**AngioFE3**

*Version 3.0*

**User’s Manual**

**Last Updated: March 8, 2021**

**Contributors**

Steven LaBelle ([labellestevena@gmail.com](file:///D:\\Temp\\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](http://mrl.sci.utah.edu/)

FEBio: <http://febio>.org

**Forum**

http://mrlforums.sci.utah.edu/forums/

Development of the FEBio project is supported in part by a grant from the U.S. National Institutes of Health (R01GM083925).



**Table of Contents**

[1. Introduction 1](#_Toc522883668)

[1.1. Overview of AngioFE 1](#_Toc522883669)

[1.2. Capabilities 1](#_Toc522883670)

[1.2.1. Usage as a Standalone Material 1](#_Toc522883671)

[1.2.2. Usage in Multiphasic Materials 1](#_Toc522883672)

[1.3. Structure of this Document 2](#_Toc522883673)

[2. Terminology 2](#_Toc522883674)

[2.1. General 2](#_Toc522883675)

[2.1.1. Extracellular Matrix (ECM) 2](#_Toc522883676)

[2.1.2. Linear Elements 2](#_Toc522883677)

[2.1.3. Higher Order Elements 2](#_Toc522883678)

[2.2. Vascular Network 3](#_Toc522883679)

[2.2.1. Initial Fragment 3](#_Toc522883680)

[2.2.2. Tip 3](#_Toc522883681)

[2.2.3. Stalk 3](#_Toc522883682)

[2.2.4. Segment 3](#_Toc522883683)

[2.2.5. Vessel 4](#_Toc522883684)

[2.3. Branching 4](#_Toc522883685)

[2.3.1. Branch Point 5](#_Toc522883686)

[2.3.2. Zenith Angle 5](#_Toc522883687)

[2.3.3. Azimuth Angle 5](#_Toc522883688)

[3. Theory 5](#_Toc522883689)

[3.1. Segment Growth 5](#_Toc522883690)

[3.1.1. Growth Velocity 5](#_Toc522883691)

[3.1.2. Position Dependent Direction(PDD) 5](#_Toc522883692)

[3.1.3. Previous Segment Contribution(PSC) 5](#_Toc522883693)

[3.1.4. Alpha(Contribution Mix) 5](#_Toc522883694)

[3.1.5. Segment Growth Equation 5](#_Toc522883695)

[3.2. AutoTimestep Adjustment 6](#_Toc522883696)

[3.3. Growth 6](#_Toc522883697)

[3.3.1. Density 6](#_Toc522883698)

[3.3.2. Cell Proliferation 6](#_Toc522883699)

[3.3.3. Anastamosis 6](#_Toc522883700)

[3.3.4. Segment Length 6](#_Toc522883701)

[3.4. Neovessel Mechanics 8](#_Toc522883702)

[3.4.1. Stress Policy 8](#_Toc522883703)

[3.4.1.1. Grown Segments Angio Stress Policy 8](#_Toc522883704)

[3.4.1.2. Load Curve Velocity Angio Stress Policy 9](#_Toc522883705)

[3.4.1.3. Load Curve Angio Stress Policy (Recomended) 9](#_Toc522883706)

[3.4.1.4. Sigmoid Angio Stress Policy(Legacy) 10](#_Toc522883707)

[3.4.1.5. Passive Stresses 10](#_Toc522883708)

[3.4.1.6. Active Stresses 10](#_Toc522883709)

[4. Internal Theory 10](#_Toc522883710)

[4.1. Vascular Growth 10](#_Toc522883711)

[4.2. Proto Growth 11](#_Toc522883712)

[5. Running FEBio+AngioFE 11](#_Toc522883713)

[5.1. Configuring FEBio to find the Plugin 11](#_Toc522883714)

[5.2. Running a Simulation with the Plugin 12](#_Toc522883715)

[5.3. Material Attributes 12](#_Toc522883716)

[5.4. Initial Segment Velocity 12](#_Toc522883717)

[5.5. Initial Modifier 13](#_Toc522883718)

[5.5.1. Fiber Randomizer 13](#_Toc522883719)

[5.5.2. Prescribed Fiber Orientation 13](#_Toc522883720)

[5.5.3. Density Initializer 13](#_Toc522883721)

[5.6. Node Data Interpolation 14](#_Toc522883722)

[5.6.1. Reference ECM Density 14](#_Toc522883723)

[5.6.2. Repulse Value 14](#_Toc522883724)

[5.7. Growth Velocity 14](#_Toc522883725)

[5.7.1. Segment Velocity Modifier 15](#_Toc522883726)

[5.7.2. Segment Growth Velocity Density Scale Modifier 16](#_Toc522883727)

[5.8. Previous Segment Contribution (PSC) 16](#_Toc522883728)

[5.8.1. Previous Segment PSC 16](#_Toc522883729)

[5.9. Position Dependent Direction (PDD) 16](#_Toc522883730)

[5.9.1. Fiber PDD 17](#_Toc522883731)

[5.9.2. ECM Density Gradient PDD 17](#_Toc522883732)

[5.9.3. Repulse PDD 17](#_Toc522883733)

[5.9.4. Concentration PDD 17](#_Toc522883734)

[5.9.5. Anastamosis PDD 18](#_Toc522883735)

[5.10. Contribution Mix(Alpha) 18](#_Toc522883736)

[5.10.1. PSC/PDD ContributionMix 18](#_Toc522883737)

[5.11. BranchPolicy 19](#_Toc522883738)

[5.11.1. Delayed Branching Policy 19](#_Toc522883739)

[5.12. Probability Distributions 20](#_Toc522883740)

[5.12.1. Normal Distribution 20](#_Toc522883741)

[5.12.2. Uniform Distribution 20](#_Toc522883742)

[5.12.3. Exponential Distribution 20](#_Toc522883743)

[5.12.4. Cauchy Distribution 21](#_Toc522883744)

[5.12.5. Chi Squared Distribution 21](#_Toc522883745)

[5.12.6. Weibull Distribution 21](#_Toc522883746)

[5.12.7. Gamma Distribution 21](#_Toc522883747)

[5.12.8. Matrix Elastic Material 22](#_Toc522883748)

[5.13. Vascular Stress Modifiers 22](#_Toc522883749)

[6. Global Constants 22](#_Toc522883750)

[6.1. Seed 23](#_Toc522883751)

[6.2. Toggle IO 23](#_Toc522883752)

[6.3. Min Scale Factor 23](#_Toc522883753)

[6.4. Bounds Tolerance 23](#_Toc522883754)

[6.5. Min Angle 23](#_Toc522883755)

[6.6. Growth Substeps 23](#_Toc522883756)

[7. Output Data 24](#_Toc522883757)

[7.1. Files 24](#_Toc522883758)

[7.1.1. Log 24](#_Toc522883759)

[7.1.2. Vessel State 24](#_Toc522883760)

[7.1.3. XPLT 24](#_Toc522883761)

[7.2. Vessel File Format 2 25](#_Toc522883762)

[8. Warnings 26](#_Toc522883763)

[8.1. Branch Bombing 26](#_Toc522883764)

[9. Sample Files 26](#_Toc522883765)

[9.1. CIF\_SAL 26](#_Toc522883766)

[9.2. Anastamosis 26](#_Toc522883767)

[9.3. Bouncy 27](#_Toc522883768)

[9.4. Multi-Domian 27](#_Toc522883769)

[9.5. Density gradient 27](#_Toc522883770)

[9.6. Multiphasic 27](#_Toc522883771)

[9.7. Tet4 Model 27](#_Toc522883772)

[9.8. Tet10 Model 27](#_Toc522883773)

[9.9. Hex20 Model 27](#_Toc522883774)

[9.10. Hex20 Cumulative Stress 27](#_Toc522883775)

[9.11. Minimal Material 27](#_Toc522883776)

[10. Extending the plugin 28](#_Toc522883777)

[10.1. Growth Modifications 29](#_Toc522883778)

[10.2. Branching Modifications 29](#_Toc522883779)

[10.3. Stress Calculation Modifications 29](#_Toc522883780)

[10.4. Cautions 29](#_Toc522883781)

[10.5. Random Distributions 30](#_Toc522883782)

# Introduction

## Overview of AngioFE

The purpose of AngioFE is to simulate the growth of angiogenic neovessels within an extracellular matrix. The two main ways that this can be used are: given some experimental data determine the material/vascular properties needed to achieve comparable deformation over a given time period, the other is to predict the deformation that will occour in a new in vivo test given the rules about vessel growth that are used, and the material properties.

## 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, Hex20, Tet4, and Tet10. AngioFE3 generates an .ang2 file which can be imported into PostView to visualize the vascular network within the model over time.

### 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.

### Usage in Multiphasic Materials

The angio material can be used the solid component of a multiphasic material. In this context the angio material has the potential to both alter vascular topology based on chemical conditions, and alter the chemical conditions around active tips.(tips altering chemical conditions is not implemented)

### 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.

e.g.

>Doxygen.exe doxyconfig.txt

### CppCheck Static Analysis

The svn repository contains a config file that allows cppcheck to perform static analysis on the source files of the plugin.

## 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.

# Terminology

This is the terminology that 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.

## General

General purpose terminology.

### Extracellular Matrix (ECM)

The extra cellular matrix is the angio material that neovessels may grow in.. In the plugin this is represented by the AngioMaterial class.

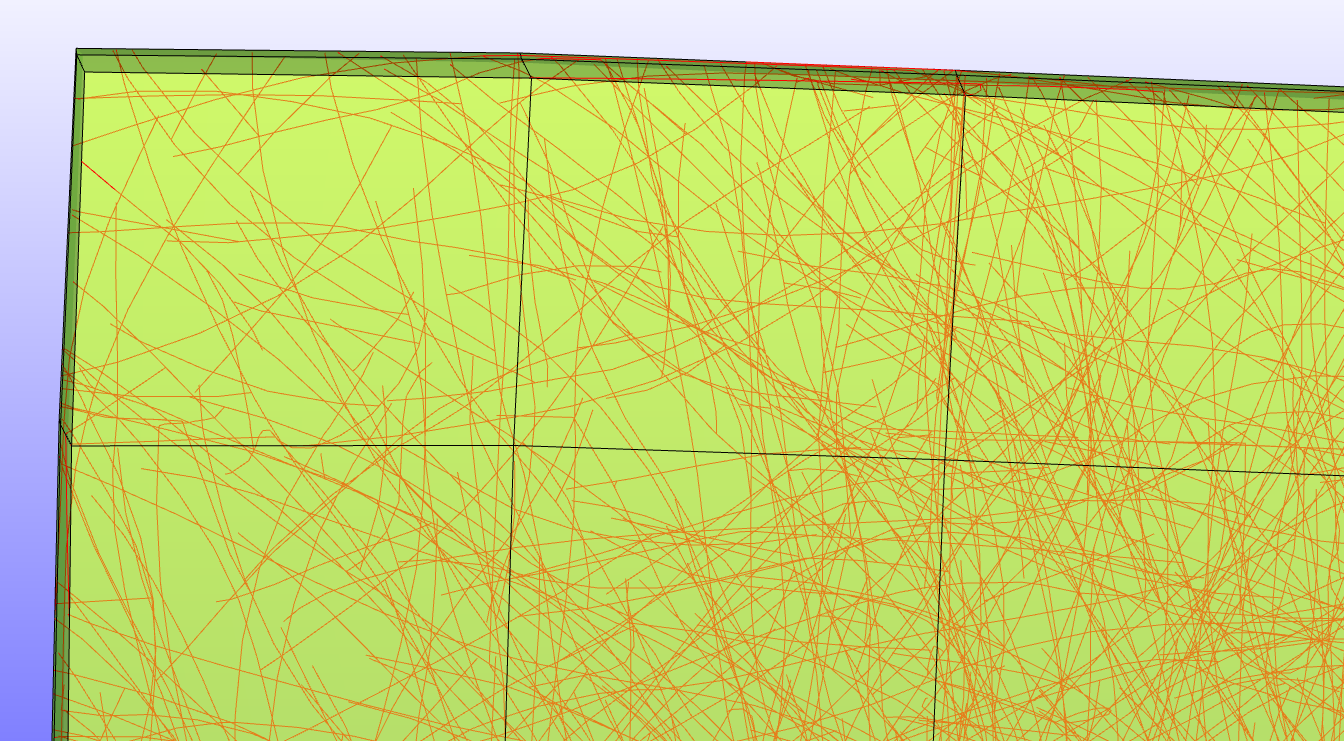
### Linear Elements

Linear elements are finite element with linear shape functions. Linear elements available in AngioFE include TET4 and HEX8 elements.

### Higher Order Elements

Any finite element that does not have linear shape functions. These include the TET10 and HEX20 elements.

## 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\*

### Initial Fragment

The microvessel that is seeded by a fragment seeder. This represents the microvessels that are mixed in the ECM at time t=0.

### Tip

An infinitesimal site where vascular growth can potentially occur. The tips have a: position, direction, and velocity. Tips may be used to calculate the stress that the ECM experiences as vessel growth occurs.

### Stalk

Any tip that has previously grown becomes a stalk, the stalk retains the position and stores the velocity of the tip that generated this point. The velocity of the stalk is recorded at the time at which the tip grew.

### 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.

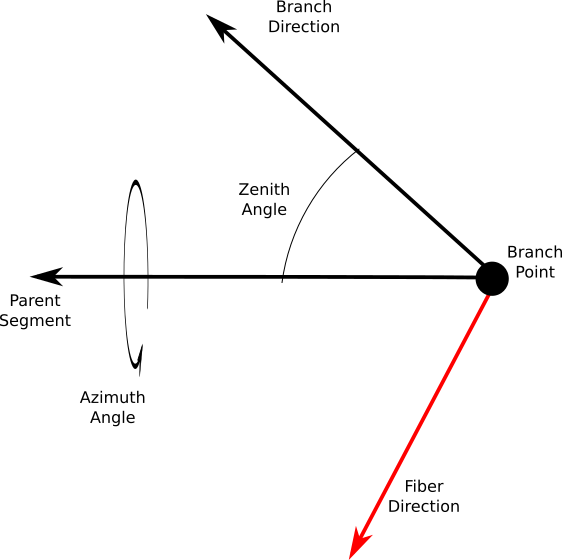
### Vessel

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

\*the segments are line segments with respect to natural coordinates of the element that contains them. In nonlinear elements this is not a line segment when viewed in the global coordinate system. The plugin assumes that the error in performing calculations as if all segments are line segments is minimal.

## Branching

The angles and the frame of reference that they are based on are important for understanding branching within AngioFE3. The three primary directions in this diagram: branch direction, parent segment direction, and fiber direction, form a basis which is used to orient any newly created branches.



### Branch Point

A point from which a branch will start growing at some point in time.

### Zenith Angle

The angle of departure from the parent segment.

### Azimuth Angle

The angle of branch departure relative to the basis formed from fiber direction and parent segment direction.

# Theory

Theoretical concepts that are needed to understand how AngioFE3 performs vascular growth. These concepts need to be understood in order to use AngioFE3 effectivly.

## Segment Growth

Segment growth is determined in a manner that ensures that the predicted neovascular network morphologies are independent of the chosen timestep.Each component of growth listed below can be composed of multiple sub components. This allows for complex interactions between the factors that determine each portion of growth.

### Growth Velocity

The magnitude at which a tip grows. Growth velocity is a function of: position, and time. This is used both in determining the timestep size to use and in vascular growth.

### Position Dependent Direction(PDD)

Numerous local field variables (e.g. ECM fiber anisotropy, cytokine gradients, strain fields) can influence the direction a neovessel grows. Position Dependent Direction is a function of position and time that accounts for the field variables that a user would like to influence growth direction.

### Previous Segment Contribution(PSC)

Persistance, i.e. growth in a direction over time is represented by the previous segment contribution. The PSC is a function of time and the previous segment’s orientation.

### Alpha(Contribution Mix)

The new direction that a segment grows is determined by a weighted sum of the PDD and PSC. Alpha is a scalar in [0-1] and is a function of position and time.

### Segment Growth Equation

The formulat for determining the direction a tip will grow depends on the user specified mixture method. This is either a linear interpolation between the components of two vectors or a linearly scaled rotation from the initial vector to the second. More detail is covered in 5.6.

## AutoTimestep 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. Safely is defined as not taking a step that would cause the stress from a tip to skip being evaluated in an element that the tip would pass through. 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. Setting dtmax to a load curve is the recommended way to limit timestep size where needed. The angio dt max parameter specifies a limit on the maximum time between angio evaluations. If contact or external loading are likely cases when FEBio will limit the timestep.

## Growth

### Density

The local ECM density that affects vascular growth is initially prescribed by the user at the angio material level. This density is scaled by the Jacobian (determinant of the local deformation gradient) at each FEBio step. Density may be prescribed on a per node basis. This will allow the user to have explicit control in areas where density changes rapidly.

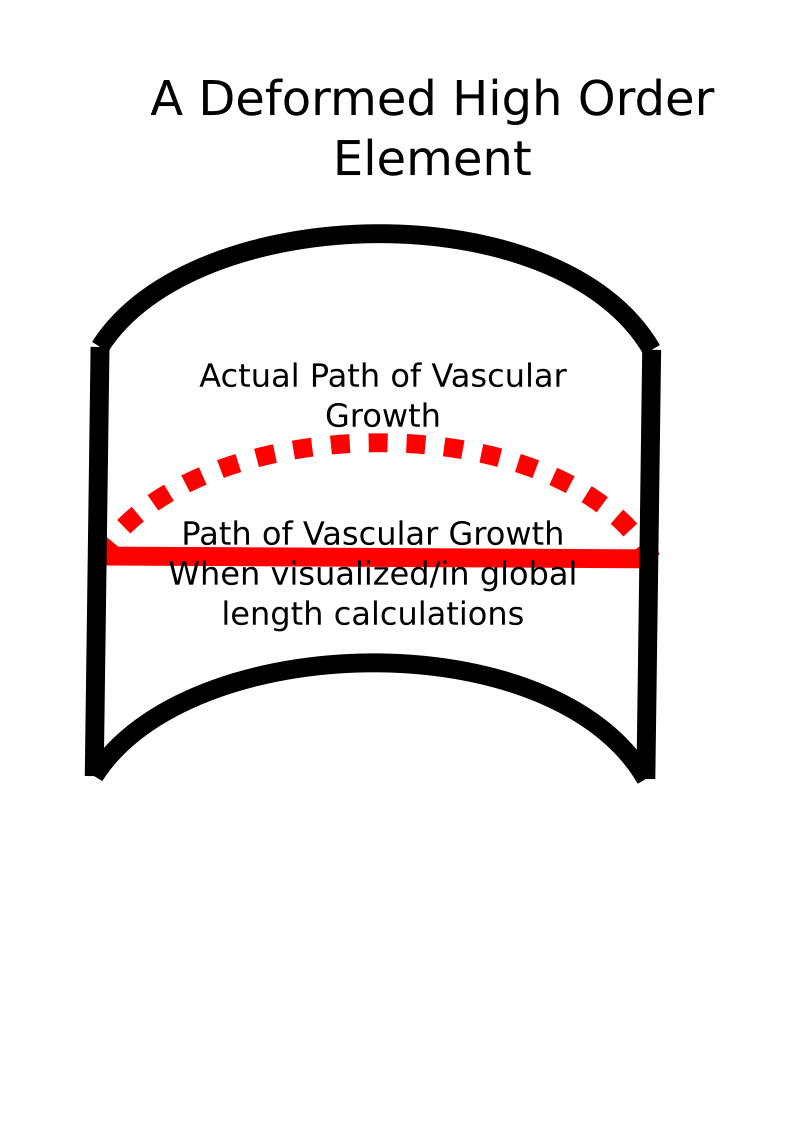
### Cell Proliferation

### Anastamosis

Within AngioFE3, anastamosis occurs when two tips near each other fuse into a single tip. Anastamosis is simulated by checking a radius around a given tip to find any active tips that do not have the same initial fragment. If an active tip like this is found the current segment will grow towards the other segment. When two tips are within a user specified fuse radius the vessels will fuse together and the two tips involed in anastomosis will cease growing. Additionally, the count of anasamoses in the element in which the tips fused will increase.

### Segment Length

Segment length is measured in the global coordinate system, and is calculated as a sum of the length of all line segments. The implication of this is that for nonlinear elements the calculated growth length may be different from the growth length of the underlying system due to the segments actually being curved if the elements the segment is in is deformed in certain ways. In this case segment length may be considered a lower bound of the actual length of segments in the simulation.



## Neovessel Mechanics

### Stress Policy

A stress policy determines how AngioFE calculates the force generated at tips. In the future it is possible to have a stress policy that includes segments in addition to active tip forces. In general, the stress is calculated at the tips. This stress depends on the distance between an active tip and the integration point where the stress is being evaluated. In an abstract sense a component of the stress at the tips is often a radial basis function. This radial basis function determines how the stress that a tip projects into the domain falls off as you move away from the tip.

#### General Stress Parameters

These parameters are used in multiple stress policies.

##### Sprout Width

##### Width of the fan of the effect of the stress.

##### Sprout Mag

The magnitude of the stress that each tip is applying.

##### Sprout Range

Used in the falloff expression as the stress decreases as the distance between the intergration point and the tip. Usually the falloff expression is e^(dist/sprout range) This is also a component of determining the radius around the current element to allow tips to effect the stress in the current element.

##### Sprout Radius Multiplier

This is the other component that determines how far to search for tips that effect stress in the current element. With the current falloff equation this cutoff is 2 standard deviations away from the tip. The search radius for tips that effect stress in the current element is sprout\_range\* sprout\_radius\_multiplier. This parameter defaults to 3 and probably should not be changed. Lowering this value may impact the accuracy of the stress fields adversely. Higher values will result in higher computation times. (accuracy will be slightly improved but probably not enough to matter)

#### Grown Segments Angio Stress Policy

Calculates the stress based on all tips and stalks within the model. The same distance calculations are used as in the Load Curve Velocity Angio Stress Policy. Not experimentally verified.

e.g.

<angio\_stress\_policy type=**"grown\_segments\_angio\_stress\_policy"**>

<sprout\_mag lc=**"2"**>**1.0**</sprout\_mag>

</angio\_stress\_policy>

#### Grown Segments Vel Angio Stress Policy

Calculates the stress based on all tips and stalks within the model. The same distance calculations are used as in the Load Curve Velocity Angio Stress Policy. The stress is scaled by the velocity of the segments grown. Not experimentally verified.

e.g.

<angio\_stress\_policy type=**"grown\_segments\_vel\_angio\_stress\_policy"**>

<sprout\_mag lc=**"2"**>**1.0**</sprout\_mag>

</angio\_stress\_policy>

#### Load Curve Velocity Angio Stress Policy

Calculates the stress based only on active tips. Stress is scaled by tip velocity. This should enable the sprout stress maginitude to be a scalar. Not experimentally verified.

<angio\_stress\_policy type=**"load\_curve\_vel\_angio\_stress\_policy"**>

<sprout\_mag>**3e-14**</sprout\_mag>

<sprout\_width>**7.0**</sprout\_width>

<sprout\_range>**250.0**</sprout\_mag>

<sprout\_radius\_multiplier>**3.0**</sprout\_radius\_multiplier>

</angio\_stress\_policy>

#### Load Curve Angio Stress Policy

Users may specify the stress exerted by active tips over time with a load curve. This may be useful for debugging purposes.

e.g.

<angio\_stress\_policy type=**"load\_curve\_angio\_stress\_policy"**>

<sprout\_mag lc=**"2"**>**1.0**</sprout\_mag>

<sprout\_width>**7.0**</sprout\_width>

<sprout\_range>**250.0**</sprout\_mag>

<sprout\_radius\_multiplier>**3.0**</sprout\_radius\_multiplier>

</angio\_stress\_policy>

#### Load Curve Density Scaled Angio Stress Policy (Recomended)

Users may specify the stress exerted by active tips over time with a load curve. This may be useful for debugging purposes. The stress is scaled by the local density scale at the current time.

e.g.

<angio\_stress\_policy type=**"load\_curve\_den\_angio\_stress\_policy"**>

<sprout\_mag lc=**"2"**>**1.0**</sprout\_mag>

<sprout\_width>**7.0**</sprout\_width>

<sprout\_range>**250.0**</sprout\_mag>

<sprout\_radius\_multiplier>**3.0**</sprout\_radius\_multiplier>

</angio\_stress\_policy>

#### Load Curve Referential Density Scaled Angio Stress Policy

Users may specify the stress exerted by active tips over time with a load curve. This may be useful for debugging purposes. The stress is scaled by the local density scale at the current time.

e.g.

<angio\_stress\_policy type=**"load\_curve\_ref\_den\_angio\_stress\_policy"**>

<sprout\_mag lc=**"2"**>**1.0**</sprout\_mag>

<sprout\_width>**7.0**</sprout\_width>

<sprout\_range>**250.0**</sprout\_mag>

<sprout\_radius\_multiplier>**3.0**</sprout\_radius\_multiplier>

</angio\_stress\_policy>

#### Sigmoid Angio Stress Policy (Legacy)

The stress is scaled by a sigmoid curve and the density scale factor. This should only be used to validate legacy models which assumed that stress didn’t develop for the first few days of growth before ramping sharply around day 4. Only recommended for reproducing results from Angio3d.

e.g.

<angio\_stress\_policy type=**"sigmoid\_angio\_stress\_policy"**>

<sprout\_mag lc=**"2"**>**1.0**</sprout\_mag>

<sprout\_width>**7.0**</sprout\_width>

<sprout\_range>**250.0**</sprout\_mag>

<sprout\_radius\_multiplier>**3.0**</sprout\_radius\_multiplier>

<a>**1.0081**</a >

<b>**0.5436**</b >

<x0>**2**</x0>

<y0>**-.004**</y0>

</angio\_stress\_policy>

#### Passive Stresses

#### Active Stresses











# Internal Theory

The theroretical background on how the plugin works internally. The information presented here is not needed to use the plugin.

## Vascular Growth

This is the portion of vascular growth that is not covered in the theory section.

Each AngioElement tracks the active tips within it’s volume. The active tips are stored in a double-buffered data structure. This data structure allows one buffer to be read from and the other to be written to. In the parallelization code, each AngioElement can be proceesed without any data conflicts from other AngioElements. In this same vein, each AngioElement checks if any tips on the faces of surrounding elements will grow into this element (adjacency in this case is defined as sharing a node with this element).

After the new growth direction of a segment is determined, the direction’s basis is transformed to the natural coordinate system of the element containing the segment. The growth direction is then projected onto the surface of the element such that the scale factor to the element surface is positive. Next, the distance is compared to the desired distance of growth during this timestep. If the potential distance to grow is greater than the desired growth length, then a segment is created and the tip is placed into a datastructure that indicates that it should continute growing on the next set of growth steps. Otherwise, a segment is created and the tip is placed in a data structure that indicates that the tip could continue growing into an adjacent element.

When tips are on the surface of the element and are sufficiently close to the edges of the element (where a number of elements meet) it is possible that the vessel could grow into one of a number of elements. To combat this, as tips cross element boundaries there is a select element function that causes the tip to be placed in at most one element. The select element function selects the element that the tip could continue growing in that the segment could grow for the most distance, and that does not change the direction of the vessel too much. Changing the direction of the tip too much is based on the tip not ending up growing in the complete opposite direction of the direction that it was previously growing.

## Proto Growth

Internally there is a set of growth steps that occour before finite element analysis begins. The purpose of these steps is to make sure that the initial fragments are the correct length. As tips are initially seeded they have no length. The proto growth steps in some sense go from time -1 to time 0. Due to choosing this time constraint it is easy for the plugin to calculate the velocity that initial tips need in order to grow to the length that is specified by the user. This designation also allows for a branching policy to be specified just for these proto timesteps. A proto branching policy allows branches to emerge from the initial fragments to make the fragments at time 0 more representative of experimental fragments that often contain a main microvessel fragment with numberous branches.

# Running FEBio+AngioFE

## Configuring FEBio to find the Plugin

A configuration file which allows FEBio to find the AngioFE is required. An example of this file febio.xml is:

<?xml version=**"1.0"** encoding=**"ISO-8859-1"**?>

<febio\_config version=**"1.0"**>

<linear\_solver type=**"pardiso"**></linear\_solver>

<import>**C:\...\AngioFE3.dll**</import>

</febio\_config>

Make sure the linear solver is supported by your build of FEBio. Make sure the import line points to the AngioFE3 shared object. Additionally, each build of the plugin can only be run with a build of FEBio that is built with the same complier for the same operating system.

## Running a Simulation with the Plugin

When run from the command line, the command should be like:

(Windows) Run febio.exe –i control.feb –cnf febio.xml task=angio angiofe.txt

(Linux) ./febio.exe –i control.feb –cnf febio.xml task=angio angiofe.txt

Angiofe.txt need not actually exist in the filesystem; if it does exist its contents will be ignored.

## Material Attributes

The current version of this plugin expects the .feb file to be in the FEBio 2.5 spec format.

## Initial Segment Velocity

Initial segment velocity may be specified. This determines how long the tips from initial fragments will be. In practice this means as long as an initial tip would not hit mesh boundary or a non angio material it will grow to this length. Note that this should be half the total fragment length desired since each tip is growing. The default value is 7.5 (total initial fragment length of 15).

e.g.

<initial\_segment\_velocity>**7.5**</initial\_segment\_velocity>

## Vessel Radius

Vessel radius specifies the radius of the vessels. This 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 occours)

e.g.

<vessel\_radius>**7**</vessel\_radius>

## Mix Method

Different methods may be used to determine a direction based on two other vector directions.

### Linear Interpolation Mix Method

The legacy method of linear interpolation between the components of the two input vectors is specified via



In the context of growth the new direction (psi) is linearly interpolated between the previous direction (psi) and the collagen fibril direction (theta). This leads to a nonlinear space of effective rotations and is no longer recommended but is still supported for legacy purposes. To use a linear interpolation of vector components include <mix\_method type= “LinInterp”></mix\_method> within the angio material.

### Linear Scaled Rotation Mix Method

The recommended method is a linear rotation or a scaling of the rotation from the first vector to the second within the plane spanned by both vectors. First the normal to the plane by the vectors is determined



The angle *φ* between the previous direction and influential direction is determined from the dot product then a rotation about the normal vector by the angle *φ* scaled by *α*:



This method guarantees that the relative change in direction is linear with respect to the scaling weight. To use a linear rotation between vectors within their shared plane include <mix\_method type=“LinRot”></mix\_method>.

## Initial Modifier

Initial Modifiers are a way to specify values to all angio elements within an angio material at the beginning of the simulation. Some examples of initial modifiers are: fiber orientation, ECM density, etc. Initial modifiers are contained within the <im\_manager> tag where each individual modification is contained within an <initial modifier> tag.

e.g.

<im\_manager type=**"im\_manager"**>

<initial\_modifier type=**"fiber\_randomizer"**>

</initial\_modifier>

<initial\_modifier type=**"density\_initializer"**>

<initial\_density>**3.0**</initial\_density>

</initial\_modifier>

</im\_manager>

### Fiber Randomizer

Multiplies the material orientation matrix by a random rotation matrix.

e.g.

<initial\_modifier type=**"fiber\_randomizer"**>

</initial\_modifier>

### EFD Fiber Randomizer

Samples a user defined ellipsoidal fiber distribution (spd i.e. symmetric positive-definite matrix) and assigns a disrete fiber to the material points based on this. The values are entered in the order (XX, YY, ZZ, XY, YZ, XZ).

e.g.

<initial\_modifier type=**"discrete\_fiber\_efd\_mat\_randomizer"**>

<spd>5,1,1,0,0,0</spd>

</initial\_modifier>

### Prescribed Fiber Orientation

Fibers may be specified by specifying a fixed material orientation from FEBio (Don’t use this with the fiber randomizer).

See FEBio Manual 4.1.1.1 or 4.1.1.2

### Density Initializer

Sets the density of all elements within an angio material to the specified value in mg/mL collagen. Note that the collagen density is always to be specified in units of mg/mL regardless of the chosen units for mass and distance. The collagen density is only used to scale the velocity of segments and not for mechanical or chemical simulations (see sections 5.9.4, 5.9.5).

e.g.

<initial\_modifier type=**"density\_initializer"**>

<initial\_density>**3.0**</initial\_density>

</initial\_modifier>

### Ellipsoidal Fibril Distribution Initializer

This prescribes an ellipsoidal fibril distribution (EFD) to each element. The same EFD is set to all elements. The EFD is prescribed as a symmetric positive-definite matrix (SPD). Due to symmetry only 6 unique values are needed. The values are entered in the order (XX, YY, ZZ, XY, YZ, XZ). Note that if a value other than a semi-positive definite matrix is entered the model may fail.

e.g.

<initial\_modifier type=**"EFD\_initializer"**>

<spd>3,1,1,0,0,0</spd>

</initial\_modifier>

## Node Data Interpolation

If you need finer control of variable used by AngioFE certain variables can be read in on a per node basis and then interpolated to the gauss points before the finite element simulation begins. This will be applied after the initial modifiers. Thus, initial modifiers can be overriden in places that deviate from the default. This is done by using a nodeset and a named mesh data section. The number of items in the nodeset and the NodeData sections must match. Node\_set\_id is the zero-indexed value of the nodeset in the order the nodesets appear in the .feb files. The name of NodeData must match what node\_interpolation value expects.

e.g.

<nodedata\_interpolation\_manager type=**"nodedata\_interpolation\_manager"**>

<node\_interpolation\_value type=**"ref\_ecm\_density"**>

<node\_set\_id>**1**</node\_set\_id>

</node\_interpolation\_value>

</nodedata\_interpolation\_manager>

**…**

<MeshData>

<NodeData name=**"ref\_ecm\_density"** node\_set=**"bc2"**>

<node lid=**"1"**>**4.0**</node>

<node lid=**"2"**>**4.0**</node>

</NodeData>

</MeshData>

### Reference ECM Density

Can be used to set the density values in the specified node set. The NodeData name this binds to is “ref\_ecm\_density”; This sets the reference density at the given nodes.

### Repulse Value

Can be used to set the repulse value on a per node basis. Unless some special behavior is desired set the repulse value to 1.0. An example of a special behavior would be repulsing part way through an element. The NodeData name this binds to is “repulse\_value”.

## Growth Velocity

The growth velocity manager is required in Angio Materials and defines the velocity of vessels within the material. The velocity modifiers are applied in the order that they are specified.

e.g.

<velocity\_manager type=**"segment\_growth\_velocity\_manager"**>

<velocity\_modifier type=**"segment\_velocity\_modifier"**>

<segment\_velocity\_over\_time>**140**</segment\_velocity\_over\_time>

</velocity\_modifier>

<velocity\_modifier type=**"segment\_velocity\_density\_scale\_modifier"**>

<interpolation\_prop type=**"per\_element\_vi"**></interpolation\_prop>

</velocity\_modifier>

</velocity\_manager>

### Segment Velocity Modifier

Allows the velocity of vessels to be set to a constant value, or have its velocity set via loadcurve.

e.g.

<velocity\_modifier type=**"segment\_velocity\_modifier"**>

<segment\_velocity\_over\_time>**140**</segment\_velocity\_over\_time>

</velocity\_modifier>

e.g.

<velocity\_modifier type=**"segment\_velocity\_modifier"**>

<segment\_velocity\_over\_time lc=”1”>**1**</segment\_velocity\_over\_time>

</velocity\_modifier>

### Sigmoid Segment Velocity Modifier

Prescribes growth per tip as a sigmoidal function given by:



Direct calculation is of the derivative of the sigmoidal curve given by



In the above equations, a is the scaling parameter equal to the total length a tip will grow, b controls the width of sigmoidal curve, and c controls the center of the sigmoidal curve where growth is at half of the total. The sigmoid policy is implemented as the following

e.g.

<velocity\_modifier type=**"sigmoid\_segment\_velocity"**>

<a>285</a>

<b>1.3</b>

<c>4.8</c>

</velocity\_modifier>

### Gompertz Segment Velocity Modifier

Prescribes growth per tip as a Gompertz function (asymmetric sigmoid) given by:



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



In the above equations a is the scaling parameter equal to the total 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 wrt time.

e.g.

<velocity\_modifier type=**"gompertz\_segment\_velocity "**>

<a>284</a>

<b>0.5</b>

<c>1</c>

<d>5</d>

</velocity\_modifier>

### Segment Growth Velocity Density Scale Modifier

Scales the segment velocity based on the density of where the current tip is. The scale is based on the current apparent density. This has been experimentally validated for cultures that do not experience external loading.

density\_scale = m\_density\_scale\_factor.x + m\_density\_scale\_factor.y \* exp(-m\_density\_scale\_factor.z \* density\_at\_point);

The values of density\_scale\_factor can be specified.

e.g.

<velocity\_modifier type=**"segment\_velocity\_density\_scale\_modifier"**>

<interpolation\_prop type=**"per\_element\_vi"**></interpolation\_prop>

</velocity\_modifier>

### Segment Growth Velocity Referential Density Scale Modifier

Scales the segment velocity based on the initial density of where the current tip is. This is currently recommended for cultures that are externally loaded since it is not known how these grow experimentally and compression/tension can greatly alter the apparent density without accurately reflecting the matrix structure.

density\_scale = m\_density\_scale\_factor.x + m\_density\_scale\_factor.y \* exp(-m\_density\_scale\_factor.z \* density\_at\_point);

The values of density\_scale\_factor can be specified.

e.g.

<velocity\_modifier type=**"segment\_velocity\_ref\_density\_scale\_modifier"**>

<interpolation\_prop type=**"per\_element\_vi"**></interpolation\_prop>

</velocity\_modifier>

## Previous Segment Contribution (PSC)

The portion of the growth represents the contribution of previous segment with respect to the direction of vascular growth.

### Previous Segment PSC

Overrides PSC as the direction of the previous segment.

e.g.

<psc\_modifier type=**"previous\_segment\_psc"**>

</psc\_modifier>

## Position Dependent Direction (PDD)

Represents the portion of vascular direction that is contributed by the environment surrounding the tip. All instances of PDD modifiers have a contribution parameter; this parameter determines how much the modifier overrides the previous PDD value. The default of contribution is 1.0 which represents completely overriding the previous value. A contribution of 0.0 will mean that the current PDD modifier will be ignored. Some PDD modifers have a Boolean parameter alpha\_override; alpha\_override if set the alpha (contribution mix parameter) will be set to the contribution value.

### Fiber PDD

Sets the PDD contribution to the local fiber direction. The <interpolation\_prop> tag determines how the values are interpolated within an element.

e.g.

<pdd\_modifier type=**"fiber\_pdd"**>

<interpolation\_prop type=**"per\_element\_vi"**></interpolation\_prop>

</pdd\_modifier>

### ECM Density Gradient PDD

If the norm of the density gradient is above the user specified threshold then set the PDD direction contribution to be perpendicular to the density gradient. Note that the density gradient will be in units of (mg/mL)/length where length is the length units chosen when creating the model geometry.

e.g.

<pdd\_modifier type=**"ecm\_density\_gradient\_pdd"**>

<interpolation\_prop type=**"per\_element\_vi"**></interpolation\_prop>

<threshold>**1e-3**</threshold>

<contribution>**1.0**</contribution>

<alpha\_override>**0**</alpha\_override>

</pdd\_modifier>

### Repulse PDD

Allows vessels to be directed away from the nodeset where this has values specified. Grad\_threshold determines whether the condition is based on the norm of the gradient or the value of repulse.

e.g.

<pdd\_modifier type=**"repulse\_pdd"**>

<interpolation\_prop type=**"per\_element\_vi"**></interpolation\_prop>

<alpha\_override>**1**</alpha\_override>

<grad\_threshold>**0**</grad\_threshold>

</pdd\_modifier>

### Concentration PDD

Allows vessels to follow concentration gradients. If the norm of the concentration gradient is above threshold for the solute then the tip direction will have a contribution of the direction of the concentration gradient.

e.g.

<pdd\_modifier type=**"concentration\_gradient\_pdd"**>

<threshold>**1e-3**</threshold>

<contribution>**1.0**</contribution>

<alpha\_override>**1**</alpha\_override>

<sol\_id>**0**</sol\_id>

</pdd\_modifier>

### Anastamosis PDD

Provides a way for tips to grow towards segments and once close enough to fuse into the segment.

Tips will not anastamose with segments that have grown from the same intial fragment. Anastamosis radius effects the radius at which the tip will start growing towards a valid segment. Tips will stop growing when: they are within the fuse radius of a tip (can be part of a grown segment), and the cos(angle) between the tips is less than fuse\_angle.

e.g.

<pdd\_modifier type=**"anastamosis\_pdd"**>

<anastamosis\_radius>**200**</anastamosis\_radius>

<fuse\_radius>**30**</fuse\_radius>

<fuse\_angle>**0.25**</fuse\_angle>

</pdd\_modifier>

## Contribution Mix(Alpha)

Specifies the mixture between the PDD and PSC vectors.

e.g.

<cm\_manager type=**"contribution\_mix\_manager"**>

<psc\_modifier type=**"psc\_pdd\_contribution\_mix"**>

<psc\_weight>**0.5**</psc\_weight>

</psc\_modifier>

</cm\_manager>

### PSC/PDD ContributionMix

Set the value of the mixture between the two direction components (will be multiplied by dt). PSC weight can be set to a load curve to change over time.

e.g.

<psc\_modifier type=**"psc\_pdd\_contribution\_mix"**>

<psc\_weight>**0.5**</psc\_weight>

</psc\_modifier>

## Interpolation Property

An interpolation property determines how values are interpolated from the integration points to all locations within an element. 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 elment 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 interpolates all of the members of a matrix to lerp between two matrices.(Lerp 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 Slerp) In the future it may be advantageous to implement a Interpolation Property based on SPR interpolation.

### Per Elment Variable Interpolation

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

e.g.

<velocity\_modifier type=**"segment\_velocity\_density\_scale\_modifier"**>

<interpolation\_prop type=**"per\_element\_vi"**></interpolation\_prop>

</velocity\_modifier>

## Tip Species Manager

The tip species manager is a feature that is currently under construction.

The tip species manager allows the user to assign solid bound molecule boundary conditions to the tips i.e. species can be represented as SBMs that move with active tips. This is currently stored per material at least for the purposes of initialization. SBMs can be converted to solutes or global (not owned by tips) SBMs from reactions specified in the material as well.

e.g.

<tip\_species\_manager type=”SBM\_Manager”>

<tip\_species\_prop type=”SBM”>

<SBM\_ID>1</SBM\_ID>

<SBM\_Production\_Rate>1e-2</SBM\_Production\_Rate>

</tip\_species\_prop>

</tip\_species\_manager>

This specifies that the tip contains SBM 1 and that the SBM concentration is 1e-2. A tip species property is required for each SBM.

## BranchPolicy

A BranchPolicy determines what BranchPoints are created and will do any modification of the branch points that is needed. A Branch Policy determines the distance between branches along a vessel, the time at which branches emerge, and the orientation of the new tip.

### Delayed Branching Policy

This policy is a simple way to determine where branch points occur. After each growth substep, if the amount of vascular growth exceeds the length to branch that has been rolled for the element, then a branch point is created and a time to emerge offset is rolled. This process is repeated until length to branch is greater than the amount of growth that has occured. This process is controlled by the probability distributions: length\_to\_branch, and time\_to\_emerge.

The direction that branch points that emerge at are controlled by the probability distributions: azimuth\_angle, and zenith\_angle.

e.g.

<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=**"uniform\_distribution"**>

<a>**0**</a>

<b>**10**</b>

<time\_clamped>**0**</time\_clamped>

</time\_to\_emerge>

</branch\_policy>

## Probability Distributions

The probability distributions are used as parameters for various parts of the branching model. The parameters of these distributions can be changed at any given mechanical point safely. The distributions will reroll any values that are produced that are below zero (branches cannot be decided retroactively, negative lengths are not allowed). If you want to be able to change the timestep size changes in probability distribution values should only occour at must points. Additionally, if timestep independence is desired the load curve must be set to use step interpolation.

### 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](https://en.wikipedia.org/wiki/Normal_distribution) [C++ Documentation](http://www.cplusplus.com/reference/random/normal_distribution/)

e.g.

<length\_to\_branch type=**"normal\_distribution"**>

<mean>**200**</mean>

<stddev>**5**</stddev>

</length\_to\_branch>

### 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](https://en.wikipedia.org/wiki/Uniform_distribution_(continuous)) [C++ Documentation](http://www.cplusplus.com/reference/random/normal_distribution/)

e.g.

<time\_to\_emerge type=**"uniform\_distribution"**>

<a>**0**</a>

<b>**10**</b>

<time\_clamped>**1**</time\_clamped>

</time\_to\_emerge>

### Exponential Distribution

The Exponential Distribution is another provided distrution. 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](https://en.wikipedia.org/wiki/Exponential_distribution) [C++ Documentation](http://www.cplusplus.com/reference/random/exponential_distribution/)

e.g.

<time\_to\_emerge type=**"exponential\_distribution"**>

<lambda>**0.5**</lambda>

<mult>**1**</mult>

</time\_to\_emerge>

### 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](https://en.wikipedia.org/wiki/Cauchy_distribution) [C++ Documentation](http://www.cplusplus.com/reference/random/cauchy_distribution/)

e.g.

<time\_to\_emerge type=**"cauchy\_distribution"**>

<a>**0.5**</a>

<b>**0.5**</b>

</time\_to\_emerge>

### Chi Squared Distribution

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

[Wikipedia Page](https://en.wikipedia.org/wiki/Chi-squared_distribution) [C++ Documentation](http://www.cplusplus.com/reference/random/chi_squared_distribution/)

e.g.

<time\_to\_emerge type=**"chi\_squared\_distribution"**>

<dof>**3.0**</dof>

<mult>**0.5**</mult>

</time\_to\_emerge>

### Weibull Distribution

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

[Wikipedia Page](https://en.wikipedia.org/wiki/Weibull_distribution) [C++ Documentation](http://www.cplusplus.com/reference/random/weibull_distribution/)

e.g.

<time\_to\_emerge type=**"weibull\_distribution"**>

<a>**2.0**</a>

<b>**0.5**</b>

</time\_to\_emerge>

### Gamma Distribution

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

[Wikipedia Page](https://en.wikipedia.org/wiki/Gamma_distribution) [C++ Documentation](http://www.cplusplus.com/reference/random/gamma_distribution/)

e.g.

<time\_to\_emerge type=**"gamma\_distribution"**>

<alpha>**2.0**</alpha>

<beta>**0.5**</beta>

</time\_to\_emerge>

### Matrix Elastic Material

This material models the effects of the matrix on the rest of the simulation. This must be a viscoelastic material. All of the inner parameters are from the underlying FEBio materials see FEBio User Manual Section 4.3.

e.g.

<matrix type=**"viscoelastic"**>

<t1>**0.00001**</t1>

<g0>**0.0**</g0>

<g1>**1.0**</g1>

<elastic type=**"EFD neo-Hookean"**>

<E>**0.00003452**</E>

<v>**0.**</v>

<beta>**2.5,2.5,2.5**</beta>

<ksi>**0.0003452,0.0003452,0.0003452**</ksi>

</elastic>

</matrix>

## Vascular Stress Modifiers

This material models the material response of the vascular network. To get the overall material properties of the angio material the final reponse is based on the weighted sum of the matrix submaterial and the vessel submaterial based on the volume ratio of vessels to ecm volume. Additionally, the tip force fields are contained in the angio stress component.

e.g.

<common\_properties type=**"angio\_properties"**>

<vessel type=**"viscoelastic"**>

<t1>**0.000005**</t1>

<g0>**0.0**</g0>

<g1>**1.0**</g1>

<elastic type=**"neo-Hookean"**>

<E>**0.003452**</E>

<v>**0.0**</v>

</elastic>

</vessel>

**…**

</common\_properties type=**"angio\_properties"**>

# Global Constants

The global constants for AngioFE go in the <Globals> <Constants> section in the .feb file. Many of the constants used to control the growth process may significantlty change when using a different element type.

## Seed

The seed is the seed for the underlying random engine. If this parameter is changed, the vascular network will change even if all other parameters remain the same. This parameter will be interpreted as an integer. A seed allows reproducibility of a given random network.

e.g.

<seed>**1393430476**</seed>

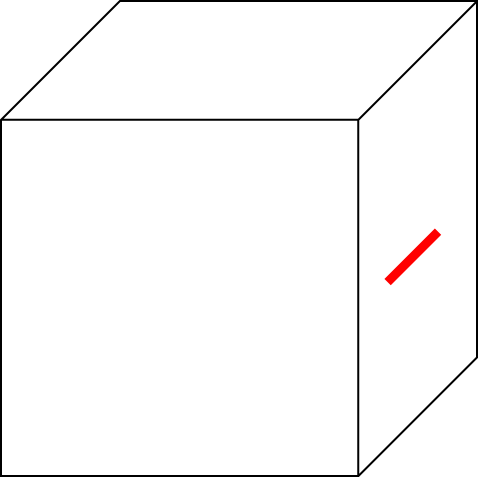
## Toggle IO

This 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.

e.g.

<no\_io>**1**</no\_io>

## Min Scale Factor

The min scale factor is used to determine if a potential segment that could grow is valid. Used to ensure that segments will not grow backwards due to raycasting roundoff. The segment will not grow back into an element if this value is set properly (recommended value of 0.01). If the scale factor is too high the vessel will become trapped in an element face rather than cross into a neighboring element (right).

e.g.

<min\_scale\_factor>**0**</min\_scale\_factor>

## Bounds Tolerance

The tolerance on natural coordinates in the growth process. Setting this value too low will result in vessels dying prematurely. If this value is too loose, segments may grow in incorrect elements.

e.g.

<bounds\_tolerance>**1e-2**</bounds\_tolerance>

## Min Angle

As tips reach the edges of an element there may be multiple elements in which it is valid for the tip to grow into. AngioFE checks the potential distance a vessel could grow in each possible element then selects the one that would allow the most growth in the current configuration based on this angle. Units are degrees.

e.g.

<min\_angle>**70**</min\_angle>

## Max Angio DT

This parameter is the maximum step size taken before the growth model will be updated. The default value is 0.25 days. This can be overridden in the control section e.g.

<max\_angio\_dt>**0.25**</max\_angio\_dt>

## Growth Substeps

The number of growth substeps that occur. This in some sense defines the maximum number of segments that may grow from a tip in a single timestep. For regular meshes this should be set to 3. If this is set to higher values than needed, the computations will take longer but the results will not change.

e.g.

<growth\_substeps>**3**</growth\_substeps>

# Output Data

## 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.

### Log

Statistics from this plugin are recorded in out\_log.csv this file can be opened by Excel. The values recorded are: Time, Material, Segments, Total Length, Vessels, Branch Points, Anastamosis, Active Tips, and Sprouts. Time is the time of this data. Material is the material id -1. Total Length is the total vessel length within this material. Vessels is the number of vessels within this material, this increases as branches are created and decreases as vessels fuse together due to anastomosis. Branch Points is the number of branches that have happened at the current time. Anastamosis 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.

### Vessel State

The file out\_vessel\_state.ang2 contains a record of the vascular network over time. To import this file in PostView click tools>Import lines and select the file. (make sure that the tools tab is enabled in PostView) If the version of PostView is new enough postview should load .ang2 files from partially completed runs.

### Final Vessel File

Contains data for each tip at the final time point recorded in final\_vessels.csv. This file contains data from the tips such as position. This will be expanded in the future to contain tip variables such as concentrations. It can be opened by the polar\_plot.py python script to generate orientation distribution functions for segment growth.

### XPLT

Some output options can be specified to show up in the heat maps in the .xplt file. These can be specified in the <Output><plotfile type="febio"> section within the .feb.file.

<var type=**"angio stress"**/>

This tag will output the stress that the vascular network creates.

<var type=**"vessel stress"**/>

This tag will output the stress that is from the vessel submaterial.

<var type=**"matrix stress"**/>

This tag will output the stress that is from the matrix submaterial.

<var type=**"vessel weight"**/>

This tag will output the weighting of the vessel material at each point.

<var type=**"matrix weight"**/>

This tag will output the weighting of the matrix material at each point.

<var type=**"matrix visco stress"**/>

This tag will output the matrix submaterial’s viscoelastic submaterial’s stress.

<var type=**"matrix elastic stress"**/>

This tag will output the matrix submaterial’s elastic submaterial stress.

<var type=**"branch\_count"**/>

This tag will output count of branches that have occoured within each element.

<var type=**"segment\_length"**/>

This tag will output the segment length in terms of the global coordinate system’s units for each element.

<var type=**"anastamoses"**/>

This tag will output the number of anastomoses within each element.

Note the fibers used in the simulation match the fibers used in FEBio so the tag to output them is from febio and is <var type=**"fiber vector"**/>

e.g.

<Output>

<plotfile type=**"febio"**>

<var type=**"displacement"**/>

<var type=**"stress"**/>

<var type=**"angio stress"**/>

<var type=**"vessel stress"**/>

<var type=**"matrix stress"**/>

<var type=**"vessel weight"**/>

<var type=**"matrix weight"**/>

</plotfile>

</Output>

## Vessel File Format 2

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 AngioFE3.

# Warnings

## Branch Bombing

It is possible to have the plugin continuously add branch points and then as those points grow add more branch points at an exponential rate. Thus, the ratio between length to branch (mean of this distribution) and vessel velocity should remain high. You will probably see runtimes increase exponentially as this ratio approaches 1.25:1. Trouble is virtually guaranteed if the ratio is 1:1.

# Sample Files

The sample files include examples that should represent the major features of AngioFE3.

## CIF\_SAL

An example of a large model. This is an example of a core in field model. The density only effects growth rate. There is no density gradient specified.

## Anastamosis

An example of growth with anastomosis turned on. Anastamosis will make tips that are sufficiently close to a segment grow towards that segment. When tips get within fuse radius of each other and the dot product of the tip’s directions is greater than fuse angle the tips will fuse.

## Bouncy

Shows of the repulse feature of AngioFE. Repulse is a way to specify a set of nodes from which vessels will be repulsed from. In this example vessels grow away from the bottom face of the model.

## Multi-Domian

Shows multiple domains. Vessels can grow between two angio materials.

## Density gradient

The density on the boundary between the materials is a slice of high ECM density. Each material handles the density gradient in a different way.

## Multiphasic

An Angio material is used as the solid component of a multiphasic material. There is a concentration gradient in the center portion of this model. In this example, the tips will grow along the concentration gradient.

## Tet4 Model

An example where vessels are growing through tet4 elements.

## Tet10 Model

An example using Tet10 elements.

## Hex20 Model

An example where vessels are growing through hex20 elements.

## Hex20 Cumulative Stress

This example uses hex20 elements, combined with a stress policy that allows the vascular network to continue to exert stress even after the tips have grown.

## Minimal Material

The following is a minimal material for AngioFE. This will not set the fiber direction or any direction, so all segments will grow the same direction for the entire time this runs. This example contains the minimum specification for the plugin to run. Refer to this example if you are encountering errors and are unsure what your angio material is missing.

<?xml version=**"1.0"** encoding=**"ISO-8859-1"**?>

<febio\_spec version=**"2.5"**>

<Material>

<material id=**"1"** name=**"Material01"** type=**"angio\_mat"**>

<angio\_stress\_policy type=**"load\_curve\_vel\_angio\_stress\_policy"**>

<sprout\_mag>**1e-13**</sprout\_mag>

</angio\_stress\_policy>

<velocity\_manager type=**"segment\_growth\_velocity\_manager"**>

</velocity\_manager>

<psc\_manager type=**"previous\_segment\_contribution\_manager"**>

</psc\_manager>

<pdd\_manager type=**"position\_dependent\_direction\_manager"**>

</pdd\_manager>

<cm\_manager type=**"contribution\_mix\_manager"**>

</cm\_manager>

<matrix type=**"viscoelastic"**>

<t1>**0.000005**</t1>

<g0>**0.0**</g0>

<g1>**1.0**</g1>

<elastic type=**"EFD neo-Hookean"**>

<E>**0.00003452**</E>

<v>**0.**</v>

<beta>**2.5,2.5,2.5**</beta>

<ksi>**0.0003452,0.0003452,0.0003452**</ksi>

</elastic>

</matrix>

<common\_properties type=**"angio\_properties"**>

<vessel type=**"viscoelastic"**>

<t1>**0.000005**</t1>

<g0>**0.0**</g0>

<g1>**1.0**</g1>

<elastic type=**"neo-Hookean"**>

<E>**0.0003452**</E>

<v>**0.0**</v>

</elastic>

</vessel>

<fragment\_seeder type=**"by\_element\_fragment\_seeder\_bidirectional"**>

<number\_fragments>**500**</number\_fragments>

</fragment\_seeder>

</common\_properties>

</material>

</Material>

</febio\_spec>

# Extending the plugin

Most new features of the plugin should be able to be implemented by creating a class that inherits from some existing class in AngioFE. The most common methods of extending the plugin cover: doing the same thing for all elements/integration points/nodes.(Initial Modifier), giving specific values for a set of nodes(Node Data Interpolation), modifying vascular growth(Growth Modifications), modifying the branching model(Branch Modifications), and modifying stress calculations(Stress Calculation Modification).

## Initial Modifier

These do the same thing for all elements within a given material.

## Node Data Interpolation

Allows the user to specify values on a per node basis. These values will then be interpolated down to the integration points before the simulation begins solve steps.

## Growth Modifications

Most modifications to tip growth should inherit from one of: ContributionMix, SegmentGrowthVelocity, PositionDependentDirection, and PreviousSegmentContribution. As long as all modifications to tip velocity are done through subclasses of SegmentGrowthVelocity the autostep functionality of the plugin will function correctly.

## Branching Modifications

A class can implement BranchPolicy to determine where and when segments will branch. Two possible examples of using this are: an immediate branch mode (branch chance per length of segment where branches appear immediately in a timestep dependent manner), or delayed environment sensitive branching (should the conditions in the element change significantly, reroll the locations and emerge times of future branches).

## Stress Calculation Modifications

How stress calculations are done can be modified by creating a subclass of AngioStressPolicy. An example of a stress modification could be implementing a stress calculation that depends exclusively on the tips in grown segments.

## Boundary Conditions

In the near future it may be advanagous to add some boundary conditions to the plugin. Currently FEBio needs to be updated to be able to parse Boundary conditions that are defined in plugins.(12/3/2018) One likely boundary condition to add is one that allows tips to modify the concentration of solutes within a multiphasic material. This boundary condition would likely inherit from FEPrescribedDOF and implement the Update() method to calculate the change in chemical concentration based on some function of the tips that reside within the elements that contain a given node.

## Cautions

The calculation of dtmin is entirely dependent on element side length and tip velocity. As long as all changes to vessel velocity are done in subclasses of SegmentGrowthVelocity nothing should need to be changed in the calculation of the largest possible timestep that AngioFE allows.

## Random Distributions

The random distributions provided by the c++ standard do not have a const operator(). This means that the distribution itself is mutable. If there are problems with random numbers, they may be related to this. Consider finding/writing a library where random numbers can be generated by only depending on the random engine.