

A.2.1 Calculating Cell and Tissue Stress

We included a two-part function to compute the stress experienced by a single cell. In the first function, `CellBase::SetStressTensor(void)`, the three stress tensor components are calculated and stored in the `CellBase` variables `sigma_xx`, `sigma_yy`, and `sigma_xy`. The second function, `CellBase::Stress(double *stress, Vector *stress_axis)`, calls on the first function and calculates the eigenvalues and eigenvectors of the stress tensor. This function takes references to two variables: `stress`, which stores the magnitude of the maximum tensile stress, and `stress_axis`, which stores the corresponding axis. `CellBase::Stress` can be called from the model plugin to set the axis for placing the new cell wall during division. Note: the `Parameter` object cannot be accessed from within the cellbase.cpp file. Therefore, the spring constant and relative perimeter thickness are currently hard-coded into `CellBase::SetStressTensor` and thus need to be manually changed. Finding a solution to this problem would be valuable.

The stress acting on the tissue as a whole is similarly calculated. We added `sigma_xx`, `sigma_yy`, and `sigma_xy` as data members to the `Mesh` class. The function `Mesh::SetStressTensor(void)` computes and sets the value of these data members by summing the individual contributions from all cells. Next, the function `Mesh::TissueStress(void)` simply returns the magnitude of maximal tensile stress based on the same eigenvalue calculation as in `CellBase::Stress`. Currently, this function is unused, however, it could be used for applying an external field.

A.2.2 Improved Energy Minimization

We enhanced the energy minimization, as described in section 3.2 of the main text, by introducing several modifications to the core simulation procedure (`TIMESTEP` block in /GUI/VirtualLeaf.cpp).

First, in the original routine, the relaxation cycle terminates as soon as the energy difference of an MC step is below the preset energy threshold: $\theta_H \ \Delta H_{MC} < \theta_H$. To make the minimization more reliable, we introduced a new input parameter: `par.mc_min_steps`, defining the number of consecutive iterations the energy check needs to be passed before the system is considered to be in equilibrium. The `equilibration_counter` keeps track of this number of consecutive iterations. To prevent infinite equilibration, we also added the input parameter `par.mc_max_steps`, the maximum number of iterations for a single relaxation cycle. Default values are set as `par.mc_min_steps` = 100 and `par.mc_max_steps` = $1e5$.

Furthermore, we implemented the option to define a scaling factor α, which links the energy threshold to tissue growth: $\theta_H = \alpha \ \theta_{init}$. Currently, α can only be modified directly in the source code (`scaling_factor`). For the results in this report, we used $\alpha = 0.3 \ln(N_{nodes})$ and $\theta_{init} = 10$.

Although unmentioned in the main text, we also revised the handling of cell wall yielding during the relaxation cycle. Originally, if an edge was marked for cell wall yielding, the yielding procedure initiated just before the energy check. Since the yielding procedure changes the force balance, the energy minimization might be terminated even though the system would need to adjust to this yielding to achieve mechanical equilibrium. We adjusted this behavior so that if cell wall yielding occurs, the `equilibration_counter` resets, providing the opportunity for the system to adapt.

We note that despite these modifications, there are instances where the energy is not properly minimized following a relaxation cycle. This problem notably seems to occur after a quick succession of many cell division events (Figure 2). Moreover, a significant number of MC steps are required to even reach this subpar state. To address this, we in-

cluded the option to limit the amount of cells that divide between two relaxation cycles, and stop inducing growth until no cells have divided before initiating a new relaxation cycle. The option can be toggled in mesh.h > `DoCellHouseKeeping()` > `bool limit_cell_division`. Whether this is an effective solution needs to be tested.

Additionally, part of the energy problem might be related to how the stress is altered after cell division. In the original VirtualLeaf framework, the target area for each daughter cell was simply halved compared to the mother cell, halving the pressure as well. We added the option to keep pressure equal in daughter cells after division. The target area of each daughter cell required for keeping the pressure equal can be calculated as:

$$A_T^{(j)} = x\left[\left(1-\frac{1}{x}\right)A^{(i)} + \frac{1}{x}A_T^{(i)}\right]$$

Here, $A_T^{(j)}$ represents the target area of the daughter cell, x is the area fraction of the daughter cell relative to the area of the mother cell. $A^{(i)}$ and $A_T^{(i)}$ denote the area and target area of the mother cell, respectively.

A.2.3 Division Rules

We proposed two interpretations of the mechanical division rule, which can be accessed in the VirtualLeaf extension. Division rules can be set in the model plugin function `Model::CellHouseKeeping`.

The shortest path rule is activated with `Cell::Divide(void)`, the default option for division in VirtualLeaf. In the case of applying mechanical division rules we use `Cell::DivideOverAxis(Vector &v)` instead, which takes a custom axis as input for placing the cell wall. For the first option, where the average stress on the cell guides the division axis, the input axis is retrieved from `CellBase::Stress(double *stress, Vector *stress_axis)` (see A.2.1). For the second option, we created the function `CellBase::FindHighestTensionEdge(Vector *edge_axis)` to retrieve the axis of the edge with the highest tension within the cell to use as the guiding axis.

A.2.4 The Equilibrium Geometry of Cells

To modify a cell's equilibrium geometry, defining the cell's shape in the absence of external forces, adjustments of edge properties are necessary. In the original version of VirtualLeaf, the spring constant (λ_S) and target length (L_T) of individual edges cannot be altered; instead, a global spring constant (`lambda_length`) is predefined in the `Parameter` object and the target length is a shared `Node` class data member. (Note that `Parameter::target_length` does *not* refer to the target length of edges but to the target length of cells!) In fact, the program does not explicitly track edges or output edge information at all. Here, we introduce the `Edge` class as a means to track individual edges and their properties.

The original VirtualLeaf code actually does define an `Edge` class (in /GUI/node.h), however, this is only used for temporarily keeping track of two connected nodes as input for the cell wall yielding procedure. In the extension, we expanded the `Edge` class's functionality. Since this implementation required many adjustments to the source code we will briefly summarize it:

- Each `Edge` object now contains a `target_length` data member to store the edge's target length. Accordingly, any part of the code requiring the target length of an edge now refers to this variable rather than the global `Node::target_length`.
- All `Edge` objects in the system are cataloged in the `Mesh` object as `edges`, similar to the `Node` and `Cell` objects.
- All `Edge` objects belonging to a cell are referenced in that `Cell` object as `edges`, storing them in the same clockwise order as the cell's `Node` objects.
- `Neighbor` objects now also store references to the two `Edge` objects linked to the edges connecting the node and its two neighboring nodes.
- The data of the `Edge` objects are included in the output under the '`edges`' XML node. We also incorporated backward compatibility, allowing input files without the '`edges`' XML node to still be read. In these cases, `Edge` objects are generated and assigned the global `Node::target_length`.
- The cell division and cell wall yielding methods manage the changes in the (number of) `Edge` objects and their references in other objects. Moreover, in both functions, we added a very ba-

sic option to change target lengths for the new or yielded cell walls. (See tag `'AIMEE A.2.4 VAR'`)

There are still many outstanding tasks to ensure control over all aspects related to the equilibrium geometry of cells during the simulation. First of all, the spring constant needs to be added as a data member to the `Edge` class. In turn, this data member needs to be referred to in the code instead of the global `Parameter::lambda_length`. This task should be relatively straightforward as the main bookkeeping of the `Edge` objects is already taken care of. A more complex challenge lies in implementing and determining how the spring constant and the target lengths should change during the simulation, and how users can change related options within the model plugin or input file.

Furthermore, to obtain full control over the equilibrium cell shape, bending properties also need to be included. The original VirtualLeaf already includes an option for calculating the bending constraint, although it is an untested implementation (see tag '`A.2.4 BENDING`'). To track a rest bending angle and a bending constant, these variables could be incorporated in the `Neighbor` objects since they already contain the two neighboring nodes to calculate the bending angle. However, similar to the `Edge` class, as `Neighbor` objects only relate to nodes and lack a direct connection to the system, a separate class for 'bending' objects might be necessary.

A.2.5 Applying External Tissue Tension

We introduced two basic methods to apply an external tension to the tissue. To simplify access to tissue-wide adjustments, we integrated a new general method into the model plugin, named `Model::TissueHouseKeeping(Mesh *m)`. This function is invoked after the relaxation cycle and `Mesh::DoCellHouseKeeping` in the main routine through `Mesh::DoTissueHouseKeeping(void)`. Within this model plugin method, functions for applying external tension can be called on.

Firstly, to simulate a tensile test by pulling fixed nodes apart, a set of nodes on opposing sides on the tissue must be set to `fixed="true"`, either in the input file or dynamically in the simulation. Subsequently, we use `Mesh::MoveFixedNodes(double dx)` in `Mesh::DoTissueHouseKeeping` to move the fixed nodes along the x-axis. Nodes on the -x side move further along the -x direction, while nodes on the +x side move further along the +x direction.

For the second option, applying tension to the boundary nodes, we use `Mesh::SetBoundaryNodeDisplacements(double dx)` to set $r^{*(k)}$ for the energy constrain as described in Eq. (8). Similar to the `Mesh::MoveFixedNodes`, this function confines the constraint's impact to the x-axis. The Langrange multiplier of the constraint is set by `Parameter::stress_lambda`. Any new boundary nodes created during the relaxation cycle through cell wall yielding will also be included in the constraint with the function `Mesh::SetSingleBoundaryNodeDisplacement`, with an $r^{*(k)}$ based on the difference in $r^{(k)}$ and $r^{*(k)}$ for its boundary node neighbors. This decision was a bit arbritary and needs to be reevalauted.

We want to note that both approaches have been implemented rather crudely, and recommend revising them for further use. Moreover, proper values for `dx` and `stress_lambda` need to be tested.

Lastly, as a potential future option, one could consider implementing an energy constraint based on tissue stress to apply a global tension to the system. Focusing on just the x component for simplicity, the adjusted Hamiltonian would then take the form of:

$$H' = H + \lambda_S \left(\sigma_{xx} - \sigma_{xx}^{ref}\right)^2 \tag{9}$$

This method requires setting a reference value σ_{xx}^{ref}, and calculating the tissue stress before and after the trial displacement of the node. Taking the difference between all elements of the stress tensor compared to a reference stress tensor into account would require finding a proper norm to determine a magnitude of the difference between σ and σ^{ref}.

A.2.6 Model Guide

We added five basic models to the extension for users to sample the features of this extension. Each of these models takes an input file 'ModelX_init.xml', with X corresponding to the model number:

- `Model1`: Division along shortest path. Also contains an extra function for setting dose-response curves. (Fig. 3a)
- `Model2`: Division along maximal tensile stress from the average over the whole cell. (Fig. 3ba)

- `Model3`: Division along the edge with maximal tension of the cell. (Fig. 3c)
- `Model4`: Application of external tension by moving fixed node. (Fig. 5a)
- `Model5`: Application of external tension by a constraint on the boundary nodes. (Fig. 5b)