**AngioFE**

*Version 3.0*

**User’s Manual**

**Last Updated: April 1, 2022**

**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>

**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**

[1. Introduction 1](#_Toc99726071)

[1.1. Overview of AngioFE 1](#_Toc99726072)

[1.2. Capabilities 1](#_Toc99726073)

[1.2.1. Usage as a Standalone Material 1](#_Toc99726074)

[1.2.2. Doxygen Documentation 1](#_Toc99726078)

[1.3. Structure of this Document 1](#_Toc99726082)

[2. Terminology 2](#_Toc99726084)

[2.1. General terms 2](#_Toc99726085)

[2.1.1. Extracellular Matrix (ECM, matrix) 2](#_Toc99726086)

[2.1.2. Linear Elements 2](#_Toc99726087)

[2.1.3. Higher Order Elements 2](#_Toc99726088)

[2.2. Vascular Network 2](#_Toc99726089)

[2.2.1. Initial Fragment 3](#_Toc99726090)

[2.2.2. Tip 3](#_Toc99726091)

[2.2.3. Stalk 3](#_Toc99726092)

[2.2.4. Segment 3](#_Toc99726093)

[2.2.5. Vessel 3](#_Toc99726094)

[2.3. Branching 3](#_Toc99726096)

[2.3.1. Branch Point 4](#_Toc99726097)

[2.3.2. Zenith Angle 4](#_Toc99726098)

[2.3.3. Azimuth Angle 4](#_Toc99726099)

[2.4. Anastamosis 5](#_Toc99726100)

[3. Theory & Implementation Details 6](#_Toc99726101)

[3.1. Segment Growth Theory 6](#_Toc99726102)

[3.1.1. Growth Velocity 6](#_Toc99726103)

[3.1.2. Position Dependent Direction (PDD) 6](#_Toc99726104)

[3.1.3. Previous Segment Contribution (PSC) 6](#_Toc99726105)

[3.1.4. Alpha (Contribution Mix) 6](#_Toc99726106)

[3.2. AutoTimestep Adjustment 7](#_Toc99726113)

[3.3. Active Stress due to Vessel Growth 7](#_Toc99726114)

[4. Internal Theory 9](#_Toc99726115)

[4.1. Vascular Growth 9](#_Toc99726116)

[4.2. Parent Vessel Initialization Step (Proto Growth) 9](#_Toc99726117)

[5. Running FEBio+AngioFE 11](#_Toc99726118)

[5.1. Configuring FEBio to find the Plugin 11](#_Toc99726119)

[5.2. Running a Simulation with the Plugin 11](#_Toc99726120)

[6. Free Format Input 12](#_Toc99726121)

[6.1. Free Format Overview 12](#_Toc99726122)

[6.2. Module Section 12](#_Toc99726123)

[6.3. Globals Section 12](#_Toc99726124)

[6.3.1. Constants 12](#_Toc99726125)

[6.3.1.1. Random Seed 12](#_Toc99726126)

[6.3.1.2. Toggle IO 13](#_Toc99726127)

[6.3.1.3. Min Scale Factor 13](#_Toc99726128)

[6.3.1.4. Bounds Tolerance 13](#_Toc99726129)

[6.3.1.5. Min and Max Angio DT 13](#_Toc99726130)

[6.3.1.6. Growth Substeps 13](#_Toc99726131)

[6.3.1.7. Bounce Condition 14](#_Toc99726132)

[6.4. Materials 14](#_Toc99726133)

[6.4.1. Constitutive Models 14](#_Toc99726134)

[6.4.2. Mix Method 15](#_Toc99726135)

[6.4.3. Vessel Radius 15](#_Toc99726136)

[6.5. Initial Modifier Manager 16](#_Toc99726137)

[6.5.1. Density Initializer 16](#_Toc99726138)

[6.5.2. Discrete Fiber Randomizer 16](#_Toc99726139)

[6.5.3. Discrete Fiber EFD Randomizer 16](#_Toc99726140)

[6.5.4. EFD Initializer 16](#_Toc99726141)

[6.5.5. EFD Rotated Initializer 17](#_Toc99726142)

[6.6. Angio Stress Policies 17](#_Toc99726143)

[6.6.1. Sigmoid Angio Stress Policy 17](#_Toc99726144)

[6.6.2. Sigmoid Density Angio Stress Policy 18](#_Toc99726145)

[6.6.3. Load Curve Angio Stress Policy 18](#_Toc99726146)

[6.6.4. Load Curve Density Scaled Angio Stress Policy 18](#_Toc99726147)

[6.6.5. Load Curve Referential Density Scaled Angio Stress Policy 19](#_Toc99726148)

[6.7. Segment Velocity Manager 19](#_Toc99726149)

[6.7.1. Segment Velocity Over Time 19](#_Toc99726150)

[6.7.2. Segment Velocity Density Scale Modifier 19](#_Toc99726151)

[6.7.3. Segment Velocity Referential Density Scale Modifier 20](#_Toc99726152)

[6.7.4. Segment Velocity Density & Fractional Anisotropy Scale Modifier 20](#_Toc99726153)

[6.7.5. Sigmoid Segment Velocity Modifier 20](#_Toc99726154)

[6.7.6. Sigmoid Adjusted Segment Velocity Modifier 21](#_Toc99726155)

[6.7.7. Gompertz Segment Velocity Modifier 22](#_Toc99726156)

[6.8. Previous Segment Contribution (PSC) Manager 22](#_Toc99726157)

[6.8.1. Previous Segment PSC Modifier 23](#_Toc99726158)

[6.9. Proto Previous Segment Contribution (PSC) Manager 23](#_Toc99726159)

[6.9.1. Proto Previous Segment PSC Modifier 23](#_Toc99726160)

[6.10. Position Dependent Direction (PDD) Manager 23](#_Toc99726161)

[6.10.1. Fiber PDD 23](#_Toc99726162)

[6.10.2. Fractional Anisotropy PDD 24](#_Toc99726163)

[6.10.3. Lagrange Principal PDD 24](#_Toc99726164)

[6.10.4. Anastomosis PDD 24](#_Toc99726165)

[6.11. Proto Position Dependent Direction (PDD) Manager 24](#_Toc99726166)

[6.11.1. Proto Fiber PDD 25](#_Toc99726167)

[6.11.2. Proto Fractional Anisotropy PDD 25](#_Toc99726168)

[6.11.3. Lagrange Principal PDD 25](#_Toc99726169)

[6.11.4. Proto Anastomosis PDD 25](#_Toc99726170)

[6.12. Contribution Mix Manager 26](#_Toc99726171)

[6.12.1. PSC-PDD Contribution Mix 26](#_Toc99726172)

[6.12.2. Density-Fractional Anisotropy Contribution Mix 26](#_Toc99726173)

[6.13. Proto Contribution Mix Manager 26](#_Toc99726174)

[6.13.1. Proto PSC-PDD Contribution Mix 27](#_Toc99726175)

[6.14. Common Properties 27](#_Toc99726176)

[6.14.1. Vessel Material Properties 27](#_Toc99726177)

[6.14.2. Fragment Seeder 27](#_Toc99726178)

[6.14.2.1. Initial segment length 27](#_Toc99726179)

[6.14.2.2. By Element Fragment Seeder Bidirectional 28](#_Toc99726180)

[6.14.2.3. By Volume Fragment Seeder Bidirectional 28](#_Toc99726181)

[6.15. Interpolation Property 28](#_Toc99726402)

[Per Elment Variable Interpolation 29](#_Toc99726403)

[6.15.1. 29](#_Toc99726404)

[6.16. Cell Species Manager 29](#_Toc99726405)

[6.17. Cell Reaction Manager 29](#_Toc99726406)

[6.18. Branch Policy 30](#_Toc99726407)

[6.18.1. Delayed Branching Policy 30](#_Toc99726408)

[6.19. Probability Distributions 30](#_Toc99726409)

[6.19.1. Normal Distribution 31](#_Toc99726410)

[6.19.2. Uniform Distribution 31](#_Toc99726411)

[6.19.3. Exponential Distribution 31](#_Toc99726412)

[6.19.4. Cauchy Distribution 31](#_Toc99726413)

[6.19.5. Chi Squared Distribution 32](#_Toc99726414)

[6.19.6. Weibull Distribution 32](#_Toc99726415)

[6.19.7. Gamma Distribution 32](#_Toc99726416)

[6.19.8. Fixed Distribution 32](#_Toc99726417)

[6.19.9. Prescribed Distribution 33](#_Toc99726418)

[7. Output Data 34](#_Toc99726483)

[7.1. Files 34](#_Toc99726484)

[7.1.1. Angio Log: out\_log.csv 34](#_Toc99726485)

[7.1.2. Vessel State: out\_vessel\_state.ang2 34](#_Toc99726486)

[7.1.3. Final Vessel File: final\_vessels.csv 34](#_Toc99726487)

[7.1.4. Cell State: final\_cells.txt 34](#_Toc99726488)

[7.1.5. Time statistics: angio\_stats.csv 34](#_Toc99726489)

[7.1.6. XPLT 35](#_Toc99726490)

[7.2. Vessel File Format 2 35](#_Toc99726536)

[Sample Files 37](#_Toc99726537)

[7.3. CIF\_SAL 37](#_Toc99726538)

[7.4. Anastamosis 37](#_Toc99726539)

[7.5. Bouncy 37](#_Toc99726540)

[7.6. Multi-Domian 37](#_Toc99726541)

[7.7. Density gradient 37](#_Toc99726542)

[7.8. Multiphasic 37](#_Toc99726543)

[7.9. Tet4 Model 37](#_Toc99726544)

[7.10. Tet10 Model 37](#_Toc99726545)

[7.11. Hex20 Model 38](#_Toc99726546)

[7.12. Hex20 Cumulative Stress 38](#_Toc99726547)

[8. Default Parameters 39](#_Toc99726548)

[9. References 40](#_Toc99726549)

# Introduction

## 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](#_ENREF_1" \o "Edgar, 2013 #14)].

## 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.

### 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.

### 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

## 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 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.

## General terms

General purpose terminology.

### Extracellular Matrix (ECM, matrix)

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

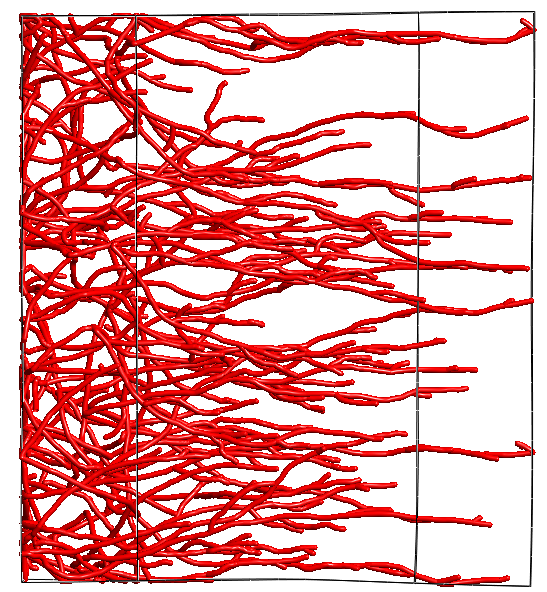
### Linear Elements

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

### Higher Order Elements

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

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

### 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.

### 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.

### Segment

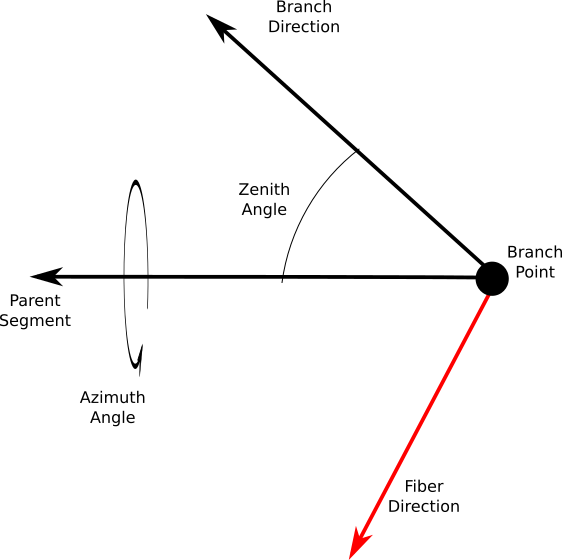
A discretized portion of a vessel this is represented by the plugin as a line segment with respect to the natural coordinate system of the element that contains it. A segment is contained entirely within a single finite element (multiple segments will be used if a vessel crosses between elements). Multiple 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.

## 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.



### 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.

### Zenith Angle

The angle of departure from the parent segment.

### Azimuth Angle

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

## Anastamosis

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 anastamosis 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.

# 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.

## 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 ***ψ*** 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 ***θ***. During each time step the new direction ***ψ****new* is calculate by partially rotating ***ψ*** to ***θ***. The rotation between the directions is scaled by the contribution mix *α.*

### 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.

### 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 ***θ*** has been tested although there is support for guidance by other state variables.

### Previous Segment Contribution (PSC)

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

### Alpha (Contribution Mix)

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

## 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. 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.

## Active Stress due to Vessel Growth

This section details the traction stress policies that can be implemented. In general, the stress field ***σ****s*(***x***) at a position ***x*** is calculated based on ***r*** which is defined as the distance vector between ***x*** and the location of the vessel tip ***x****s*



The angle between the vessel direction ***ψ*** and the distance vector ***r*** is indicated by *φ*. Stresses are applied at vessel tips as body forces. The resulting stress due to the body force is given by



The components of the stress field are detailed below.

nitude (*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 ***ψ***. 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 controls how the stress field dissipates with radial distance *r* from the sprout tip. The default value for this is 200 μm.

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.











# 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.

## Parent Vessel Initialization Step (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

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

## Configuring FEBio to find the Plugin

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">

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

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

</febio\_config>

Linux example configuration file:

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

<febio\_config version="3.0">

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

<import>C:\...\AngioFE3.so</import>

</febio\_config>

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

## Running a Simulation with the Plugin

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

# 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.

## 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.

## 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”/>

## 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 but can be modified by the user as needed.

### Constants

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

#### 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.

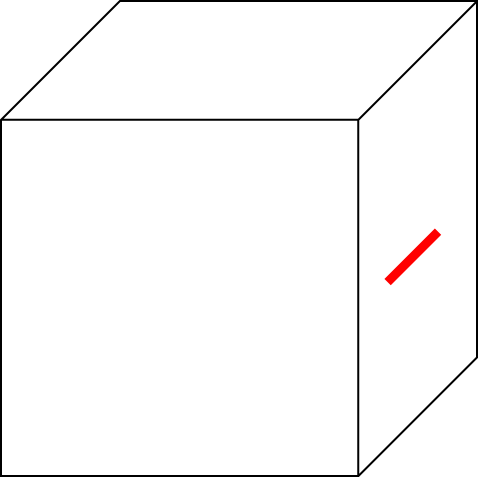
<seed>1393430476</seed>

#### 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>

#### 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., < 0.1) 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>0.1</min\_scale\_factor>

#### 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 elemtn 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>

#### 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>

#### 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>

#### Bounce Condition

The bounce condition handles how vessels behave when they encounter an external face of the model. Vessels can either 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 be balanced out by growth of vessels 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>

## 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>

### 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 contains the definition of the solid portion of the mechanical properties of the matrix which are stored in a matrix tag. The solid tag also contains the properties of the vessels as well as common angio properties that are shared externally with AngioFE and FEBio.

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>

<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>

<!-- the rest of the specification of parameters relevant to the growth model go here -->

</material>

### Mix Method

The mix method determines how ***ψ****new* is calculated from ***ψ*** and ***θ***. Previously, this method was calculated as a weighted sum of two vectors with the weight *α:*

.

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 ***ψ*** and ***θ*** does not vary linearly with *α*. To overcome this, a new method is used which scales the rotation of ***ψ*** to ***θ*** with the rotation matrix ***R***

.

In both cases, setting *α* = 0.0 results in ***ψ****new* = ***ψ*** and setting *α* = 1.0 results in ***ψ****new*= ***θ***.

The legacy approach can be used by specifying:

<mix\_method type=”LinInterp”/>

The default method can be used by specifying :

<mix\_method type=”LinRot”/>

### 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 occours)

<vessel\_radius>6.3</vessel\_radius>

## 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>

### 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>

### 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”/>

### Discrete Fiber EFD Randomizer

The discrete fiber EFD randomizer generates a vector field of collagen directions derived from a user-provied 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>

### 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>

### 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 allows the users to subtly vary the orientation as desired. 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>

## 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.

### 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



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>

### 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



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>

### 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 loadcurve 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>

### 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](#_ENREF_4" \o "Edgar, 2015 #22)].

<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>

### 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>

## 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>

### 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>

### Segment Velocity Density Scale Modifier

The segment velocity density scale modifier scales the segment growth rate by *ν*(***x***) which is based on the local matrix density. The scale is defined previously [[4](#_ENREF_4" \o "Edgar, 2015 #22)] as

.

Here, *ρ*(***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>

### Segment Velocity Referential Density Scale Modifier

The segment velocity referential density scale modifier scales the segment growth rate by *ν*(***x***) which is based on the local initial matrix density. The scale is defined previously [[4](#_ENREF_4" \o "Edgar, 2015 #22)] as

.

Here, *ρ*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>

### 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 *ν*(***x***,*ρ*,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.

.

Here, *ρ*1/2 is the center of the Lorentzian function along the *ρ* axis and FA1/2 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 *ρ* 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>

### 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.

.

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

.

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>

### 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

.

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 *sa* which is defined as

.

Here, *qs* is the ratio of the current vessel volume fraction *ws* to the threshold volume fraction *wthresh* which has a minimum of 1.0:

.

The values *as*, *bs*, and *s*0 were selected so that when the vessel volume fraction was below the threshold *wthresh, sa* = 1.0. Once *qs* > 1.0, *sa* rapidly decreases until *sa* = *s*0. The parameters *as,* *qs*, *bs*, *s*0, and *wthresh* 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>

### Gompertz Segment Velocity Modifier

The Gompertz segment velocity modifier prescribes growth 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 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>

## Previous Segment Contribution (PSC) Manager

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

<psc\_manager type=“previous\_segment\_contribution\_manager”>

…

</psc\_manager>

### Previous Segment PSC Modifier

The previous segment PSC modifier is the default PSC modifier. This returns the prior direction ***ψ*** that a vessel grew.

<psc\_modifier type=”previous\_segment\_psc”/>

## Proto Previous Segment Contribution (PSC) Manager

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

<proto\_psc\_manager type=“previous\_segment\_contribution\_manager”>

…

</proto\_psc\_manager>

### Proto Previous Segment PSC Modifier

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

<proto\_psc\_modifier type=”proto\_previous\_segment\_psc”/>

## 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>

### 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>

### 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”/>

### 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 1st 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>

### Anastomosis PDD

Anastamoses 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 anastamosis 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="anastamosis\_pdd">

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

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

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

</pdd\_modifier>

## 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>

### 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>

### 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>

### 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 1st principal Lagrange strain.

<pdd\_modifier type=”lagrange\_principal\_pdd”/>

### Proto Anastomosis PDD

Anastamoses 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 anastamosis 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\_anastamosis\_pdd">

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

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

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

</proto\_pdd\_modifier>

## Contribution Mix Manager

The contribution mix manager handles how the parameter *α* is set when calculating the mixture between two directions such as ***ψ*** and ***θ***. 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 *α.*

### PSC-PDD Contribution Mix

The PSC-PDD contribution mix modifier allows the user to specify the value of *α*.

<cm\_manager type=”contribution\_mix\_manager>

<psc\_modifier type=”psc\_pdd\_contribution\_mix”>

<psc\_weight>*α*</psc\_weight>

</psc\_modifier>

</cm\_manager>

### Density-Fractional Anisotropy Contribution Mix

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

.

In the above equation, *α*0 is the minmum value of *α*, *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>

## Proto Contribution Mix Manager

The proto contribution mix manager handles how the parameter *α* is set during the initialization step when calculating the mixture between two directions such as ***ψ*** and ***θ***. 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 *α.* This also allows the value of α to differ during the initialization step used to generate the initial microvessel fragments.

### Proto PSC-PDD Contribution Mix

The proto PSC-PDD contribution mix modifier allows the user to specify the value of *α* during the initialization step.

<proto\_cm\_manager type=”proto\_contribution\_mix\_manager>

<proto\_psc\_modifier type=”proto\_psc\_pdd\_contribution\_mix”>

<proto\_psc\_weight>*α*</proto\_psc\_weight>

</proto\_psc\_modifier>

</proto\_cm\_manager>

## 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>

### 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>

### 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 (*nfrag*) and the initial segment length.

#### 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 loadcurve by using the “prescribed\_distribution” distribution type. For the latter option, the loadcurve points should be formatted as:

<point>length,probability</point>

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

#### 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>

#### 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>

## 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.

### 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>

## 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 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 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>

## 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 moledules 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.

## 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>

## 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/)

<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/)

<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/)

<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/)

<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/)

<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/)

<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/)

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

<alpha>2.0</alpha>

<beta>0.5</beta>

</time\_to\_emerge>

### 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>

### 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>

# 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.

### 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, Anastamosis, 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. 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: 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.

### 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.

### 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.

### Time statistics: angio\_stats.csv

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

### XPLT

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. |

## 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 AngioFE.

# Sample Files

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

## 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.

# 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 μm |  |  |
|  | Sprout radius multiplier | <sprout\_radius\_multiplier> | 3 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 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.