

# AngioFE

*Version 3.0*

## User's Manual

**Last Updated: April 19, 2022**

**Contributors**

Steven LaBelle ([labellestevena@gmail.com](mailto:labellestevena@gmail.com))

Marsh Poulson ([therealmarshpoulson@gmail.com](mailto:therealmarshpoulson@gmail.com))

Steve Maas ([steve.maas@utah.edu](mailto:steve.maas@utah.edu))

Dr. Jeffrey Weiss ([jeff.weiss@utah.edu](mailto:jeff.weiss@utah.edu))

**Contact address**

Musculoskeletal Research Laboratories, University of Utah  
72 S. Central Campus Drive, Room 2646  
Salt Lake City, Utah

**Website**

MRL: <http://mrl.sci.utah.edu>

FEBio: <http://febio.org>

**Github**

FEBio: <https://github.com/febiosoftware/FEBio>

AngioFE: <https://github.com/febiosoftware/AngioFE>

**Forum**

<https://forums.febio.org/forum/plugin-forums>

Development of the FEBio project is supported in part by a grant from the U.S. National Institutes of Health.



# Table of Contents

<b>1. INTRODUCTION.....</b>	<b>1</b>
1.1. OVERVIEW OF ANGIOFE.....	1
1.2. CAPABILITIES.....	1
1.2.1. Usage as a standalone material.....	1
1.2.2. Doxygen documentation.....	1
1.3. STRUCTURE OF THIS DOCUMENT .....	1
<b>2. TERMINOLOGY .....</b>	<b>2</b>
2.1. GENERAL TERMS.....	2
2.1.1. Extracellular matrix (ECM, matrix).....	2
2.1.2. Linear elements .....	2
2.1.3. Higher order elements.....	2
2.2. VASCULAR NETWORK .....	2
2.2.1. Initial fragment .....	2
2.2.2. Tip .....	2
2.2.3. Stalk.....	2
2.2.4. Segment .....	3
2.2.5. Vessel .....	3
2.3. BRANCHING .....	3
2.3.1. Branch point.....	4
2.3.2. Zenith angle.....	4
2.3.3. Azimuth angle.....	4
2.4. ANASTAMOSIS .....	<b>ERROR! BOOKMARK NOT DEFINED.</b>
<b>3. THEORY &amp; IMPLEMENTATION DETAILS.....</b>	<b>6</b>
3.1. SEGMENT GROWTH THEORY.....	6
3.1.1. Growth velocity .....	6
3.1.2. Position dependent direction (PDD).....	6
3.1.3. Previous segment contribution (PSC) .....	6
3.1.4. Alpha (contribution mix) .....	6
3.2. AUTO TIMESTEP ADJUSTMENT .....	7
3.3. ACTIVE STRESS DUE TO VESSEL GROWTH.....	7
<b>4. INTERNAL THEORY .....</b>	<b>8</b>
4.1. PARENT VESSEL INITIALIZATION STEP (PROTO GROWTH) .....	8
4.2. VASCULAR GROWTH .....	8
<b>5. RUNNING FEBIO+ANGIOFE .....</b>	<b>9</b>
5.1. CONFIGURING FEBIO TO FIND ANGIOFE .....	9
5.2. RUNNING A SIMULATION WITH ANGIOFE .....	9
<b>6. FREE FORMAT INPUT .....</b>	<b>10</b>
6.1. FREE FORMAT OVERVIEW .....	10
6.2. MODULE SECTION .....	10
6.3. GLOBALS SECTION .....	10
6.3.1. Constants.....	10
6.3.1.1. Random seed.....	10
6.3.1.2. Toggle IO .....	11
6.3.1.3. Min scale factor.....	11
6.3.1.4. Bounds tolerance.....	11
6.3.1.5. Min and max Angio dt .....	11

6.3.1.6. Growth substeps .....	11
6.3.1.7. Bounce condition .....	11
6.4. MATERIALS.....	12
6.4.1. Constitutive models .....	12
6.4.2. Mix method.....	13
6.4.3. Vessel radius .....	13
6.5. INITIAL MODIFIER MANAGER.....	13
6.5.1. Density initializer.....	14
6.5.2. Discrete fiber randomizer .....	14
6.5.3. Discrete fiber EFD randomizer.....	14
6.5.4. EFD initializer .....	14
6.5.5. EFD rotated initializer.....	14
6.6. ANGIO STRESS POLICIES.....	15
6.6.1. Sigmoid angio stress policy.....	15
6.6.2. Sigmoid density angio stress policy .....	15
6.6.3. Load curve angio stress policy.....	16
6.6.4. Load curve density scaled angio stress policy .....	16
6.6.5. Load curve referential density scaled angio stress policy.....	16
6.7. SEGMENT VELOCITY MANAGER .....	16
6.7.1. Segment velocity over time .....	17
6.7.2. Segment velocity density scale modifier.....	17
6.7.3. Segment velocity referential density scale modifier .....	17
6.7.4. Segment velocity density & fractional anisotropy scale modifier .....	17
6.7.5. Sigmoid segment velocity modifier.....	18
6.7.6. Sigmoid adjusted segment velocity modifier .....	18
6.7.7. Gompertz segment velocity modifier.....	19
6.8. PREVIOUS SEGMENT CONTRIBUTION (PSC) MANAGER.....	20
6.8.1. Previous segment PSC modifier.....	20
6.9. PROTO PREVIOUS SEGMENT CONTRIBUTION (PSC) MANAGER.....	20
6.9.1. Proto previous segment PSC modifier .....	20
6.10. POSITION DEPENDENT DIRECTION (PDD) MANAGER .....	20
6.10.1. Fiber PDD.....	21
6.10.2. Fractional anisotropy PDD .....	21
6.10.3. Lagrange principal PDD .....	21
6.10.4. Anastomosis PDD .....	21
6.11. PROTO POSITION DEPENDENT DIRECTION (PDD) MANAGER.....	21
6.11.1. Proto fiber PDD.....	22
6.11.2. Proto fractional anisotropy PDD.....	22
6.11.3. Lagrange principal PDD .....	22
6.11.4. Proto anastomosis PDD.....	22
6.12. CONTRIBUTION MIX MANAGER .....	23
6.12.1. PSC-PDD contribution mix.....	23
6.12.2. Density-fractional anisotropy contribution mix .....	23
6.13. PROTO CONTRIBUTION MIX MANAGER .....	23
6.13.1. Proto PSC-PDD contribution mix.....	24
6.14. COMMON PROPERTIES .....	24
6.14.1. Vessel material properties.....	24
6.14.2. Fragment seeder.....	24
6.14.2.1. Initial segment length.....	24
6.14.2.2. By element fragment seeder bidirectional.....	25
6.14.2.3. By volume fragment seeder bidirectional.....	25
6.15. BRANCH POLICY .....	25
6.15.1. Delayed branching policy .....	25
6.16. INTERPOLATION PROPERTY .....	26
6.16.1. Per element variable interpolation.....	26
6.17. CELL SPECIES MANAGER .....	26

6.18. CELL REACTION MANAGER .....	27
6.19. PROBABILITY DISTRIBUTIONS .....	27
6.19.1. Normal distribution.....	27
6.19.2. Uniform distribution.....	28
6.19.3. Exponential distribution.....	28
6.19.4. Cauchy distribution.....	28
6.19.5. Chi squared distribution .....	28
6.19.6. Weibull distribution.....	29
6.19.7. Gamma distribution .....	29
6.19.8. Fixed distribution.....	29
6.19.9. Prescribed distribution.....	29
<b>7. OUTPUT DATA.....</b>	<b>30</b>
7.1. FILES.....	30
7.1.1. Angio log: <i>out_log.csv</i> .....	30
7.1.2. Vessel state: <i>out_vessel_state.ang2</i> .....	30
7.1.3. Final vessel file: <i>final_vessels.csv</i> .....	30
7.1.4. Cell state: <i>final_cells.txt</i> .....	30
7.1.5. Time statistics: <i>angio_stats.csv</i> .....	30
7.1.6. XPLT .....	31
7.2. VESSEL FILE FORMAT INFORMATION.....	31
<b>8. EXAMPLE PROBLEMS .....</b>	<b>33</b>
8.1. SIMPLE PROBLEMS .....	33
8.2. ANISOTROPY GRADIENTS .....	33
8.3. VARIATIONS IN INTERFACE STRUCTURE.....	34
<b>9. DEFAULT PARAMETERS.....</b>	<b>36</b>
<b>10. REFERENCES.....</b>	<b>37</b>

# 1. Introduction

## 1.1. Overview of AngioFE

The purpose of AngioFE is to simulate the growth of angiogenic neovessels and interactions between neovessels and the extracellular matrix. AngioFE was developed to mimic microvascular growth from microvessel fragments in collagen hydrogels. The default parameters for the growth rules are based on results from prior experiments, but can be modified by the user as desired [1-7].

## 1.2. Capabilities

This version of the plugin can simulate the growth of vascular networks in a variety of element types while considering a range of field variables that modify growth and matrix deformation. Additionally, the material the plugin provides can be used as the solid component of a multiphasic material. Currently the following element types are supported: Hex8 and Hex20. Support for Tet4 and Tet10 is in development. AngioFE generates an .ang2 file, which can be imported into PostView to visualize the vascular network within the model over time.

### 1.2.1. Usage as a standalone material

The angio material supplied by this plugin may be used a material for a finite element domain in a mechanical analysis. This material can theoretically be used as a parent material for any non-rigid material in FEBio. The angio material must be the head material for any domain in which neovessel growth is to be simulated. In the absence of an angio material, AngioFE will perform similarly to a model in FEBio.

### 1.2.2. Doxygen documentation

The doxygen documentation may be generated from the source code and the doxygen config file. All classes and public members of the plugin are commented with comments that can be read with doxygen.

```
>Doxygen.exe doxyconfig.txt
```

## 1.3. Structure of this document

The structure of this document is meant to mirror the structure the angio material in the .feb. If a section is a subsection of another within this document this means that, the tag for this subsection can be put in the section that it is contained in. The only exception to this is probability distributions, which are used as inputs to multiple other sections.

## 2. Terminology

This terminology will be used throughout the plugin. Some of the objects for which definitions are given include the mathematical properties of object in the way they are simulated by the plugin. These terms are what the user is expected to know and will not fully cover the internal implementation of the plugin.

### 2.1. General terms

General-purpose terminology.

#### 2.1.1. Extracellular matrix (ECM, matrix)

The extra cellular matrix is the angio material that neovessels may grow within. In the plugin, this is represented by the angio material class. This material has generally been specified to mimic the mechanical properties of type-I collagen hydrogels.

#### 2.1.2. Linear elements

Linear elements are finite element with linear shape functions. Linear elements available in AngioFE include HEX8 elements.

#### 2.1.3. Higher order elements

Any finite element that does not have linear shape functions. Currently this is the HEX20 element.

### 2.2. Vascular network

The vascular network is the microvessels that are contained within the ECM. The plugin represents each portion of the vessel as a line segment.

#### 2.2.1. Initial fragment

The parent microvessels that are seeded by the fragment seeder. These are generated during the initialization step before time  $t = 0$ .

#### 2.2.2. Tip

The ends of the vessels are the tips. These are where vascular growth occurs. The tips have a: position, direction, growth rate, and traction.

#### 2.2.3. Stalk

Any tip that has previously grown becomes a stalk, the stalk retains the position and stores the growth rate of the stalk is recorded at the time at which the tip grew.

### **2.2.4. Segment**

A discretized portion of a vessel this is represented by the plugin as a line segment with respect to the natural coordinate system of the element that contains it. A segment is contained entirely within a single finite element (multiple segments will be used if a vessel crosses between elements). Multiple segments may be used in the growth of a single tip in the growth process during a single timestep. The neovascular network is rendered as connected segments in PostView.

### **2.2.5. Vessel**

A collection of segments that originate from a single initial fragment.

## **2.3. Branching**

The angles and the frame of reference that they are based on are important for understanding branching within AngioFE. The three primary directions in this diagram: branch direction, parent segment direction, and fiber direction, form a basis that is used to orient any newly created branches.



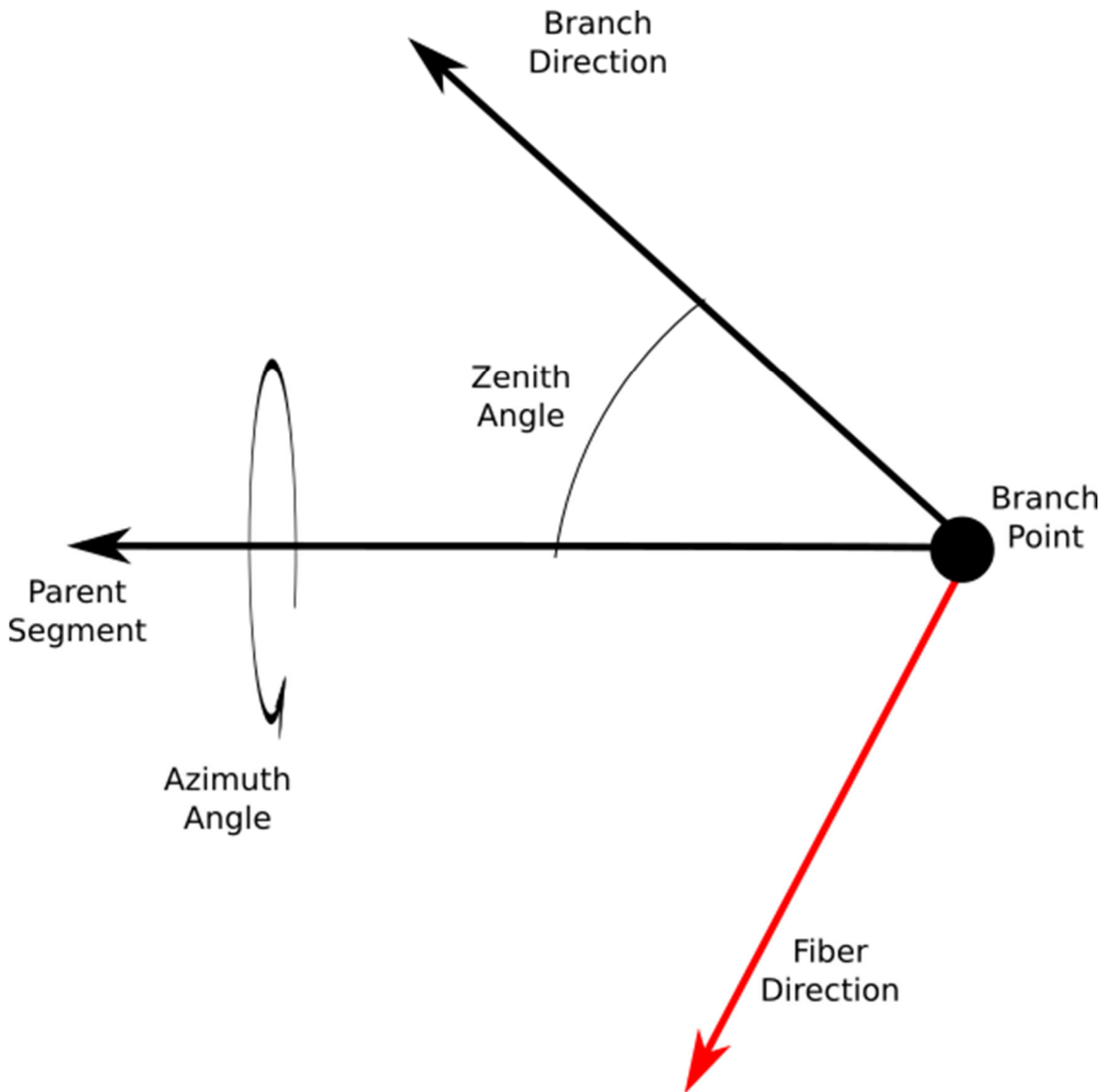


Figure 1: Branch angle specification.

### 2.3.1. Branch point

A point from which a branch will start growing at some point in time. These are generated along growing vessels based on user-input rules.

### 2.3.2. Zenith angle

The angle of departure from the parent segment.

### 2.3.3. Azimuth angle

The rotation of the branch about the axis of the parent segment.

## 2.4. Anastomosis

When two or more vessels grow close to each other, they can sometimes fuse via a phenomenon called anastomosis. Within AngioFE, users can choose to simulate anastomosis when two vessel tips grow towards each other. When anastomosis is selected AngioFE the growth model checks if there are any active tips near a vessel that do not come from the same initial parent microvessel. If an active tip is found then the segments will grow towards each other and fuse.

## 3. Theory & implementation details

Theoretical concepts that are needed to understand how AngioFE performs vascular growth. These concepts are fundamental to proper usage of AngioFE. Users are referred to our recent publications for detailed examples of the capabilities of the plugin.

### 3.1. Segment growth theory

AngioFE simulate sprouting angiogenesis from parent microvessel fragments. Each simulation begins with an initialization step that produces the initial parent microvessel fragments. New segments are sequentially added to the tips of the parent microvessel fragments. The growth rate and direction of the segments can be tied to local features of the matrix structure or mechanics. Growth rate and direction are controlled by managers that apply each modifier in the sequence that they are listed by the user in the input file.

The direction a vessel tip grows is determined by a combination of persistence and guidance. The persistence direction  $\psi$  is the direction that the vessel grew on the prior growth step. The guidance direction is currently only affected by the local collagen orientation direction  $\theta$ . During each time step the new direction  $\psi_{new}$  is calculate by partially rotating  $\psi$  to  $\theta$ . The rotation between the directions is scaled by the contribution mix  $\alpha$ .

#### 3.1.1. Growth velocity

The growth velocity manager determines the magnitude of the growth velocity (i.e., the growth rate). The effective growth rate is the product of each individual growth rate modifier. Each modifier can scale the effective growth rate based on either the current simulation time or local state variables e.g. matrix density, anisotropy, strain, local vessel volume fraction, etc.

#### 3.1.2. Position dependent direction (PDD)

The position dependent direction determines the direction that a vessel will grow due to local stimuli (e.g., matrix anisotropy, strain fields, etc.). Currently, only the collagen orientation direction  $\theta$  has been tested although there is support for guidance by other state variables.

#### 3.1.3. Previous segment contribution (PSC)

The previous segment contribution accounts for the persistence direction  $\psi$  of a vessel along the direction it grew in the last growth step.

#### 3.1.4. Alpha (contribution mix)

The new direction that a segment grows is determined by a partial rotation from the persistence direction  $\psi$  to the position dependent direction (generally  $\theta$ ). The partial rotation is scaled by  $\alpha$  which is in the range of [0-1]. Alpha can be set as a constant or modified to change based on the matrix anisotropy.

## 3.2. Auto timestep adjustment

The plugin adjusts the timestep before it is taken by FEBio to ensure the timestep is valid for vessel growth. This is to prevent vessels from growing through too many elements before the stress from those vessels is evaluated. The adjustment of timestep size will only cut down the step size to the maximum step that can be taken safely. This allows the growth of vessels to limit timesteps size in most cases but if mechanics are a limiting factor, the step size will remain unchanged. The plugin is meant to be used with FEBio's autostepper.

## 3.3. Active stress due to vessel growth

This section details the traction stress policies that can be implemented. In general, the stress field  $\sigma_s(\mathbf{x})$  at a position  $\mathbf{x}$  is calculated based on  $\mathbf{r}$  which is defined as the distance vector between  $\mathbf{x}$  and the location of the vessel tip  $\mathbf{x}_s$

$$\mathbf{r} = \mathbf{x}_s - \mathbf{x} \quad (2.1)$$

The angle between the vessel direction  $\boldsymbol{\psi}$  and the distance vector  $\mathbf{r}$  is indicated by  $\varphi$ . Stresses are applied at vessel tips as body forces. The resulting stress due to the body force is given by

$$\boldsymbol{\sigma}_x(\mathbf{x}) = ae^{-b\|\mathbf{r}\|} \cos^N\left(\frac{\varphi}{2}\right) \hat{\mathbf{r}} \otimes \hat{\mathbf{r}} \quad (2.2)$$

The components of the stress field are detailed below:

### Sprout Magnitude ( $a$ )

The parameter  $a$  is the effective magnitude of the stress field. This value can be scaled based on the current time and local state variable values. The sprout magnitude is specified by the type of the stress policy.

### Sprout Fan Exponential ( $N$ )

The directionality of the traction stress field is controlled by parameter  $N$ . As  $N$  increases, the stress field becomes increasingly polarized along the vessel direction  $\boldsymbol{\psi}$ . The default value of  $N$  is set to 2; alternative values could be used to increase the polarization of the stress field although the sprout magnitude,  $a$ , will likely need to be adjusted as well to account for these changes.

### Sprout Range ( $b$ )

The exponential in equation (2.2) controls how the stress field dissipates with radial distance  $r$  from the sprout tip. The default value for this is 200  $\mu\text{m}$ .

### Sprout Radius Multiplier

The sprout radius multiplier is a parameter affecting the implementation of the stress field but is not explicitly used in calculation of the stress field. This parameter limits how far the stress is evaluated from a vessel tip. Thus, the stress at positions farther than the product of the sprout range and the sprout radius multiplier are not evaluated since the stress is assumed negligible at such distances. Lowering this value may affect the accuracy of the stress fields adversely. Higher values will result in higher computation times.

## 4. Internal theory

This section reviews the implementation of the growth model including the parent vessel initialization step (proto-growth) and the ray-tracing method used to pass vessel tips between elements.

### 4.1. Parent vessel initialization step (proto growth)

AngioFE version 3.0 introduces a new initialization step that provides the user more control of the initial vascular network. Previously, parent microvessel fragments were seeded as straight lines of uniform length. However, vessels in *in vitro* cultures and *in vivo* are characterized by curvature and a distribution of initial lengths.

In version 3.0, the initialization step (proto-growth step) allows the users to specify initial vessel parameters based on probability distributions. Specifically, users prescribe distributions for the initial vessel length as well as parameters that affect the initial vessel curvature. Internally, tips from the same segment are seeded  $\sim 1 \mu\text{m}$  apart from each other. Next, the tips “grow” apart from each other. The length of this initial segment is derived from a user-defined probability distribution. The direction that a tip grows during the initialization step is defined by an ellipsoidal collagen orientation distribution function (ODF).

### 4.2. Vascular growth

Vessel tips are stored within the finite element containing them. A ray-tracing method is used to determine when growing tips will encounter faces between adjacent elements. The tip is then passed from one element to the next, allowing the tip’s growth to quickly be modified in response to changes in matrix structure and other state variables.

## 5. Running FEBio+AngioFE

This section details the steps to run AngioFE within FEBio as well as the free-format input

### 5.1. Configuring FEBio to find AngioFE

A configuration file is required to point the location of the AngioFE plugin to FEBio.

Windows example configuration file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<febio_config version="3.0">
  <import>C:\...\AngioFE.dll</import>
</febio_config>
```

Linux example configuration file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<febio_config version="3.0">
  <import>C:\...\AngioFE.so</import>
</febio_config>
```

It is important to check that the same version of FEBio that the plugin was built against is used.

### 5.2. Running a simulation with AngioFE

AngioFE is classified as a task plugin within the FEBio framework. The syntax for running AngioFE+FEBio from the command line is:

Windows:

```
febio.exe -i input_file_name.feb -cnf febio.xml task=angio
```

Linux:

```
febio.exe -i input_file_name.feb -cnf febio.xml task=angio
```

Alternatively, the plugin can be called after opening the FEBio terminal:

```
run -i input_file_name.feb -cnf febio.xml task=angio
```

## 6. Free format input

This chapter describes the XML-based input format used by FEBio with AngioFE. This format follows standard XML conventions and can be viewed with any file viewer that supports XML files. The free format input file can also be edited with any text editor. Please see the FEBio user manual for more details.

### 6.1. Free format overview

The different sections that support specification of policies, parameters, etc. in AngioFE are currently defined:

**Module** Defines the physics module for solving the model.

**Globals** Defines the global variables in the model

**Material** Specifies the materials used in the problem and the material parameters. For AngioFE this section will include both the description of the material model as well as most of the specification of vessel initialization, growth, behavior, and stresses.

**Output** Defines additional data that is to be stored including certain data associated with AngioFE.

### 6.2. Module section

The module section defines the type of analysis to perform within FEBio. Currently, AngioFE requires that the module be set to multiphasic:

```
<Module type="Multiphasic"/>
```

### 6.3. Globals section

The Globals section is used to define global variables used by both FEBio as well as those used by AngioFE. In addition to the constants required by the multiphasic module, there are a few parameters, which are specific to AngioFE. These parameters have default values that can be modified by the user as needed.

#### 6.3.1. Constants

In addition to the Global constants defined within FEBio, AngioFE has some unique constants.

##### 6.3.1.1. Random seed

The seed is the seed for the random engine. This ensures that the same input file will generate consistent results when rerun across different machines. If this parameter is changed, the vascular network will change even if all other parameters remain the same. This parameter is an integer.

```
<seed>1393430476</seed>
```

### 6.3.1.2. Toggle IO

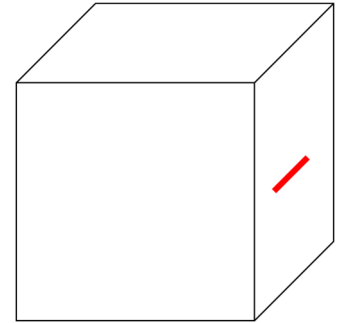
This optional parameter disables all of the custom files that are created by this plugin. Set this parameter to a nonzero value. If this parameter is not specified all of the files will be created.

```
<no_io>1</no_io>
```

### 6.3.1.3. Min scale factor

The min scale factor is used by the ray-tracing algorithm during segment growth. This factor affects how vessels behave when they encounter element faces. Generally, low values (e.g.,  $< 0.01$ ) are used. If the scale factor is too high, the vessel will become trapped in an element face rather than cross into a neighboring element (right).

```
<min_scale_factor>0.01</min_scale_factor>
```



### 6.3.1.4. Bounds tolerance

The bounds tolerance is another value used by the ray-tracing algorithm. This value is used in cases such as when a vessel grows across a vertex where it is ambiguous which adjacent element the segment should grow into. Setting this value too low will result in vessels becoming stuck along element boundaries. If this value is too loose, segments may grow into incorrect elements. Values between  $1e-3$  and  $1e-1$  are generally ok. The default value is  $1e-2$ . This can be overridden in the control section

```
<bounds_tolerance>1e-2</bounds_tolerance>
```

### 6.3.1.5. Min and max angio dt

These parameters modify the minimum and maximum step size taken before the growth model will be updated. The default maximum value is 0.25 days and there is no default minimum. These can be overridden in the control section

```
<min_angio_dt>0.1</min_angio_dt>
<max_angio_dt>0.25</max_angio_dt>
```

### 6.3.1.6. Growth substeps

Segment growth must be evaluated multiple times within a single time step if the segment changes elements. In this case, the first evaluation grows the vessel to the face of the element. Subsequent evaluations are performed to grow the segment in the new element. The default value for this is 3, which should allow a vessel to grow through at least 2 elements. Values less than 2 are not recommended, as segment growth will be paused at element faces until the next FEBio timestep is taken. If the growth velocity is high and segments are likely to grow through more than 2 elements, this value can be overridden in the control section

```
<growth_substeps>3</growth_substeps>
```

### 6.3.1.7. Bounce condition

The bounce condition handles how vessels behave when they encounter an external face of the model. Vessels can grow along the external face or they can bounce off the face. The former case is useful when simulating a physical boundary that the vessels would not be able to grow through. Growth along external faces is specified



by setting the bounce value to 0. The latter case is useful for simulating symmetry conditions in simplified models where growth of vessels out of the simulation domain is expected to match vessel growth into the simulation domain. Bouncing on external faces is the default case and can be specified by setting the bounce value to 0. This parameter can be overridden in the control section

```
<bounce>0</bounce>
```

## 6.4. Materials

The majority of the controls for the growth model within AngioFE are prescribed in the angio material. The angio material acts as a parent material to the constitutive model as well as stress policies associated with vessel tips, growth velocity modifiers, and position dependent modifiers. The angio material also defines the initialization step (i.e., the proto-growth step).

The syntax for declaring the angio material is:

```
<material id="[id number]" name="[material name]" type="multiphasic">
...
</material>
```

### 6.4.1. Constitutive models

The constitutive model is generally placed first within the angio material. Biphasic and multiphasic material parameters are specified immediately after the head `<Material>` tag. The remaining information is stored within a `<solid>` tag. The `<solid>` tag type is always set to “angio.” This section details both the matrix mechanical properties as well as all specifications of the growth model for the material. Matrix mechanical properties are stored within the `<matrix>` subtag. This subtag contains parameters that are normally detailed in the `<solid>` tag for standard FEBio analyses.

Example:

```
<material id="1" name="MaterialCore" type="multiphasic">
  <phi0>0.00567</phi0>
  <permeability type="perm-const-iso">
    <perm>3.588e4</perm>
  </permeability>
  <osmotic_coefficient type="osm-coef-const">
    <osmcoef>1</osmcoef>
  </osmotic_coefficient>
  <fixed_charge_density>0</fixed_charge_density>

  <solid type="angio">
    <matrix type="viscoelastic">
      <t1>1.25e-5</t1>
      <g0>0.0</g0>
      <g1>1.0</g1>
      <elastic type="EFD neo-Hookean">
        <E>2.38e5</E>
        <v>0.</v>
        <beta>2.5,2.5,2.5</beta>
        <ksi>4.1045e6,4.1045e6,4.1045e6</ksi>
      </elastic>
    </matrix>
  </solid>
</material>
```

```

    </matrix>
    <!-- the rest of the specification of parameters relevant to the growth model go
    here -->
  </solid>
</material>

```

### 6.4.2. Mix method

The mix method determines how  $\psi_{new}$  is calculated from  $\psi$  and  $\theta$ . Previously, this method was calculated as a weighted sum of two vectors with the weight  $\alpha$ :

$$\psi_{new} = (1 - \alpha)\psi + \alpha\theta. \quad (2.3)$$

This was the legacy method for determining the contribution of two directions in the calculation of a new direction. This method is no longer used since the angle between  $\psi$  and  $\theta$  does not vary linearly with  $\alpha$ . To overcome this, a new method is used which scales the rotation of  $\psi$  to  $\theta$  with the rotation matrix  $R$

$$\psi_{new} = R(\psi, \theta, \alpha). \quad (2.4)$$

In both cases, setting  $\alpha = 0.0$  results in  $\psi_{new} = \psi$  and setting  $\alpha = 1.0$  results in  $\psi_{new} = \theta$ .

The legacy approach can be used by specifying:

```
<mix_method type="LinInterp"/>
```

The default method can be used by specifying:

```
<mix_method type="LinRot"/>
```

### 6.4.3. Vessel radius

The vessel radius is used to calculate the relative contribution of the matrix and vessel submaterials. (This calculation is done assuming that the vessels do not change in volume as deformation occurs)

```
<vessel_radius>6.3</vessel_radius>
```

## 6.5. Initial modifier manager

The initial modifier manager is used to set material properties of the matrix used by AngioFE. These are different from those within FEBio. Generally, these allow the user to specify the matrix density, fibril vector orientation, or collagen ellipsoidal fibril distribution (EFD).

Each initial modifier is contained within the initial modifier manager tag:

```

<im_manager type="im_manager">
  ...

```

```
</im_manager>
```

### 6.5.1. Density initializer

The density initializer is used to set the matrix density of the material. The units are mg/mL. Users may specify a constant value for the whole material or reference an elemental data map:

```
<initial_modifier type="density_initializer">
  <initial_density>3.0</initial_density>
</initial_modifier>
```

### 6.5.2. Discrete fiber randomizer

The discrete fiber randomizer generates a vector field of randomly oriented collagen directions. This initial modifier is compatible with the fiber position dependent direction policy.

```
<initial_modifier type="fiber_randomizer"/>
```

### 6.5.3. Discrete fiber EFD randomizer

The discrete fiber EFD randomizer generates a vector field of collagen directions derived from a user-provided symmetric positive-definite (SPD) matrix defining an EFD. This differs from the discrete fiber randomizer that assumes a uniformly distributed EFD. This initial modifier is compatible with the fiber position dependent direction policy. Users may specify a constant value for the whole material or reference an elemental data map:

```
<initial_modifier type="discrete_fiber_efd_randomizer">
  <spd>xx,yy,zz,xy,yz,xz</spd>
</initial_modifier>
```

### 6.5.4. EFD initializer

The EFD initializer generates a map of EFDs for each element. The EFD is prescribed via a symmetric positive-definite (SPD) matrix. This initial modifier is compatible with the fractional anisotropy position dependent direction policy. Users may specify a constant value for the whole material or reference an elemental data map:

```
<initial_modifier type="efd_initializer">
  <spd>xx,yy,zz,xy,yz,xz</spd>
</initial_modifier>
```

### 6.5.5. EFD rotated initializer

The EFD rotated initializer generates a map of EFDs for each element. The EFDs are randomly rotated within the XY plane. This may be desirable for simulating heterogeneous matrices with some local alignment but overall isotropy. The EFD is prescribed via a symmetric positive-definite (SPD) matrix. This initial modifier is compatible with the fractional anisotropy position dependent direction policy. Users may specify a constant value for the whole material or reference an elemental data map:

```
<initial_modifier type="rot_efd_initializer">
  <spd>xx,yy,zz,xy,yz,xz</spd>
```

</initial\_modifier>

## 6.6. Angio stress policies

The angio stress policies define the stress fields surrounding the growing vessels. Different policies are available which allow the stress to be influenced by the simulation time and local state variables. Users will need to be careful with the units for stress evaluations particularly when days are used for time.

### 6.6.1. Sigmoid angio stress policy

The sigmoid angio stress policy scales the stress based on a sigmoidal curve evaluated at the current simulation time. The sigmoid curve has the form

$$y = y_0 + \frac{a}{1 + e^{\frac{t-x_0}{b}}} \quad (2.5)$$

Here,  $a$  is the amplitude of the curve,  $b$  affects the width of the curve,  $x_0$  is the center of the curve, and  $y_0$  is the minimum value of the curve. This curve is generally set to mirror the sigmoidal increase in vessel length.

```
<angio_stress_policy type="sigmoid_angio_stress_policy">
  <fan_exponential>2.0</fan_exponential>
  <sprout_range>200.0</sprout_range>
  <sprout_radius_multiplier>3.0</sprout_radius_multiplier>
  <a>0.0252</a>
  <b>0.5435</b>
  <x0>7</x0>
  <y0>0</y0>
</angio_stress_policy>
```

### 6.6.2. Sigmoid density angio stress policy

The sigmoid density angio stress policy scales the stress based on the local matrix density and a sigmoidal curve evaluated at the current simulation time. The sigmoid curve has the form

$$y = y_0 + \frac{a}{1 + e^{\frac{t-x_0}{b}}} \quad (2.6)$$

Here,  $a$  is the amplitude of the curve,  $b$  affects the width of the curve,  $x_0$  is the center of the curve, and  $y_0$  is the minimum value of the curve. This curve is generally set to mirror the sigmoidal increase in vessel length.

```
<angio_stress_policy type="sigmoid_dens_angio_stress_policy">
  <fan_exponential>2.0</fan_exponential>
  <sprout_range>200.0</sprout_range>
  <sprout_radius_multiplier>3.0</sprout_radius_multiplier>
  <a>0.0252</a>
  <b>0.5435</b>
```

```

    <x0>7</x0>
    <y0>0</y0>
  </angio_stress_policy>

```

### 6.6.3. Load curve angio stress policy

The load curve angio stress policy allows the user to scale the stress over time using a load curve. The magnitude of the stress in the example below is the product of  $x$  and the value of load curve 1 at the current simulation time:

```

<angio_stress_policy type="load_curve_angio_stress_policy">
  <sprout_mag lc="1">x</sprout_mag>
  <fan_exponential>2.0</fan_exponential>
  <sprout_range>200.0</sprout_range>
  <sprout_radius_multiplier>3.0</sprout_radius_multiplier>
</angio_stress_policy>

```

### 6.6.4. Load curve density scaled angio stress policy

The load curve density scaled angio stress policy allows the user to scale the stress based on the local matrix density and the time. The current matrix density scales the stress so that it decreases with matrix density as done previously [4].

```

<angio_stress_policy type="load_curve_den_angio_stress_policy">
  <sprout_mag lc="1">x</sprout_mag>
  <fan_exponential>2</fan_exponential>
  <sprout_range>200.0</sprout_range>
  <sprout_radius_multiplier>3.0</sprout_radius_multiplier>
</angio_stress_policy>

```

### 6.6.5. Load curve referential density scaled angio stress policy

The load curve referential density scaled angio stress policy is implemented similarly to the load curve density scaled angio stress policy. However, this policy uses the initial density for calculations of the density scale rather than the current density.

```

<angio_stress_policy type="load_curve_ref_den_angio_stress_policy">
  <sprout_mag lc="1">x</sprout_mag>
  <fan_exponential>7.0</fan_exponential>
  <sprout_range>200.0</sprout_range>
  <sprout_radius_multiplier>3.0</sprout_radius_multiplier>
</angio_stress_policy>

```

## 6.7. Segment velocity manager

The segment velocity manager organizes the segment velocity modifiers. The effective segment growth rate is the product of all the segment velocity modifiers. Each modifier is contained within the segment velocity manager tag:

```

<velocity_manager type="segment_growth_velocity_manager">
  ...
</velocity_manager>

```

### 6.7.1. Segment velocity over time

The segment velocity over time modifier scales the segment growth rate based on a user input value  $x$ :

```
<velocity_modifier type="segment_velocity_modifier">
  <segment_velocity_over_time>x</segment_velocity_modifier>
</velocity_modifier>
```

### 6.7.2. Segment velocity density scale modifier

The segment velocity density scale modifier scales the segment growth rate by  $v(x)$  which is based on the local matrix density. The scale is defined previously [4] as

$$v(x) = a + be^{-c\rho(x)}. \quad (2.7)$$

Here,  $\rho(x)$  is the local matrix density in mg/mL. This modifier requires that the interpolation property be specified.

```
<velocity_modifier type="segment_velocity_density_scale_modifier">
  <interpolation_prop type="per_element_vi"></interpolation_prop>
  <density_scale_factor>a,b,c</density_scale_factor>
</velocity_modifier>
```

### 6.7.3. Segment velocity referential density scale modifier

The segment velocity referential density scale modifier scales the segment growth rate by  $v(x)$  which is based on the local initial matrix density. The scale is defined previously [4] as

$$v(x) = a + be^{-c\rho_0(x)}. \quad (2.8)$$

Here,  $\rho_0(x)$  is the local initial matrix density in mg/mL. This modifier requires that the interpolation property be specified.

```
<velocity_modifier type="segment_velocity_ref_density_scale_modifier">
  <interpolation_prop type="per_element_vi"/>
  <density_scale_factor>a,b,c</density_scale_factor>
</velocity_modifier>
```

### 6.7.4. Segment velocity density & fractional anisotropy scale modifier

The segment velocity density and fractional anisotropy scale modifier scales the segment growth rate by the Lorentzian function  $v(x, \rho, FA)$  which is based on the local matrix density and the anisotropy based on the fractional anisotropy calculated from the primary and secondary semiprincipal axes of the local collagen fibril EFD.

$$v(\rho(x), FA(x)) = \frac{a}{\left(1 + \left(\frac{\rho(x) - \rho_{1/2}}{b}\right)^2\right) \left(1 + \left(\frac{FA(x) - FA_{1/2}}{c}\right)^2\right)} + d. \quad (2.9)$$

Here,  $\rho_{1/2}$  is the center of the Lorentzian function along the  $\rho$  axis and  $FA_{1/2}$  is the center of the Lorentzian function along the FA axis. The parameter  $a$  is the amplitude of the Lorentzian function,  $b$  and  $c$  are the spread of the function along the  $\rho$  and FA axes respectively, and  $d$  is the minimum value of the function. This modifier requires that the interpolation property be specified.

```
<velocity_modifier type="segment_velocity_density_fa_scale_modifier">
  <interpolation_prop type="per_element_vi"/>
  <rFA_a>3.413</rFA_a>
  <rFA_b>1.759</rFA_b>
  <rFA_c>0.6155</rFA_c>
  <rFA_d>0.1</rFA_d>
  <rFA_r0>2.0</rFA_r0>
  <rFA_f0>0.85</rFA_f0>
</velocity_modifier>
```

### 6.7.5. Sigmoid segment velocity modifier

The sigmoid segment velocity modifier scales the segment growth rate so that the total segment length increases in a sigmoidal fashion.

$$g(t) = g_0 + \frac{a}{1 + e^{-\frac{t-c}{b}}} \quad (2.10)$$

The actual growth rate is scaled by the time derivative of the sigmoid curve

$$\frac{dg(t)}{dt} = \frac{ae^{-\frac{t-c}{b}}}{b \left( 1 + e^{-\frac{t-c}{b}} \right)^2} \quad (2.11)$$

Here,  $a$  is the amplitude of the sigmoidal curve,  $b$  is the spread of the curve,  $c$  is the center of the curve.

```
<velocity_modifier type="sigmoid_segment_velocity">
  <a>100</a>
  <b>1.3</b>
  <c>5</c>
</velocity_modifier>
```

### 6.7.6. Sigmoid adjusted segment velocity modifier

The sigmoid adjusted segment velocity modifier allows the growth rate to increase at a sigmoidal rate over time. This modifier differs from the sigmoid segment velocity modifier in 2 ways. First, the center of the curve shifts for new vessels or branches that emerge after the initialization step. This is done so that when vessels emerge from branches their initial growth rate is slightly slower than that of vessels originating during the initialization step. Second, this modifier takes into account the local vessel volume fraction. Once the local vessel volume fraction exceeds a user-defined threshold the growth rate exponentially decreases. This was done so that the growth rate does not decrease over time unless vessels become crowded. The growth rate is defined as

$$g_a = s_a \frac{a_a}{1 + e^{\frac{t-c_a}{b_a}}} . \quad (2.12)$$

Here,  $a$  is the maximum growth rate,  $b$  is the spread of the curve, and  $c$  is the center of the curve. The local vessel volume fraction is accounted for by the scale  $s_a$  which is defined as

$$s_a = a_s e^{\frac{q_s}{b_s}} + s_0 . \quad (2.13)$$

Here,  $q_s$  is the ratio of the current vessel volume fraction  $w_s$  to the threshold volume fraction  $w_{thresh}$  which has a minimum of 1.0:

$$q_s = \min \left( \frac{w_v}{w_{thresh}}, 1.0 \right) . \quad (2.14)$$

The values  $a_s$ ,  $b_s$ , and  $s_0$  were selected so that when the vessel volume fraction was below the threshold  $w_{thresh}$ ,  $s_a = 1.0$ . Once  $q_s > 1.0$ ,  $s_a$  rapidly decreases until  $s_a = s_0$ . The parameters  $a_s$ ,  $q_s$ ,  $b_s$ ,  $s_0$ , and  $w_{thresh}$  are not currently exposed to the user.

```
<velocity_modifier type="sigmoid_adjusted_segment_velocity">
  <a>100</a>
  <b>2.572</b>
  <c>5</c>
</velocity_modifier>
```

### 6.7.7. Gompertz segment velocity modifier

The Gompertz segment velocity modifier prescribes growth as a Gompertz function (asymmetric sigmoid) given by:

$$g(t) = g_0 + a e^{-b e^{-c(t-d)}} . \quad (2.15)$$

Note that the gompertz function symmetric (similar to a sigmoid) when  $b$  is assigned  $\ln(2)$ ,  $c$  is assigned 1, and  $d$  is assigned 0.

Direct calculation is of the derivative of the Gompertz function given by

$$\frac{dg(t)}{dt} = abc e^{-b e^{-c(t-d)}} e^{-c(t-d)} . \quad (2.16)$$

In the above equations  $a$  is the scaling parameter equal to the total new length a tip will grow,  $b$  and  $c$  are controls of the Gompertz function for growth and decay, and  $d$  is a shifting parameter to better enforce shifts in the curve with respect to time.



```

<velocity_modifier type="gompertz_segment_velocity ">
  <a>284</a>
  <b>0.5</b>
  <c>1</c>
  <d>5</d>
</velocity_modifier>

```

## 6.8. Previous segment contribution (PSC) manager

The previous segment contribution (PSC) manager keeps track of modifiers to the prior direction  $\psi$  that a vessel grew.

```

<psc_manager type="previous_segment_contribution_manager">
  ...
</psc_manager>

```

### 6.8.1. Previous segment PSC modifier

The previous segment PSC modifier is the default PSC modifier. This returns the prior direction  $\psi$  that a vessel grew.

```

<psc_modifier type="previous_segment_psc"/>

```

## 6.9. Proto previous segment contribution (PSC) manager

The proto previous segment contribution (PSC) manager keeps track of modifiers to the prior direction  $\psi$  that a vessel “grew” during the initialization step.

```

<proto_psc_manager type="previous_segment_contribution_manager">
  ...
</proto_psc_manager>

```

### 6.9.1. Proto previous segment PSC modifier

The proto previous segment PSC modifier is the default proto PSC modifier. This returns the prior direction  $\psi$  that a vessel “grew” during the initialization step.

```

<proto_psc_modifier type="proto_previous_segment_psc"/>

```

## 6.10. Position dependent direction (PDD) manager

The position dependent direction manager manages the position dependent direction (PDD) modifiers. PDDs control the direction vessels grow based on local state variables. Each modifier is applied sequentially in the order they are specified in the input file. Each modifier is contained within the pdd manager tag

```

<pdd_manager type="position_dependent_direction_manager">
  ...
</pdd_manager>

```

### 6.10.1. Fiber PDD

The fiber PDD guides vessels based on the local fibril orientation. The local fibril orientation for this pdd is determined from the vector field of fibril orientation vectors stored at element nodes. The interpolation property must be defined for this tag.

```
<pdd_modifier type="fiber_pdd">
  <interpolation_prop type="per_element_vi"/>
</pdd_modifier>
```

### 6.10.2. Fractional anisotropy PDD

The fractional anisotropy PDD guides vessels based on the local collagen orientation EFD. At each time step, a direction is randomly sampled from the EFD.

```
<pdd_modifier type="fractional_anisotropy_pdd"/>
```

### 6.10.3. Lagrange principal PDD

The Lagrange principal PDD guides vessels based on the SPD defining the Lagrange strain tensor. This biases growth along the direction of the 1<sup>st</sup> principal Lagrange strain. The interpolation property must be defined for this tag.

```
<pdd_modifier type="lagrange_principal_pdd">
  <interpolation_prop type="per_element_vi"/>
</pdd_modifier>
```

### 6.10.4. Anastomosis PDD

Anastomoses occur when two vessel tips grow towards each other then fuse into a single vessel.

Tips will not anastomose with segments that have grown from the same initial parent microvessel fragment. The anastomosis PDD parameters include 1) the anastomosis radius that affects the radius at which the tip will start growing towards a segment, 2) the fuse radius that determines how close vessels must be before the anastomosis occurs, and 3) the fuse angle. The cosine of the fuse angle is required as an input. This prevents vessels from anastomosing at right angles.

```
<pdd_modifier type="anastomosis_pdd">
  <anastomosis_radius>200</anastomosis_radius>
  <fuse_radius>30</fuse_radius>
  <fuse_angle>0.25</fuse_angle>
</pdd_modifier>
```

## 6.11. Proto position dependent direction (PDD) manager

The proto position dependent direction manager manages the proto position dependent direction (PDD) modifiers. Proto PDDs control the direction vessels “grow” during the initialization step based on local state variables. These state variables may differ from those specified by the initial modifier manager. If they do, they must be specified within the declaration of the proto PDD modifiers unlike the syntax for the PDD modifiers. Not all of the standard

PDDs are available for the proto PDD manager. Each proto modifier is applied sequentially in the order they are specified in the input file. Each proto modifier is contained within the proto PDD manager tag.

```
<proto_pdd_manager type="proto_position_dependent_direction_manager">
...
</proto_pdd_manager>
```

### 6.11.1. Proto fiber PDD

The proto fiber PDD guides vessels based on the local fibril orientation as defined by the initial modifier manager. The local fibril orientation for this proto pdd is determined from the vector field of fibril orientation vectors stored at element nodes. The interpolation property must be defined for this proto PDD.

```
<proto_pdd_modifier type="proto_fiber_pdd">
  <interpolation_proptype="per_element_vi"/>
</proto_pdd_modifier>
```

### 6.11.2. Proto fractional anisotropy PDD

The proto fractional anisotropy PDD guides vessels based on the local collagen orientation EFD. At each proto time step, a direction is randomly sampled from the EFD. The proto fractional anisotropy PDD must have the SPD defining the EFD included. This is done since the initial orientation of the parent microvessel network may differ from the orientation of the collagen matrix containing the vessels.

```
<proto_pdd_modifier type="proto_fractional_anisotropy_pdd">
  <proto_efd>xx,yy,zz,xy,yz,xz</proto_efd>
</proto_pdd_modifier>
```

### 6.11.3. Lagrange principal PDD

The Lagrange principal PDD guides vessels based on the SPD defining the Lagrange strain tensor. This biases growth along the direction of the 1<sup>st</sup> principal Lagrange strain.

```
<pdd_modifier type="lagrange_principal_pdd"/>
```

### 6.11.4. Proto anastomosis PDD

Anastomoses occur when two vessel tips grow towards each other then fuse into a single vessel.

Tips will not anastomose with segments that have grown from the same initial parent microvessel fragment. The proto anastomosis PDD parameters include 1) the anastomosis radius that affects the radius at which the tip will start growing towards a segment, 2) the fuse radius that determines how close vessels must be before the anastomosis occurs, and 3) the fuse angle. The cosine of the fuse angle is required as an input. This prevents vessels from anastomosing at right angles.

```
<proto_pdd_modifier type="proto_anastomosis_pdd">
  <anastomosis_radius>200</anastomosis_radius>
  <fuse_radius>30</fuse_radius>
  <fuse_angle>0.25</fuse_angle>
</proto_pdd_modifier>
```

## 6.12. Contribution mix manager

The contribution mix manager handles how the parameter  $\alpha$  is set when calculating the mixture between two directions such as  $\psi$  and  $\theta$ . The contribution mix manager takes a couple of PSC modifiers as children, which are not supported by the PSC manager. This is done since the PSC manager keeps track of the direction of the previous direction while the contribution mix manager keeps track of  $\alpha$ .

### 6.12.1. PSC-PDD contribution mix

The PSC-PDD contribution mix modifier allows the user to specify the value of  $\alpha$ .

```
<cm_manager type="contribution_mix_manager">
  <psc_modifier type="psc_pdd_contribution_mix">
    <psc_weight> $\alpha$ </psc_weight>
  </psc_modifier>
</cm_manager>
```

### 6.12.2. Density-fractional anisotropy contribution mix

The density-fractional anisotropy contribution mix directly sets the value of  $\alpha$  based on the local matrix density and the fractional anisotropy calculated from the primary and secondary semiprincipal axes. The parameter  $\alpha(\text{FA})$  is defined by the sigmoidal curve

$$\alpha(\text{FA}) = \alpha_0 + \frac{a}{1 + e^{b(\text{FA} - c)}}. \quad (2.17)$$

In the above equation,  $\alpha_0$  is the minimum value of  $\alpha$ ,  $a$  is the amplitude of the curve,  $b$  is the spread of the curve, and  $c$  is the center.

```
<cm_manager type="contribution_mix_manager">
  <psc_modifier type="density_FA_contribution_mix"/>
</cm_manager>
```

## 6.13. Proto contribution mix manager

The proto contribution mix manager handles how the parameter  $\alpha$  is set during the initialization step when calculating the mixture between two directions such as  $\psi$  and  $\theta$ . The proto contribution mix manager takes one proto PSC modifier as a child, which is not supported by the proto PSC manager. This is done since the proto PSC manager keeps track of the direction of the previous direction while the proto contribution mix manager keeps track of  $\alpha$ . This also allows the value of  $\alpha$  to differ during the initialization step used to generate the initial microvessel fragments.

### 6.13.1. Proto PSC-PDD contribution mix

The proto PSC-PDD contribution mix modifier allows the user to specify the value of  $\alpha$  during the initialization step.

```
<proto_cm_manager type="proto_contribution_mix_manager">
  <proto_psc_modifier type="proto_psc_pdd_contribution_mix">
    <proto_psc_weight> $\alpha$ </proto_psc_weight>
  </proto_psc_modifier>
</proto_cm_manager>
```

## 6.14. Common properties

The common properties tag is used to assign mechanical properties to the vessels as well as to specify how the fragment seeder generates the initial vessels.

```
<common_properties type="angio_properties">
  ...
</common_properties>
```

### 6.14.1. Vessel material properties

The simulation domain in AngioFE assumes that the solid portion of the domain is a mixture of collagen fibrils and microvessel fragments. This mixture is weighted by the volume fraction of each solid component. While the collagen fibril properties are specified in the matrix section, vessel properties are specified within the common properties tag.

```
<vessel type="viscoelastic">
  <t1>1.25e-5</t1>
  <g0>0.0</g0>
  <g1>1.0</g1>
  <elastic type="neo-Hookean">
    <E>2.38e6</E>
    <v>0.0</v>
  </elastic>
</vessel>
```

### 6.14.2. Fragment seeder

The fragment seeder dictates how the initial vessel fragments are created as well as how many are generated per material. There are a few options for specifying this. Each type of fragment seeder requires definition of the number of fragments ( $n_{frag}$ ) and the initial segment length.

#### 6.14.2.1. Initial segment length

The initial segment length is prescribed as a probability distribution. This probability distribution can either be a fixed distribution (i.e. all vessels are the same initial length) or they can be sampled from a probability distribution function (PDF) input as a load curve by using the “prescribed distribution” distribution type. For the latter option, the load curve points should be formatted as:

```
<point>length,probability</point>
```

Refer to section 6.19 for a list of supported probability distributions.

#### 6.14.2.2. By element fragment seeder bidirectional

The by element fragment seeder bidirectional type randomly selects elements in the material to insert the new vessel. A point within the element is chosen as the center of the vessel. During the initialization step, each end of the vessel “grows” until the specified initial segment length is reached.

```
<fragment_seeder type="by_element_fragment_seeder_bidirectional">
  <number_fragments>135</number_fragments>
  <initial_segment_length type="prescribed_distribution">
    <distribution lc="1"/>
  </initial_segment_length>
</fragment_seeder>
```

#### 6.14.2.3. By volume fragment seeder bidirectional

The by volume fragment seeder bidirectional type randomly selects elements in the material to insert the new vessel. This method differs from the by element seeder in that the probability of randomly selecting an element is biased based on the volume of the element. Thus, this method is better suited to materials where the initial volume of each element is not the same. A point within the element is chosen as the center of the vessel. During the initialization step, each end of the vessel “grows” until the specified initial segment length is reached.

```
<fragment_seeder type="by_volume_fragment_seeder_bidirectional">
  <number_fragments>135</number_fragments>
  <initial_segment_length type="prescribed_distribution">
    <distribution lc="1"/>
  </initial_segment_length>
</fragment_seeder>
```

### 6.15. Branch policy

The branch policy determines when and where are created and will do any modification of the branch points that is needed. The branch policy determines the distance between branches along a vessel, the orientation of the new tip, and the time to emerge (i.e., how long after the branch is generated should it start to grow). Parameters in branch policies can be assigned to one of the probability distributions outlined in Section 6.19. A schematic of some of these parameters involved in branching is presented in Section 2.3.

#### 6.15.1. Delayed branching policy

This policy is a simple way to specify where branch points occur. After each growth step, if a vessel has grown the specified length to branch, then a branch point is created.

```
<branch_policy type="delayed_branching_policy">
  <interpolation_prop type="per_element_vi"></interpolation_prop>
  <azimuth_angle type="azimuth_angle_probability_distribution">
  <angle type="uniform_distribution">
    <a>0</a>
    <b>6.2831</b>
```

```

<time_clamped>0</time_clamped>
</angle>
</azimuth_angle>
<zenith_angle type="zenith_angle_probability_distribution">
<angle type="fixed_distribution">
<value>1.3</value>
</angle>
</zenith_angle>
<length_to_branch type="uniform_distribution">
<a>0</a>
<b>350</b>
<time_clamped>0</time_clamped>
</length_to_branch>
<time_to_emerge type="fixed_distribution">
<value>0.5</value>
</time_to_emerge>
</branch_policy>

```

## 6.16. Interpolation property

An interpolation property determines how values are interpolated from the integration points to all locations within an element. Currently there is only one option for this which is the per element variable interpolation. This has the possibility of being used in many other materials. In general, this will determine whether a given property is interpolated in manner that is continuous/discontinuous on element boundaries. This is often used to determine the value of properties at tips, which can be at any location within an element. Classes that implement this must interpolate doubles and quaternions. Interpolation of doubles can be used to do linear interpolation of various properties. In some cases this may be used to interpolate all of the members of a matrix to lerp between two matrices. (Linear interpolation may or may not be desirable for certain calculations) The interpolation of quaternions is used to interpolate directions. This can be advantageous as the resulting vector of a quaternion interpolation will have the same length as the initial vector (this operation may be similar to spherical linear interpolation). In the future, it may be advantageous to implement an interpolation property based on SPR interpolation.

### 6.16.1. Per element variable interpolation

Interpolate values on a per element basis. This interpolation is fast but can be discontinuous on element boundaries.

```

<velocity_modifier type="segment_velocity_density_scale_modifier">
  <interpolation_prop type="per_element_vi"></interpolation_prop>
</velocity_modifier>

```

## 6.17. Cell species manager

Note: The cell species manager is a feature that is currently under construction.

The cell species manager allows the user to assign solutes and solid bound molecules to the tips. This is currently stored per material for the purposes of initialization.

```

<cell_species_manager type="species_manager">
<cell_solute_prop type="Solute">
<Solute_ID>1</Solute_ID>
  </cell_solute_prop>
<cell_SBM_prop type="SBM">
<SBM_ID>1</SBM_ID>
  </cell_SBM_prop>
</cell_species_manager>

```

This example specifies that each cell contains a solute and an SBM which have the same ID as global solutes and species.

## 6.18. Cell reaction manager

The cell reaction manager is a feature that is currently under construction.

The cell species manager allows the user to define chemical reactions that take place inside of cells. These reactions do not directly affect chemical species within the mesh.

```

<cell_reaction_manager type="reaction_manager">
<cell_reaction name="production" type="cell mass-action-reversible">
<Vbar>0</Vbar>
<forward_rate type="cell constant reaction rate">
  <k>1e3</k>
</forward_rate>
<reverse_rate type="cell constant reaction rate">
  <k>1e-2</k>
</reverse_rate>
<vP sol="1">1</vP>
  </cell_reaction>
</cell_species_manager>

```

## 6.19. Probability distributions

The probability distributions are used as parameters for including uncertainty and natural variation in models.

### 6.19.1. Normal distribution

The Normal Distribution is one of the provided distributions. This distribution has 2 parameters: mean, and stddev. The mean is the mean of the distribution. The stddev is the standard deviation of the distribution.

[Wikipedia Page](#) [C++ Documentation](#)

```

<length_to_branch type="normal_distribution">
  <mean>200</mean>
  <stddev>5</stddev>
</length_to_branch>

```



### 6.19.2. Uniform distribution

The Uniform Distribution is one of the provided distributions. This distribution has 3 parameters: a, b, and time\_clamped. If time\_clamped is false a and b are the ends of the distribution. Otherwise the ends of the distribution are: a, and b-time. (1 is for time\_clamped true, 0 is for time\_clamped false)

[Wikipedia Page](#) [C++ Documentation](#)

```
<time_to_emerge type="uniform_distribution">
  <a>0</a>
  <b>10</b>
  <time_clamped>1</time_clamped>
</time_to_emerge>
```

### 6.19.3. Exponential distribution

The Exponential Distribution is another provided distribution. It has 2 parameters: mult and lambda. Lambda is the rate of this distribution. Mult is a scale parameter that scales the result of the distribution.

[Wikipedia Page](#) [C++ Documentation](#)

```
<time_to_emerge type="exponential_distribution">
  <lambda>0.5</lambda>
  <mult>1</mult>
</time_to_emerge>
```

### 6.19.4. Cauchy distribution

The Cauchy Distribution is another provided distribution. It has 2 parameters: a, and b. A is location. B is scale which must be greater than 0.

[Wikipedia Page](#) [C++ Documentation](#)

```
<time_to_emerge type="cauchy_distribution">
  <a>0.5</a>
  <b>0.5</b>
</time_to_emerge>
```

### 6.19.5. Chi squared distribution

The Chi Squared Distribution is another provided distribution. It has 2 parameters: dof, and mult. dof is degrees of freedom. Mult is scale.

[Wikipedia Page](#) [C++ Documentation](#)

```
<time_to_emerge type="chi_squared_distribution">
  <dof>3.0</dof>
  <mult>0.5</mult>
</time_to_emerge>
```

### 6.19.6. Weibull distribution

The Weibull Distribution is another provided distribution. It has 2 parameters: a, and b. A is shape. B is scale.

[Wikipedia Page](#) [C++ Documentation](#)

```
<time_to_emerge type="weibull_distribution">
  <a>2.0</a>
  <b>0.5</b>
</time_to_emerge>
```

### 6.19.7. Gamma distribution

The Gamma Distribution is another provided distribution. It has 2 parameters: alpha, and beta. Alpha is shape. Beta is rate.

[Wikipedia Page](#) [C++ Documentation](#)

```
<time_to_emerge type="gamma_distribution">
  <alpha>2.0</alpha>
  <beta>0.5</beta>
</time_to_emerge>
```

### 6.19.8. Fixed distribution

The fixed distribution only returns one value. This is used when a constant value is desired.

```
<initial_segment_length type="fixed_distribution">
  <value>100</value>
</initial_segment_length>
```

### 6.19.9. Prescribed distribution

The prescribed distribution allows users to define a probability distribution using a loadcurve. The loadcurve points are formatted so that the first component is the value and the second component is the probability associated with that value.

```
<initial_segment_length type="prescribed_distribution">
  <distribution lc="1"/>
</initial_segment_length>
```

## 7. Output Data

### 7.1. Files

This section describes the possible outputs of this plugin. The two main modes of output are creating files in the same directory the run was started from, or adding additional data to the .xplt file which can be viewed with heat maps.

#### 7.1.1. Angio log: out\_log.csv

Statistics from this plugin are recorded in out\_log.csv. The values recorded are: Time, Material, Segments, Total Length, Vessels, Branch Points, Anastomosis, and Active Tips. Time is the time of this data. Material is the material id (0 indexed). Total Length is the total vessel length within this material. Vessels is the number of vessels within this material which increases as branches are created and decreases as vessels fuse together due to anastomosis. Branch Points is the cumulative number of branches that have happened at the current time. Anastomosis is the number of tips that have fused to another vessel. Active tips are the number of tips within the current material that will grow in the next grow step.

#### 7.1.2. Vessel state: out\_vessel\_state.ang2

The vessel state file is a binary file containing a record of the vascular network over time. This file is used to visualize vessels in FEBioStudio. To import, select Post>Import lines and select the file. The “smooth lines 3D” render mode option is recommended for visualizing vessels.

#### 7.1.3. Final vessel file: final\_vessels.csv

The final vessel file contains data for each tip at the final time point. This file contains data from the tips such as position. This data may be analyzed to determine vessel network orientation and polarization.

#### 7.1.4. Cell state: final\_cells.txt

The final cell file is a text file that is similar to the vessel state file. This file contains a record of cell positions over time. This file is used to visualize cells in FEBioStudio. To import, select Post>Import points and select a file. The cell feature in FEBio is still in development. This file currently tracks cells at vessel tips and can output cellular concentrations of chemical species.

#### 7.1.5. Time statistics: angio\_stats.csv

The angio statistics file contains time information for how much time the model spent on each task.

### 7.1.6. XPLT variables

Some output options can be specified to show up in the heat maps in the FEBioStudio .xplt file. These can be specified in the same manner as other state variables within FEBio. Variables relevant to AngioFE are presented in the table below.

Variable Type	Description
fiber vector	The local collagen fibril orientation used with the vector field method for representing collagen orientation. This is used with the discrete fiber initial modifiers and PDDs.
angio SPD	The local collagen ellipsoidal fibril distribution stored as a symmetric positive-definite (SPD) matrix. This is used with the EFD initial modifiers and fractional anisotropy PDD.
angio ECM density	The elemental value of the collagen concentration in mg/mL.
angio stress	The stress due to vessel tip contractions.
vessel stress	The stress experienced by the vessel submaterial.
matrix stress	The stress experienced by the matrix submaterial.
vessel weight	The volume fraction of the vessels within each element.
matrix weight	The volume fraction of the matrix within each element.
matrix visco stress	The matrix submaterial's viscoelastic stress.
matrix elastic stress	The matrix submaterial's elastic stress.
branch count	The number of branches within each element.
segment length	The total segment length in each element.
anastomoses	The number of anastomoses in each element.

## 7.2. Vessel file format information

This format is the output format for the vessel file that is output by this plugin. The file has extension .ang2. All of the coordinates that are stored in this file are relative to the reference configuration. All numbers in this file are written in a little endian format. All integers in this file are unsigned. All floating-point numbers are in the IEEE 754 format. This file starts with the magic number 0xfdb97531. Next in the file is the version which is 4 bytes, the current version is 0. The remainder of the file is organized into sections with one section per mechanical (FEBio) timestep. Each section starts with the number of segments created in this timestep stored as a 4-byte integer. Next in the section is a floating-point number is the start time of the current mechanical timestep. Next in the file is the collection of segments that grew this timestep. Each segment is represented with 6 floating point numbers, x\_0, y\_0, z\_0, x\_1, y\_1, z\_1.

Revision 1. The file version number is now 1. Immediately following the version number is the number of 32 bit integers that are used as bitmasks to denote whether vessels can grow in a given material, this is a 4 byte integer. The bitmasks are sequential for each group of masks (the first bitmask integer is for materials 1 through 32, the second bitmask has data for materials 33 through 64, and so on). All mask groups are represented with a 4-byte integer. Within each mask, each bit represents whether or not vessels can grow in the given material. If the mask bit for a given material is 1 then vessels are allowed to grow within this material. If the mask bit for a given material is 0 then vessels are not allowed to grow within the material. Within each mask group the ones place is

the mask for the first material within the mask group; the two's place in the mask group is the mask for the second material within the mask group, and so on.

Revision 1 files will be generated by the current version of AngioFE.

The sample files include examples that should represent the major features of AngioFE.

## 8. Example problems

### 8.1. Simple problems

Four example problems are included that feature a basic setup. The simulation domain is a cube with constant material properties throughout the domain. Each model includes either the fiber pdd or the fractional anisotropy pdd.

- 1) Isotropic fiber pdd .
  - Low fibril alignment.
- 2) Anisotropic fiber pdd.
  - High fibril alignment.
- 3) Isotropic fractional anisotropy pdd.
  - Low fibril alignment.
- 4) Anisotropic fractional anisotropy pdd.
  - High fibril alignment.

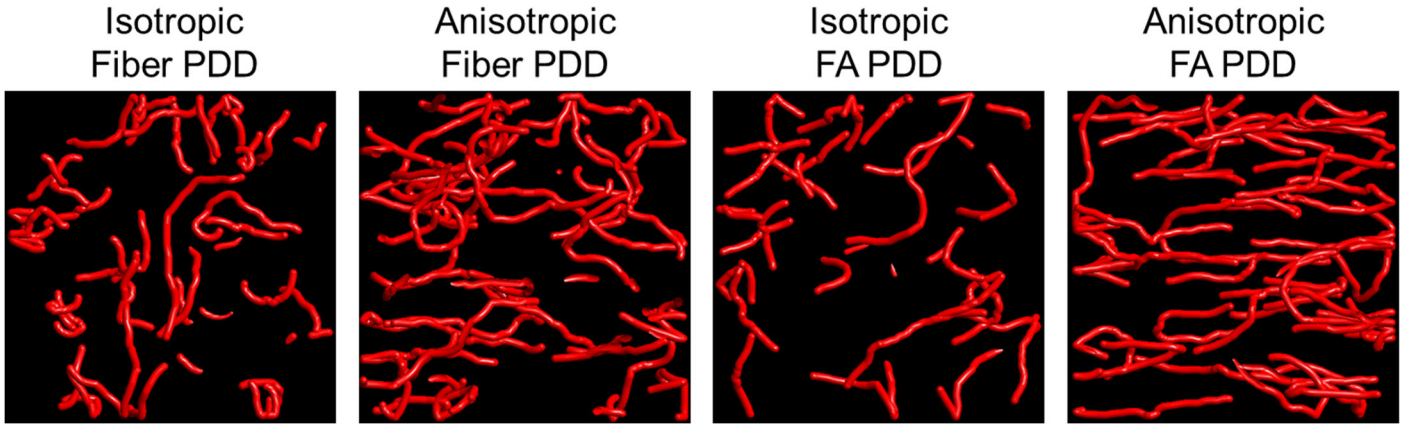


Figure 2: Representative images of growth in unaligned (isotropic) or aligned (anisotropic) matrices with either the fiber pdd (vector field) or fractional anisotropy pdd (EFD).

These problems feature

- Boundary condition where vessels bounce off external faces of the simulation.
- Prescribed distribution of initial microvessel lengths.

### 8.2. Anisotropy gradients

Three example problems with differing anisotropy gradients are included. The simulation domain is a rectilinear grid with vessels seeded on one end of the simulation domain.

- 1) Baseline model
  - Vessels are seeded on the left end. The anisotropy everywhere is set to low.
- 2) Positive gradient
  - Vessels are seeded on the left end. The anisotropy increases from left to right.

### 3) Negative gradient

- Vessels are seeded on the right end. The anisotropy decreases from right to left.

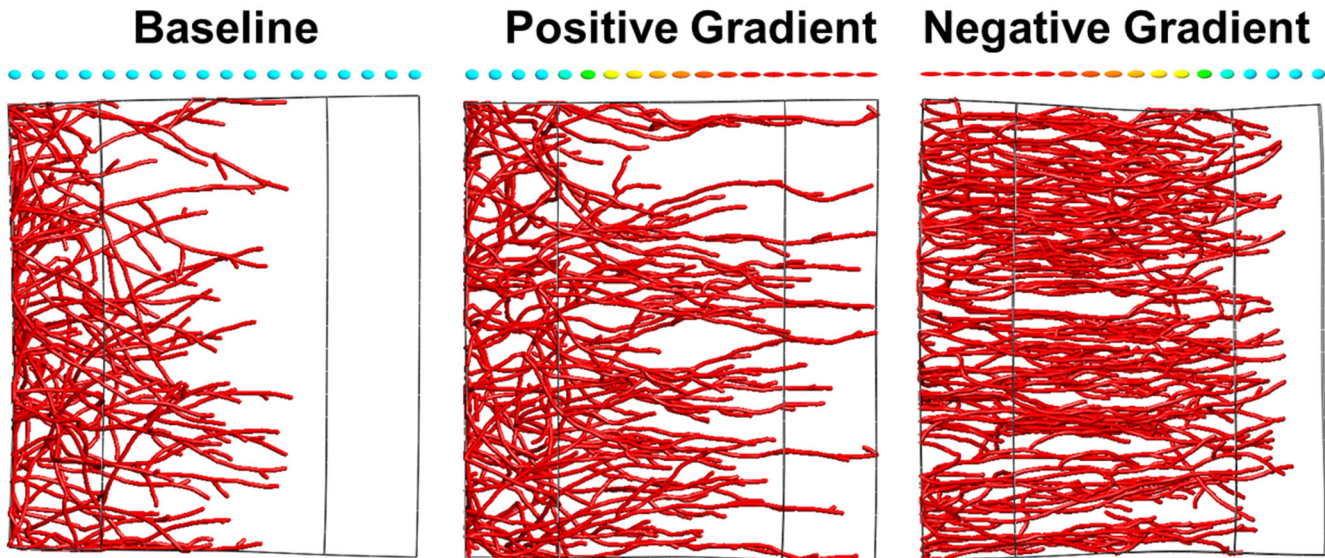


Figure 3: Representative results for anisotropy gradient simulations. The local anisotropy is represented by the ellipsoidal glyphs above each simulation.

These problems feature

- Data maps to specify the matrix anisotropy on a per-element basis.
- Branching.
- The sigmoid adjusted segment velocity modifier.
- Boundary condition where vessels grow along the external faces of the simulation.
- Fractional anisotropy PDD.
- Density-fractional anisotropy contribution mix.
- Prescribed distribution of initial microvessel lengths.
- Boundary condition restricting proto-growth to a single material.

## 8.3. Variations in interface structure

Six example problems with differing structure in an interface between a tumor and its periphery are included. The simulation domain is rectilinear with vessels seeded in the periphery. A thin interface separates the periphery from a tumor. Each model has a different specification of matrix density and anisotropy in the interface. The periphery and tumor are both relatively isotropic with a density of 3 mg/mL.

### 1) Baseline model

- The interface matrix density and alignment are the same as the periphery.

### 2) High density interface

- The interface matrix density is 5.0 mg/mL.

### 3) Alignment along the interface

- The collagen in the interface is aligned along the direction of the interface.

### 4) Alignment across the interface

- The collagen in the interface is aligned across the direction of the interface.
- 5) High density with alignment along the interface
- The interface matrix density is 5.0 mg/mL with fibrils along the direction of the interface.
- 6) High density with alignment across the interface
- The interface matrix density is 5.0 mg/mL with fibrils across the direction of the interface.

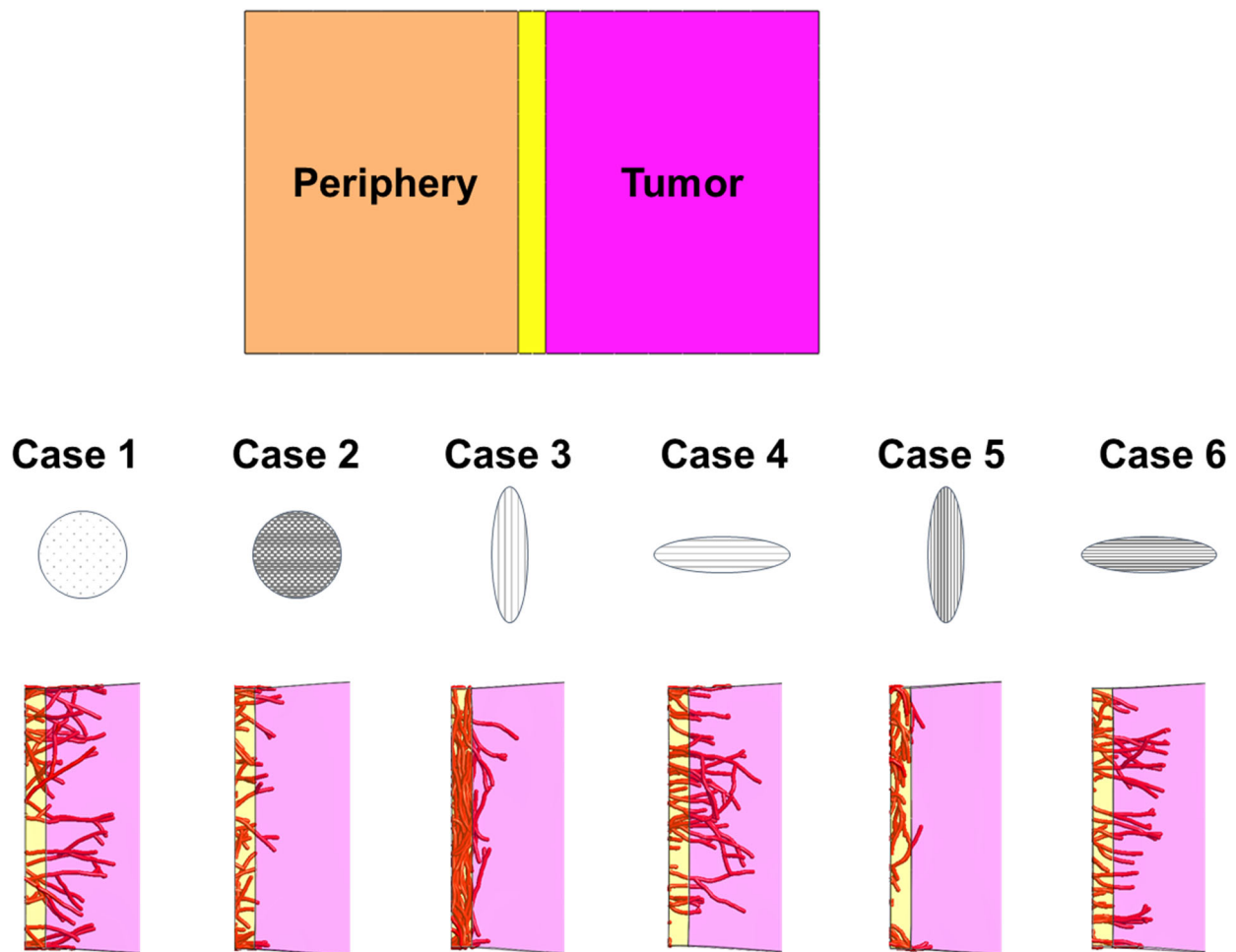


Figure 4: Simulation of vessels crossing an interface between a tumor and its periphery. Cutouts for each case are presented at the bottom.

These problems feature

- Branching.
- The segment velocity density & fractional anisotropy modifier.
- Boundary condition where vessels grow along the external faces of the simulation.
- Fractional anisotropy PDD.
- Density-fractional anisotropy contribution mix.
- Prescribed distribution of initial microvessel lengths.
- Boundary condition restricting proto-growth to a single material.



## 9. Default parameters

The table below summarizes the default parameters associated with each tag, whether the parameter is required, and references where relevant.

Section	Parameter	Tag	Default Value	Required?	Reference
Constants	Min scale factor	<min_scale_factor>	1e-2		
	Bounds tolerance	<bounds_tolerance>	1e-2		
	Growth substeps	<growth_substeps>	3		
	Min angio timestep	<min_angio_dt>	None		
	Max angio timestep	<max_angio_dt>	0.25 days		
	Bounce	<bounce>	1		
Angio stress policy	Sprout fan exponential	<fan_exponential>	2		
	Sprout range	<sprout_range>	200 $\mu\text{m}$		
	Sprout radius multiplier	<sprout_radius_multiplier>	3		

## 10. References

- [1] Edgar, L. T., Sibole, S. C., Underwood, C. J., Guilkey, J. E., and Weiss, J. A., 2013, "A computational model of in vitro angiogenesis based on extracellular matrix fibre orientation," *Comput Methods Biomech Biomed Engin*, 16(7), pp. 790-801.
- [2] Edgar, L. T., Underwood, C. J., Guilkey, J. E., Hoying, J. B., and Weiss, J. A., 2014, "Extracellular matrix density regulates the rate of neovessel growth and branching in sprouting angiogenesis," *PLoS One*, 9(1), p. e85178.
- [3] Underwood, C. J., Edgar, L. T., Hoying, J. B., and Weiss, J. A., 2014, "Cell-generated traction forces and the resulting matrix deformation modulate microvascular alignment and growth during angiogenesis," *Am J Physiol Heart Circ Physiol*, 307(2), pp. H152-164.
- [4] Edgar, L. T., Hoying, J. B., and Weiss, J. A., 2015, "In Silico Investigation of Angiogenesis with Growth and Stress Generation Coupled to Local Extracellular Matrix Density," *Ann Biomed Eng*, 43(7), pp. 1531-1542.
- [5] Edgar, L. T., Maas, S. A., Guilkey, J. E., and Weiss, J. A., 2015, "A coupled model of neovessel growth and matrix mechanics describes and predicts angiogenesis in vitro," *Biomech Model Mechanobiol*, 14(4), pp. 767-782.
- [6] Strobel, H. A., LaBelle, S. A., Krishnan, L., Dale, J., Rauff, A., Poulson, A. M., Bader, N., Beare, J. E., Aliaj, K., Weiss, J. A., and Hoying, J. B., 2020, "Stromal Cells Promote Neovascular Invasion Across Tissue Interfaces," *Frontiers in Physiology*, 11(1026).
- [7] LaBelle, S. A., Dinkins, S. S., Hoying, J. B., Budko, E. V., Rauff, A., Strobel, H. A., Lin, A. H., and Weiss, J. A., 2022, "Matrix anisotropy promotes angiogenesis in a density-dependent manner," *American Journal of Physiology-Heart and Circulatory Physiology*.