Development of Elastic Resistive Force Theory & Applications to Uprooting

by

Lale Yılmaz

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2024

Authored by: Lale Yılmaz
Department of Mechanical Engineering
August 23, 2024

Certified by: Ken Kamrin
Professor of Mechanical Engineering and Applied Math, Thesis Supervisor

Accepted by: Nicolas Hadjiconstantinou
Chairman
Department Committee on Graduate Theses

Development of Elastic Resistive Force Theory & Applications to Uprooting

by

Lale Yılmaz

Submitted to the Department of Mechanical Engineering
on August 23, 2024 in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

ABSTRACT

Granular intrusion processes such as sand locomotion, uprooting, and digging are commonly present. While these phenomena can be accurately modeled via discrete element methods and continuum models, this accuracy comes at a great computational cost, especially for large systems. Granular Resistive Force Theory (RFT) is a reduced-order, rate-independent model that has been shown to successfully capture the motion of rigid intruders in granular media, with a reduced computational cost. RFT is based on a rate-independent theory that calculates the force experienced by a body using its direction of velocity. This makes it difficult to handle scenarios that are near-stagnant which occur frequently in uprooting of plants. To overcome this limitation, we introduce elastic RFT (eRFT) which is based on a rate-independent plasticity flow-rule–like criterion, and pair it with deformable intruders. We focus on modeling uprooting processes which inherently have flexible intruders and are often dynamically controlled. This allows us to address both previously mentioned shortcomings of RFT (stagnancy and flexible intruders) at once. By combining eRFT with a nonlinear beam theory to represent slender, inextensible roots we create a speedy computational tool. Using MATLAB, we simulate various uprooting scenarios to better understand anchoring mechanisms of different root geometries. We showcase the validity of eRFT results by comparing them to experimental data. To implement eRFT in ABAQUS, we make use of an existing user subroutine which allows the study of a broader range of intruder materials and shapes. While the subroutine has its limitations, initial comparisons to computational and experimental results are demonstrative.

Thesis supervisor: Ken Kamrin
Title: Professor of Mechanical Engineering and Applied Math

Acknowledgments

There are many people I owe gratitude to for their help and support throughout my academic journey and particulary as I worked to complete this Master's thesis. I would like to begin by thanking my advisor Ken for his support throughout my graduate studies. Besides his vast knowledge and expertise in his field, he inspires me as a generous advisor who allows his students to explore their own ideas on their own pace, but is always there to help. A crucial part of the computational work in this thesis was validated experimentally thanks to the dedicated efforts by Madison Douglas and Taylor Perron, whom I am grateful towards for their collaboration. I am also thankful for the current and past members of the Kamrin group, particularly Will and Shashank, who have made me feel like a part of the group and were more than willing to lend a helping hand whenever I was stuck.

There are many mentors and instructors that have shaped me as a researcher and student. I would like to thank Professor Rohan for his kindness both within and outside the classroom. I had a rough start to graduate school and his kind approach and frequent check-ins with his students was a great factor in making me feel welcome. I would like to thank my mentors from my undergraduate studies at University of Rochester. Professor Perucchio taught me how to conduct research at a high standard and how every detail matters. Professor Muir has taught me how to think like an engineer and how to lead a team. Anytime I am stuck during my studies, I remind myself of his wise words: "remember, you chose this."

Of course none of this academic work would be possible without my dear support network. Anne, Babik, without your unwavering support I would not have been able to achieve any of this. Even though living far from home is difficult, knowing you are always there to pick up my calls anytime means a lot to me.

My close friends from back home have perfected the art of remote friendship now entering my 6th year of international studentship. Burcu, Doğax2, Oğuzhan, Berke, and many more, your support and calls that come out of nowhere mean the world to me. My dear friends from undergrad who are still present in my life both through their visits to Boston and remotely, Anna, Holli, thank you for always being within reach when I needed. The amazing people I have befriended at Boston and MIT that have made the rigorous work more bearable through our game nights and bar crawls, Nina, Hana, Sav, Vanessa, Xavier, Andrew, Chris, Theo, thank you all for sharing both the misery and the fun with me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Studying the interaction of solids with granular media is of interest for a multitude of areas such as terramechanics [1, 2] , impact penetration [3, 4, 5], or uprooting of trees [6, 7]. Various computational tools have been developed to model the phenomena governing the interaction of grains and solids. To capture the behavior of a granular medium accurately through its various phases as it moves, discrete element methods have been developed where each grain is modeled individually. DEM can yield extremely accurate results given that the accurate contact models and interaction properties are applied [8]. The problem with DEM arises when we try to scale up our study area of interest. For problems such as modeling part of a dirt race track or a large tree root structure, DEM approaches become quite costly [9, 10]. Also, often in intrusion problems such as terramechanics or surface locomotion of robots and animals, a detailed response of the granular medium such as a continuously defined stress field or detection of fly-away grains may not be essential to the problem at hand. Oftentimes, the resistance the grains apply to the intruder in response to its motion is sufficient for the application. To reduce the computational cost while preserving the ability to calculate the resistance experienced by an intruder, Li et al [11] adapted resistive force theory to granular media. For applications where the detailed response of the granular continuum is not required, RFT has become the ideal tool for computationally efficient modeling of intrusion problems in grains obeying frictional plasticity [12].

In this work, we will first cover how resistive force theory works and its shortcomings. We will then propose and explain elastic RFT to address these shortcomings. Finally, we will present the applications of elastic RFT to various scenarios to prove it addresses the shortcomings of RFT and matches experimental results. We will also cover the progress on an ABAQUS user subroutine implementation of resistive force theory which will allow for wider and easier usage of RFT.

Besides theoretical extension of resistive force theory to elastic regimes, in this work we also explore its applications to uprooting problems by pairing it with the nonlinear inextensible beam model we developed to model the behavior of roots. Studying uprooting can help understand the underlying mechanisms of plant root anchorage. Plant roots improve terrain stability by anchoring the soil [13, 14] stabilizing river banks [15, 16], and reducing the harm of landslides [17]. The anchoring effect of root systems can be used in geoengineering applications to prevent erosion [18, 14]. However, there is not yet a general theory of anchoring mechanisms. Resistive force theory can be a useful and rapid tool in investigating the effects

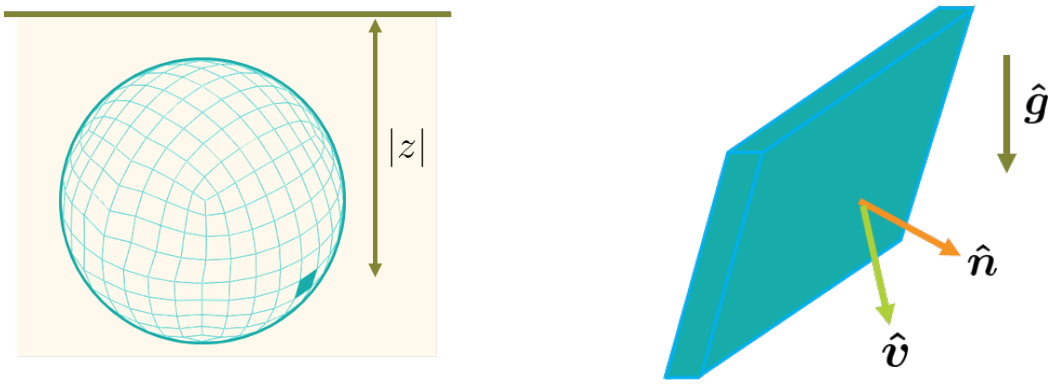
of various root systems. To pair resistive force theory with plant roots, we make use of the nonlinear inextensible beam model we developed to model the behavior of the roots. The numerical results obtained from the simulations combining these beams and elastic RFT to mimic uprooting scenarios were then compared to experiments performed both by our collaborators from the Perron Group in the MIT EAPS department and by the author.

1.1 Granular Resistive Force Theory

Resistive force theory was initially developed to model the motion of intruders through viscous media [19, 20]. In recent years, it has been adapted to granular media [11]. RFT has been shown to accurately model motion of legged robots on granular terrain [11], terramechanics of rigid wheels [9], impact intrusion [21], and burrowing motions mimicking animals [22]. The accuracy of the predictions of resistive force theory in these works when compared to experimental and computational results are higher than those for resistive force theory in viscous fluids [23]. The simplicity and high accuracy of granular resistive force theory is attributed to the fact that the continuum level response of a granular medium can be captured by a frictional plasticity model [24, 25]. The high accuracy paired with the simplicity of calibrations and reduced computational cost of granular resistive force theory has inspired the exploration of the limits of granular RFT, along with work on expanding its applicability which is part of the work in this thesis. In this section we will briefly go over the core concepts and equations governing granular resistive force theory. For more in depth derivations, experimental and computational validations, and discussions, the reader can refer to the aforementioned references.

For an arbitrary intruder moving below the surface of a granular medium, the total resistive force experienced by the intruder can be expressed as the integral of the traction applied onto the intruder by the grains along the intruder's external surface [equation 1.1]. Granular resistive force theory argues that, if we are to discretize the surface of the intruder into small plate surface elements (as can be seen in figure 1.1a), the traction on each surface element should be independent of other surface elements on the intruder [11, 26].

$$\mathbf{f}^{\mathbf{total}} = \int_{\text{surf}} \mathbf{t}\, dA \tag{1.1}$$



(a) (b)

Figure 1.1: Visual overview of rigid RFT assumptions: (a) an intruder moving through granular media with discretized surface elements (b) Surface normal $\hat{\mathbf{n}}$, velocity direction $\hat{\mathbf{v}}$, and gravity direction $\hat{\mathbf{g}}$ highlighted on a single surface element of interest

Furthermore according to granular resistive force theory, the traction on each surface element can be determined via a traction per depth vectorial function, called the $\boldsymbol{\alpha}$ function in this work and some others, and the depth of the surface element. If the surface element is above the surface level, the corresponding resistive force is zero since the element is not in the granular medium. This vectorial $\boldsymbol{\alpha}$ function depends on the surface normal $\hat{\mathbf{n}}$ and velocity direction $\hat{\mathbf{v}}$ of the plate element, along with gravity $\mathbf{g}$ and material parameters [equation 1.2].

$$\mathbf{t} = \boldsymbol{\alpha}(\hat{\mathbf{n}}, \hat{\mathbf{v}}, \mathbf{g}; \text{mat})|z| \tag{1.2}$$

To determine the $\boldsymbol{\alpha}$ function, experimental (as in [11]) or higher order computational (as in [26]) datasets are generated by moving a small rigid plate element with a set surface normal $\hat{\mathbf{n}}$ and velocity direction $\hat{\mathbf{v}}$ in the granular medium of interest. The test is repeated as these parameters are varied. Once enough data points are obtained, the $\boldsymbol{\alpha}$ function values can be plotted as a heat map, as can be seen in figure 1.2 for the 2D case. To be able to plot the independent variables velocity direction and face normal in single axes, we define scalar angles that correspond to the direction of these vectors. As can be seen in figure 1.3, the variable β corresponds to the angle from the horizontal and the face of the plate element, while γ is the angle from the horizontal to the direction of velocity. These definitions apply to 2D-RFT. In 3D a third angle is introduced for the tilt of the plate element [26, 22]. With these angles defined, the $\boldsymbol{\alpha}$ values are plotted with γ, the angle of attack, varying along the horizontal axis and β, the orientation, changing along the vertical axis. Once these heat maps are established with enough data points, we can perform fits to the values of the $\boldsymbol{\alpha}$ function in terms of its input parameters. In [11] trigonometric functions were used to calculate these fits while Agarwal et al. [26] used polynomial functions. For most of the work in this thesis, linear fits are used.

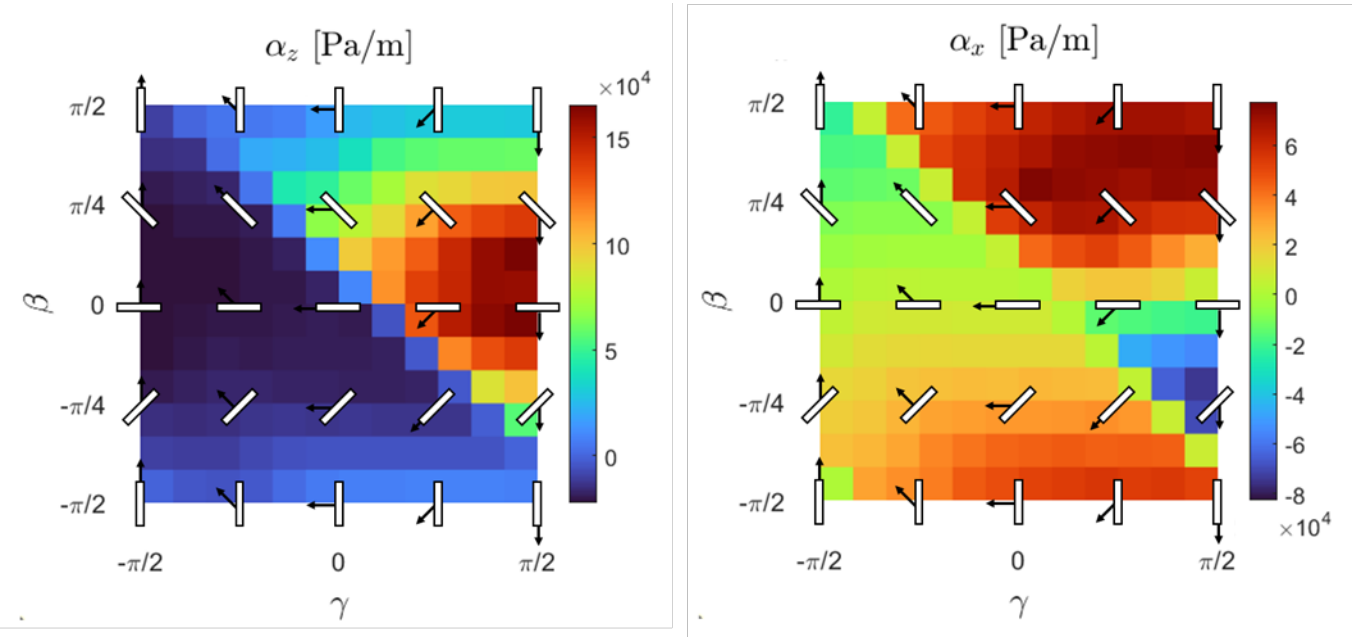


Figure 1.2: $\boldsymbol{\alpha}$ heat maps for horizontal and vertical components for 2D-RFT

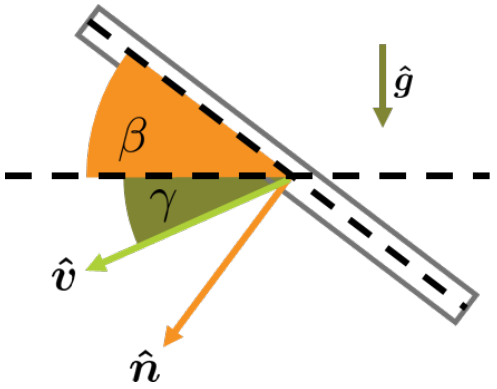


Figure 1.3: Visual definitions of the β and γ angles corresponding to the plate direction and velocity direction, respectively.

Upon investigating the values of the $\boldsymbol{\alpha}$ function for various materials experimentally, Li et al. have discovered that the overall shape of the heat maps for the components of the $\boldsymbol{\alpha}$ functions do not vary when the material of the grains is modified [11]. Thus, a generic $\boldsymbol{\alpha}$ function can be defined that only needs to be adjusted via a single scalar function of material parameters when the granular medium of interest changes [equation 1.3]. This significantly reduces the effort required to apply granular RFT. The scaling material function ξ depends on the critical density of the granular medium ρ_c, the magnitude of gravitational acceleration g, and f: a function of surface friction between the intruder and the granular medium μ_s and the internal friction of the grains μ_{int} [equation 1.4][11, 26].

$$\mathbf{t} = \boldsymbol{\alpha}_{\text{gen}}(\hat{\mathbf{n}}, \hat{\mathbf{v}}, \hat{\mathbf{g}})\xi(\text{mat}, |\mathbf{g}|)|z| \tag{1.3}$$

$$\xi = \rho_c \, g \, f(\mu_s, \mu_{\text{int}}) \tag{1.4}$$

With continuous definitions of traction per area per depth in terms of the plate orientation and velocity direction, any intruder can be prescribed a resistive force, provided that all plate elements have a well-defined velocity direction.

1.1.1 Leading Edge Hypothesis

The resistive force on a surface element is calculated as outlined above only if the surface element is on the leading edge. The leading edge is the side of the intruder at a point that is moving into the grains surrounding itself. Resistive forces are only applied to the leading edge since in the wake of the motion the "back side" or "following edge" should experience negligible forces from the grains. The evaluation of the leading edge is performed by calculating the dot product of the face normal of the surface element $\hat{\mathbf{n}}$ and velocity direction $\hat{\mathbf{v}}$. If this dot product is positive or zero on the surface element, the element is considered in the leading edge and the resistive force on the element is calculated as outlined above. If the dot product is negative, the surface element is assigned zero resistive force.

1.1.2 Calibration of RFT Data for Different Materials

As previously mentioned, the shape of the $\boldsymbol{\alpha}$ functions are generic for any material. When we want to adjust it for a new set of intruder material - grain combination, we need to follow a calibration process. To do so, one must pick a single point on the $\boldsymbol{\alpha}$ plots that corresponds to a plate orientation and motion direction. Typically tests that are easier to perform are chosen for experimental ease such as a horizontal plate being pulled out or pushed in a bed of grains. Then, an experiment mimicking the desired point on the $\boldsymbol{\alpha}$ plot should be conducted. The $\boldsymbol{\alpha}$ values should then all be linearly scaled such that the point chosen on the $\boldsymbol{\alpha}$ plot matches the experimentally measured resistive force.

1.2 Shortcomings of Granular/Rigid RFT

The ability of resistive force theory to capture the forces experienced by intruders in granular media under steady state kinematically controlled motions is remarkable and has been verified against both experimental results and continuum models [11, 26, 21]. However, since the definition of the resistive force relies on the direction of velocity, in cases where a surface element has zero velocity the resistive force is not well defined. The absence of velocity can occur in various scenarios including the stagnant point at the bottom of a rotating wheel and any static equilibria. Using granular RFT, any object being dropped into grains never reaches equilibrium since once the resistive forces balance the weight of the object, the object has zero velocity. In the absence of velocity there is no resistive forces and the self-weight of the object mobilizes it again. The lack of a well-defined traction in the absence of velocity for granular resistive force theory is similar to the absence of a well-defined stress tensor in the absence of plastic strains in rigid plasticity models. Motivated by this similarity, we rename granular RFT that has been developed so far "Rigid RFT".

We propose "elastic RFT" to mitigate this problem. With elastic RFT we treat the steady-state resistive forces which come from the $\boldsymbol{\alpha}$ functions as yield tractions that apply once the granular media flows steadily around the intruder element. Before yield, and thus in the absence of velocity, we define an elastic traction which arises from the bulk stiffness of the grains. We mimic this by introducing an RFT stiffness.

When comparing the results from rigid RFT to DEM simulations for reciprocal motion of an intruder, Zhang et al. have found that while for rigid RFT the resistive force would switch directions and magnitudes following a square wave pattern, the DEM results provided resistive forces that gradually built up to the steady-state value of the resistive force [27]. With elastic RFT we aim to mimic this gradual build-up of the resistive forces to their steady-state values.

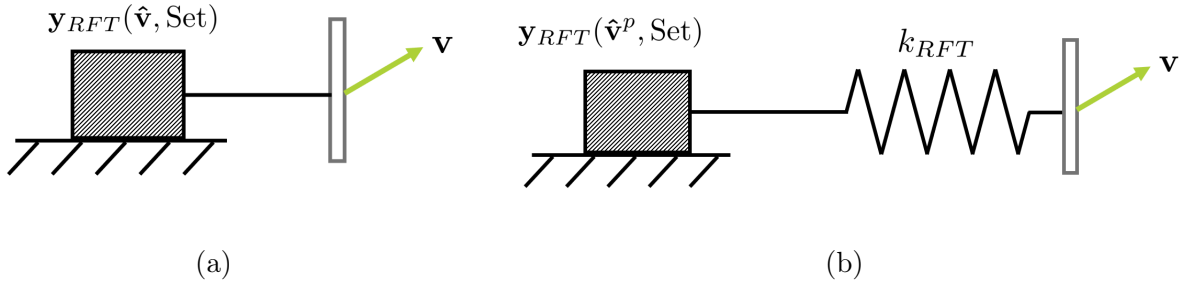
Another unexplored avenue with rigid RFT is that thus far it has only been paired with either rigid intruders [1], or deformable intruders with prescribed kinematics [11, 22]. Many common intruders in grains such as snakes, plant roots, soft robots are elastic and deform in response to the forces applied by the grains onto them. We anticipate the resulting change in shape of the intruder to in turn affect the resistive force calculations. In this work, we first explore the effect of intruder flexibility via beams modeling plant roots, which is outlined in chapter 3. Chapter 4.4 discusses the effect of intruder flexibility on resistive forces. We then attempt to implement resistive force theory as a user subroutine in the commercial finite element analysis software ABAQUS so the intruder can be modeled with a wider variety of material models in chapter 4.6.

Chapter 2

Elastic RFT

To resolve the issue of an unknown traction in the absence of velocity observed in rigid RFT, we propose elastic resistive force theory, which introduces an elastic traction to the existing RFT formulation. This also allows RFT to transition from a purely plastic model to a hypoelastic-plastic one which has been shown to accurately capture the continuum behavior of granular media and is thus used widely [28, 29]. Plasticity theory has been used as a successful parallel for deriving and explaining constitutive relations where the tractions are governed by frictional yield conditions [30].

2.1 Analog Models for Rigid and Elastic RFT



(a) (b)

Figure 2.1: Analog models for rigid (a) and elastic (b) RFT.

We make use of one-dimensional analog models to help visualize the constitutive models discussed in this section for resistive force theory, similar to their use for plasticity models [31]. Both models are used to model the traction applied to the plate element of interest with velocity $\mathbf{v}$ and and Set parameters corresponding to the plate which are the face normal $\hat{\mathbf{n}}$, depth $|z|$, gravity $\mathbf{g}$, and material parameters. The rigid RFT analog model (figure 2.1a) consists of a slider connected to the plate element. The slider captures the plastic response due to the flow of grains. Below the yield force set by rigid RFT corresponding to the velocity and Set parameters corresponding to the plate, the slider does not move. Thus, when the plate is moving the force in the system is that governing the slider. However, when

the slider is not moving, we cannot determine the traction experienced by the plate element. To remedy this, we introduce an elastic spring with stiffness k_{RFT}. This way, even when the plate is not moving or the slider is not sliding, by keeping track of the displacement the spring has experienced from its neutral position we can calculate an elastic traction. The spring captures the bulk elastic response of the grains, and allows for a well-defined traction below the yield of grains.

The sliders in both analog models for RFT are governed by a yield strength that depends on the velocity direction and Set parameters which includes the face normal of the plate, the material parameters, depth of the plate, and gravity. Holding the Set parameters constant, which are typically known for a plate element at a time of interest, we can create yield traction surfaces that govern the sliders. For rigid RFT, these surfaces vary as the velocity direction changes. For elastic RFT (eRFT) we construct closed yield surfaces by holding the Set parameters constant and varying the plastic velocity direction. Examples of yield surfaces for different plate normal directions can be seen in figure 2.2. If the traction vector the plate is experiencing is within the locus corresponding to its parameters, it is still within the elastic regime and thus there is no flow of grain around the plate. If the traction grows and reaches the yield surface, then there is plastic flow, corresponding to the sliding of the slider in the analog models. For each locus, the vertices are created by holding the plate orientation and other Set parameters constant while varying the velocity direction. Between loci, the plate orientation is varied while other Set parameters are still constant. Essentially, these are points along a horizontal line of the alpha plots since the plate orientation angle β does not vary along a horizontal line on these heat maps. Besides the experimentally measured points, the points generated by the fits, such as the trigonometric [11, 22] or polynomial [26] fits, defining the alpha functions can be used to generate these loci. The continuous alpha functions generated by these fits are useful for rigid RFT, however since we evaluate our yield condition by checking whether the traction vector is within the yield locus, it can often be difficult with higher order fits which might have loops and self intersections (figure 2.3).

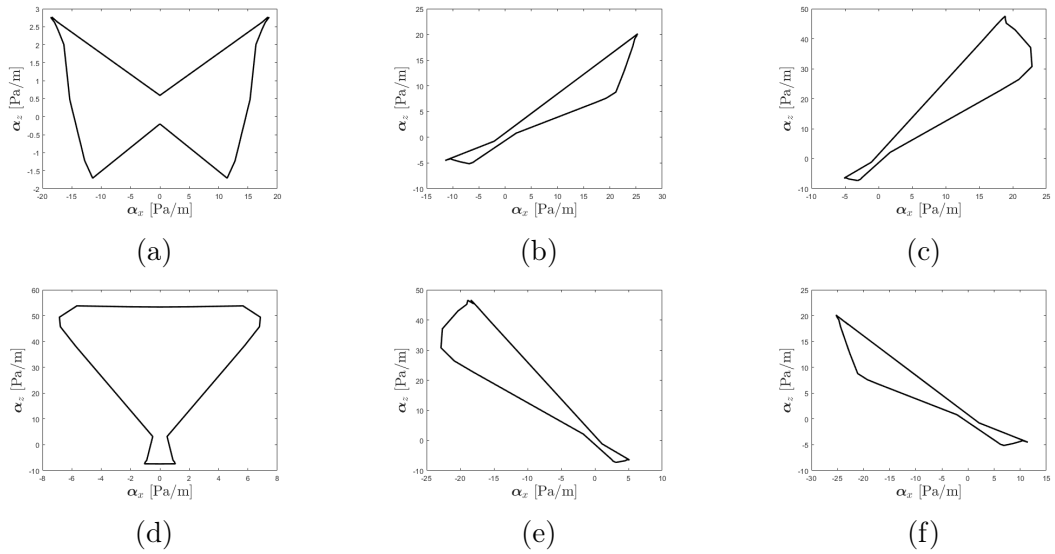


Figure 2.2: Linearly interpolated loci for different plate orientations

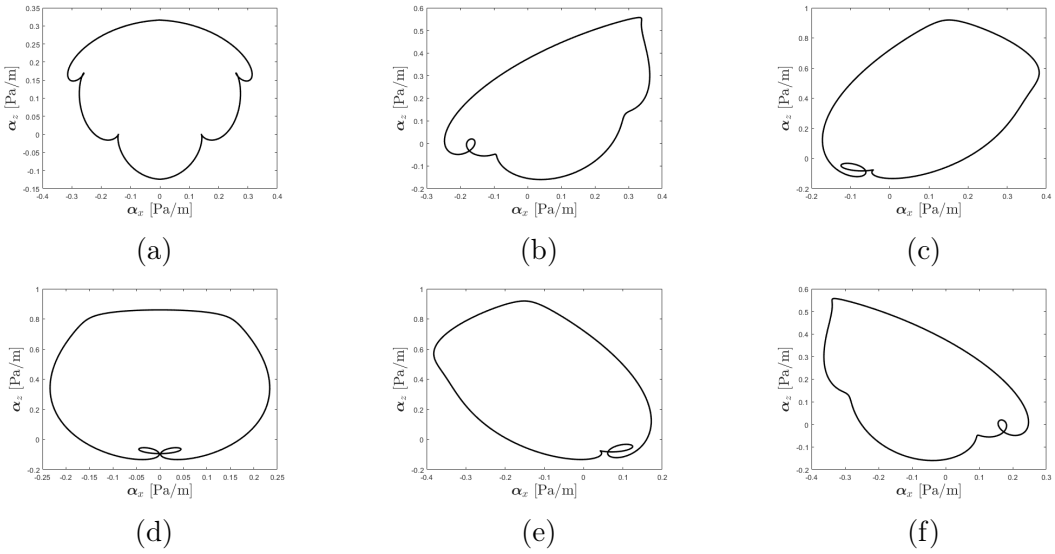


Figure 2.3: Loci generated via 3^{rd} order polynomial interpolations for different plate orientations

The spring in elastic RFT allows us to assign a well-defined traction to the system below the yield threshold of the slider, corresponding to the region inside the yield surfaces. We have followed 2 different approaches to define the manner in which the elastic traction approaches the yield:

1. Non-codirectional Linear Elastic RFT
2. Codirectional Nonlinear Elastic RFT

For these two versions, linear and nonlinear describe the elastic response, while codirectional and non-codirectional refer to the relationship between elastic displacement and plastic velocity. This is slightly different than the usage of these words in plasticity theory, however it is useful for us when distinguishing between the two models. For both of these methods we follow the same kinematic steps initially.

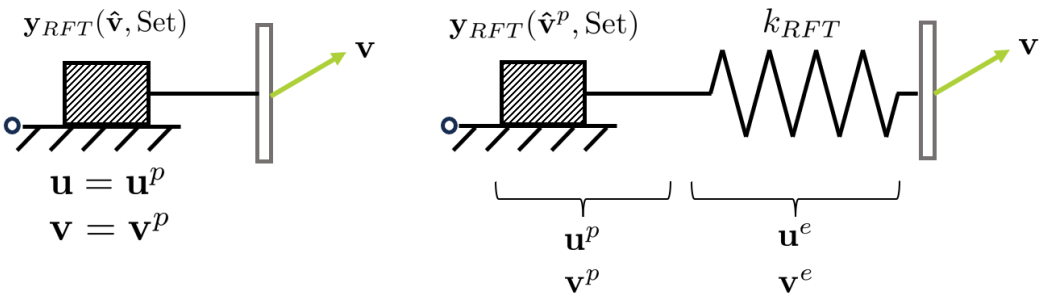


Figure 2.4: Decomposition of displacements and velocities into elastic and plastic components for rigid and elastic RFT

For rigid RFT, any displacement of the plate will result in the slider moving, thus any displacement and velocity are plastic. For elastic RFT, we assume an additive kinematic decomposition of the displacement $\mathbf{u}$ and velocity $\mathbf{v}$ into elastic and plastic components over the spring and slider, respectively. The elastic displacement over the spring is recovered once the plate is released, however the plastic displacement over the slider will remain unless further plasticity is achieved. The difference between the two elastic RFT models arise from the definition of the direction of the elastic traction.

For a displacement $\mathbf{u}$ from a reference configuration, the elastic and plastic components are defined as in equation 2.1 and velocity as in equation 2.2 where the superscripts e and p denote the elastic and plastic components.

$$\mathbf{u} = \mathbf{u}^e + \mathbf{u}^p \tag{2.1}$$

$$\mathbf{v} = \mathbf{v}^e + \mathbf{v}^p \tag{2.2}$$

2.2 Non-codirectional Linear Elastic RFT

Once the displacement is decomposed into elastic and plastic components, the traction in the system is defined as the RFT stiffness times the elastic displacement as can be seen in equation 2.3. We assume a linear elastic response such that the magnitude of the traction in the elastic regime scales linearly with the elastic displacement, and the direction of the traction is along the direction of the elastic displacement.

$$\boldsymbol{t} = k_{RFT}\,\mathbf{u}^e = k_{RFT}\,(\mathbf{u} - \mathbf{u}^p) \tag{2.3}$$

We keep track of the rate of change in traction due to elastic velocity [equation 2.4]. This expression is a material time derivative as long as the velocity is defined as the relative velocity of the plate with respect to the background material, which is typically stationary. Since RFT defines traction on material plate elements, after this step, material frame indifference is satisfied.

$$\dot{\boldsymbol{t}} = k_{RFT}\left(\mathbf{v} - \mathbf{v}^{p}\right) \tag{2.4}$$

2.3 Codirectional Nonlinear Elastic RFT

Codirectional eRFT assumes that the elastic tractions within the system grow in the direction of the yield strength that corresponds to the direction of elastic displacement.

For this version of RFT, we assume that the elastic traction grows in the direction of the flow strength corresponding to the direction of elastic displacement. The flow strength is not necessarily along the elastic displacement, which results in a nonlinear elastic relationship. To evaluate the direction in which the traction should be, we evaluate the yield strength function y_{RFT} by substituting the velocity direction with the elastic displacement direction.

$$\boldsymbol{t} = k_{RFT}|\mathbf{u}^{e}|\frac{\mathbf{y}_{RFT}(\hat{\mathbf{u}}^{e}, \mathrm{Set})}{|\mathbf{y}_{RFT}(\hat{\mathbf{u}}^{e}, \mathrm{Set})|} \tag{2.5}$$

As can be seen in equation 2.5, below the yield threshold, the magnitude of the traction in the system scales linearly with the elastic displacement magnitude and RFT stiffness. The nonlinearity arises from the direction of the traction which is embedded in the fractional term which defines the direction of the traction. During flow, this traction is also equal to the yield strength corresponding to the plastic velocity direction (equation 2.7). The yield surfaces for linear and nonlinear elastic RFT are identical.

$$\mathbf{t} = \mathbf{y}_{RFT}(\hat{\mathbf{v}}^{p}, \hat{\mathbf{n}}, \hat{\mathbf{g}}, |z|, \mathrm{mat}) \tag{2.6}$$

2.4 Flow Rules for Elastic RFT

Continuing the parallels between resistive force theory and plasticity theory, we can define a flow rule governing the traction and velocity in the system, similar to plastic flow rules relating the stresses to the strains [32]. During plastic flow, the resistive traction is equal to the yield strength $\mathbf{y}_{RFT}$ which forms identical yield surfaces as those defined by rigid RFT, however for elastic RFT we assume that the yield strength depends not on the total velocity direction, but on the direction of the plastic velocity (equation 2.7). During yield, the traction will be defined in terms of the plastic velocity and the Set parameters.

$$\mathbf{t} = \mathbf{y}_{RFT}(\hat{\mathbf{v}}^{p}, \hat{\mathbf{n}}, \hat{\mathbf{g}}, |z|, \mathrm{mat}) \tag{2.7}$$

We can then define the yield condition h (equation 2.8) which evaluates whether the traction is causing yield or is in the elastic regime. Here, $\hat{\mathbf{v}}^{p*}$ defines the direction of plastic velocity that would be obtained if the traction $\mathbf{t}$ were scaled in magnitude so as to reach the yield locus. If traction $\mathbf{t}$ is less than the yield strength, thus the traction vector is within the

locus and the value of h is less than zero, the grains around the plate element are still in the elastic regime. If the traction $\mathbf{t}$ is equal to the yield strength, the traction vector is on the yield locus and $h = 0$, plastic flow can occur.

$$h = |\mathbf{t}| - |\mathbf{y}_{RFT}(\hat{\mathbf{v}}^{p*}(\mathbf{t}), \hat{\mathbf{n}}, \hat{\mathbf{g}}, |z|, \text{mat})| \leq 0 \tag{2.8}$$

where $\hat{\mathbf{v}}^{p*}$ returns the point on the yield surface corresponding to the direction of the traction $\mathbf{t}$.

During flow, the traction must be on the yield surface and remain on the yield surface. This corresponds to both $h = 0$ and its time derivative $\dot{h} = 0$. The time derivative of h can be evaluated as in equation 2.9.

$$\dot{h} = \frac{\mathbf{t}}{|\mathbf{t}|} \cdot \dot{\mathbf{t}} - \frac{\mathbf{y}_{RFT}}{|\mathbf{y}_{RFT}|} \cdot \dot{\mathbf{y}}_{\mathbf{RFT}} \tag{2.9}$$

We can apply the chain rule to expand $\dot{\mathbf{y}}_{\mathbf{RFT}}$

$$\dot{\mathbf{y}}_{RFT} = \frac{\partial \mathbf{y}_{RFT}}{\partial \hat{\mathbf{v}}^{p*}} \left(\frac{\partial \hat{\mathbf{v}}^{p*}}{\partial \mathbf{t}} \dot{\mathbf{t}} \right) + \frac{\partial \mathbf{y}_{RFT}}{\partial \hat{\mathbf{n}}} \dot{\hat{\mathbf{n}}} \tag{2.10}$$

Plugging the expression for $\dot{\mathbf{y}}_{\mathbf{RFT}}$ into the equation for $\dot{h} = 0$ and rearranging to isolate $\dot{\mathbf{t}}$ we obtain equation 2.11

$$\left[\frac{\mathbf{t}}{|\mathbf{t}|} - \left(\frac{\partial \hat{\mathbf{v}}^{p*}}{\partial \mathbf{t}} \right)^{\mathrm{T}} \left(\frac{\partial \mathbf{y}_{RFT}}{\partial \hat{\mathbf{v}}^{p*}} \right)^{\mathrm{T}} \frac{\mathbf{y}_{RFT}}{|\mathbf{y}_{RFT}|} \right] \cdot \dot{\mathbf{t}} = \frac{\mathbf{y}_{RFT}}{|\mathbf{y}_{RFT}|} \cdot \left(\frac{\partial \mathbf{y}_{RFT}}{\partial \hat{\mathbf{n}}} \dot{\hat{\mathbf{n}}} \right) \tag{2.11}$$

During yield, $\hat{\mathbf{v}}^{p*}$ is equal to $\hat{\mathbf{v}}^{p}$. We can determine $\hat{\mathbf{v}}^{p}$ by inverting the flow strength function. For a particular element of interest, the Set parameters Set $= \{\hat{\mathbf{n}}, \hat{\mathbf{g}}, |z|, \text{mat}\}$ are all known for a plate of interest at a set time from geometry, material parameters, and experimental parameters. Then, $\hat{\mathbf{v}}^{p}$ is the inverse of the flow strength function returning the traction $\mathbf{t}$ during yield.

$$\hat{\mathbf{v}}^{p} = \mathbf{y}^{-1}_{RFT}(\mathbf{t}, \text{Set}) \tag{2.12}$$

With the direction of plastic velocity known, once we can get an expression for the magnitude of plastic velocity we can combine the two for a flow rule. If we express the plastic velocity as a magnitude and direction, $\mathbf{v}^{p} = v^{p}\hat{\mathbf{v}}^{p}$, where v^{p} is the unknown plastic velocity magnitude we will obtain from the explicit form of the flow rule.

After this step, the approach to obtain the flow rule for the two different formulations of elastic RFT differ.

2.4.1 Flow Rule for Non-codirectional Linear elastic RFT

To continue deriving the flow rule for non-codirectional linear elastic RFT, in the $\dot{h} = 0$ equation, we can substitute the time rate of change of the traction as defined by 2.4.

$$\left[\frac{\mathbf{t}}{|\mathbf{t}|} - \left(\frac{\partial \hat{\mathbf{v}}^{p*}}{\partial \mathbf{t}} \right)^{\mathrm{T}} \left(\frac{\partial \mathbf{y}_{RFT}}{\partial \hat{\mathbf{v}}^{p*}} \right)^{\mathrm{T}} \frac{\mathbf{y}_{RFT}}{|\mathbf{y}_{RFT}|} \right] \cdot \dot{\mathbf{t}} = \frac{\mathbf{y}_{RFT}}{|\mathbf{y}_{RFT}|} \cdot \left(\frac{\partial \mathbf{y}_{RFT}}{\partial \hat{\mathbf{n}}} \dot{\hat{\mathbf{n}}} \right) \tag{2.13}$$

The time derivative of the traction for non-codirectional linear elastic RFT (equation 2.4) can be substituted in.

$$\left[\frac{\mathbf{t}}{|\mathbf{t}|} - \left(\frac{\partial \hat{\mathbf{v}}^{p*}}{\partial \mathbf{t}}\right)^{\mathrm{T}} \left(\frac{\partial \mathbf{y}_{RFT}}{\partial \hat{\mathbf{v}}^{p*}}\right)^{\mathrm{T}} \frac{\mathbf{y}_{RFT}}{|\mathbf{y}_{RFT}|}\right] \cdot k_{RFT}(\mathbf{v} - v^p \hat{\mathbf{v}}^p) = \frac{\mathbf{y}_{RFT}}{|\mathbf{y}_{RFT}|} \cdot \left(\frac{\partial \mathbf{y}_{RFT}}{\partial \hat{\mathbf{n}}} \dot{\hat{\mathbf{n}}}\right) \tag{2.14}$$

In this equation, the only unknown is the magnitude of plastic velocity v^p. To write the explicit flow rule succinctly, everything in the square brackets is renamed as $\mathbf{m}$ [equation 2.15]. Once isolated, we can solve for the explicit form of the magnitude of plastic velocity 2.16. Combined with the direction of plastic velocity solved for earlier in equation 2.12, we have the complete flow rule for non-codirectional linear elastic RFT.

$$\mathbf{m} = \left[\frac{\mathbf{t}}{|\mathbf{t}|} - \left(\frac{\partial \hat{\mathbf{v}}^{p*}}{\partial \mathbf{t}}\right)^{\mathrm{T}} \left(\frac{\partial \mathbf{y}_{RFT}}{\partial \hat{\mathbf{v}}^{p*}}\right)^{\mathrm{T}} \frac{\mathbf{y}_{RFT}}{|\mathbf{y}_{RFT}|}\right] \tag{2.15}$$

$$\mathbf{v^p} = v^p \hat{\mathbf{v}}^{\mathbf{p}} \tag{2.16}$$

$$v^p = \frac{\mathbf{m} \cdot \mathbf{v}}{\mathbf{m} \cdot \hat{\mathbf{v}}^p} - \frac{1}{k_{RFT} \mathbf{m} \cdot \hat{\mathbf{v}}^p} \frac{\mathbf{y}_{RFT}}{|\mathbf{y}_{RFT}|} \cdot \left(\frac{\partial \mathbf{y}_{RFT}}{\partial \hat{\mathbf{n}}} \dot{\hat{\mathbf{n}}}\right) \tag{2.17}$$

2.4.2 Flow Rule for Codirectional Nonlinear elastic RFT

To derive the flow rule for codirectional nonlinear eRFT, we expand equation 2.11 by substituting $\dot{\mathbf{t}}$ with equation 2.18.

$$\dot{\mathbf{t}} = \frac{\partial \mathbf{t}}{\partial \mathbf{u}^e} \dot{\mathbf{u}}^e \tag{2.18}$$

We can then substitute the $\dot{\mathbf{u}}^e$ term with the plastic velocity subtracted from the total velocity. Since the direction of plastic velocity $\hat{\mathbf{v}}^p$ can be determined from 2.12 directly, after this substitution the only unknown is the magnitude of plastic velocity.

$$\dot{\mathbf{u}}^e = \mathbf{v}^e = \mathbf{v} - \mathbf{v}^p = \mathbf{v} - v^p \hat{\mathbf{v}}^p \tag{2.19}$$

To write the final version of the flow rule shortly, we introduce another shorthand for one of the terms. The tensorial quantity $\mathbf{N}$ is the transpose of the derivative of the traction with respect to the elastic displacement direction, which can be evaluated from equation 2.5.

$$\mathbf{N} = \left(\frac{\partial \mathbf{t}}{\partial \mathbf{u}^e}\right)^{\mathrm{T}} \tag{2.20}$$

Isolating the plastic velocity magnitude we get the final expression for the flow rule governing codirectional nonlinear elastic RFT [equation 2.21].

$$v^p = \frac{\mathbf{v} \cdot (\mathbf{N}\mathbf{m})}{\hat{\mathbf{v}}^p \cdot (\mathbf{N}\mathbf{m})} - \frac{1}{\hat{\mathbf{v}}^p \cdot (\mathbf{N}\mathbf{m})} \frac{\mathbf{y}_{RFT}}{|\mathbf{y}_{RFT}|} \cdot \left(\frac{\partial \mathbf{y}_{RFT}}{\partial \hat{\mathbf{n}}} \dot{\hat{\mathbf{n}}}\right) \tag{2.21}$$

2.5 Numerical Implementation of Elastic RFT

To numerically simulate an intruder's motion via discrete timesteps, we calculate the resistive forces with a trial return, explicit/implicit Euler algorithm [33].

With the traction, position, velocity, elastic velocity, plastic velocity and external loads known at time t, we first assume a purely elastic step and calculate a trial traction. We then check if the trial traction is inside the yield locus corresponding to the Set parameters of the plate element at timestep t. This is done by the *inpoly* command available in MATLAB which evaluates whether the point defined by the traction vector is inside the closed polygon generated by the vertex points defining the locus. If the trial traction vector is inside the locus, the elastic step is accepted and the solver advances to time $t + dt$. If however the trial traction is outside the yield locus, yield has occured in this timestep and we must correct the trial traction so it lands on the locus. This step is different for the two different elastic RFT methods.

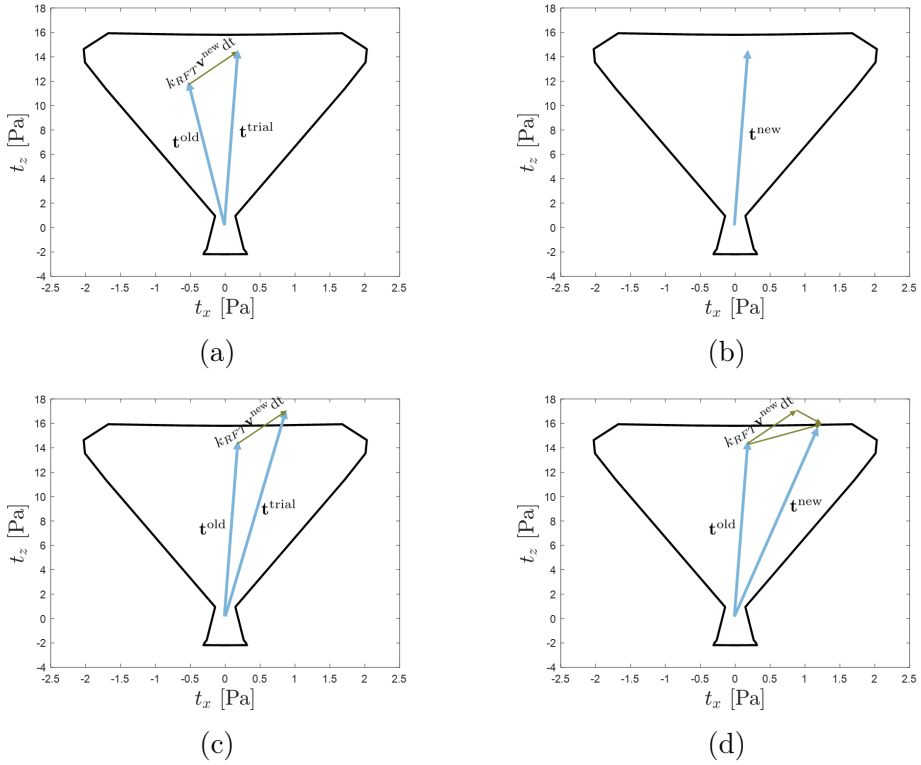


Figure 2.5: Visual representation of the steps for the numerical of the traction update through time and evaluation of the flow rule

2.5.1 Linear Elastic RFT: Numerical Solution

Algorithm 1 Algorithm for evaluating the flow rule numerically for linear elastic RFT

knowing $\mathbf{t}, \mathbf{u}, \mathbf{v}^{new}, \mathbf{v}^{e}, \mathbf{v}^{p}$ at timestep t
take an elastic trial step ▷ figure 2.5a
set the new elastic velocity to the new velocity
$\mathbf{v}^{e,new} = \mathbf{v}^{new}$
$\mathbf{t}^{trial} = \mathbf{t} + k_{RFT}\,\mathbf{v}^{new}\,\mathrm{dt}$
if $\mathbf{t}^{trial}$ is inside or on the locus **then** ▷ figure 2.5b
  $\mathbf{t}^{new} = \mathbf{t}^{trial}$
  $\mathbf{v}^{p,new} = \mathbf{v}^{p,old}$
else $\mathbf{t}^{trial}$ is outside the locus ▷ figure 2.5c
  solve for $\mathbf{v}^{p,new}$ such that ▷ figure 2.5d
  $\mathbf{v}^{new} = \mathbf{v}^{p,new} + \mathbf{v}^{e,new}$ & $\mathrm{t}^{new} = \mathbf{t}^{old} + k_{RFT}\mathbf{v}^{e,new}\,\mathrm{dt} = \mathbf{y}_{RFT}(\hat{\mathbf{v}}^{p,new}, \mathrm{Set})$
end if

Solving for the Plastic Velocity Direction for Linear Elastic Velocity

For linear noncodirectional elastic RFT, we can solve for the plastic velocity direction following the steps below. Assuming a fully elastic step, we update the resistive force as:

$$\mathbf{t}^{trial} = \mathbf{t}^{old} + k_{RFT}\,\mathbf{v}^{new}\,dt \tag{2.22}$$

If this trial force lies outside the yield locus, then it means that the grains are flowing and the trial traction must be returned back such that it lies on the yield locus at the new timestep. To perform this "return" step, we assume that the change in Set parameters, such as the face normal and depth, are negligible within the timestep.

$$\mathbf{t}^{new} = \mathbf{t}^{old} + k_{RFT}\,(\mathbf{v} - \mathbf{v}^{p})\,\mathrm{dt} = \mathbf{y}_{RFT}(\hat{\mathbf{v}}^{p}, \mathrm{Set}) \tag{2.23}$$

Rearranging and decomposing equation 2.23, we obtain the following three equations.

$$\mathbf{t}^{trial} - \mathbf{y}_{RFT}(\hat{\mathbf{v}}^{p}, \mathrm{Set}) = k_{RFT}\,\mathbf{v}^{p}\,\mathrm{dt} \tag{2.24}$$

$$\mathbf{t}_{x}^{trial} - \mathbf{y}_{RFT,x}(\hat{\mathbf{v}}^{p}, \mathrm{Set}) = k_{RFT}|\mathbf{v}^{p}|\cos(\gamma^{p} - \pi)dt \tag{2.25}$$

$$\mathbf{t}_{y}^{trial} - \mathbf{y}_{RFT,y}(\hat{\mathbf{v}}^{p}, \mathrm{Set}) = k_{RFT}|\mathbf{v}^{p}|\sin(\gamma^{p} - \pi)dt \tag{2.26}$$

The $-\pi$ comes from the definition of γ in the RFT formulation. We can now divide the x-direction equation by the y-direction equation and rearrange to obtain the following:

$$(\mathbf{t}_{x}^{trial} - \mathbf{y}_{RFT,x}(\mathbf{v}^{p}))\sin(\gamma^{p} - \pi) = (\mathbf{t}_{y}^{trial} - \mathbf{y}_{RFT,y}(\mathbf{v}^{p}))\cos(\gamma^{p} - \pi) \tag{2.27}$$

We can then solve this by finding the instersections of these two curves which can be done using the *polyxpoly* command in MATLAB. Sometimes these curves have multiple intersections. In such cases the intersection that is closest to the trial traction is selected. If there are no intersections, the trial traction is accepted for that timestep which often results in a solution in the next iteration.

2.5.2 Linear Codirectional Elastic RFT: Numerical Solution

Algorithm 2 Algorithm for evaluating the flow rule numerically for nonlinear elastic RFT

knowing $\mathbf{t}, \mathbf{u}, \mathbf{v}^{new}, \mathbf{u}^e, \mathbf{u}^p$ at timestep t
take an elastic trial step ▷ figure 2.5a
set the new elastic velocity to the new velocity
$\mathbf{v}^{e,new} = \mathbf{v}^{new}$
$\mathbf{u}^{e,new} = \mathbf{u}^e + \mathbf{v}^{new}\,\mathrm{dt}$
$\mathbf{t}^{trial} = \mathbf{t} + k_{RFT}|\mathbf{v}^{new}\mathrm{dt}|\frac{\mathbf{y}_{RFT}(\hat{\mathbf{u}}^e,\mathrm{Set})}{|\mathbf{y}_{RFT}(\hat{\mathbf{u}}^e,\mathrm{Set})|}$
if $\mathbf{t}^{trial}$ is inside or on the locus **then** ▷ figure 2.5b
  $\mathbf{t}^{new} = \mathbf{t}^{trial}$
  $\mathbf{v}^{p,new} = \mathbf{v}^p$
  $\mathbf{u}^{p,new} = \mathbf{u}^p$
else $\mathbf{t}^{trial}$ is outside the locus ▷ figure 2.5c
  solve for $\mathbf{v}^{p,new}$ such that ▷ figure 2.5d
  $\mathbf{t}^{new} = \mathbf{t}^{old} + k_{RFT}|\mathbf{v}^{e,new}\mathrm{dt}|\frac{\mathbf{y}_{RFT}(\hat{\mathbf{u}}^e,\mathrm{Set})}{|\mathbf{y}_{RFT}(\hat{\mathbf{u}}^e,\mathrm{Set})|}\,\mathrm{dt} = \mathbf{y}_{RFT}(\hat{\mathbf{v}}^{p,new},\mathrm{Set})$ &
  $\mathbf{v}^{new} = \mathbf{v}^{p,new} + \mathbf{v}^{e,new}$
  adjust and update the rest of the variables
  $\mathbf{u}^{e,new} = \mathbf{u}^e + \mathbf{v}^{e,new}\mathrm{dt}$
  $\mathbf{u}^{p,new} = \mathbf{u}^p + \mathbf{v}^{p,new}\mathrm{dt}$
end if

Solving for the Plastic Velocity Direction for Nonlinear Elastic Velocity and Proof of Codirectionality

For the nonlinear codirectional formulation of elastic RFT, the elastic traction is defined following equation 2.5. After the elastic trial step for this formulation, if the trial displacement leads to a traction outside the yield locus, it must be corrected by some correction factor $\mathbf{R}$ such that the new corrected elastic displacement at the end of the timestep yields the correct elastic traction which matches the yield traction due to the direction of plastic velocity [equation 2.7].

$$\mathbf{u}^{e,new} = \mathbf{u}^{e,trial} - \mathbf{R} = \mathbf{u}^{e,trial} - \mathbf{u}^{e,old} + \mathbf{v}^{e,new}\,\mathrm{dt} = \mathbf{u}^{e,trial} - \mathbf{v}^{p,new}\,\mathrm{dt} \tag{2.28}$$

$$\mathbf{R} = \mathbf{v}^{p,new}\,\mathrm{dt} \tag{2.29}$$

$$\mathbf{t}^{elastic}(\mathbf{u}^{e,new}) = \mathbf{y}_{RFT}(\hat{\mathbf{v}}^{p,new}) = \mathbf{y}_{RFT}\left(\frac{\hat{\mathbf{R}}}{\mathrm{dt}}\right) \tag{2.30}$$

The elastic traction is defined following equation 2.5, such that its direction is already determined by the $\mathbf{y}_{RFT}$ function. Thus for the elastic and plastic traction to be equal to each other as in equation 2.30, the direction provided from the $\mathbf{y}_{RFT}$ function must be the same for both inputs $\hat{\mathbf{u}}^{\mathbf{e,new}}$ and $\mathbf{v}^{p,new}$. For this to be true, the direction of both of these inputs must be the same since the yield traction function is one-to-one. The direction of the corrected elastic displacement must be along the direction of the updated plastic velocity.

Both the trial and corrected elastic displacements are along the plastic velocity direction, earning this formulation of elastic RFT the "co-directional" title.

$$k_{RFT}|\mathbf{u}^{e,new}|\frac{\mathbf{y}_{RFT}(\hat{\mathbf{u}}^{e,new},\text{Set})}{|\mathbf{y}_{RFT}(\hat{\mathbf{u}}^{e,new},\text{Set})|} = \mathbf{y}_{RFT}(\hat{\mathbf{v}}^{p,new},\text{Set}) \tag{2.31}$$

$$\mathbf{u}^{e,new} \parallel \mathbf{v}^{p,new} \tag{2.32}$$

$$\mathbf{u}^{e,trial} - \mathbf{R} \parallel \frac{\mathbf{R}}{\text{dt}} \tag{2.33}$$

$$\mathbf{u}^{e,trial} \parallel \mathbf{R} \tag{2.34}$$

2.6 Viscoelastic RFT

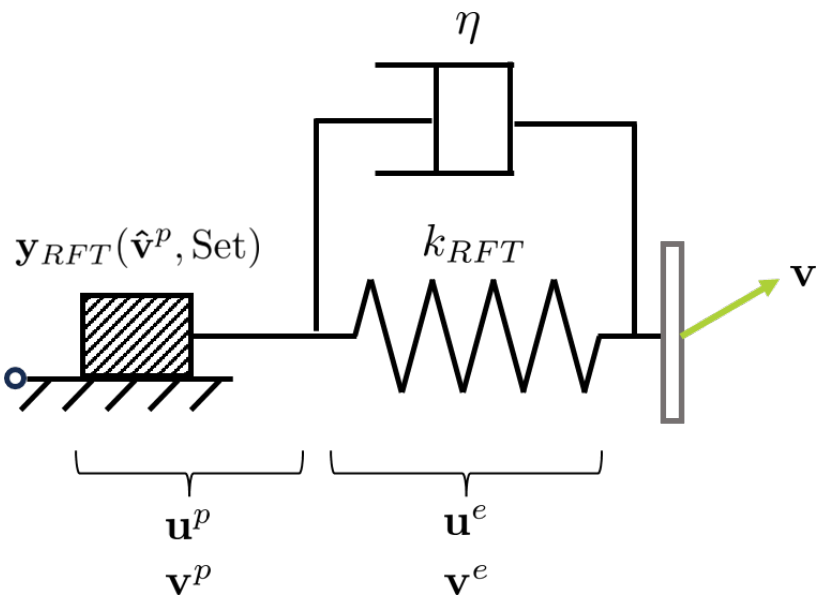


Figure 2.6: Elasto-Visco-Plastic Resistive Force Theory Analog Model

The framework for viscoelastic RFT is similar to elastic RFT. To account for the viscous dissipation and slight rate dependent response in the elastic response of grains, we introduce some damping into the model. In the analog model we represent this as a damper parallel to only the elastic spring portion of the model. While a large amount of visco-elasto-plastic models for granular media are similar to a Bingham plasticity model[28], it is essential for the viscosity introduced to RFT to not affect the slider or equivalently the plastic response since resistive force theory (unless dynamic-RFT is used [21]) applies at low velocities, where the plastic response of noncohesive grains are effectively rate-independent [12]. Besides the physical slight rate dependent elastic behavior of grains [34], viscoelastic RFT will also reduce some of the oscillatory behavior of the intruder. Once we have established the manner in which viscosity should be introduced into the model, we can follow the same steps as before to define the elastic traction. In this section, we will refer to the traction over the section

of the analog model containing the viscous damper and elastic spring as the viscoelastic traction. In this portion of the model, we can see that by having them parallel to each other the viscoelastic traction [equation 2.36] is the linear superposition of the elastic traction [equaiton 2.3 or 2.5] and viscous traction [equation 2.35]. Both linear and nonlinear elastic RFT can be used in the viscoelastic RFT formulation. For the rest of this section, we will use the linear non-codirectional elastic RFT.

$$\mathbf{t}^{viscous} = \eta\,\dot{\mathbf{u}}^{e} = \eta\,\mathbf{v}^{e} \tag{2.35}$$

$$\mathbf{t}^{viscoelastic} = \mathbf{t}^{elastic} + \eta\,\mathbf{v}^{e} \tag{2.36}$$

To numerically implement this we follow the same trial-return algorithm as in the elastic RFT solvers.

Algorithm 3 Algorithm for evaluating the flow rule numerically for viscoelastic RFT

knowing $\mathbf{t}, \mathbf{u}, \mathbf{v}^{new}, \mathbf{v}^{e}, \mathbf{v}^{p}$ at timestep t
take an elastic trial step ▷ figure 2.5a
set the new elastic velocity to the new velocity
$\mathbf{v}^{e,new} = \mathbf{v}^{new}$
$\mathbf{t}^{trial} = \mathbf{t} + k_{RFT}\,\mathbf{v}^{new}\,\mathrm{dt} + \eta\mathbf{v}^{new}$
$\mathbf{t}^{e}_{trial} = \mathbf{t} + k_{RFT}\,\mathbf{v}dt$
$\mathbf{t}^{v}_{trial} = \eta\mathbf{v}$
if $\mathbf{t}^{trial}$ is inside or on the locus **then** ▷ figure 2.5b
    $\mathbf{t}^{new} = \mathbf{t}^{trial}$
    $\mathbf{v}^{p,new} = \mathbf{v}^{p,old}$
else $\mathbf{t}^{trial}$ is outside the locus ▷ figure 2.5c
    solve for $\mathbf{v}^{p,new}$ such that ▷ figure 2.5d
    $\mathbf{v}^{new} = \mathbf{v}^{p,new} + \mathbf{v}^{e,new}$ & $\mathrm{t}^{new} = \mathbf{t}^{old} + k_{RFT}\mathbf{v}^{e,new}\,\mathrm{dt} = \mathbf{y}_{RFT}(\hat{\mathbf{v}}^{p,new}, \mathrm{Set})$
    $\tilde{k} = k + \eta/dt$
    $\mathbf{t}^{new} = \mathbf{t}_{trial} - \tilde{k}\mathbf{v}^{p,new}\mathrm{dt} = \mathbf{y}_{RFT}(\hat{\mathbf{v}}^{p,new}, \mathrm{Set})$
end if

If the trial traction is outside the locus, we can follow section 2.5.1 by decomposing the plastic velocity to obtain 2 different equation to solve for the plastic velocity direction. The trial traction is then corrected a new timestep can be taken following the same algorithm.

Chapter 3

Nonlinear Inextensible Beam Solver for Root Models

To model the motion of the roots as they are being pulled out of granular solids, we choose beam elements due to the roots' slender nature. Since we anticipated large deformations of the roots as they move through the grains, we diverge from the commonplace linear Euler-Bernoulli beam theory and make use of nonlinear beam dynamics as well as introduce inextensibility in the roots. To satisfy the inextensibility kinematic constraint, we introduce the tension within the beams/roots as a Lagrange multiplier to satisfy this constraint. One of the benefits of modeling the roots as inextensible beams is that monitoring the tension inside the roots allows us to evaluate whether the roots would break during the uprooting process. Root structures mainly fail by either root breakage or root pullout [35]. Root pullout is addressed by using resistive force theory, while the tension calculated inside the root structures can be compared to the plant's axial strength to consider breakage. The second beneift of modeling the roots as inextensible beams is that from a scaling argument, the inextensible bending process is governed by a time step of the order $1/E$ instead of $1/EI$ if we were to solve for stretch of the beams. This way, the numerical solver is able to take larger stable timesteps and produce results quicker.

While the assumption of inextensibility of roots was initially made for the computational time savings, it is well justified when considering that most green wood (wood that has at least 50% moisture) have Young's moduli of the order of gigapascals [36]. Typically tree roots, branches, and wood in general are considered to be orthotropic materials meaning that they have different elastic properties in 3 orthogonal directions [37, 38]. Wagner et al.'s inspection of the elastic properties of a root in three orthogonal directions, longitudinal, radial, and tangential, demonstrate that the longitudinal stiffness of green wood is much larger than the other two, suggesting that an inextensibility assumption of roots along their length is realistic [39].

3.1 Governing Equations for Nonlinear Beams

From the position of the points along the beam, we can define tangent and normal vectors along the beam. These can then be used to calculate the curvature as a function of the axial

coordinate s 3.1

$$K = \frac{d\hat{\mathbf{t}}}{ds} \cdot \hat{\mathbf{n}} \tag{3.1}$$

For the constitutive relationship between the moment and curvature, we use the linear elastic relationship [40].

$$M = EI\,K \tag{3.2}$$

The equations of motion governing a nonlinear beam with linear mass λ subject to bending moments M, tension T, shear forces V, and external loading f can be derived by examining the free body diagram in figure 3.1a

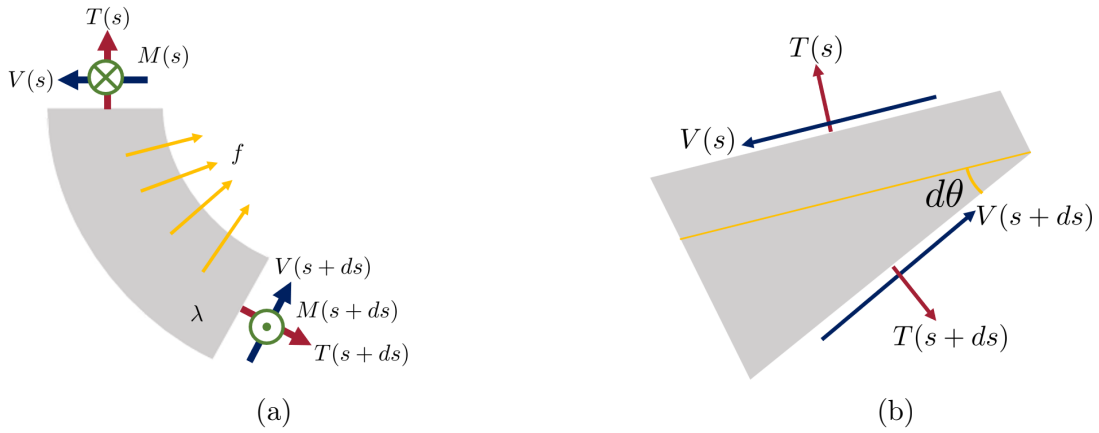


Figure 3.1: (a) Free Body Diagram for Deriving Governing Equations. (b) Detailed Image for $d\theta$

Balance of angular momentum at axial coordinate s returns:

$$\Sigma M_s = -M(s) + M(s+ds) + V(s+ds)ds = 0 \tag{3.3}$$

Since we want the beam model to be able to capture large deformations, we define our balance of linear momentum equations in local tangential and normal directions instead of a global coordinate system. The balance of linear momentum in these two directions returns:

$$\Sigma F_t = -T(s) + T(s+ds) + N(s+ds)\sin(d\theta) + f_t ds = \lambda ds \ddot{\mathbf{x}} \cdot \hat{\mathbf{t}} \tag{3.4}$$

$$\Sigma F_n = -V(s) + V(s+ds) + T(s+ds)\sin(d\theta) + f_n ds = \lambda ds \ddot{\mathbf{x}} \cdot \hat{\mathbf{n}} \tag{3.5}$$

Rearranging these three we get the three governing equations of motion:

1. Tangential direction equation of motion:

$$\frac{dT}{ds} + VK + f_t = \lambda \dot{\mathbf{v}} \cdot \mathbf{t} \tag{3.6}$$

2. Normal direction equation of motion:

$$\frac{dV}{ds} + TK + f_n = \lambda \dot{\mathbf{v}} \cdot \mathbf{n} \tag{3.7}$$

3. Balance of moments, relating shea forces to moments:

$$V = -\frac{dM}{ds} \tag{3.8}$$

These 3 equations govern the motion of the beam elements. These are similar to linear Euler-Bernoulli beam governing equations. We can see that for small deformations (which is the regime in which linear beam theory applies) the curvature can be approximated to be zero dropping the nonlinear terms in both equations 3.6 and 3.7. At low deformations, we can neglect the extension of the beam and thus any tension developing in response, dropping out the tension T. Then we recover linear Euler-Bernoulli theory.

3.2 Inextensibility Constraint

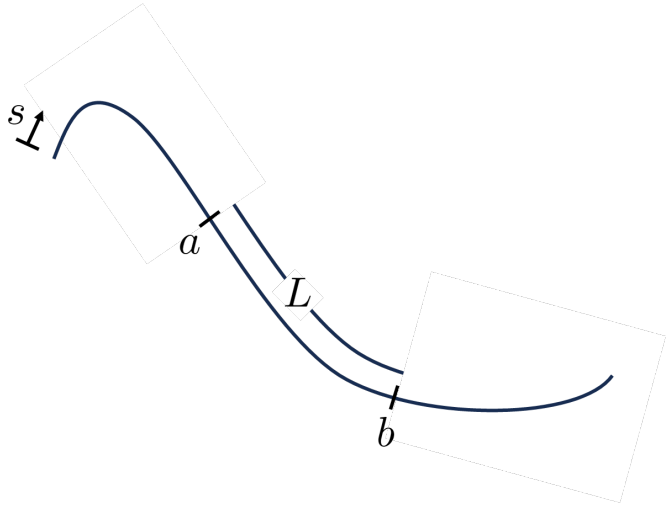


Figure 3.2: Sample parametric 2-dimensional curve for the inextensibility constraint derivation

This section derives the governing equations to satisfy the inextensibility constraint applied to the beam elements modeling the roots. The length L of a 2-dimensional parametric curve defined by $x(s)$ and $y(s)$ such as the one in figure 3.2 in the range $s \in [a, b]$ can be calculated using the integral in equation 3.9.

$$L = \int_a^b ds = \int_a^b \sqrt{\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2}\, ds \tag{3.9}$$

Since we want the length of a section of the beam to not change during its motion, we set its time rate of change to zero.

$$\frac{dL}{dt} = \frac{d}{dt}\int_a^b \sqrt{\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2}\, ds = 0 \tag{3.10}$$

Using the Leibniz integral rule we can take the derivative inside the integral [equation 3.11].

$$0 = \int_a^b \frac{2\left(\frac{dx}{ds}\frac{d}{dt}\left(\frac{dx}{ds}\right) + \frac{dy}{ds}\frac{d}{dt}\left(\frac{dy}{ds}\right)\right)}{2\sqrt{\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2}} ds \tag{3.11}$$

Since time t and length parameter s have no dependency we can flip the order of differentiation.

$$0 = \int_a^b \frac{\frac{dx}{ds}\frac{d}{ds}\left(\frac{dx}{dt}\right) + \frac{dy}{ds}\frac{d}{ds}\left(\frac{dy}{dt}\right)}{\sqrt{\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2}} \tag{3.12}$$

The (not-normalized) tangent vector on a parametric curve can be written as in equation 3.13.

$$\mathbf{t} = \left\langle \frac{dx}{ds}, \frac{dy}{ds} \right\rangle \tag{3.13}$$

The velocity vector in terms of time derivatives can be written as:

$$\mathbf{v} = \left\langle \frac{dx}{dt}, \frac{dy}{dt} \right\rangle \tag{3.14}$$

The expression in the numerator of the integral in equation 3.12 can be expressed as the dot product of the derivative of the velocity vector v with respect to the parameter s with the tangent vector t. Equation 3.12 can be re-written as:

$$0 = \int_a^b \frac{\mathbf{t} \cdot \frac{d\mathbf{v}}{ds}}{\sqrt{\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2}} \tag{3.15}$$

Since the denominator in equation 3.15 cannot be zero, to satisfy the inextensibility constraint, the numerator must equal zero, giving us the final form of the inextensibility constraint in equation 3.16. The dot product of the tangent vector at a point and the derivative of velocity along the curve coordinate s must equal zero to satisfy inextensibility.

$$\mathbf{t} \cdot \frac{d\mathbf{v}}{ds} = 0 \tag{3.16}$$

3.3 The Projection Method

The projection method is a numerical method primarily developed and popularized by Chorin to reduce the difficulties of numerically solving the incompressible Navier-Stokes Equation for fluids [41]. In this approach, the pressure and velocity fields which are coupled through the incompressibility constraint. To mediate this, Chorin decouples the two fields and first calculates a mid-step star velocity that does not necessarily satisfy incompressibility[42]. Since the the calculation of the star velocity splits the time-step into fractions, the projection method falls under the broader umbrella of fractional step methods [41]. After the star step,

the pressure is calculated via the incompressibility constraint. Once the incompressibility-satisfying pressures are known, the velocity is updated from the mid-step star values to the new velocities at the new time step.[43]

We make use of the projection method to separate the calculations of velocities and tensions for the beams representing the roots. At the start of a time-step, from the positions of the nodes we calculate the curvature 3.1. Once the curvature is known, the moments are defined following 3.2. Then the shear forces along the beam are defined from the derivative of the moments as 3.8. Then, knowing the old velocities, external line loads (due to resistive forces, gravity etc.), curvatures, and shear forces we can take a semi-explicit half step following equations 3.17, 3.18 to define mid-step star velocities, as is typical for the projection method.

$$v_t^* = (VK + f_t)\frac{dt}{\lambda} + v_t^{old} \tag{3.17}$$

$$v_n^* = (\frac{dV}{ds} + f_n)\frac{dt}{\lambda} + v_n^{old} \tag{3.18}$$

Using these star velocities, we solve the inextensibility constraint for the tensions. This is detailed in the following section. Once the tensions are known, the new velocities satisfying inextensibility are calculated using 3.19 & 3.20.

$$v_t^{new} = \left(\frac{dT}{ds}\right)\frac{dt}{\lambda} + v_t^* \tag{3.19}$$

$$v_n^{new} = TK\frac{dt}{\lambda} + v_n^* \tag{3.20}$$

Thus we are able to calculate the tensions and velocities in separate steps to satisfy the inextensibility constraint by making use of the projection method.

3.4 Calculations for Projection-Correction: Tensions Satisfying Inextensibility

In order to successfully implement the projection method, we need an expression of the inextensibility constraint and governing equations in terms of tension T. To do so we first take the derivative of equation 3.19 with respect to s and rearrange:

$$\frac{d^2T}{ds^2} = \frac{\lambda}{dt}\left[\frac{d}{ds}\left(\mathbf{v}^{new}\cdot\mathbf{t} - \mathbf{v}^*\cdot\mathbf{t}\right)\right] \tag{3.21}$$

By applying the chain rule to the first term we obtain equation 3.22.

$$\frac{d^2T}{ds^2} = \frac{\lambda}{dt}\left[\frac{d\mathbf{v}^{new}}{ds}\cdot\mathbf{t} + \mathbf{v}^{new}\cdot\frac{d\mathbf{t}}{ds} - \frac{dv_t^*}{ds}\right] \tag{3.22}$$

Due to inextensibility [equation 3.16] the first term on the right hand side of equation 3.22is zero. We can also write the second term as a product of components rather than a dot

product. In equation 3.23 and the following, the subscripts t and n denote the components of the derivative of the tangent of the beam in the tangential and normal directions, respectively.

$$\frac{d^2T}{ds^2} = \frac{\lambda}{dt}\left[v_t^{new}\left(\frac{d\mathbf{t}}{ds}\right)_t + v_n^{new}\left(\frac{d\mathbf{t}}{ds}\right)_n - \frac{dv_t^*}{ds}\right] \tag{3.23}$$

Since we want to solve for the tension before we know the new tangential and normal velocities, we replace v_t^{new} and v_n^{new} with their expressions in equations 3.19 and 3.20 to substitute the star velocities and other quantities known before tension is calculated.

$$\frac{d^2T}{ds^2} = \frac{\lambda}{dt}\left[\left(\left(\frac{dT}{ds}\right)\frac{dt}{\lambda} + v_t^*\right)\left(\frac{d\mathbf{t}}{ds}\right)_t + \left(TK\frac{dt}{\lambda} + v_n^*\right)\left(\frac{d\mathbf{t}}{ds}\right)_n - \frac{dv_t^*}{ds}\right] \tag{3.24}$$

Rearranging we get a second order ordinary differential equation for T (equation 3.25) which can easily be numerically solved.

$$\frac{d^2T}{ds^2} - \frac{dT}{ds}\left(\frac{d\mathbf{t}}{ds}\right)_t - TK\left(\frac{d\mathbf{t}}{ds}\right)_n = \frac{\lambda}{dt}\left[v_t^*\left(\frac{d\mathbf{t}}{ds}\right)_t + v_n^*\left(\frac{d\mathbf{t}}{ds}\right)_n - \frac{dv_t^*}{ds}\right] \tag{3.25}$$

Combining equation 3.25 for T, equation 3.19 for v_t^{new} and equation 3.20 for v_n^{new} along with the boundary conditions, we can solve for the unknowns at all nodes.

3.5 Discrete Numerical Beams

So far, the derivation of the inextensible nonlinear beam theory has been a discussion based on a continuous beam with axial coordinate s. To numerically solve for the motion of these beams, we discretize them into nodes and elements.

The black squares denote the element centers while the orange circles denote the nodes. There are ghost elements at the free ends of the beams to help calculate certain derivatives at the nodes. The positions, velocities, curvatures, moments, and line forces are stored in and applied to the nodes. The tension T, shear force V are stored at the element centers. The ghost nodes have 0 shear force and typically have 0 tension (can be modified for force-controlled boundary conditions). Since the positions and velocities are stored at the nodes, the governing equations are solved for the nodes. Storing tensions and shear forces in the element centers allows for the derivative terms dT/ds and dV/ds to be centered at the nodes. For the tension and shear force terms that appear directly (paired with curvature K in the governing equations) we average the elemental values to the node centers. The existence of ghost elements makes the calculation of averages and centered finite difference derivatives more straightforward.

The tangents are first calculated for the elements from the positions of their 2 neighboring nodes. The normals are then defined for the elements via a 90 degree rotation of the tangent clockwise. Once the element tangent and normal vectors are defined, they are averaged onto the nodes as nodal normal and tangential vectors.

3.6 Is the discrete inextensibility the same as the continuous?

The inextensibility equation 3.16 is derived for the continuous beam. It should hold for the discretized beam as well, and the numerical solution should approach the analytical as the discretization gets finer and finer. However a key issue/difference for the continuous and discrete implementation of the inextensibility equation is that in the derivation of the constraint, both terms in equation 3.16, the tangent vector and the velocity derivative, are located at the same position on the beam. However, for the discrete beam, the tangents are calculated at the element centers and averaged to the node centers. This section goes over the procedure we follow to ensure that both terms for the inextensibility constraint are in consistent positions and the discrete and continuous results match.

The equivalent of equation 3.16 in a discrete element with 2 surrounding nodes can be visualized as the following:

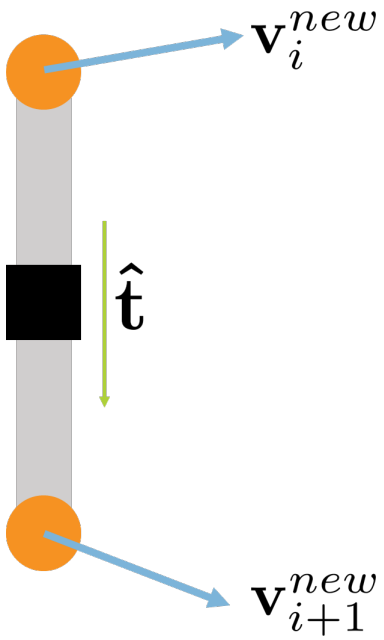


Figure 3.3: One discrete beam element with velocities visualized at the surrounding nodes

Rewriting the inextensibility equation 3.16 as a refresher:

$$\mathbf{t} \cdot \frac{d\mathbf{v}}{ds} = 0$$

The $\hat{\mathbf{t}}$ in the continuous equation is the green element tangent vector $\hat{\mathbf{t}}$ in figure 3.3. The velocity to satisfy this equation is the new velocities defined by equations 3.19 and 3.20. We can express the velocities at the nodes as:

$$\mathbf{v}_i^{new} = v_{n,i}^{new}\hat{\mathbf{n}}_i + v_{t,i}^{new}\hat{\mathbf{t}}_i \tag{3.26}$$

$$\mathbf{v}_{i+1}^{new} = v_{n,i+1}^{new}\hat{\mathbf{n}}_{i+1} + v_{t,i+1}^{new}\hat{\mathbf{t}}_{i+1} \tag{3.27}$$

Where $\hat{\mathbf{n}}_i$ is the normal vector at the i^{th} node. We can now write the derivative term in equation 3.16 which becomes:

$$\hat{\mathbf{t}} \cdot \frac{d\mathbf{v}}{ds} = \hat{\mathbf{t}} \cdot \frac{1}{ds}\left(v^{new}_{n,i+1}\hat{\mathbf{n}}_{i+1} + v^{new}_{t,i+1}\hat{\mathbf{t}}_{i+1} - v^{new}_{n,i}\hat{\mathbf{n}}_i - v^{new}_{t,i}\hat{\mathbf{t}}_i\right) = 0 \tag{3.28}$$

There are 4 separate dot products in this equation. Since the tangential and normal vectors are averaged to the nodes and $\hat{\mathbf{t}}$ is defined for the element none of them cancel out. Now we can replace the v^{new} components for the nodes with equations 3.20 and 3.19 to incorporate T. One detail here is how we average T to the nodes to be able to calculate equation 3.20 and how we take the 1st order difference in equation 3.19 will change the final form. For this explanation we will follow equation 3.29 which averages the elemental tension values to the node. We follow equation 3.30 which is a first order central difference approximation to calculate the first derivative of the tension at the nodes. In both of these equations, the right hand side indices i and $i-1$ refer to element indices.

$$T_{averaged}\Big|_i = \frac{T_i + T_{i-1}}{2} \tag{3.29}$$

$$\left.\frac{dT}{ds}\right|_i = \frac{T_i - T_{i-1}}{ds} \tag{3.30}$$

The normal and tangential velocities at the i^{th} node can be expressed as equations **??** once the previously calculated average tension values and derivatives are subsituted in.

$$v^{new}_{n,i} = \frac{T_i + T_{i-1}}{2} * K_i * \frac{dt}{\lambda} + v^*_{n,i} \tag{3.31}$$

$$v^{new}_{t,i} = \frac{T_i - T_{i-1}}{ds}\frac{dt}{\lambda} + v^*_{t,i} \tag{3.32}$$

Plugging these expressions and the new velocity component expressions into the discrete inextensibility equation 3.28 and cancelling out $1/ds$ we obtain equation 3.33.

$$\begin{aligned}\left(\frac{T_{i+1} + T_i}{2}K_{i+1}\frac{dt}{\lambda} + v^*_{n,i+1}\right)\mathbf{n}_{i+1} \cdot \hat{\mathbf{t}} + \left(\frac{T_{i+1} - T_i}{ds}\frac{dt}{\lambda} + v^*_{t,i+1}\right)\mathbf{t}_{i+1} \cdot \hat{\mathbf{t}} \\ - \left(\frac{T_i + T_{i-1}}{2}K_i\frac{dt}{\lambda} + v^*_{n,i}\right)\mathbf{n}_i \cdot \hat{\mathbf{t}} - \left(\frac{T_i - T_{i-1}}{ds}\frac{dt}{\lambda} + v^*_{t,i}\right)\mathbf{t}_i \cdot \hat{\mathbf{t}} = 0\end{aligned} \tag{3.33}$$

This equation evaluated for each element is going to form the system of equations governing the beam structure as the solver steps through discrete timesteps. In the next few steps we will isolate the coefficients for each of the terms so entries into the matrix representing this linear system of equations can be reviewed. We place the known parameters, the star velocities, at the right hand side equation and factor out the coefficients for the elemental tension forces.

For the left hand side of the equation becomes:

$$T_{i+1}\left[\frac{1}{2}K_{i+1}\frac{dt}{\lambda}\hat{\mathbf{t}}\cdot\mathbf{n}_{i+1}+\frac{1}{ds}\frac{dt}{\lambda}\hat{\mathbf{t}}\cdot\mathbf{t}_{i+1}\right]$$
$$T_{i}\left[\frac{1}{2}K_{i+1}\frac{dt}{\lambda}\hat{\mathbf{t}}\cdot\mathbf{n}_{i+1}-\frac{1}{ds}\frac{dt}{\lambda}\hat{\mathbf{t}}\cdot\mathbf{t}_{i+1}-\frac{1}{2}K_{i}\frac{dt}{\lambda}\hat{\mathbf{t}}\cdot\mathbf{n}_{i}-\frac{1}{ds}\frac{dt}{\lambda}\hat{\mathbf{t}}\cdot\mathbf{t}_{i}\right]$$
$$T_{i-1}\left[\frac{-1}{2}K_{i}\frac{dt}{\lambda}\hat{\mathbf{t}}\cdot\mathbf{n}_{i}+\frac{1}{ds}\frac{dt}{\lambda}\hat{\mathbf{t}}\cdot\mathbf{t}_{i}\right]$$

On the right hand side, we gather the known or previously evaluated variables:

$$-(v^{*}_{n,i+1})\hat{\mathbf{t}}\cdot\mathbf{n}_{i+1}-(v^{*}_{t,i+1})\hat{\mathbf{t}}\cdot\mathbf{t}_{i+1}+(v^{*}_{n,i})\hat{\mathbf{t}}\cdot\mathbf{n}_{i}+(v^{*}_{t,i})\hat{\mathbf{t}}\cdot\mathbf{t}_{i}$$

Using these values and equations 3.20 and 3.19 we construct the matrix representing the governing equations for the beam structure. Once this system is solved, we obtain the updated velocities which satisfy the inextensibility equation.

3.7 Curvature Rate Based Damping

The inextensible beams are able to capture the overall elastic response of tree roots. However, in reality the plant material also displays some time-dependent behavior [44, 45]. To mimic this effect as well as to damp out the oscillations of the roots in our simulations, we introduce a curvature-rate-based damping into our system of equations. This approach helps mimic the resistance of a root to being bent and the damping increases the quicker and more intense the change in shape occurs. We do so by augmenting the moment response in the beam in response to the curvature change rate.

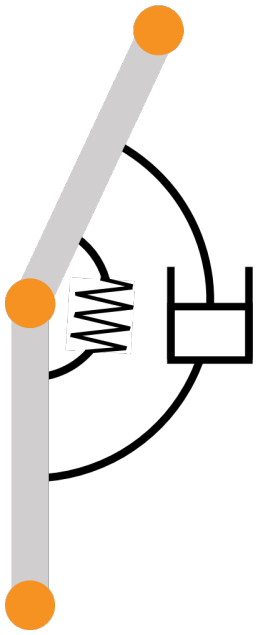


Figure 3.4: Analog model visualization of viscoelastic beam elements

We modify equation 3.2 as the following to account for the curvature rate based damping, where η is the damping coefficient.

$$M = EI\,K + \eta\dot{K} \tag{3.34}$$

Evaluating the time rate of change of curvature numerically requires us to store the past values of curvature. To avoid this and work with only the present values of variables we follow the steps outlined below. We can plug in the definition of the curvature along the beam as stated in equation 3.1.

$$\dot{K} = \frac{d}{dt}\left(\frac{d\hat{\mathbf{t}}}{ds}\cdot\hat{\mathbf{n}}\right) \tag{3.35}$$

When we apply the chain rule we obtain:

$$\left(\frac{d}{dt}\frac{d^2\mathbf{x}}{ds^2}\right)\cdot\hat{\mathbf{n}} + \frac{d^2\mathbf{x}}{ds^2}\cdot\frac{d\hat{\mathbf{n}}}{dt} \tag{3.36}$$

The first term scales with the rate of change of the curvature while the second term scales with the present curvature magnitude. If we rearrange the the order of the derivatives in the first term we get the second derivative of the normal velocity (equation 3.37). This can be evaluated in a single time step and is able to capture the resistance of the beam. This term is used in place of $\dot{K}$ in our beam code due to its computational ease. The second term can be accounted for by storing the face normal vectors between timesteps, however it has not been necessary to capture a realistic damping response thus far. This second term can be rearranged into the form in equation 3.37, which can be evaluated easily for all nodes by simply using the normal velocity values at the current timestep.

$$\frac{d^2v_n}{ds^2} \tag{3.37}$$

3.8 Multi-Branch Equations

To model elementary 2D root structures, we combine multiple beams. Besides each branch's individual governing equations, we introduce 2 new kinematic constraints at the intersection of any 2 beams/branches.

1. The nodes at the connection interface need to have the same velocity and position at all timesteps

2. The initial angle between the connecting branches does not change.

The second kinematic constraint comes from the assumption that junctions of roots are much stronger than the rest of the roots.

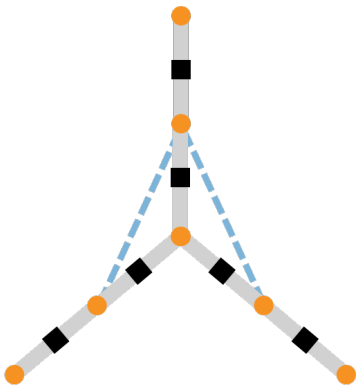


Figure 3.5: Inverted Y shaped branch made up of 3 combined beams. At the center orange node, kinematic constraint 1 applies. The angular constraint is achieved via blue massless rigid links

With all the equations governing the multi-branch structure, we assemble them into a linear system of equations which is solved at each time step as we iterate through time in our simulations. The linear system of equations is visualized in figure 3.6 where the unknowns are listed as $\mathbf{T}$ for tension, $\mathbf{f}_{top}$ for the force required at the top node to apply a velocity constraint, $\mathbf{f}_{rigid}$ for the forces in the rigid links satisfying the angular constraint between branches, and $\mathbf{f}_{conn}$ for connection forces at the center node between connecting branches ensuring they move together.

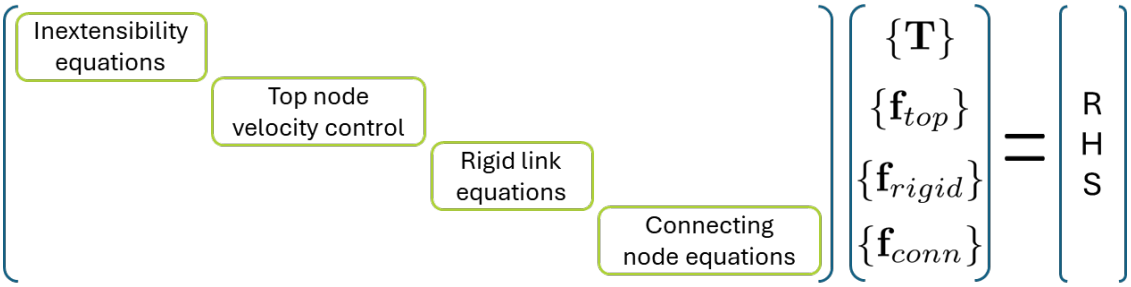


Figure 3.6: Final form of the linear system of equations for a multi-branch root structure. Top node velocity controls are emitted for force controlled uprooting scenarios.

3.9 Boundary Conditions & Implementation

We have considered 3 types of boundary condition options for the single beam or combined root structures.

1. top node imposed velocity boundary condition
2. top element imposed tension boundary condition
3. top node angular velocity constraint paired with curvature constraint for the first 2 nodes.

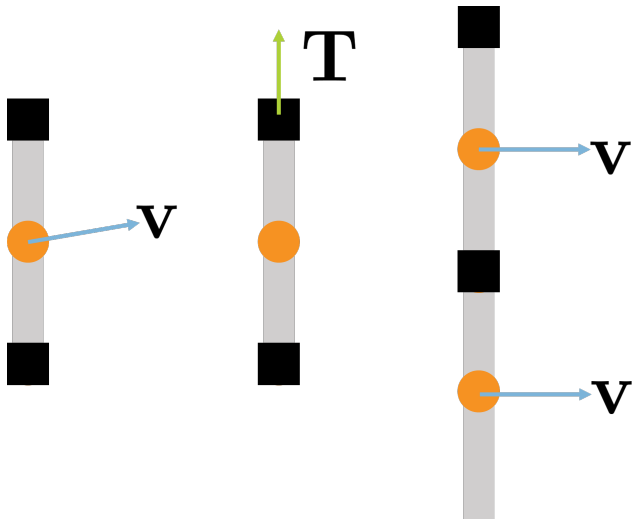


Figure 3.7: Three boundary condition options visually represented. Velocity boundary condition at the top node (left), tension boundary condition at the top element (center), angular velocity and curvature constraint (right)

Velocity Boundary Condition

One scenario we were interested in modeling was a root structure being pulled with constant or controlled velocity from the top. To mimic this, we make use of an imposed velocity boundary condition at the top node (figure 3.7-left). The boundary conditions for the top node are obtained by imposing the desired velocity in equations 3.7 and 3.6, resulting in equations 3.38,3.39.

$$\frac{T_2 - T_1}{ds} + V_1 K_2 = \frac{\lambda}{dt}(v_{t,2}^{new} - v_{t,2}^{old}) - f_{t,2} \tag{3.38}$$

$$\frac{-1}{ds} V_1 + K_2 T_1 = \frac{\lambda}{dt}(v_{n,2}^{new} - v_{n,2}^{old}) - \frac{V_2}{ds} - f_{n,2} \tag{3.39}$$

Solving this set of equations gives us the tension T at all the elements and the shear force V_1 within the first element.

Tension Boundary Condition

The simplest boundary condition to enforce is the assigned tensile force applied at a desired position along the structure. This force is typically placed at the top. We mimic this by applying the tension to the ghost element before the first real element. This is simply an equation of type T = assigned value which modifies the prior $T = 0$ for the top ghost element.

Velocity and Curvature Boundary Condition

To pull a single beam buried vertically in grains sideways we have made use of this boundary condition. To ensure that the top of the beam where it is dragged sideways, we assign the

same velocity to the two first nodes so they move together and the curvature at the first real element (second black box from the top in figure 3.7-right).

3.10 Summary of the Beam Solver

Algorithm 4 Algorithm for solving the beam equations numerically

knowing the positions of the nodes $\mathbf{x}(s)$, current velocities at the nodes, and the external loads at some timestep t:

calculate the tangent vectors $\hat{\mathbf{t}}$ for the elements. ▷ equation 3.13
Rotate the tangent vectors 90 degrees clockwise to define the elemental normal vectors $\hat{\mathbf{n}}$

Average the elemental tangent and normal vectors to the nodes.
calculate curvatures at the nodes $K = \hat{\mathbf{n}} \cdot \frac{\mathbf{d}\hat{\mathbf{t}}}{\mathbf{ds}}$ ▷ equation 3.1

Calculate the moments from curvatures $M = EI\,K$, ▷ equation 3.2

if damping is used **then**
    add damping term ▷ equation 3.34
end if

Calculate the shear forces at the elements ▷ equation 3.8
and the derivative of shear forces at the nodes

Take the star step with the known quantities. ▷ equations 3.17, 3.18

Assemble the linear system of equations ▷ sections 3.6, 3.8, 3.9
and solve for tension T at the elements, along with rigid link forces, connection forces, and boundary condition satisfying forces.

Perform the corrector step to obtain the velocities ▷ equations 3.20, 3.19
at the nodes at the new timestep $t + dt$

Update the positions

3.11 Complex Branch Generator & Solver

Most of this project focused on experiments and simulations of Y-shaped simple root structures. However, realistic roots contain fractal-like more complex and higher number of branches. The beam solver developed for the Y-shaped roots was written specifically for a 3-branch model, however to avoid having to customize the code for each root geometry, we

developed a complex root structure generator and solver. This was then paired with the elastic RFT solver to complete uprooting simulations.

With the complex root generator and solver, it is possible to model root branches with different lengths and number of elements. Inclusion of tapering branches or different cross-sections and thus bending stiffnesses is also possible, but not yet incorporated. To generate any root geometry of interest, the user needs to provide the complex root generator with the number of branches, the angles of those branches measured clockwise from the -y axis, the number of nodes on each branch, the lengths of each branch and the connections list. The connections list is a 2-by-N matrix, where N is the number of connections. A connection is a top to bottom linking of two branches. For each connection the user must first enter the number corresponding to the top branch and then the bottom one. With all this information, the complex branch generator creates two simple text files for the solver to read. The first, the branch data file, is a file that contains four columns. The first column has the node IDs which are unique sequential integer numbers assigned to each node to keep track. The second column is the branch ID column which shows the ID number of the branch that a node corresponds to. The third and fourth columns contain the x and y coordinates of the nodes, respectively.

The second file, the connection file, has as many row as there are total nodes and as many columns as the total number of connections. In each column, the bottom node of the top branch for that connection gets a +1 entry while the top node for the bottom branch gets a -1 entry. There can be rows with multiple +1 entries since multiple bottom branches can connect to a top branch.

The complex branch solver can take this geometry data and run the same uprooting tests as the inverted Y-shaped root structures. It follows the same governing equations. The challenge of the complex solver is handling different number of nodes for different branches and creating the matrix entries corresponding to the linear system of equations to apply the corrector step after the star step calculation for velocities.

Chapter 4

Applications of eRFT & Discussion

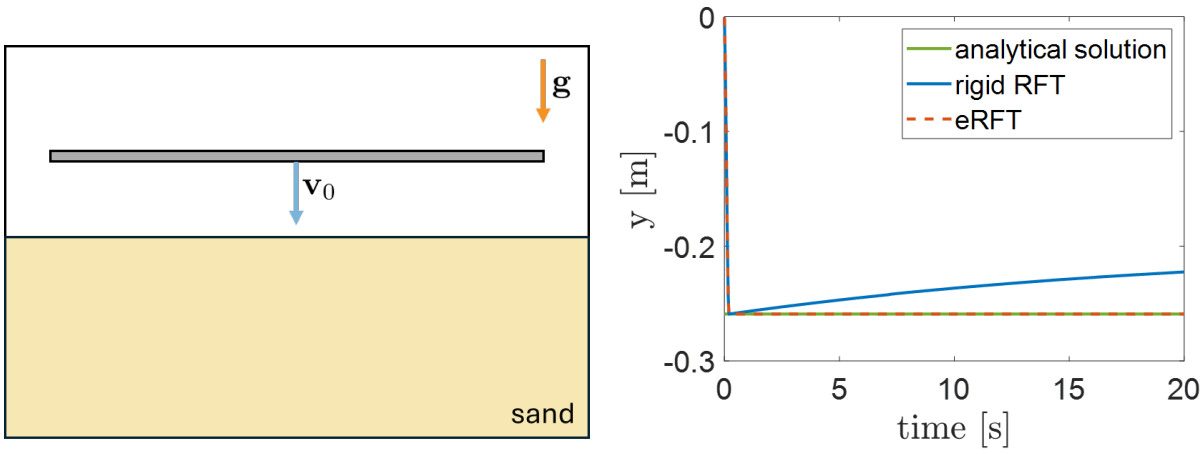
4.1 Beam Drop Test

One of the key shortcomings of rigid RFT is its inability to achieve static equilibrium in load controlled scenarios. To highlight this shortcoming and demonstrate that eRFT is able to address it, we look at the test case of dropping a horizontal beam with initial velocity into a bed of grains under its self-weight. The experimental setup can be seen in figure 4.1a. These results are compared to an analytical solution acquired from a work-energy balance (equations 4.1 4.2) in figure 4.1b. Both rigid RFT and eRFT first hit the depth corresponding to the analytical solution. However, once a balance between the resistive force and the beam's weight is achieved, the beam should have zero velocity in this static equilibrium. Without any velocity, rigid RFT is unable to provide a resistive force to hold up the beam's weight, so it begins to move. The beam keeps moving upward in the rigid RFT scenario. After the initial "dip" of the beam below the analytical depth, the beam is pushed up and accelerates. In general, the resistive forces are much larger for intrusion than extraction, thus the beam faces less resistance moving upward, which results in the rigid RFT solution slowly lifting the beam up. The eRFT solution on the other hand, remains at the analytical solution, with small oscillations if undamped eRFT is used. This example clearly highlights elastic RFT's ability to fill in the gap in intrusion processes corresponding to no velocity/load control/static equilibrium cases which cannot be captured accurately by rigid RFT.

The analytical solution for this section is derived from a work energy balance between the initial kinetic energy of the beam, change in potential energy, and the work done by resistive forces as the beam travels through the grains.

$$KE_0 = \Delta PE + W_{RFT} \tag{4.1}$$

$$\frac{1}{2}m|\mathbf{v}_0|^2 = mg(z_0 - z_{\text{final}}) + \int_{z_{\text{surface}}}^{z_{\text{final}}} |\mathbf{f}_z| \, \mathrm{dz} \tag{4.2}$$

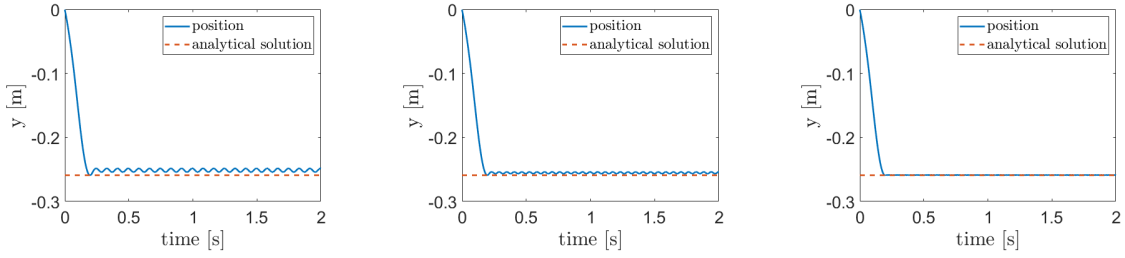


(a) (b)

Figure 4.1: (a) Setup for Beam Drop Test. (b) time-position results of the beam drop test comparing the analytical solution (green), elastic RFT solution (orange), and rigid RFT solution (blue)

4.2 Effects of Elastic RFT Stiffness

To investigate the effects of the RFT stiffness parameter k_{RFT}, we ran the same beam drop test at various values of k_{RFT}. The results (figure 4.2) show that for any value of RFT stiffness, elastic RFT solution hits the analytical depth and begins to oscillate. The frequency of the oscillations increase and the amplitude of the oscillations decrease with increasing stiffness as expected. The oscillations are effectively indiscernible to the naked eye for the stiffest solution. The solutions tend to oscillate above the analytical solution instead of centered around the analytical solution due to the fact that the resistive forces are higher during intrusion, thus a net upward bias is born from a short distance of intrusion resistive forces balancing a larger distance of extrusion forces.

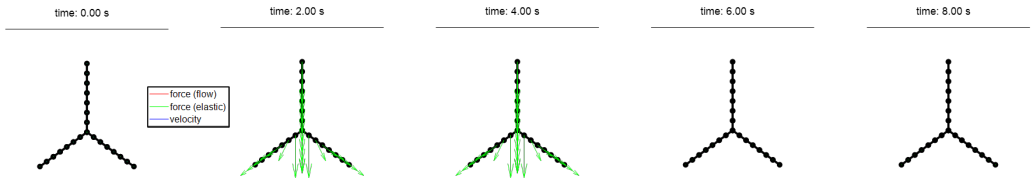


(a) (b) (c)

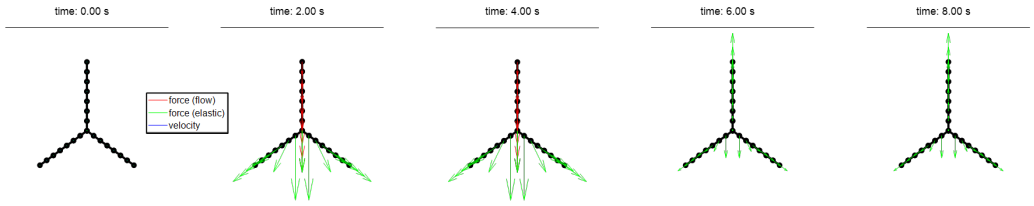
Figure 4.2: Computational beam drop experiment results for three different RFT stiffnesses. (a)k_{RFT}=500N/m (b) k_{RFT}=1kN/m (c) k_{RFT}=10kN/m

4.3 Force Controlled Uprooting: What is the threshold for uprooting?

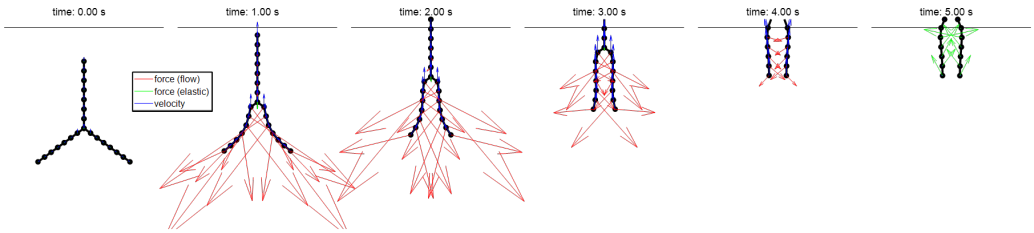
Mimicking an uprooting scenario by loading our root structures vertically from the top, we investigate the threshold force requried to uproot a root structure. We tested the response of the same root under 5N, 100N, and 4kN applied to the top of the root. All three roots were pulled for 4 seconds and then released for 4 seconds. Chronological screencaps can be seen in figure 4.3.



(a) Purely Elastic Case

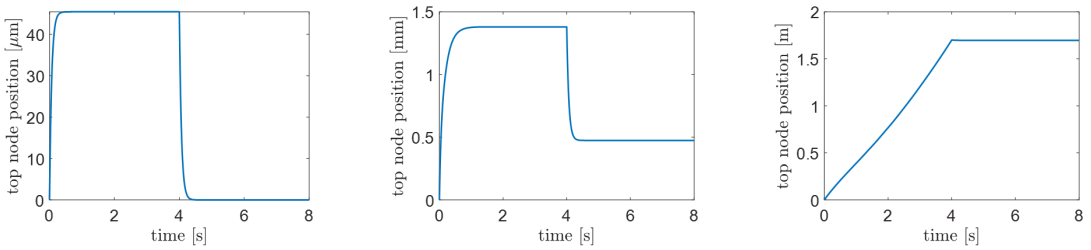


(b) Small Plasticity Case



(c) Large Plasticity Case

Figure 4.3: Response of an identical root to three different applied load magnitudes at the top node.



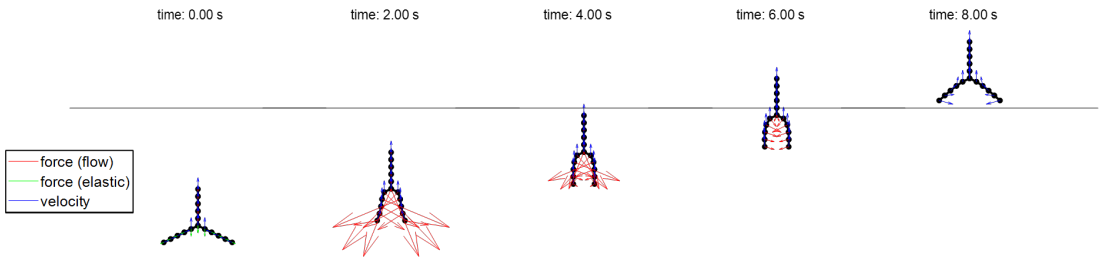
(a) Purely Elastic Case (b) Small Plasticity Case (c) Large Plasticity Case

Figure 4.4: time vs the position of the top node for the three identical roots pulled at different load magnitudes.

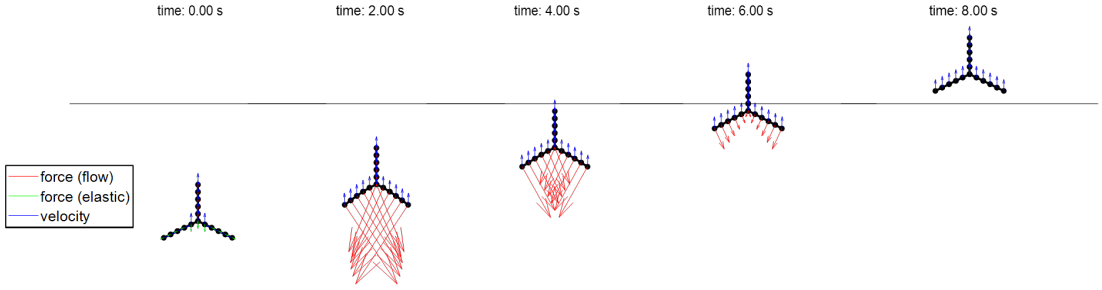
For the smallest applied load, the root structure barely deforms and eRFT is able to balance the applied load via only elastic tractions. This corresponds to none of the grains around the root flowing, but simply responding elastically. When we increase the load to 100N, we see the small plastic response. Around the vertical portion of the root red traction arrows corresponding to plastic flow of grains are present. This load is not enough to completely uproot the structure in 4 seconds however when the load is removed we can still see its effect via the residual tractions required to balance the new position of the root, similar to residual stresses in a partially plastically deformed metal after unloading due to lack of compatibility of strains in the plastic part. For loads large enough, such as 4kN for this root, the grains quickly yield resulting in red plastic traction arrows in the visualization. The roots bend significantly under the large load, and most of the root is above the surface by the end of the 4 seconds of loading.

4.4 Effects of Intruder Flexibility

Previous studies using RFT mainly focused on either rigid intruders or flexible intruders whose kinematics were assigned. This portion of the thesis reviews the effects of accounting for the flexibility of the intruder on the resistive forces calculated. The same velocity-controlled uprooting scenario was repeated for 2 roots with identical geometries but different elastic moduli. The geometric and material parameters for the large aluminum sample and the 50A rubber sample can be seen in tables 4.2 & 4.1. In figure 4.5 the difference between the response of the two samples can be seen qualitatively. The rubber sample bends a lot more in response to the resistive forces, while the aluminum sample essentially maintains its initial shape.



(a) Rubber Branch



(b) Aluminum Branch

Figure 4.5: Response of identical roots with different material properties to constant velocity pull at the top node

In figure 4.6 the force required at the top node to apply a constant velocity is shown for the two samples. Overall, since the aluminum sample bends less, it maintains a larger area that is perpendicular to its direction of motion resulting in a larger peak resistive force. The rubber root structure requires a peak load that is over 30% lower than that required for the aluminum sample. Since the rubber sample bends significantly, its ends remain in the sand for a larger total displacement. It can be seen in figure 4.6 that the aluminum sample is freed from the grains around 15 cm of displacement and no longer experiences any resistance from the grains while the rubber sample continues to experience traction from the grains until it travels 17 cm. These results highlight that both qualitatively and quantitatively, taking the flexibility of the intruder into account heavily influences the results one obtains from resistive force theory.

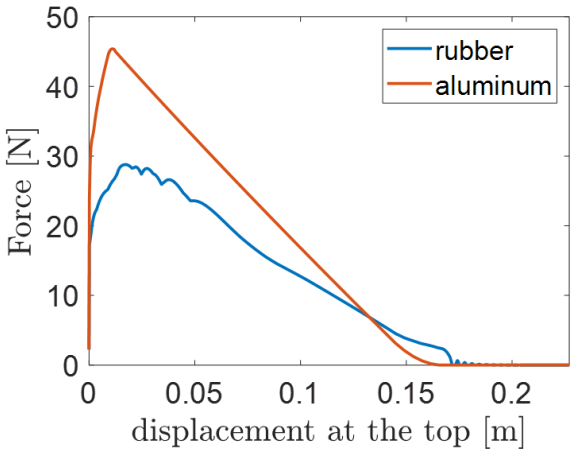


Figure 4.6: force displacement response of the rubber and aluminum Y-shaped roots

4.5 Experimental Comparison

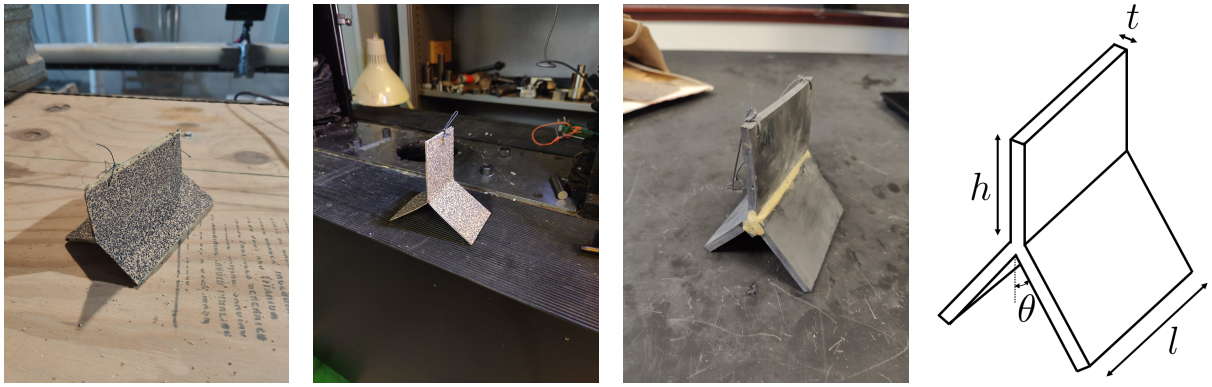
To validate our numerical elastic RFT beam solver uprooting results, 2 separate sets of experiments have been conducted. As part of the Perron group's efforts to investigate the strength and contribution of primitive roots to landscape formation, Madison Douglas, PhD, has constructed the large sandbox testing rig seen in image 4.7a. In these experiments, the sample is pulled out of the sandbox with a fishing line that is tied to a constant torque drill that is operated manually. The force required for the uprooting is captured via a load cell in series with the fishing line used in the pulley uprooting mechanism. The initial results from these experimental and computational results showed qualitative agreement, however we struggled to achieve better quantitative matching.

To try out another method of running similar uprooting tests, a second set of experiments were conducted by the author using a 5 gallon bucket as the sand bed (figure 4.7b). In these sets of experiments, the bucket was placed in an INSTRON machine to control the velocity boundary condition and measure the total resistive force required to impose said motion. This setup allowed for more precise control when imposing the velocity boundary condition at the top of the root structure.

Between the two different setups, 4 inverted Y shaped root structures have been manufactured and tested. The geometric and material parameters defining these samples can be found in tables 4.1 & 4.2. The elastic modulus for the rubber samples are converted from their shore A hardness scale following the empirical relationship in [46]. The aluminum samples manufactured using Amerimax flashing shingles were covered in 60 grit sandpaper so that the surface friction between the root structure and the grains would be similar to the internal friction of the grains. The rubber samples were sanded with a coarse grit sander to remove their smooth finish.

(a) (b)

Figure 4.7: Two experimental setups used.(a) Large sand bed with drill-controlled pulley mechanism, (b) INSTRON controlled bucket



(a) (b) (c) (d)

Figure 4.8: Experimental inverted Y-shaped root structures. (a) large aluminum (b) small aluminum (c) rubber 50A (d) drawing for dimension references

	length [mm]	height [mm]	angle [rad]	thickness [mm]
large aluminum	126	45	1.0949	3
small aluminum	60	60	1.0475	3
rubber 50 A	95	70	0.8029	7

Table 4.1: Table of experimental branch structure geometric specifications

	linear density $\lambda[kg/m]$	Young's Modulus
large aluminum	1.0125	0.1-1 GPa
small aluminum	.486	0.1-1 GPa
rubber 50 A	3.0429	0.82 MPa

Table 4.2: Table of experimental branch structure material parameter specifications

4.5.1 Large Sandbox Uprooting Tests

2 sets of roots and their corresponding calibration tests have been completed in this setup. An aluminum sample coated in 60 grit sandpaper and a shore 50 A rubber sample, have been manufactured into the inverted Y root shapes. Flat plates made of these materials have been uprooted and the resulting forces have been used to calibrate the $\boldsymbol{\alpha}$ functions as mentioned in the introduction section.

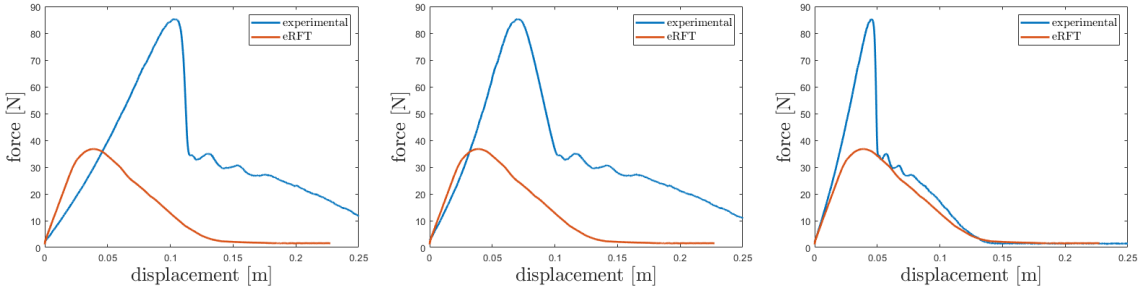


Figure 4.9: Comparison of experimental and computational force-displacement results for the large aluminum root structure. (a) raw force-time data converted to force-displacement assuming constant velocity (b) results from (a) adjusted by the fishing line extension (c) data from (a) adjusted to match the distance covered under the surface

After the calibration, the initial comparison between the numerical and experimental results for the aluminum sample showed general qualitative agreement as can been seen in figure 4.9a. The initial peak in the force response for the experimental data was assumed to be due to the preparation process in which the sand is poured over the top of the sample. The RFT forces are calculated at a steady flow state, thus they will not be able to capture a more densely packed state. If desired, this can be built into the ξ parameter so it accounts for the packing density, however this would require a lot more experimental data to construct the $\boldsymbol{\alpha}$ functions. Besides the initial peak, the experimental and computational results also differed in the amount of time the sample spent under the sand surface. Assuming the experiment was conducted under constant velocity conditions via the drill, this difference was attributed to the fact that the fishing line used in the experiments was elastic. To adjust for the stretching of the fishing line, we have completed tensile testing for the material using an INSTRON machine. The fishing line showed some hardening behavior as we repeated tensile tests. The results can be seen in figure 4.10. We have used the stiffer elastic modulus which was achieved after a couple tests since the line used in the experimental setup was

reused for all repetitions of the test. The force-time-displacement data from experiments were adjusted by taking the stretching of the fishing line into account. This only created a minor difference in the results, as can be seen in figure 4.9 b.

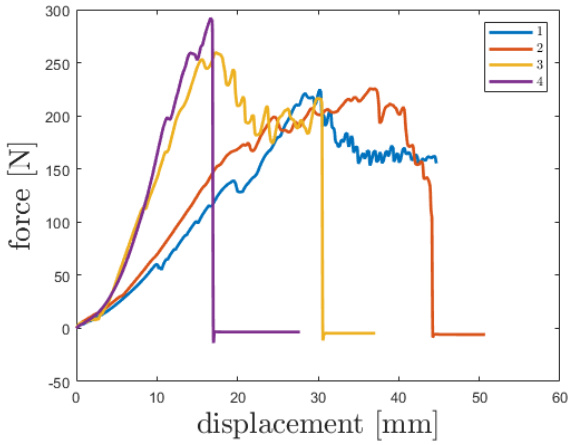


Figure 4.10: Fishing line tensile test force displacement results. Legend shows sequence of tests.

According to these experimental results, if the drill velocity was held constant, the duration that the sample experienced a resistive force times that velocity was larger than the depth of the sample. This lead us to believe that the drill used in the setup was in fact not able to provide constant velocity to the sample. To fix this mismatch in time spent under the surface level between experimental and computational results, we tried adjusting the experimental data such that the constant velocity multiplied by the time the sample experienced a force would result in the total distance the sample needs to travel to exit the sand. With this adjustment, the experimental and computational results show much better agreement (figure 4.9b) besides the initial peak due to preparation. This process strengthened our beliefs that the drill was not an adequate tool for the tests we wanted to conduct, which lead to the next set of experiments.

4.5.2 INSTRON Machine Sandbucket Uprooting Tests

To ensure that the sample is pulled at a constant rate from the top, we made use of an INSTRON tensile testing machine. Since the space available to run uprooting tests with this machine was limited, we scaled down the sandbox into a 5-gallon bucket. To ensure no edge effects are present, we scaled down the aluminum root sample such that all dimensions would be less than a third of the bucket diameter and sand height. By doing so, according to slip line theory, the flow around the samples should not experience the effects of the bucket walls [47, 48].

To run the uprooting and calibration tests with the INSTRON machine and sandbucket setup, first the test bucket is filled at least one sample height tall with sand. Then the sample is placed at the center of the bucket. The bucket is then filled with the sand from the top by pouring through a sieve. During this stage, attention must be paid to not bury the fishing

line loop that must be tied to the INSTRON machine crosshead. Once the bucket is filled to desired sand height, it is then placed into the INSTRON machine. Filling the bucket on the INSTRON machine or using alternative methods could yield results that better match the assumptions of resistive force theory, however to limit the disruption of use of the INSTRON and labspace, this was the best method at the time of these experiments. Once the bucket is in the INSTRON, the fishing line loop of the buried sample is attached to the fishing line tied to the crosshead via a carabiner. The INSTRON is then run following the program corresponding to the user's parameters.

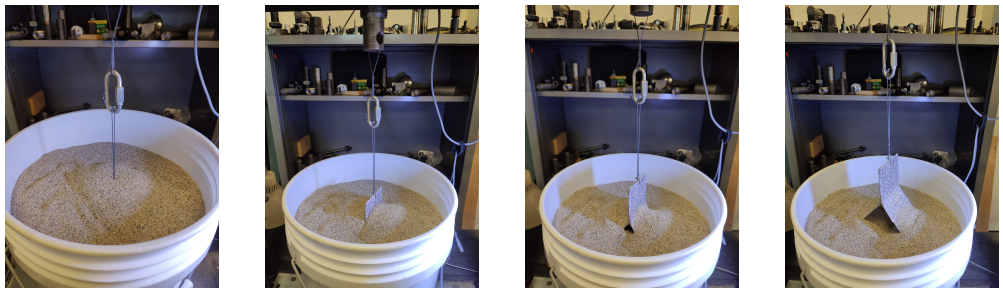
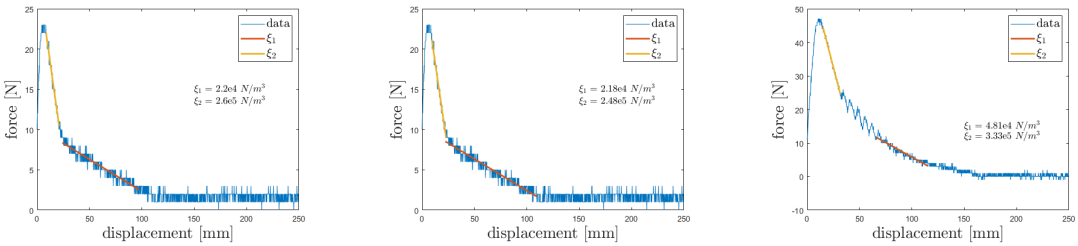


Figure 4.11: Sequential images of the bucket uprooting experiment for the small aluminum sample

To calibrate the $\boldsymbol{\alpha}$ functions via the ξ parameter, we first performed pull tests on a flat 6cm*6cm aluminum plate covered by 60 grit sand paper. The INSTRON was able to provide extremely reproducible displacement and force data. To calculate the ξ parameter, we intended to use the backslope of the force displacement data from these tests. However, as can be seen in figure 4.12, there were 2 distinct linear regions in the force-displacement data past the initial peak as we loaded the test specimens. The ξ parameters calculated off of these two slopes differ by one order of magnitude which resulted in drastically different results when used in the computational solutions.



(a) (b) (c)

Figure 4.12: Three separate flat pull calibration tests for the INSTRON-bucket setup

Previous applications and calibrations of resistive force theory either worked with large intruders at the surface where the depth scaled with the lengthscale of the part of the intruder below the surface, or with fully submerged plate elements buried deep. In this study of uprooting we are combining both where the root structures first are buried somewhat deep compared to their length, similar to the latter case and as they are being extracted from the

sand, they interact with the grains similar to the large intruders at the surface. This fact, paired with the 2 clearly distinct slopes on the calibration tests leads us to consider that perhaps the depth of the intruder with respect to its characteristic length is an important factor to take into consideration when applying RFT to intruders that traverse multiple regimes.

To explore this hypothesis, we tested a smaller flat calibration plate. The new plate, made from aluminum coated with 60 grit sandpaper as the previous samples, had a length and width of 2 cm. Thus, the same start depth of the previous calibration tests, 10 or 15 cm depending on the test, corresponded to a much larger depth relative to the lengthscale of the sample. The force-displacement plots for these tests [figure 4.13] reproduced the larger ξ parameter as measured by the higher relative depth portions of the previous calibration tests.

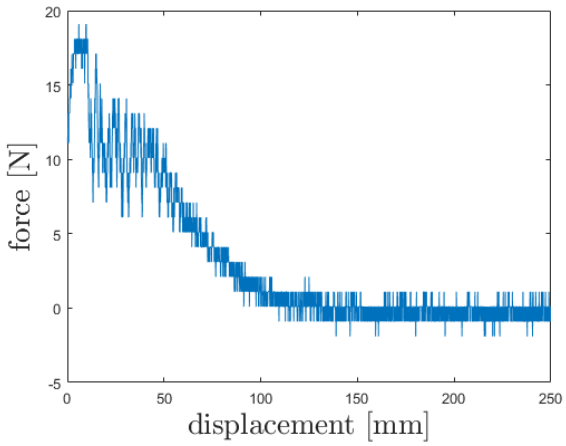


Figure 4.13: Force displacement data for a small flat plate being extracted

When deriving an analytical explanation as to why resistive force theory works well for materials governed by frictional plasticity, Askari & Kamrin made use of dimensional analysis for garden hoe typed intrusion problems [24]. Their derivation, which used the length of the gardenhoe as one of the parameters governing the resistive force experienced by the gardenhoe, did not need to consider the depth as a separate lengthscale governing the resistive forces since for a gardenhoe type intrusion, the depth scales directly with the length of the hoe that has intruded the grains. In many of the previous successful uses of resistive force theory [1, 11], similar surface intrusion of large bodies have been studied. However, motivated by the clearly different ξ parameters we see in our bucket extraction experiments, we expect that the relative depth, a depth to characteristic lengthscale ratio, should also be considered as a parameter in resistive force theory formulations.

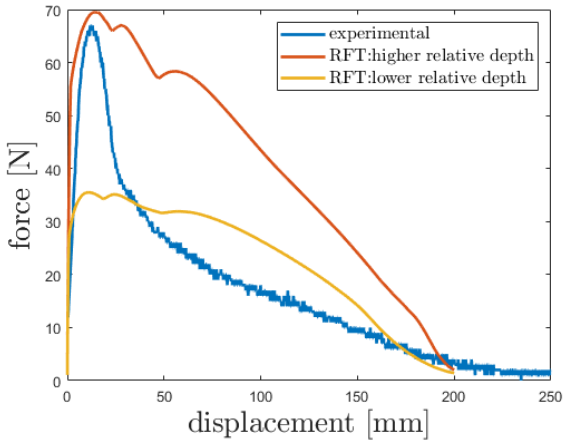


Figure 4.14: Experimental force displacement results plotted along numerical results calibrated at two different relative depths

In figure 4.14, we see the experimental force displacement data alongside two sets of computational results. The orange curve calculated using the ξ parameter at the higher relative depth (the higher ξ value) matches the initial portion of the experiments where the root structure was buried deeper. As the structure is extracted, after a few centimeters of extraction, the numerical solution deviates from the experimental. However, if we are to use a ξ parameter calculated at a lower relative depth, the yellow numerical curve is closer to the later parts of the experimental data.

4.6 ABAQUS Implementation of RFT

While the inextensible beam and elastic RFT combination is able to capture uprooting scenarios realistically as has been discussed in prior sections, the capabilities of the beam-based intruder solver we generated is limited when modeling intrusion and extraction problems. To be able to model more complex geometries, as well as account for a wider range of material models we are working on an ABAQUS user subroutine implementation of resistive force theory.

4.6.1 VDLOAD 3D Rigid RFT

In this section we go over the implementation of rigid 3D RFT in ABAQUS using the explicit user subroutine VDLOAD. For an overview of 3D rigid RFT, the reader can refer to source [26].

ABAQUS Explicit has a subroutine named VDLOAD that allows the user to define a surface pressure as a function of position, velocity, face normals and more at each surface element [49]. We have written a FORTRAN subroutine with the 3D-RFT coefficients, gravity magnitude, and surface level position built-in. The subroutine code can be found in Appendix A. The polynomial fit coefficients used in [26] are typed directly into the FORTRAN file.

To use this code on any explicit ABAQUS simulation of interest, the reader can follow the steps listed below.

1. Define your solid model in ABAQUS with desired geometry, material model, and mesh. Make sure that the direction of gravity is aligned with the -y axis (Or modify the definition of variable g in the code). It is recommended to align the top center at the origin or at y=0 so specifying the surface level height will be more straightforward in the following steps.

2. Under the load module, select the distributed load option. Apply a user defined load (VDLOAD) on external surfaces you want to apply RFT forces onto.

3. Modify the FORTRAN code to fit your model for the following parameters:

   (a) surface level height

   (b) ξ material scaling parameter

4. Create a job. Under the "general" tab add the Select the FORTRAN file containing the subroutine.

5. Submit job and verify results.

VDLOAD Uprooting Test

We have followed the process outlined in the previous section on a solid model of the large aluminum y-root sample in ABAQUS. We have used a linear elastic material model for the aluminum with elastic modulus E=0.1 GPa and Poisson's ratio ν=0.3. The solid was also assigned a density of 2700 kg/m^3. We then assigned a constant velocity boundary condition at the top surface in the y direction via a reference point rigidly tied to the surface. The pressure load distribution from 3D Rigid RFT partway through the solution can be seen in figure 4.15a. Since both gravity and resistive forces are applied instantaneously on the model, the initial results had plenty of oscillations. To damp these out we introduced mass scaling and increased linear bulk damping viscosity as suggested by the ABAQUS manual [49]. These results were clearer, however for ease of visualization we have further smoothed the data for the comparison in figure 4.15b. We have also made significant use of mass scaling to reduce some of the dynamic oscillations [49] Since the VDLOAD subroutine was only able to apply the perpendicular portion of the traction due to resistive forces, we modified the MATLAB solver to do the same. Despite the frequent oscillations even after multiple rounds of smoothing of the ABAQUS data, we see that the force required at the top of the root structure to uproot it at the assigned constant velocity generally agrees between the two solvers.

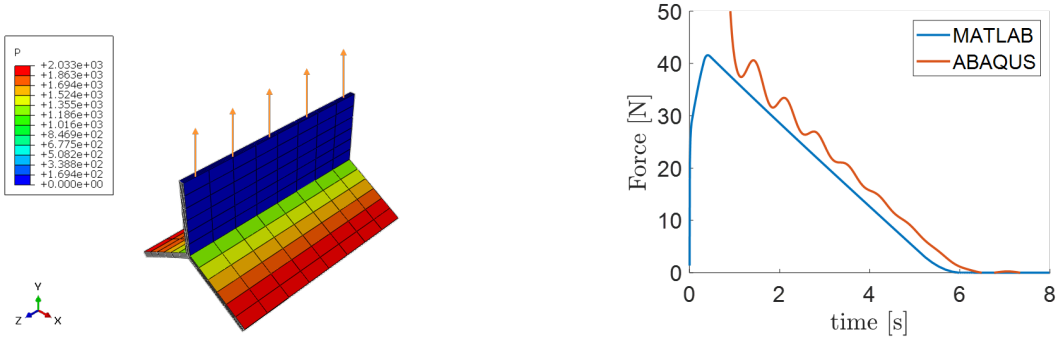


Figure 4.15: (a) Loading and geometry of aluminum root simulated in ABAQUS (b) Comparison of ABAQUS VDLOAD results with MATLAB eRFT results (with the tangential loads removed)

VDLOAD Complex Geometry Test

The main motivation in creating an ABAQUS RFT implementation was to create a more easily and widely usable RFT code that can be paired with the strong finite element modeler and solver in ABAQUS. To highlight the ability of our implementation to capture this goal even with just VDLOAD, we present a more complex geometry and boundary conditions in this section.

(a) (b) (c) (d) (e) (f) (g)

Figure 4.16: Sequential snapshots of the pressure loads from ABAQUS VDLOAD RFT an inverted V-shape being twisted and extracted

An inverted V-shaped soft aluminum structure has been modeled as a linear elastic solid with Young's modulus E=1 MPa and Poisson's ratio ν=0.3. The structure is tied to a rigid point above the center of its top face where a vertical linear velocity boundary condition and an angular velocity boundary condition are applied such that the structure is rotated as it is being pulled up. A sequence of the normal component of the resistive forces calculated via the VDLOAD subroutine can be seen in figure 4.16. The deformation of the structure is realistic. The pressure loads and thus deflections are larger earlier on in the simulation when the structure is deeper below the surface level. As the structure gets closer to the surface, its "arms" open further apart due to the lower resistive tractions with lower depth. We cna also see the leading edge hypothesis being applied correctly in the snapshots where both the rear edge of the arms are visible with no load applied. Although a linear elastic model is not the most appropriate for such large deformations, this initial test demonstrates the feasibility of modeling a wide range of geometries and materials in ABAQUS paired with RFT.

4.7 Conclusions & Future Work

This work proposed elastic RFT to improve upon rigid granular RFT. The theory for codirectional nonlinear and non-codirectional linear formulations of elastic RFT were presented. Steps for numerical implementation of both versions were outlined. Proof of elastic RFT's ability to address rigid RFT's shortcomings was demonstrated via the beam drop example and a study of intruder flexibility. With the introduction of elasticity to resistive force theory, it can now be applied to load-controlled scenarios and achievement of equilibrium or temporary no velocity positions can be captured.

A nonlinear inextensible beam theory is discussed and numerically implemented to model root structures. Numerical simulations of uprooting scenarios combining these beam structures with elastic RFT are compared to two sets of experimental results.

The ABAQUS implementation of resistive force theory has the potential to make resistive force theory accessible to a wider range of users. Currently, a user element subroutine to implement elastic RFT completely in ABAQUS is under way. This can become a great tool for engineers and scientists working on intrusion problems or designs that interface with flowing grains.

Granular resistive force theory still has a wide range of unexplored application scenarios. Although resistive force theory has been used for swimmers fully immersed in viscous fluids [50], it has not been expanded for viscous solids or granular media. Viscous resistive force theory a future area of consideration as we seek to represent the soils for the uprooting simulations more realistically.

The most recent set of experiments also strongly suggests that the relative depth of the intruder with respect to its characteristic length could have an influence on the resistive forces. The transition between deeply buried intruders and surface intrusions remains to be explored. W anticipate that this transition can be built into resistive force theory via the inclusion of a relative depth parameter in the formulations.

Appendix A

VDLOAD RFT FORTRAN Code

```
      subroutine vdload (
C Read only (unmodifiable)variables -
     1 nblock, ndim, stepTime, totalTime,
     2 amplitude, curCoords, velocity, dirCos, jltyp, sname,
C Write only (modifiable) variable -
     1 value )
C
      include 'vaba_param.inc'
C
      dimension curCoords(nblock,ndim), velocity(nblock,ndim),
     1  dirCos(nblock,ndim,ndim), value(nblock)
      character*80 sname
C

!declaring variables
  real :: coeffs(20,3)
  real :: RFTforce(3)
  real,  dimension(20) :: Tk !coefficients/terms

  real :: p1, p2, p3 !g dot v !g dot n ! n dot v
  real :: f1, f2, f3
  integer :: i
  real :: ksi, depth, area

  real :: n1, n2, n3
  real, dimension(3) :: v, n, g
  real, dimension(3) :: alfa_gen, alfa_gen_norm,alfa_gen_tang,alfa
  real :: vmag
  integer :: deneme

  logical :: negNV, negGN
```

```
  g = [0.0,-1.0,0.0] !gravity taken along y=-1

!###############fit coefficients #############
  coeffs(1,:) = [0.00212,   -0.06796,   -0.02634]
  coeffs(2,:) = [-0.02320,  -0.10941,   -0.03436]
  coeffs(3,:) = [-0.20890,  0.04725,    0.45256]
  coeffs(4,:) = [-0.43083,  -0.06914,   0.00835]
  coeffs(5,:) = [-0.00259,  -0.05835,   0.02553]
  coeffs(6,:) = [0.48872,   -0.65880,   -1.31290]
  coeffs(7,:) = [-0.00415,  -0.11985,   -0.05532]
  coeffs(8,:) = [0.07204,   -0.25739,   0.0679]
  coeffs(9,:) = [-0.02750,  -0.26834,   -0.16404]
  coeffs(10,:) = [-0.08772,  0.02692,    0.02287]
  coeffs(11,:) = [0.01992,   -0.00736,   0.02927]
  coeffs(12,:) = [-0.45961,  0.63758,    0.95406]
  coeffs(13,:) = [0.40799,   0.08997,    -0.00131]
  coeffs(14,:) = [-0.10107,  0.21069,    -0.11028]
  coeffs(15,:) = [-0.06576,  0.04748,    0.01487]
  coeffs(16,:) = [0.05664,   0.20406,    -0.02730]
  coeffs(17,:) = [-0.09269,  0.18519,    0.10911]
  coeffs(18,:) = [0.01892,   0.04934,    -0.04097]
  coeffs(19,:) = [0.01033,   0.13527,    0.07881]
  coeffs(20,:) = [0.1512,    -0.33207,   -0.27519]


  ksi= 133282; !media-dependent scaling coeff
  surfLevel = 3 !surface level

        do i = 1, nblock
        n1 = dirCos(i,3,1)
        n2 = dirCos(i,3,2)
        n3 = dirCos(i,3,3)
        n = [-n1,-n2,-n3]
        v = velocity(i,:)
vmag = sqrt(v(1)**2+v(2)**2+v(3)**2)+.0001;
v = [v(1)/vmag,v(2)/vmag,v(3)/vmag]
  !calculating dot products
        p1 = dot_product(g,v)
        p2 = dot_product(g,n)
        p3 = dot_product(n,v)

        negNV = (p3 .lt. 0)
        negGN = (p2 .lt. 0)
if ((p2 .lt. 0)) then
p2 = -p2
```

```
end if
        Tk = [1.0,p1,p2,p3,p1**2,p2**2,p3**2,p1*p2,p2*p3,p3*p1,
     +         p1**3,p2**3,p3**3,p1*p2**2,p2*p1**2,p2*p3**2,
     +         p3*p2**2,p3*p1**2,p1*p3**2,p1*p2*p3]
        f1 = sum(Tk*coeffs(:,1))
        f2 = sum(Tk*coeffs(:,2))
        f3 = sum(Tk*coeffs(:,3))
        alfa_gen = f1*n + f2*v + f3*g
        alfa_gen_norm = dot_product(alfa_gen,n)*n
depth = surfLevel - curCoords(i,2)
        RFTforce = ksi*alfa_gen_norm*depth
RFT_pres = dot_product(RFTforce,n)
if ((depth .lt. 0) .or. (p3 .lt. 1e-3)) then
RFT_pres = 0
end if

  value(i) = -RFT_pres*amplitude
  enddo

      return
      end
```

References

[1] S. Agarwal, C. Senatore, T. Zhang, M. Kingsbury, K. Iagnemma, D. I. Goldman, and K. Kamrin. "Modeling of the interaction of rigid wheels with dry granular media". *Journal of Terramechanics* 85 (2019), pp. 1–14.

[2] A. Hosoi and D. I. Goldman. "Beneath Our Feet: Strategies for Locomotion in Granular Media". *Annual Review of Fluid Mechanics* 47.Volume 47, 2015 (2015).

[3] N. Artemieva and B. Ivanov. "Launch of martian meteorites in oblique impacts". *Icarus* 171.1 (2004).

[4] S. Dunatunga and K. Kamrin. "Continuum modeling of projectile impact and penetration in dry granular media". *Journal of the Mechanics and Physics of Solids* 100 (Mar. 2017).

[5] R. D. Maladen, Y. Ding, C. Li, and D. I. Goldman. "Undulatory Swimming in Sand: Subsurface Locomotion of the Sandfish Lizard". *Science* 325 (2009).

[6] Kurstjens, Perdok, and Goense. "Selective uprooting by weed harrowing on sandy soils". *Weed Research* (2000).

[7] C. Hartmann, D. Tessier, and G. Pédro. "Changes in sandy Oxisols microfabric after mechanical up-rooting of an oil palm plantation". *Soil Micromorpohlogy: Studies in Management and Genesis*. Developments in Soil Science. Elsevier, 1993.

[8] R. P. Jensen, P. J. Bosscher, M. E. Plesha, and T. B. Edil. "DEM simulation of granular media—structure interface: effects of surface roughness and particle shape". *International Journal for Numerical and Analytical Methods in Geomechanics* 23.6 (1999).

[9] S. Agarwal, A. Karsai, D. I. Goldman, and K. Kamrin. "Efficacy of simple continuum models for diverse granular intrusions". *Soft Matter* 17.30 (2021), pp. 7196–7209.

[10] C. Kelly, N. Olsen, and D. Negrut. "Billion degree of freedom granular dynamics simulation on commodity hardware via heterogeneous data-type representation". *Multibody System Dynamics* 50 (Dec. 2020).

[11] C. Li, T. Zhang, and D. I. Goldman. "A Terradynamics of Legged Locomotion on Granular Media". *Science* 339.6126 (2013), pp. 1408–1412.

[12] K. Kamrin. "A hierarchy of granular continuum models: Why flowing grains are both simple and complex". *EPJ Web of Conferences* 140 (2017). Ed. by F. Radjai, S. Nezamabadi, S. Luding, and J. Delenne, p. 01007.

[13] G. Easson. "The Effects of Riparian Vegetation on Bank Stability". *Environmental Engineering Geoscience - ENVIRON ENG GEOSCI* 8 (Nov. 2002).

[14] D. H. Gray and R. Sotir. *Biotechnical and Soil Bioengineering Slope Stabilization: A Practical Guide for Erosion Control*. 1996.

[15] A. Ielpi, M. Lapôtre, C. Boyce, and M. Gibling. "The impact of vegetation on meandering rivers". *Nature Reviews Earth Environment* 3 (Jan. 2022).

[16] B. Abernethy and I. D. Rutherfurd. "The distribution and strength of riparian tree roots in relation to riverbank reinforcement". *Hydrological Processes* 15 (2001).

[17] S. Baets, J. Poesen, B. Reubens, K. Wemans, J. De Baerdemaeker, and B. Muys. "Root tensile strength and root distribution of typical Mediterranean plant species and their contribution to soil shear strength". *Plant and Soil* 305 (Apr. 2008).

[18] P. P. Capilleri, M. Cuomo, E. Motta, and M. Todaro. "Experimental Investigation of Root Tensile Strength for Slope Stabilization". *Indian Geotechnical Journal* 49 (Nov. 2019).

[19] J. Gray and G. J. Hancock. "The Propulsion of Sea-Urchin Spermatozoa". *Journal of Experimental Biology* 32.4 (1955), pp. 802–814.

[20] C. J. Brokaw. "FLAGELLAR PROPULSION". *Journal of Experimental Biology* 209.6 (Mar. 2006).

[21] S. Agarwal, A. Karsai, D. I. Goldman, and K. Kamrin. "Surprising simplicity in the modeling of dynamic granular intrusion". *Science Advances* 7.17 (2021), eabe0631.

[22] L. K. Treers, C. Cao, and H. S. Stuart. "Granular Resistive Force Theory Implementation for Three-Dimensional Trajectories". *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1887–1894.

[23] B. Rodenborn, C.-H. Chen, H. L. Swinney, B. Liu, and H. P. Zhang. "Propulsion of microorganisms by a helical flagellum". *PNAS : Proceedings of the National Academy of Sciences of the United States of America* 110.5 (2013).

[24] H. Askari and K. Kamrin. "Intrusion rheology in grains and other flowable materials". *Nature Materials* 15.12 (2016).

[25] H. Askari and K. Kamrin. "Intrusion in heterogeneous materials: Simple global rules from complex micro-mechanics". *Nature Materials* 15 (Dec. 2016).

[26] S. Agarwal, D. I. Goldman, and K. Kamrin. "Mechanistic framework for reduced-order models in soft materials: Application to three-dimensional granular intrusion". *Proceedings of the National Academy of Sciences* 120.4 (2023), e2214017120.

[27] T. Zhang and D. I. Goldman. "The effectiveness of resistive force theory in granular locomotiona)". *Physics of Fluids* 26.10 (2014), p. 101308.

[28] J. D. Goddard. "Continuum Modeling of Granular Media". *Applied Mechanics Reviews* 66 (Sept. 2014).

[29] S. Dunatunga and K. Kamrin. "Continuum modelling and simulation of granular flows through their many phases". *Journal of Fluid Mechanics* 779 (Sept. 2015).

[30] L. Anand. "A constitutive model for interface friction". *Computational Mechanics* 12 (1993).

[31] L. Anand, K. Kamrin, and S. Govindjee. *Introduction to mechanics of solid materials*. Oxford, England: Oxford University Press, 2022.

[32] L. Anand and S. Govindjee. *Continuum Mechanics of Solids*. Oxford University Press, July 2020.

[33] L. Anand and S. Govindjee. *Example Problems for Continuum Mechanics of Solids*. 2020.

[34] F. Maddalena and M. Ferrari. "Viscoelasticity of granular materials". *Mechanics of Materials* 20.3 (1995).

[35] A. R. ENNOS. "The mechanics of anchorage in seedlings of sunflower, Helianthus annuus L." *New Phytologist* 113.2 (1989).

[36] K. J. Niklas and H.-C. Spatz. "Worldwide correlations of mechanical properties and green wood density". *American Journal of Botany* 97.10 (2010), pp. 1587–1594.

[37] A. Tabiei and J. Wu. "Three-dimensional nonlinear orthotropic finite element material model for wood". *Composite Structures* (2000).

[38] K. Hasebe and S. Usuki. "Application of Orthotropic Failure Criterion to Wood". *Journal of Engineering Mechanics* (1989).

[39] L. Wagner, T. K. Bader, D. Auty, and K. de Borst. "Key parameters controlling stiffness variability within trees: a multiscale experimental–numerical approach". *Trees* 27.1 (2013), pp. 321–336.

[40] S. P. Timoshenko. *Elastic Curves*. 1953.

[41] J. Guermond, P. Minev, and J. Shen. "An overview of projection methods for incompressible flows". *Computer Methods in Applied Mechanics and Engineering* 195.44-47 (2006), pp. 6011–6045.

[42] A. J. Chorin. "The numerical solution of the Navier-Stokes equations for an incompressible fluid". *Bulletin of the American Mathematical Society* 73.6 (1967), pp. 928–931.

[43] A. J. Chorin. "A numerical method for solving incompressible viscous flow problems". *Journal of Computational Physics* 2.1 (1967), pp. 12–26.

[44] T. Zhang, S. L. Bai, Y. F. Zhang, and B. Thibaut. "Viscoelastic properties of wood materials characterized by nanoindentation experiments". *Wood Science and Technology* (Sept. 2012).

[45] A. P. Schniewind and J. D. Barrett. "Wood as a linear orthotropic viscoelastic material". *Wood Science and Technology* (Mar. 1972).

[46] A. N. Gent. "On the Relation between Indentation Hardness and Young's Modulus". *Rubber Chemistry and Technology* 31.4 (Sept. 1958).

[47] V. V. (V. Sokolovskiĭ. *Statics of granular media*. Completely rev. and enl. ed. Translated by J.K. Lusher. English translation edited by A.W.T. Daniel. [1st ed.] Pergamon Press, 1965.

[48] W. Kang, Y. Feng, C. Liu, and R. Blumenfeld. "Archimedes' law explains penetration of solids into granular media". *Nature Communications* (Mar. 2018).

[49] *ABAQUS User Guide*. Available at https://help.3ds.com/HelpDS.aspx?V=2021&P=DSSIMULIA_Established&L=English&contextscope=all&F=SIMACAESUBRefMap/simasub-c-vdload.htm. Dassault Systemes. 2021.

[50] L. Ohm. "Well-Posedness of a Viscoelastic Resistive Force Theory and Applications to Swimming". *Journal of Nonlinear Science* 34.5 (2024), p. 82.