A picture containing food, drawing

Description automatically generated

Department of Mechanical Engineering  
FACULTY OF ENGINEERING AND DESIGN

**FINAL YEAR Meng PROJECT REPORT**

*Generating synthetic coronary vessel trees for simulating blood flow in the heart*

*Theodore Lloyd Morgan*

*13/05/2020*

A close up of a logo

Description automatically generated

“I certify that I have read and understood the entry in the Student Handbook for the Department of Mechanical Engineering on Cheating and Plagiarism and that all material in this assignment is my own work, except where I have indicated with appropriate references.”



Author’s Signature: …………………………………

Supervisor: Andrew Cookson

Assessor: Steve Cayzer

## Summary

Computer generated vessel trees offer an innovative approach to improving the diagnosis of cardiovascular diseases through the extension of vessel trees imaged from CT angiography. The synthetically generated vessels extend the coronary arteries from the CT images down to the arteriole level allowing for improved blood flow simulations.

In this report a method to simulate realistic coronary vascular growth within a defined 2-dimensional area has been proposed and assessed. The method aims to minimise the total volume of the network whilst abiding by certain physical principles. These include angle rules to dictate the direction of vessel growth, as well as intersection rules to prevent vessels from growing within each other. This method has been used to generate vessel trees of up to 4000 terminal segments in an average time of 2 hours. The proposed method has been assessed against other leading research within the field as well as morphometric data taken from porcine hearts. In comparison to the leading research the model fits the shape of diameter plots against bifurcation level and Strahler order. However, discrepancies within the initial vessels are apparent due to inaccuracy within the seeding data. As well as this consistently the model’s vessels are smaller than that of stated literature likely due to the assumptions made within the pressure drop across vessels within the method.

Due to these discrepancies improvements have been suggested for future work. With these improvements to the method it will produce an accurate representation of vascular growth of the coronary arteries within a computation time feasible for clinical diagnoses.

## Acknowledgements

The author would like to thank Dr Andrew Cookson for his support and guidance throughout the project and constant assistance with development. Dr Steve Cayzer for providing insight into possible optimisation methods and planning of work timeline. Erica Jane Morgan for inspiring me after her bypass operation to take up this project to assist in any way I can in fighting cardiovascular disease.

Contents

[Summary i](#_Toc40241117)

[Acknowledgements i](#_Toc40241118)

[List of Figures iii](#_Toc40241119)

[List of Tables iv](#_Toc40241120)

[Introduction 1](#_Toc40241121)

[Literature Review 3](#_Toc40241122)

[Computer-Optimisation of Vascular trees Wolfgang Schreiner and Peter Franz Buxbaum, 1993 3](#_Toc40241123)

[Generation of patient-specific cardiac vascular networks – 2018 6](#_Toc40241124)

[Development of a Model of the Coronary Arterial Tree for the 4-D XCAT Phantom 7](#_Toc40241125)

[Murray’s Law 8](#_Toc40241126)

[Summary of Literature 8](#_Toc40241127)

[Aim and Objectives 9](#_Toc40241128)

[Method 10](#_Toc40241129)

[Key Terms 10](#_Toc40241130)

[Vessel Tree Seeding and Data Structure 11](#_Toc40241131)

[Generating Points 12](#_Toc40241132)

[Connection Optimisation 13](#_Toc40241133)

[Results and Discussion 16](#_Toc40241134)

[Parameter Tests and Study 16](#_Toc40241135)

[Investigation 18](#_Toc40241136)

[Primary Results 19](#_Toc40241137)

[Future Work 32](#_Toc40241138)

[Conclusions 33](#_Toc40241139)

[References 34](#_Toc40241140)

[Appendix 35](#_Toc40241141)

[Appendix 1.1 Diameters and lengths of vessel segments and elements in each order of vessels in pig RCA (G S Kassab, 1993) 35](#_Toc40241142)

[Appendix 1.2 Diameters and lengths of vessel segments and elements in each order of vessels in pig LAD (G S Kassab, 1993) 35](#_Toc40241143)

[Appendix 1.3 Diameters and lengths of vessel segments and elements in each order of vessels in pig RCX (G S Kassab, 1993) 36](#_Toc40241144)

[Appendix 1.4 Patient data set 1 36](#_Toc40241145)

[Appendix 1.5 Patient data set 2 36](#_Toc40241146)

[Appendix 1.6 Patient data set 3 37](#_Toc40241147)

[Appendix 1.7 Patient data set 4 38](#_Toc40241148)

[Appendix 1.8 Main Tree Generation Script 38](#_Toc40241149)

[Appendix 1.9 Strahler order Script 47](#_Toc40241150)

[Appendix 1.10 Function pointcheck 50](#_Toc40241151)

[Appendix 1.11 Function LineIntersectPar 51](#_Toc40241152)

[Appendix 1.12 Function LineIntersectDau 51](#_Toc40241153)

[Appendix 1.13 Function CircBound 51](#_Toc40241154)

[Appendix 1.14 Function Circle 51](#_Toc40241155)

## List of Figures

[Figure 1: Share of deaths by cause in the world 2017 (IHME, 2017) 1](#_Toc40241156)

[Figure 2: Process Diagram from Imaging to results 2](#_Toc40241157)

[Figure 3: Visual plot of tree with 4000 terminal segments 3](#_Toc40241158)

[Figure 4 : Average Segment Diameter at each bifurcation level 4](#_Toc40241159)

[Figure 5: Pressure profile of model against feline heart 4](#_Toc40241160)

[Figure 6: Results from Schreiner's 1999 Paper (Wolfgang Schreiner R. K., 1999) 5](#_Toc40241161)

[Figure 7: Visual representation of tree development within (Wolfgang Schreiner R. K., 1999) 5](#_Toc40241162)

[Figure 8: Illustration of forest growth. (a); inputs provided (b): distributing ﬁrst segment on left ventricle surface for each tree. (c): surface growth (stage 1) reaching 1000 terminal segments. (d): inner growth (stage 2) reaching 6000 terminal segments (Clara Jaquet, 2018) 6](#_Toc40241163)

[Figure 9: Results from two patient cases 7](#_Toc40241164)

[Figure 10: Results From fractal coronary vessel generation (George S K Fung, 2011) 7](#_Toc40241165)

[Figure 11: Diagram depicting relevant terms 10](file:///C:\Users\user\Desktop\Fyp%20Final.docx#_Toc40241166)

[Figure 12: Patient seeding data visual plot 11](#_Toc40241167)

[Figure 13: 250 Terminal points visual plot 12](#_Toc40241168)

[Figure 14: 250 Terminal points connected visual plot 12](#_Toc40241169)

[Figure 15: Connection optimisation method 14](#_Toc40241170)

[Figure 16: Angle Constraints applied to model 14](#_Toc40241171)

[Figure 17 Murrays 2.55 17](#_Toc40241172)

[Figure 18: Murrays 3 17](#_Toc40241173)

[Figure 19: Murrays Range 18](#_Toc40241174)

[Figure 20 A: conventional Strahler ordering scheme, B: Diameter defined Strahler order system (G S Kassab, 1993) 19](#_Toc40241175)

[Figure 21: 4000 Terminal segment Tree 20](#_Toc40241176)

[Figure 22: Plot of average diameter against Strahler order with mean upper and lower bound presented 20](#_Toc40241177)

[Figure 23: Plot of average length against Strahler order with mean upper and lower bound presented 20](#_Toc40241178)

[Figure 24: 4000 Terminal Segments Tree (Wolfgang Schreiner P. B., 1993) 21](file:///C:\Users\user\Desktop\Fyp%20Final.docx#_Toc40241179)

[Figure 25: 4000 Terminal segment Tree 21](file:///C:\Users\user\Desktop\Fyp%20Final.docx#_Toc40241180)

[Figure 26: Plot against Schreiners Model 22](#_Toc40241181)

[Figure 27: Comparison with data from (Chee, 1987) 23](#_Toc40241182)

[Figure 28: 6000 Segment 3D Visual plot (Clara Jaquet, 2018) 24](#_Toc40241183)

[Figure 29: Comparison of LAD vs Clara et al 25](#_Toc40241184)

[Figure 30: 4000 Terminal points model against morphometric data diamters 26](#_Toc40241185)

[Figure 31: 4000 Terminal points model against morphometric data lengths 27](#_Toc40241186)

[Figure 32: Patient data seeding 1 29](#_Toc40241187)

[Figure 33: Patient 1 Results 29](#_Toc40241188)

[Figure 34 Patient data seeding 2 29](#_Toc40241189)

[Figure 35: Patient 2 Results 29](#_Toc40241190)

[Figure 36: Patient data seeding 3 29](#_Toc40241191)

[Figure 37: Patient 3 Results 29](#_Toc40241192)

[Figure 38: Patient data seeding 4 30](#_Toc40241193)

[Figure 39: Patient 4 Results 30](#_Toc40241194)

[Figure 40: Average segment diameter of each patient generation 30](#_Toc40241195)

[Figure 41: Computation time for with various physical constraints removed 31](#_Toc40241196)

## List of Tables

[Table 1: Data needed for seeding segments with Segment 1,2,3 and 21 shown 11](#_Toc40241197)

[Table 2: Equation variables 13](#_Toc40241198)

[Table 3: Results of Point Selection Parameter Investigation with N = 250 16](#_Toc40241199)

[Table 4: Parameters used in generation 21](#_Toc40241200)

[Table 5: Parameters Used in generation 26](#_Toc40241201)

## Introduction

Coronary artery disease is currently the leading cause of death worldwide. Improvement to diagnostic tools is therefore critical in the fight against it. Currently medical imaging of the heart is limited by the time taken to produce the image and the image resolution as well as availability of equipment. Commonly computed tomography scans (CT) are used however, this imaging modality is limited to a resolution of 0.5mm which can only view the initial coronary arteries (Akmal Sabarudinm, 2013) (Jesionek, 2017). There is potential however to use computer generated models in conjunction with CT scans to provide greater detail to imaging whilst still providing a fast and accessible diagnoses. Other imaging techniques can be used on the coronary arteries however lack the resolution of high-end non-invasive CT scans whilst also being far less commonly used. For example, echocardiography can achieve a resolution of around 2mm (Marek Krzanowski, 2003), MRI around 1.5mm to 5.5mm (al, 2007) and PET with around 5mm - 6.3mm (Marcelo F. Di Carli, 2011).

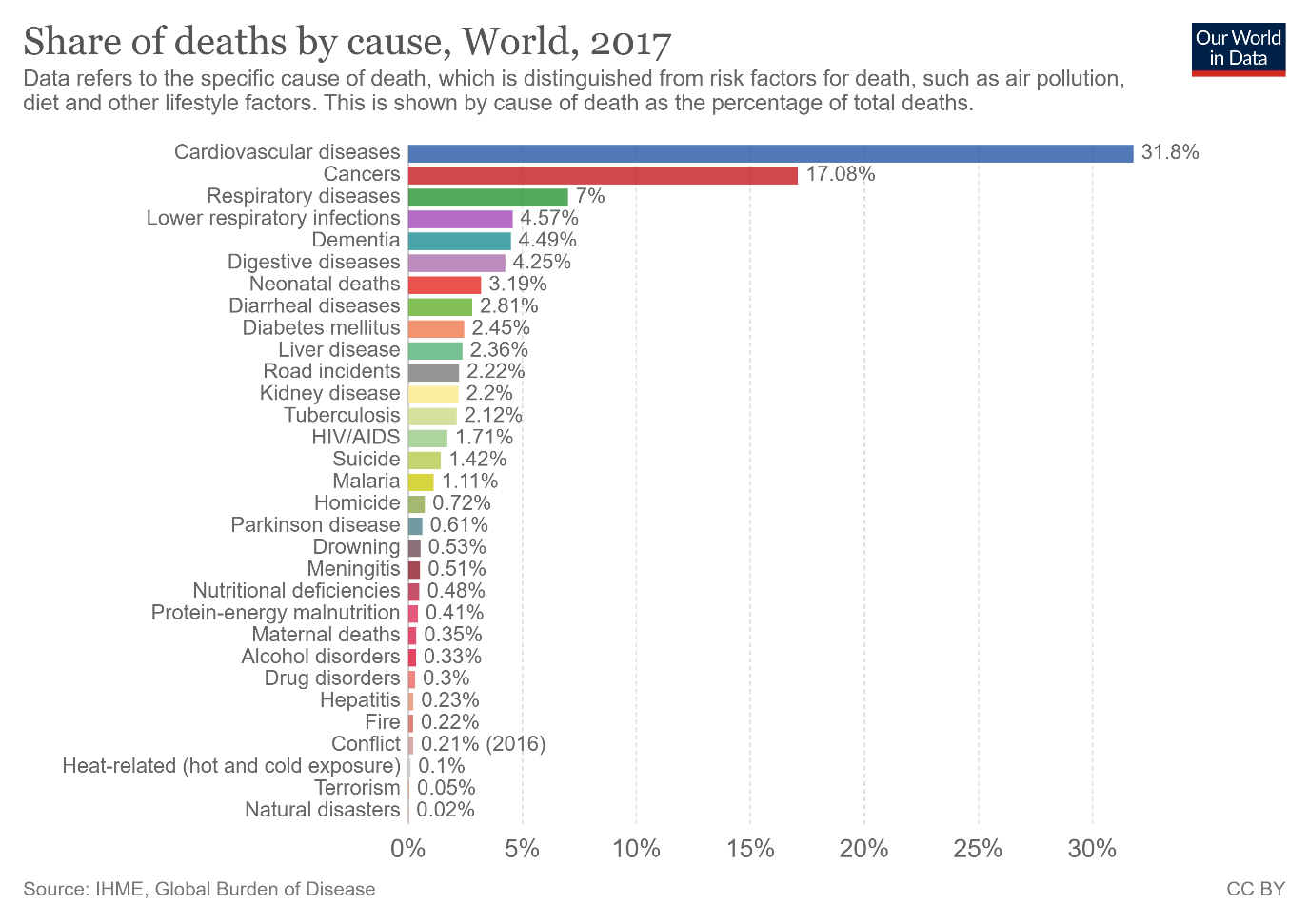


Figure : Share of deaths by cause in the world 2017 (IHME, 2017)

The current research in this field is limited to a few select studies which have been able to create synthetically generated models of the coronary artery system to varying degrees. These include Schreiner’s work in a 1993 paper which created the first synthetically generated model using his Constrained constructive optimisation (CCO) method to create a basic 2d model (Schreiner, 1993). The CCO method has become a staple in synthetic models being used in many optimisation methods that have followed. Culminating in the most recent paper from 2018 which was able to create a model of the left ventricle down to the arteriole level at around 60-microns in diameter (Clara Jaquet, 2018). These models have been verified against porcine morphometric data but have not been implemented into the clinic due to further validation being required as well as improvements to computation speed.

This project aims to create a program within MATLAB that will be able to take initial coronary vessel seeding such that a CT scan may provide and generate a synthetic model of vessels to a microvascular level whilst providing useful data previously not available. This project focuses on generation within a 2d circular area due to time constraints of the project timescale and limitations due to COVID – 19. However, this has allowed for focus on the accuracy and speed of the model as well as further numerical investigation. Further development will be required to transfer the model into a 3d generation. Although, this project should establish a useful starting point for further work with the mathematical principles and method for generating a synthetic tree established and validated.

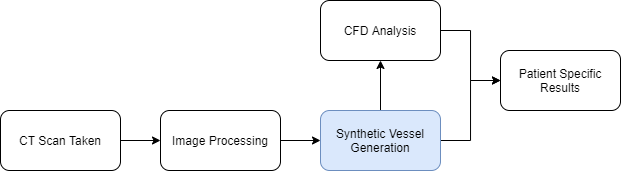


Figure : Process Diagram from Imaging to results

The model will fall into the diagnoses process as shown above. Starting with a CT scan taken of the patient which will then be processed to provide the seeding data for the model. The model will then simulate vessel growth of this initial seeding data to create a network of synthetic vessels that fills the area. This vessel data will then be used within a CFD model which can then be used to provide further analysis as well as other useful data for a patients diagnosis.

## Literature Review

Several papers have been conducted into this field and contributed to this project. One of the key pieces of literature cited by many papers researching synthetic vessel trees is (G S Kassab, 1993) who collected morphometric data across numerous studies on porcine hearts. This data establishes many of the rules and constraints applied in latter papers such as the constraints of the branching angle of vessels as well as the key observation that almost all vessels bifurcate (split into two) and very rarely trifurcate. It is also used in many studies to compare the results of the models to validate them.

Following this research synthetic models could be developed with the necessary rules to create them and data to validate them. Three papers will be focussed on for the purposes of this literature review as they are the primary sources for the work conducted in this project. These include Schreiner’s first paper which laid the groundwork for synthetic models as well as detailing parts of the algorithm and methods used to create these trees. A 2018 paper which took Schreiner’s work further into a full 3d model with greater functionality and features and finally a 2011 paper which took a completely different approach to other methods to generate a synthetic tree.

### Computer-Optimisation of Vascular trees Wolfgang Schreiner and Peter Franz Buxbaum, 1993

This paper laid the groundwork for synthetic models with the development of the constrained constructive optimisation method. Whilst rudimentary compared to later models it provided the proof that these models were possible.

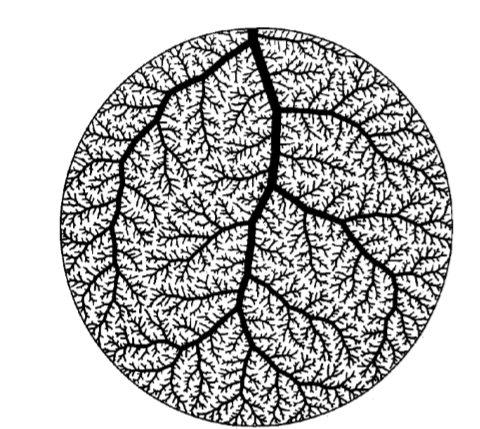


Figure : Visual plot of tree with 4000 terminal segments

The CCO method aims to minimise the total volume of the vessels whilst abiding by physical constraints and rules to simulate realistic vascular growth. One of these physical constraints used is an angle limit to bifurcations such that they are limited to bifurcate to a maximum of 80 degrees as found in morphometric data. Another physical constraint is that vessels may not intersect each other as this would represent vessels growing through each other although this rule is only necessary within 2D models as it is far less likely to occur in 3D generation. This method validated its data both against (Chee, 1987) shown below and (Moise, 1988) which compared the pressure profile of the model with various parameters to a feline heart. The plot in Figure 4 shows a good fit to real life vasculature with some differences in values across the earlier vessels, which could be due to the method employed.

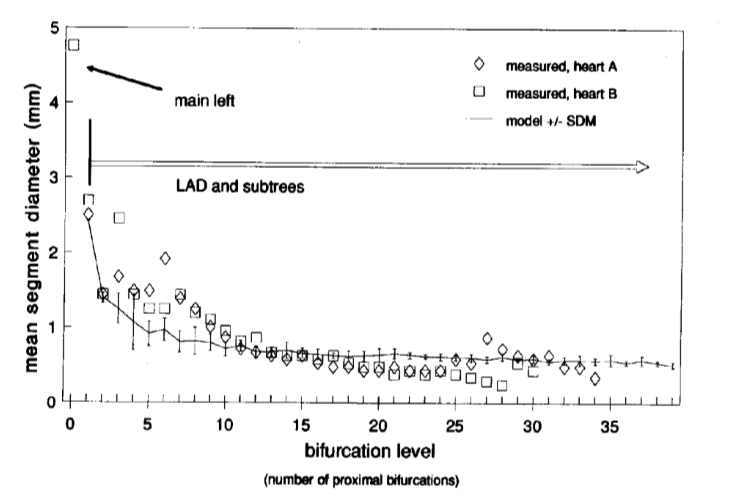


Figure : Average Segment Diameter at each bifurcation level

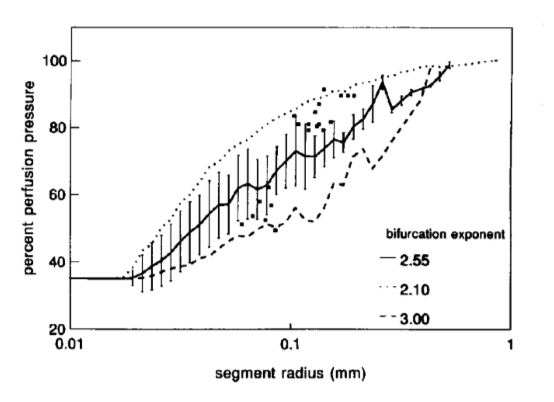


Figure : Pressure profile of model against feline heart

The method causes issues with early vessels due to the pseudorandom nature of the generation method. Whilst the model is connected to minimise volume the points that are connected are random, only constrained to fill the volume evenly with a distribution criterion. This causes uncertainty within the tree as early segments generated usually become major branches within the tree and the location of these major segments change between tree’s created by the model. Another issue with this model is the method of optimisation only optimises at each new bifurcation. This causes all upstream branches to be sub optimal as the conditions they are under have changed. This issue however, is acknowledged within the paper and is taken as an acceptable inaccuracy possibly due to taxing computation cost of re-optimising the entire tree for each new segment.

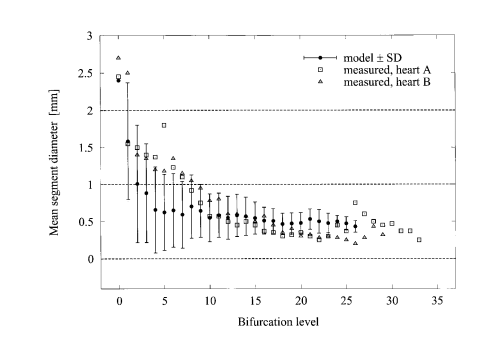


Figure : Results from Schreiner's 1999 Paper (Wolfgang, 1999)

Above a plot from a paper completed 2 years later can be seen. This made some novel improvements to the model adapting it to grow within a convex three-dimensional piece of tissue seen below in Figure 7. However, on Figure 6 it can be seen that the standard deviation of the model has greatly increased with this change to 3D possibly indicating consistency problems with 3D models. Excluding this graph, no further validation of the model was completed other than a few visual inspections comparing the figures below with corrosion casts.

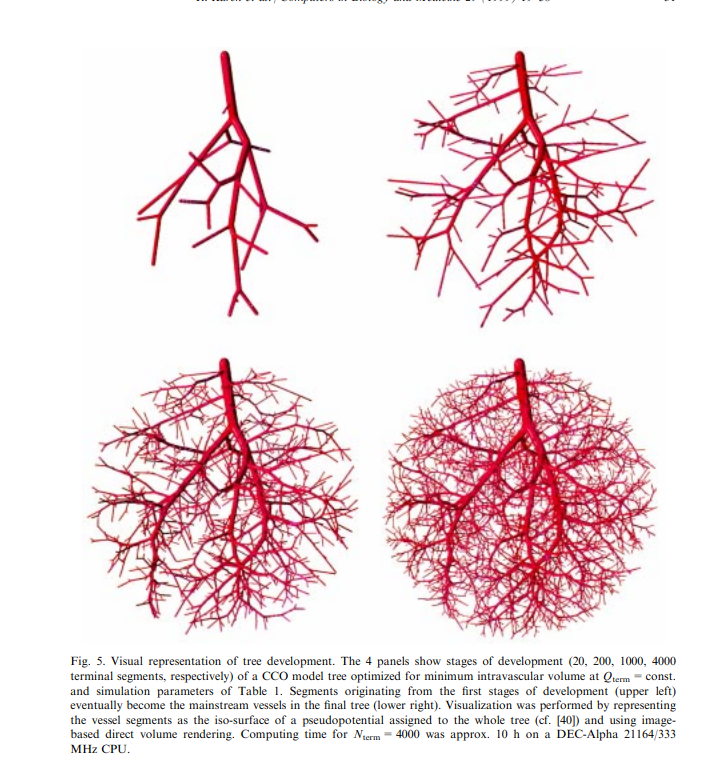


Figure : Visual representation of tree development within (Wolfgang, 1999)

### Generation of patient-specific cardiac vascular networks – 2018

The most recent and thorough paper within this field was completed by Clara Jaquet et al in 2019 published within IEEE Transactions on biomedical engineering. This paper created a 3d generation method which would fill the entire left ventricle. First beginning with vessel generation on the surface followed by simulating internal vascular growth within the myocardium as shown below.

This method follows on from Schreiner’s method applying the CCO method and expanding upon it to build a 3D model of the left ventricle. To create these, CT scans of 6 patients’ hearts were taken and converted to create image A shown below. This could then be used to seed vascular growth. Primary innovations to the method for this model include an algorithm for different vessel trees to compete. This is necessary due to the seeding from the CT scans which provide multiple starting points from epicardial coronaries for vessels to grow from within the same territory. Another addition to Schreiner’s method is the adaptation to 3 dimensions with the volume minimisation being adapted for the myocardium.

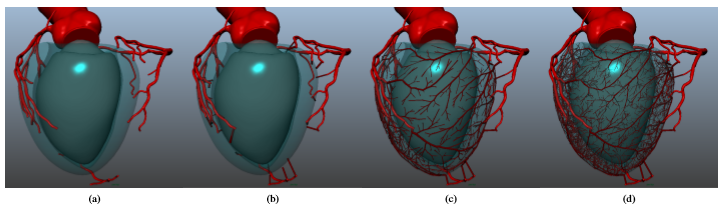


Figure : Illustration of forest growth. (a); inputs provided (b): distributing ﬁrst segment on left ventricle surface for each tree. (c): surface growth (stage 1) reaching 1000 terminal segments. (d): inner growth (stage 2) reaching 6000 terminal segments (Clara Jaquet, 2018)

However, this model suffers from the primary issue faced by most synthetic models, computation time. The Models produced below depict 6000 terminal segment models for patients which took several days of computation time each. This computation time may not be relevant for some diagnoses but in could be critical to patient’s survival in others. The level of computation power required would also be difficult to adopt on a large scale for clinics easily as for each patient several days of run time would be needed on a suitably powerful machine not including any CFD analysis which may be needed for further useful diagnosis information. Further validation of this model is also still required before it can be practically used with the model’s human generations currently being compared with Kassab’s pig morphometry. The paper does state however that the physiological validity of the generated vasculature is being assessed with a multiscale hemodynamic model to help further prove its validation.

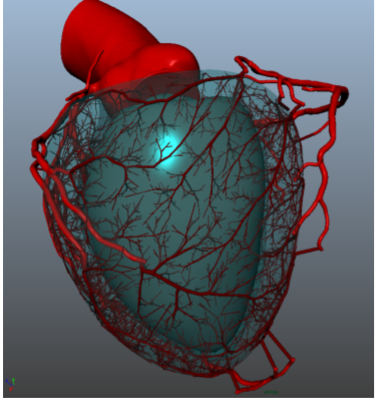
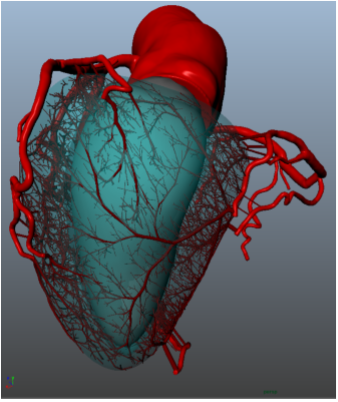


Figure : Results from two patient cases

### Development of a Model of the Coronary Arterial Tree for the 4-D XCAT Phantom

An alternative to the optimisation methods is the fractal method carried out by a group in 2011 which were able to produce the synthetic model shown below. This method is done using morphometric data to set physical parameters the model must abide by such as radii and length of vessels as a function of its Strahler order.

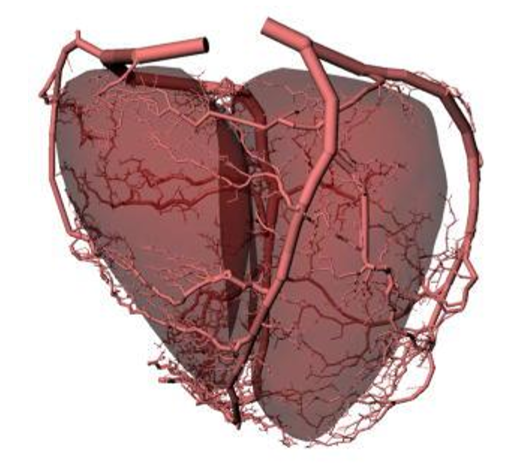


Figure : Results From fractal coronary vessel generation (George S K Fung, 2011)

This method takes a CT scan of a normal human subject and uses it to seed the initial segments of the coronary arteries. The scan also defines the boundaries of the ventricles for the model. Statistical guidelines are applied to it based upon statistical morphometric data taken from various investigations into porcine hearts (Kassab et al 1993, 1994, 1997, Kassab and Fung 1994). An iterative rule-based generation method is then applied to extend the tree beyond its initial seeding. This includes the optimal branching angle rule which determines the diameter of a vessel branch based upon the flow rate through it. A self-avoidance algorithm for vessels is also employed as well as a boundary avoidance algorithm to prevent vessels growing outside the defined volume.

This model produces results much more quickly but suffers from validation issues as it uses Kassab’s work to generate its model. Fractal trees can also not accurately represent anatomic features of real vascular networks (Johannes H G M Van Beek, 1989). There is also difficulty with the implementation of this method as the morphological statistical data needed was scaled to fit humans as it is not possible to recreate this data within humans. This is due to the methods used to create the casts of the vasculature, where the mould is pumped in by the heart at the point of death.

### Murray’s Law

Another key piece of literature defines the nature of the bifurcations of a parent segment into two daughter segments. Murray’s law presented in the equation below is used in all models to define the relationship of vessels during a generation. Larger vessels have been found to have a smaller exponent of around 2 with smaller vessels having larger exponents of around 3.

Equation : Murray's Law (Murray, 1926)

### Summary of Literature

Synthetic vessel tree generation methods currently fall under two categories those are: fractal models and optimised models. Fractal models use statistical distributions to determine the angles, lengths and radii of vessels based on its Strahler order. These are based on the morphometric data taken from Kassab’s work such as in the 2011 paper described earlier. Optimised models use the CCO method detailed in Schreiner’s paper. These optimised models simulate vessel growth within a defined space based on observed principles.

Based on the literature found an optimised model approach will be taken for the following reasons. Firstly, it is important that this model is validated which cannot be easily done with the fractal model due to the validation method of comparing to morphometric statistical data being made redundant as it is being used to generate the model. These models whilst being highly accurate in sizes of their diameters and lengths of the vessels also suffer from accuracy of the connectivity and locations of the vessels which will be important for future CFD analysis. Another key factor is that there are many more optimised models which also detail their overall methods. This greatly helps with the development of this reports model as the understanding of how these models function as well as their shortcomings assists in development of a new optimised model. It should be noted that these papers are a part of journals and so are very brief rarely going into detail of their methods. Most mention briefly the order in which things take place however, details are left vague such as it being stated that the tree is subjected Murray’s law without details of its implementation. This affects the implementation of these features into this projects model as there are a multitude of ways to implement them. These different methods will also need to be assessed to find the most appropriate one.

## Aim and Objectives

Aim: This project aims to develop an accurate simulation of vascular growth for the surface of the heart within a 2D area with the following objectives.

1. Develop a CCO method for vessel tree growth simulation
2. Produce a well-documented model for future research and further development.
3. Validate results against existing morphometric data
4. Analyse variations in results based on physical constraint changes within the model

The project’s aim and objectives have been adjusted greatly since the initial project scoping and planning. Initially this was due to a greater understanding requiring a re-assessment of what was feasible within the timespan. However, the objectives were further restricted due to limitations imposed by COVID – 19. This has resulted in the final model being in 2d within a circle as opposed to modelling the surface of the left ventricle due to the time it would take to create this model and run it to create the results due to the extra computation power required. As well as this, computational fluid dynamics analysis of the model will not be included in this project due to the lack of access to these tools whilst operating from home. However, more detailed analysis of the results produced by the model as well as greater validation using morphometric data has been conducted to compensate. COVID - 19 also limited other aspects of the project as meetings with cardiologists to discuss the project had to be cancelled. The meetings with Dr Davidson, and Dr Mennim would have been useful to gain insight into potential clinical uses as well as features they may have found useful from the model. It was also hoped that certain aspects of the model could be discussed as to help with the understanding of physical aspects such as blood flow and pressure drop within vessels.

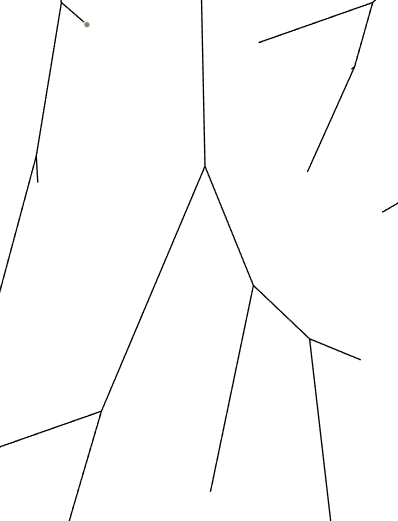
## Method

### Key Terms

Within this method specific terms will be used so it is necessary to detail these for clarity to begin with. Firstly, a segment refers to a single vessel between two branches shown below. A bifurcation is where a segment branches and creates two daughter segments at its distal end. The proximal end refers to the end where its parent is connected whilst the distal refers to the end where its daughters bifurcate from. A terminal segment is a segment at which the vessel terminates, each terminal vessel shares the flow rate of the entire leading artery evenly. This means that in say a 4000 terminal segment generation each vessel supports 1/4000th of the flow rate which is then assumed to distribute in the area surrounding its distal end. For clarity it should be noted that the model is a binary branching tree made up of rigid cylindrical tubes that behave under steady state laminar flow conditions.

Proximal End

Figure : Diagram depicting relevant terms



Segment

Distal End

Segment

Terminal segment

Branch or Bifurcation

Terminal segment

Terminal segment

### Vessel Tree Seeding and Data Structure

Firstly, the data needed for seeding the tree is loaded. An example of this can be seen in the table below where the first 21 segments of a tree are loaded. This seeding data represents the vessels capable of being seen from a CT scan of a patient’s coronary arteries. It also serves the purpose of reducing the effect of the pseudorandom nature of the point selection the tree is generated from as the main vessels have already been set.

Furthermore, the data structure can be seen below necessary for generating the tree. Further data is created in the latter portion for the plotting of results however this will be discussed later for simplicity. The proximal x and y are the co-ordinates of the beginning of the segment with the distal being the end of the segment. The co-ordinates of the points are between 0-2 for both the x and y and are scaled to the appropriate size for calculations. The parent is the ID of the segment prior to it with daughter 1 and 2 being the two segments that branch from the segment. The flow rate is the number of terminal segments that this segment supplies with each terminal segment supplying an equal amount of the total flow rate as detailed previously.

Table : Data needed for seeding segments with Segment 1,2,3 and 21 shown

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Proximal x** | **Proximal y** | **Distal x** | **Distal y** | **Parent** | **Daughter 1** | **Daughter 2** | **Flow Rate** |
| 1 | 1 | 2 | 1.1 | 1.6 | 0 | 2 | 3 | 11 |
| 2 | 1.1 | 1.6 | 1.2 | 1.4 | 1 | 6 | 7 | 8 |
| 3 | 1.1 | 1.6 | 0.85 | 1.4 | 1 | 5 | 4 | 3 |
| ... | ... | ... | … | … | … | … | … | … |
| 21 | 1.3 | 1.15 | 1.45 | 1.15 | 6 | 0 | 0 | 1 |

A picture containing clock

Description automatically generated

Figure : Patient seeding data visual plot

### Generating Points

The model then generates the terminal point to be connected. The point selection criteria used was developed from Schreiner’s work and adapted to work within this project’s model. The point selection criteria work by generating random points within an assigned area and selecting one based on a distance threshold which is the minimum distance between a point and all previous segments. This essentially attempts to fill the area as evenly as possible by selecting points that are spread out from segments. Below the plot of the points selected can be seen which appear to still cause clusters but when this is connected in an appropriate way the selection appears to fill the area evenly.

A close up of a logo

Description automatically generated

Figure : 250 Terminal points visual plot

A close up of a piece of paper

Description automatically generated

Figure : 250 Terminal points connected visual plot

Schreiner’s method first determines whether the point lies somewhere along the segment. If the point does the orthogonal distance is calculated between the point and the segment. Otherwise the distance between the point and the segments distal and proximal ends is taken. This is where the methods differ as the minimum distance between the point and the segment is calculated by finding the distance between the point and several positions along the line as opposed to the orthogonal distance. This was done as the original method was not compatible with this projects model. When it was implemented it would select points close to other segments and ignore many empty spaces. As well as this computation time was saved by using the new simplified method. This minimum distance is found for all segments and the smallest one is saved. This distance is then checked against a distance threshold calculated using the equation shown below. kterm in the equation refers to the number of terminal segments currently in the tree. If this distance is greater than the threshold this then it is selected. However, if it does not further points are tested up to 200 points at which point the threshold is reduced by 10% until a point is found.

Equation : Distance threshold equation (Schreiner, 1993)

### Connection Optimisation

Once the point has been selected it must be connected to a segment. This connection will be selected based on minimising the total volume of the vessels concerned. However, physical constraints are taken into consideration.

Firstly, the point is connected to a segment. This is done by generating random points in the area between the new point and the segment shown below in Figure 15. One of these connections is then tested first calculating the total volume of the connection using Poiseuille solution to find the radius for each of the segments. The equations for this calculation can be seen below in Equation 3. If this new connection’s volume is less than the previous smallest volume found or is the first connection its physical constraints are then tested.

Equation : Rearranged Poiseuille’s equation

Table : Equation variables

|  |  |  |
| --- | --- | --- |
| Symbol | Description | Unit |
| r | Radius of segment | m |
| Q | Flow rate through segment | m3/s |
|  | Blood Viscosity | Pa.s |
| l | Segment length | m |
|  | Pressure drop along segment | Pa |

A picture containing game

Description automatically generated

A picture containing table

Description automatically generatedA picture containing table

Description automatically generated

Figure : Connection optimisation method

The angles of the new connection are tested with the angle’s constraints shown below. This prevents vessels growing backwards on themselves with the constraints for angles A and B. The constraint of angle C applies as to assist in volume optimisation as an angle of greater than this is not optimal within volume optimisation (Clara Jaquet, 2018).

A picture containing table

Description automatically generated

**C < 80°**

**B > 120°**

**A > 120°**

Figure : Angle Constraints applied to model

Next the connection is tested for Murray’s Law shown below. This states that the radii of the daughters’ vessels must obey a power law relationship to the parent vessel. The value of gamma varies between 3 for smaller vessels and 2 for larger vessels as found in various works (E VanBavel, 1992; J C Schwarz, 2013; Y. Zhou, 1999; Kassab, 2006; I. A Lubashevsky, 1999). This value will be explored within the results section to find the optimal value to be used or range of values to be used. This check also makes sure that the vessels’ diameter does not grow which was possible with longer connecting bifurcations.

Equation : Murrays Law

Finally, the new connection is tested to see if it intersects with any of the previous segments. This is done by comparing the line equations of the new segments and all previous segments and checking if they intersect and if so, if it is within the bounds of both lines. This check includes the new parent and the two daughters created from this new connection as they are potentially all shifted with the new bifurcation point.

This process is repeated with the new point tested connecting to every segment. Once the optimal connection is found it is saved, with the appropriate connectivity data adjusted for the segments it connects to as to reflect this. This entire process of selecting a new point and optimally connecting it then repeats until the number of segments desired to be created has been completed usually around 250 – 4000 times.

The results from the generation are then plotted. This includes a plot of the vessels themselves so that the generation can be examined providing insight into the behaviour of the generation method as well as its connectivity. A plot of the vessel radii as a function of the bifurcation level of vessels was also produced for comparison with other literature as well as a plot of average radii against diameter-based Strahler order.

## Results and Discussion

### Parameter Tests and Study

#### Generated Points

An Investigation into the model’s parameters will first need to be conducted as to find the optimal set up for consistent and accurate results. This will include an investigation into the number of randomly generated points selected from, as well as the number of bifurcation points generated. For these two parameters a larger value should represent a more accurate solution as it has a greater number of points to choose from allowing for better solutions to be found at each step. However, larger point selection should cause a great increase in the computation time. Results that will be investigated are the average total volume of the generated trees, the standard deviation of the volume as well as the standard deviation of the radii of vessels at each bifurcation level. These factors are commonly used within other studies of synthetic vessel trees. The average volume has been selected as it indicates which solutions are the most optimal. The Standard deviations indicate the consistency of the results produced by the set parameters. Average computation time will also be recorded.

Table : Results of Point Selection Parameter Investigation with N = 250

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Points** | **Average Volume** | **Volume SD** | **Average SD of Radii (mm)** | **Computation Time** |
| 50 | 1.00016x10-6 | 9.34859 x10-8 | 0.1137 | 11.407 |
| 100 | 9.98526 x10-07 | 8.67891 x10-8 | 0.1094 | 23.028 |
| 200 | 9.95478 x10-07 | 8.5584 x10-8 | 0.1028 | 38.870 |
| 400 | 9.47929 x10-07 | 7.19754 x10-8 | 0.1049 | 113.978 |

200 points has been selected as it has the smallest average volume excluding 400 points as well as having a reasonable standard deviation showing it can produce consistent results. The computation time for this is also reasonable which should contribute greatly to a reduced computation time for larger vessel trees that will be conducted later. 400 points appears to perform slightly better than 200 points however, the computation time is 4 times longer which would not be a worthwhile trade especially for a larger number of segments.

#### Murrays law Implementation

Murrays law dictates a rule that the daughter vessels’ radii must obey. However various values for the exponent (gamma) can be used within the model as well as various methods of implementation. Gamma varies between 2.55 to 3 within the coronary vessels. Murray’s law within the model currently checks whether the new connection obeys the rule by finding the radii of the vessels of the proposed connection and then calculating what the parent vessel radii should be through Murray’s law. This then compares the proposed parent vessel against the value calculated through Murray’s law. The check allows for 10% error as without this an exact solution would need to be found which at best slows the model down greatly and at worst causes no solutions to be found. Another option however was to allow a range of answers between 2.55 and 3 for gamma. To investigate this the same values described in the points investigation will be observed as well as a visual plot so that the structure of the tree can be observed to be behaving correctly. Plots of the average diameter against the bifurcation level for model have been created against two hearts taken from (Chee, 1987). Changes in Murray’s law only appears to have a noticeable affect one the number of bifurcations taking place. An exponent of 3 appears to be most consistent in this however, with a range from 18 -24 across the tests with both 2.55 and 2.55-3 having around 18 -27 maximum bifurcations. However, this is not conclusive enough to determine which exponent should be used. The exponent will instead be determined by the work it is being compared to. For Schreiner’s work this will be 2.55 whilst for comparisons to Clara Jaquet a value of 3 will be used as these are the values proposed in their respective works and should thus help match the parameters set for a better comparison. A value of 3 will also be used to compare to the morphometric data as this was not mentioned within Jaquet’s work to affect the comparison.

A close up of a map

Description automatically generated

Figure Murrays 2.55

A close up of a map

Description automatically generated

Figure : Murrays 3

A close up of a map

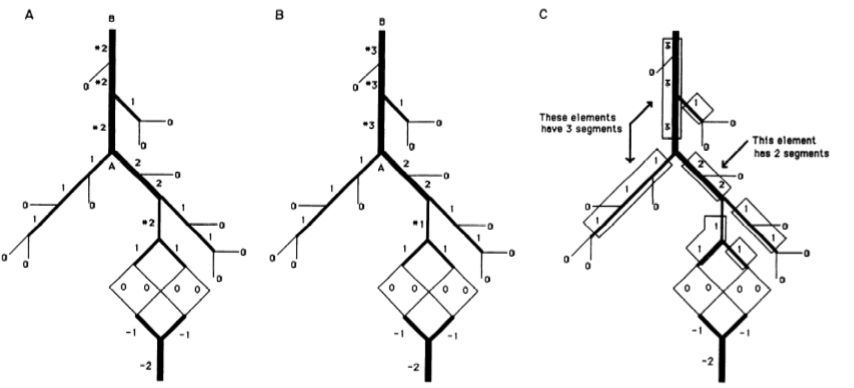
Description automatically generated

Figure : Murrays Range

### Investigation

For comparison to other academic papers the diameter-based Strahler order has been implemented created by (G S Kassab, 1993). This allows for a more accurate representation of the results as well as allowing a direct comparison with morphometric data and other synthetic trees.

The diameter-based Strahler order is used within Kassab’s morphometric data as well as Jaquet’s work. The diameter-based Strahler order is a method of giving vessels an order to observe diameter and length changes of vessels within hearts. This method works by giving every vessel a Strahler order given by their position in the tree where terminal segments have an order of 5. Kassab’s measurements start at Strahler order 1. However, these vessels are much smaller than any generation of the model would be likely to proceed to as they are at the capillary level at around 9.4 microns. The order from above the terminal segments are then increased up the tree when two segments of the same order meet at a bifurcation point. If two segments have the same order, their parent segment’s order is increased by one otherwise the largest order of the two daughters is given to it shown in Figure 20. Using this Strahler order the means and standard deviations are found for each order and used to provide ranges of diameters for them. Each vessel is then re assigned to each order based on the range they fit in. The mean and standard deviation of these new orders are then found and the process is repeated until it has converged usually after around 3 iterations. This provides a diameter-based Strahler order for each vessel.



Vessel diameter too small for order 2

Vessel diameter large enough for order 3

Figure A: conventional Strahler ordering scheme, B: Diameter defined Strahler order system (G S Kassab, 1993)

### Primary Results

Below can be seen the results collected from 10 simulations run from one patient data set. These include one of the visual plots of the trees as well as a plot of the average diameter for each Strahler order with upper and lower bound ranges based on their standard deviation. A plot of the average diameter against the bifurcation level has also been included. Each of these plots will be compared in turn with academic papers for validation and investigation. However, it should be noted that for the bifurcation level plot there are 250 terminal segments for the generation so that it can directly be compared with Schreiners work. All other comparisons will be completed with 4000 terminal segments. The average computation time was around 2.5 hours for a 4000 terminal segment plot on an Intel i7 -4790 CPU 3.6 GHz processor. However, usage of this processor was limited to 13% due to MATLAB’s limited utilisation of it.

Some notable problems with the model can be seen visually with the changes in diameter between vessel bifurcations being noticeable. This presents a physical error as nothing is taken into the account for flow splitting at the bifurcation. For the purpose of volume calculation, the connection is treated as three separate perfectly cylindrical vessels meeting at a set point which would not be the case in reality.

A close up of a logo

Description automatically generated

Figure : 4000 Terminal segment Tree



Figure : Plot of average diameter against Strahler order with mean upper and lower bound presented



Figure : Plot of average length against Strahler order with mean upper and lower bound presented

#### Comparison with Schreiner’s work

Firstly, results will be compared to work from Schreiner’s paper. This is because both models are 2d generations and can facilitate the same input parameters shown below. These parameters are for the left anterior descending (LAD) coronary artery within humans. It should be noted that these parameters will be used throughout the results excluding the Murray’s law exponent. Firstly, a generation can be seen below with 4000 terminal Segments. Immediately differences between the two models can be seen visually. Plotting of the vessels appears smoother on Schreiner’s due to the method employed within this projects model for plotting the vessels which can appear somewhat jagged as the vessel diameter suddenly changes at bifurcations.

Table : Parameters used in generation

|  |  |
| --- | --- |
| Parameter | Value |
| Radius to be filled | 0.05 m |
| Flow through leading artery | 8.33 x 10-6 m3/s |
| Pressure at lead artery | 1.33 x 104 Pa |
| Pressure at distal end N = 250 | 8.38 x 103 Pa |
| Pressure at distal end N = 4000 | 7.98 x 103 Pa |
| Blood Viscosity | 3.6 x 10-3 Pa.s |
| Murray’s Law exponent | 2.55 |

A close up of a map

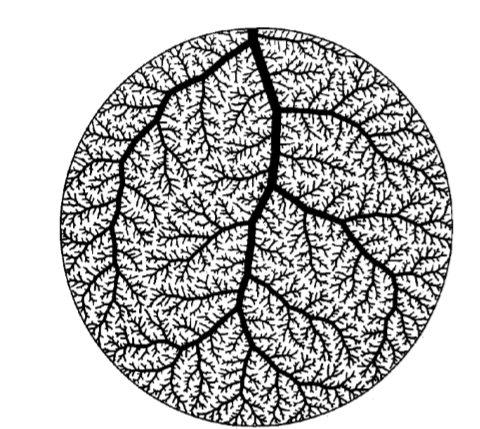
Description automatically generated

Figure : 4000 Terminal Segments Tree (Schreiner, 1993)

Figure : 4000 Terminal segment Tree

This projects model appears to tend towards curving to branch towards points whilst Schreiner’s prefers a more direct straight vessel. This would represent longer distances that blood would be required to take to reach the terminal vessels at some points as the route is very indirect. This is potentially due to assumptions made within the pressure drop across vessels which is that pressure drop is evenly distributed along each segment towards the terminal segment. For example, if there are 5 segments between the lead segment and the terminal segment each segment will have an equal pressure drop that will be the total pressure drop over 5. This would cause the distance travelled to not be considered for the connectivity of the tree, as the pressure drop across longer paths would be much greater. Longer paths will then tend to occur as the radius reduces along them meaning a smaller total volume than a more direct path which may need to attach to a larger vessel increasing total volume of this new connection. This reveals a physical error within the model that will need to be improved upon in future work and considered when discussing results. This should also lead to this projects model creating a reduced total volume as it can more optimally fill the radius due to it not being bound by as many physical constraints. The total volume of the model for 4000 terminals was found to be 1420 mm3. However, it is never stated within Schreiners work what the total volume was found to be so there can be no direct comparison of this data.



Figure : Plot against Schreiners Model

Above a plot of average segment diameter against bifurcation level has been plotted. The bounds presented were found using the standard deviation and means from 10 models with identical parameters. Two key points can be taken from this plot with observation. First is that the bifurcation of Schreiner’s model continues to much greater values than in this projects model. With 39 bifurcations compared to around 20 – 25. It is unclear within the results presented in the paper as to whether many of the terminal segments proceed to this bifurcation level or whether this is only a few terminal segments reaching such large amounts. If this is assumed to not be an anomaly and in fact a common occurrence the difference in maximum bifurcation level reveals a key difference within the generation method that lies within the topology of the trees. It appears from Figure 24 and Figure 25 that this project’s model prefers longer segments with fewer bifurcations to fill the area leading to a reduced amount of bifurcations to reach the terminal segment. Schreiners model above appears to prefer smaller segment lengths especially when reaching the terminal segments with far greater amounts of bifurcations. Thesecond is that this project’s diameters are consistently lower than Schreiner’s. This is likely due to the implementation of pressure drop described earlier which allows for more optimal selection of connections as it has fewer physical restrictions applied to it.



Figure : Comparison with data from (Chee, 1987)

From the results above it can also be seen that the two models generate in similar ways shown from the matching curves for the diameters with bifurcation level excluding the increased number of bifurcation level of Schreiner’s work. It also consistently shows a smaller range due to its smaller standard deviation showing a better consistency in generation. This is potentially due to the seeding described earlier removing uncertainty from the early segment generations.

Some issues presented in this work as well have been solved by the model. For example, in the paper they state that the pseudorandom nature of the points generated can cause uncertainty within the model leading to the larger standard deviations in results. This is primarily an issue early in the generation as these points tend to become the major vessels within the tree. In the model presented in this project sample patient specific models have been created. These seed an initial 21 segments within the model representing what could be seen by a CT scan. This seeding should essentially guide the model reducing the uncertainty due to the first key points being pre-set and not needing to be pseudorandom. However, this improvement is not immediately apparent in the results as the seeding used is not that of the measured heart presented in the result. Hence the difference in values for the early bifurcation levels. If more data from the study by (Chee, 1987) was available the larger initial vessels could be input as the seeding data increasing the accuracy and consistency of the model. Without this data the model is still essentially guessing at where the larger vessels will be as they are generated. Due to the vessels generated later in the procedure becoming minor vessels the random nature of the points generated has less of an affect as the connectivity becomes more important.

The reason for the initial vessels causing uncertainty is due to the fact of them becoming major vessels. Any error with this vessels location and size will have a much greater effect on the tree as a result. This major vessel will also affect the generation of the tree further down the line as new vessels points will be connected considering intersection with these major vessels. This will also affect the volume optimisation as connections to the major vessels are less feasible due to the increased volume of them. This should be the analysis of your results.

The computation time for Schreiner’s model was stated to be 2 days on an IBM 3090 mainframe. Whilst being a high-end processing unit this system is now around 40 years old at the point of writing and so is hard to compare to the processor currently used. The computation time is obviously significantly less for the proposed model however this reduction could be significantly due to increased processing power available.

#### Comparison with Clara Jaquet’s work

Below visual plots can be seen taken from (Clara Jaquet, 2018). This poses an immediate problem for comparison of data in that these plots are 3 dimensional whilst this project’s is in 2 dimensions. This will be apparent within plotted results when compared as the generation of vessels will be greatly different. For example, within 3 dimensions intersections are much less likely to occur between vessels as they are able to pass over or under each other whereas in 2 dimensions they are restricted to cross. Seeding data can also not be replicated for comparison as it is 3-dimensional data that cannot be converted to 2 dimensions due to it surrounding the entirety of the heart.

Another key difference within the generation methods is that within Jaquet’s model multiple trees compete with each other due to the seeding data containing multiple separate arteries. These arteries start at various points around the heart competing for space and blood flow and should cause large differences within results due to this behaviour. For example, different trees are assigned different flow rates to achieve. This causes trees to grow to different sizes, both in internal volume of the vessels and in the area that they cover.

However, given these differences vessel data can still be compared using diameter-based Strahler ordering. As the sizes and dimensions of these will only be somewhat affected due to the generation methods of the vessels themselves remaining somewhat similar even with the extra dimension. The main factor that would be affected is the locations of the vessels as they cannot be compared in any way due to the nature of both the models.

|  |  |
| --- | --- |
| Clara Jaquet | Model |
| Volume = 1261 ± 43 mm3 | Volume = 1420 ± 160 mm3 |
| Diameter = 171 ± 61 μm | Diameter = 250 ± 14.3 μm |

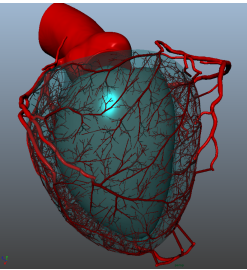


Figure : 6000 Segment 3D Visual plot (Clara Jaquet, 2018)



Figure : Comparison of LAD vs Clara et al

On the plot above the model from the paper (Clara Jaquet, 2018) can be seen compared to the model from this report. Both models appear to follow a similar curve much closer than expected. The differences are much greater at Strahler order 11 which represents the leading vessels of the LAD with around only 32 vessels within this order. This difference can be attributed to the lack of seeding data for the model causing uncertainty within these first few generations. The range of values also appears to vary greatly even with the first 21 segments input for the model showing even with the addition, uncertainty still occurs greatly within the beginning generations causing large discrepancies within diameters. This should be considered for future work as it shows weak areas within the model which could be improved upon with accurate seeding data or refinement of the current method within point selection and connection. It should be noted that the seeding data is not exactly represented within the final model. This is because due to the nature of the generation method the vessels can still be moved and adjusted when connecting a new terminal segment. This connection is restricted to be within the coordinates of the distal and proximal ends of the segment but will still cause differences between the original and the final model. In future work the seeding could be locked in or restricted further to only connect to the segment rather than moving the connection point however with the current method it was not able to be easily implanted.

#### Morphometric data validation

Morphometric data taken from (G S Kassab, 1993) has been a key tool for many synthetic generation methods to validate models. Using the diameter-based Strahler ordering, plots have been created to compare the average diameter and length from vessel generations with the porcine morphometric data. The data from Kassab’s work can be seen in appendices 1.1 - 1.3.



Figure : 4000 Terminal points model against morphometric data diameters

To do this the parameters shown below were used and repeated for 10 different tree generations and plotted against the data for the left anterior descending coronary artery. It can be seen that the shape of both the morphometric data plots and the projects model are somewhat similar with the gradient of the morphometric data being much steeper towards the higher Strahler order. This indicates that within the model the reduction in diameter of the vessels is much steadier than is found in real life examples. This relates back to the earlier point from comparisons with Schreiners work where the path to a terminal segment could be seen to be much longer in this projects model than in their work. These longer vessels and more indirect connections could lead to the more gradual decrease in gradient as the new terminal vessels tend to connect to the longer chains of vessels than the larger vessels that may be closer to them. This again is due to the assumption of the pressure drop being even over a longer chain of vessels than a smaller one.

Table : Parameters Used in generation

|  |  |
| --- | --- |
| Parameter | Value |
| Radius to be filled | 0.05 m |
| Flow through leading artery | 8.33 x 10-6 m3/s |
| Pressure at lead artery | 1.33 x 104 Pa |
| Pressure at distal end N = 250 | 8.38 x 103 Pa |
| Pressure at distal end N = 4000 | 7.98 x 103 Pa |
| Blood Viscosity | 3.6 x 10-3 Pa.s |
| Murray’s Law exponent | 3 |

Another difference is that the morphometric data tends to have much larger vessels at the higher orders. This indicates discrepancies in accuracy for the leading vessels. Some of these higher order vessels in the model will be those seeded initially. These seeding vessels could be adjusted easily with the correct data if the model was in 3d which should remove this inaccuracy. Another cause in the difference in these values will be due to the flow rate not being known for the various pig arteries being modelled. This data however could be easily found in practice on a patient to patient basis for higher accuracy. The flow rate currently used is that carried through from Schreiner’s work for the LAD of humans of 8.33 x10-6 m3/s.



Figure : 4000 Terminal points model against morphometric data lengths

|  |  |
| --- | --- |
| Morphometric data | Model |
| Volume = 1400 ± 150 mm3 | Volume = 1420 ± 160 mm3 |

A plot of the average length of vessels against Strahler order can also be seen above. The plots of both sets of results shows very little similarities although, the results from Kassab’s morphometric data do not seem to represent a consistent shape with the ranges of values being very large. The range of values for the model however, appears to be very small in comparison further proving the models consistency with results. Again, the large discrepancies at the higher Strahler orders can be accounted for by the inaccurate seeding data used within the models. The volumes of the model and Kassab’s Morphometric data are also very similar closer than Jaquet’s total volume. However, the size of the pig heart is not exactly the same so it could be the case that a human LAD is slightly smaller reducing the total volume.

The primary cause for the difference in results is likely to be the difference in a 2d generation and a 3d physical pig heart. Generation methods differ for surface vessels than that of internal vessels within the myocardium. Within other 3d synthetic generation methods the model usually creates the surface vessels for the first 1000 segments where it then penetrates into the myocardium and changes its method for generation. A key difference is that vessels are much less likely to intersect allowing for more freedom within the connections to pick the optimal choice for minimising blood volume. This is due to it being very unlikely for vessels to intersect in 3d where they can go around each other within the volume whereas in 2d from any point to another it is forced to intersect if there are vessels in-between. This intersection limits the connectivity of the 2d model leading to more inefficient connections than in 3 dimensions. The volume optimisation method is also adjusted with an extra weighting added related to the location of the mid-point of the vessel for the total volume calculation.

Equation : Percentages of vessels at each Strahler order

|  |  |  |
| --- | --- | --- |
| Strahler order | Percentage of total Vessels | |
|  | Model | Morphometric Data |
| 5 | 48.9% | 38.5% |
| 6 | 32.3% | 31.8% |
| 7 | 13.0% | 17.4% |
| 8 | 4.1% | 8.0% |
| 9 | 1.5% | 2.6% |
| 10 | 0.1% | 1.3% |
| 11 | 0.09% | 0.4% |

A break down of the percentages of vessels at each Strahler order is also shown below. The vessels of Strahler orders 1-5 have been removed from the morphometric data so that they can be directly compared. These percentages are consistently different across all Strahler orders other than 6 where they match somewhat precisely. It seems that from Kassab’s research a significant number of vessels tend towards a Strahler order of 5. This shows a difference between the structure of the models with this model tending to prefer larger vessels over smaller vessels to distribute the blood around the heart. This is interesting as the total volumes are still very similar although, this could be due to the smaller vessels which are not accounted for in the model contributing to Kassab’s total volume

#### Patient Specific Data Tests

Below 4 visual plots can be seen from 4 different seeds with the seeding also shown below. These different seeding plots represent an example of CT scans from 4 distinct patients. It is clear that the initial vessels seeded become major vessels in all cases showing the importance of the seeding data to be accurate. It is also interesting to note that the seeded vessels tend to be larger based on how many vessels are downstream of it at the point of seeding. This should be the case as they support a larger flow rate initially however, it is interesting that the initial number of vessels are key as 3990 more terminal segments are added after the initial seeding. This would increase the number of terminal vessels downstream by a sizeable amount depending on which connections are made.

A plot of the radius and lengths against Strahler order can be seen showing that each have distinct differences between the early vessels. This is as expected due to these vessels being primarily made up of the seeding data set. However, they all tend towards similar values at lower Strahler orders due to the generation methods still remaining the same between each method.

|  |  |
| --- | --- |
| A picture containing clock  Description automatically generated  Figure : Patient data seeding 1 | A close up of a logo  Description automatically generated  Figure : Patient 1 Results |
| A picture containing object, clock  Description automatically generated  Figure Patient data seeding 2 | A close up of a logo  Description automatically generated  Figure : Patient 2 Results |
| A picture containing clock  Description automatically generated  Figure : Patient data seeding 3 | A close up of a logo  Description automatically generated  Figure : Patient 3 Results |
| A picture containing object, clock  Description automatically generated  Figure : Patient data seeding 4 | Figure : Patient 4 Results |



Figure : Average segment diameter of each patient generation

#### Computation time

Below a plot of computation time against number of terminal segments can be seen. This plot shows a curve with a gradually increasing gradient when all constraints are applied. The gradient increase will be because as more segments make up the network more have to be checked both as a possible connections and as possible intersections. This increase in gradient may start to reduce passed 2000 because passed a certain point the area is so filled with vessels that the intersection check is failed much earlier as it checks through every point.

A close up of a map

Description automatically generated

Figure : Computation time for with various physical constraints removed

When the intersection check is removed the computation time significantly drops showing that it is one of the larger factors taking up computing power. The increasing gradient as terminal segments are added is also much less with the intersection check removed. The fit is now much more linear as now each new terminal segment only adds another possible connection to be tested rather than two new checks in the form of another intersection check.

Plots of angle and Murrays law reduce the computation time although not as significantly as the intersection check as their calculations are much easier. They are also earlier in the checks and so if the connection fails them it skips the intersection check which takes longer. The angle check can be seen to be speeding up the model. This is potentially due to it narrowing down the search area by removing unlikely solutions earlier with a quick calculation.

Another potential reason for the increased amount of physical constraints increasing computation time is that with more physical constraints it increases the chances of a point not being able to be connected and the process needing to be repeated with an new point. For 1000 terminal segments a counter was put in place to count every time the model determined a point had no way of being connected. It found that 100 times the point had not been connected.

## Future Work

Firstly, a better calculation for pressure drop along the vessels could be done. This area is lacking within the model due to the lack of knowledge within the field. This caused difficulty within the implementation as the pressure drop is pivotal in calculating the radii of vessels necessary for the volume minimisation the model is based on. However, it could be the case that this method for pressure drop is necessary to save on computation time due to the reduced complexity. Although, this is unknown as the methods for other models detailed in papers is vague at best and non-existent at worse. Schreiners method details the pressure at the lead vessel and terminal vessels for 250 segments and 4000 segments. Within (Clara Jaquet, 2018) pressure drop is defined in Poiseuille’s solution to be Resistance multiplied by flow but how these values are determined is left unknown. It may be possible to define radius in another way in the method and then use this to define the pressure drop across the vessel however this would make pressure drop redundant for generation as volume minimisation is all that is concerned currently.

The next major step will be to adapt the model into 3 dimensions similar to Jaquet’s work. This implementation would use the same method presented in this project with a few changes. Namely, changing the intersection check as this would be much less likely and potentially redundant. The target function of the volume minimisation would also need to be changed for the different layers of myocardium due to the varying nature of the hemodynamics between them. However, the surface generation method should remain the same before penetrating into the myocardium with some adaptation to 3D coordinates. It should be noted that computation time should significantly rise with this improvement due to the increased complexity caused. Although it is hoped that with the simplified method proposed that this time is less than that in other papers as to provide a more practical clinical solution.

For future work a different coding language could potentially be used such as python as to more efficiently carry out generations. This may also allow for a better visual representation of the vessels as the current plots could be improved upon to remove the jagged edges caused by the rectangles that represent the vessels.

Another improvement that focuses on the computation time is an improvement to the volume optimisation. Currently a selection of random points are tested in turn. This could be improved upon by narrowing the selection area to a feasible area based on the physical constraints. Finding the position with the smallest possible volume could be then sped up by implementing a more sophisticated method. Such as some sort genetic algorithm or potentially through methods discussed in Schreiners 1993 paper which uses small incremental changes in the bifurcation point to find the gradient of the target function which the bifurcation point is then moved along the descending gradient.

Future implementation of Murrays law could investigate changing the exponents value with size of radii to more accurately match what occurs in reality as stated in (Clara Jaquet, 2018). However, in this model Murray’s law exponent is set to 3 which could indicate that this method is not necessary or inaccurate in generations. The exponent could also vary with other factors such as the ratio of metabolic to viscous power dissipation of the tree of interest as suggested in (Kassab, 2006).

## Conclusion

In this report a method to simulate realistic coronary vascular growth within a defined 2-dimensional area has been proposed. This method has been validated against morphometric data and compared to previous papers within the field. The model has been shown to satisfy the fits of Schreiner and Jaquet’s work as well as Kassab’s morphometric data. Through the documentation of this report and detailed annotation of the MATLAB code, future work detailed previously will be able to be carried out. This will allow the model to be adapted into a fully 3-dimensional generation of the coronary vessels. The improvements to the pressure drop along the vessels should correct the discrepancies in vessel dimensions between the model and morphometric data. A key improvement needed to be made in the field of synthetic vessel generation is with the computation time of models so that it can be useful within a clinic. Because of this the model’s computation time was focused on for this project and is one of its key strengths with a vessel tree generation taking 2 hours for 4000 terminal segments. In comparison to Jaquet’s work where a 6000 terminal segment simulation would take several days to complete. The improvements to the computation time are due to the implementation of physical constraints which are checked first to narrow down realistic solutions from possible connections. However, this could be improved further by making use of a faster coding language such as python as well as more powerful and improved utilisation of CPU’s.

(Clara Jaquet, 2018) (Schreiner, 1993) (G S Kassab, 1993)

# References

Akmal Sabarudinm, Z. S. (2013). Coronary CT angiography: Diagnostic value and clinical challenges. *World Jounral of Cardiology*, 473-483.

al, S. G. (2007). Magnetic resonance imaging of the coronary arteries. *Cardiovascular Journal of Africape*, 248-259.

Chee, M. Z. (1987). Segment analysis of human coronary arteries. *Journal of Vasscular Research*, 76-84.

Clara Jaquet, L. N.-C. (2018). Generation of patient-specific cardiac vascular networks:a hybrid image-based and synthetic geometric model. *Transactions on Biomedical Engineering, Institute of Electrical and Electronics Engineers, 2019*, 946-955.

E VanBavel, J. S. (1992). Branching patterns in the porcine coronary arterial tree. Estimation of flow heterogeniety. *Circulation research 71.5*, 1200-1212.

G S Kassab, Y. C. (1993). Morphometry of pig coronary arterial trees. *American Journal of PHYSIOLOGY*, H350 - H365.

I. A Lubashevsky, V. V. (1999). Analysis of the optimality principles responsible for vascular network architectonics.

IHME. (2017). *What does the world die from*. Retrieved from Our World in Data: https://ourworldindata.org/burden-of-disease

J C Schwarz, J. P. (2013). 3D Imaging of vascular networks for biophysical modeling. *Journal of Biomechanics*, 229-239.

Johannes H G M Van Beek, S. A. (1989). Regional myocardial flow heterogeneity explained with fractal networks. *American jounal of physiology*, H1670-80.

Kassab G S, B. J. (1997). Analysis of pig's coronary arterial blood flow and detailed anatomical data. *Biomed Eng 25*, 204-217.

Kassab G S, L. D. (1994). Morphometry of pig coronary venous netowrk. *american joual of physiology*, H2100-13.

Kassab GS, F. Y. (1994). Topoglogy and dimensions of pig coronary capillary network. *American Jounral of physiology*, H319-25.

Kassab, G. S. (2006). Scaling laws of vascular trees: of form and function. *American journal of Physiology-Heart and Circulatory Physiology*, H894-H903#.

Marcelo F. Di Carli, V. L. (2011). Cardiac PET/CT for the Evaluation of Known or Suspected Coronary Artery Disease. *Radiographics*, 1239-1254.

Marek Krzanowski, W. B. (2003). *Imaging of all three coronary arteries by transthoracic echocardiography. an illustrated guide.* BMC.

Moise, N. S. (1988). Echocardiography in canine and feline Cardiology. 113-156.

Murray, C. D. (1926). THE PHYSIOLOGICAL PRINCIPLE OF MINIMUM WORK APPLIED TO THE ANGLE OF BRANCHING OF ARTERIES. *Journal of General Physiology*, 835-841.

Rudolf Karch, F. N. (1999). A three-dimensional model for arterial tree representation, generated by constrained constructive optimization. *Computers in Biology and Medicine, Volume 29, Issue 1*, 19-38.

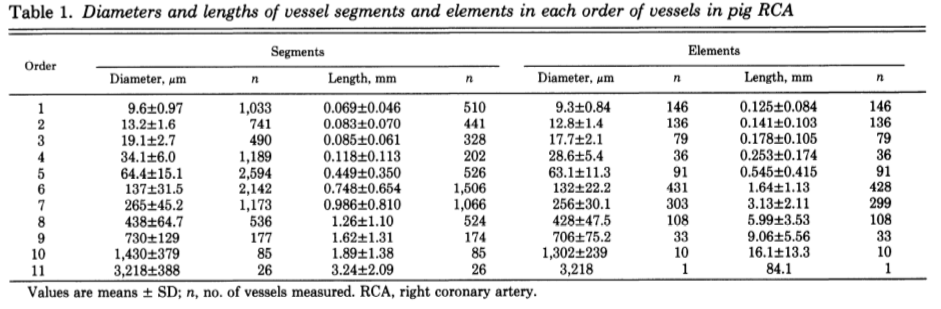
Wolfgang Schreiner, P. B. (1993). Computer-Optimization of Vascular Trees. *IEEE Transactions on Biomedical Engineering*, 482-491.

Wolfgang Schreiner, R. K. (1999). A three-dimensional model for arterial tree representation, generated by constrained constructive optimisation. *Computers in Biology and Medicine*, 19-38.

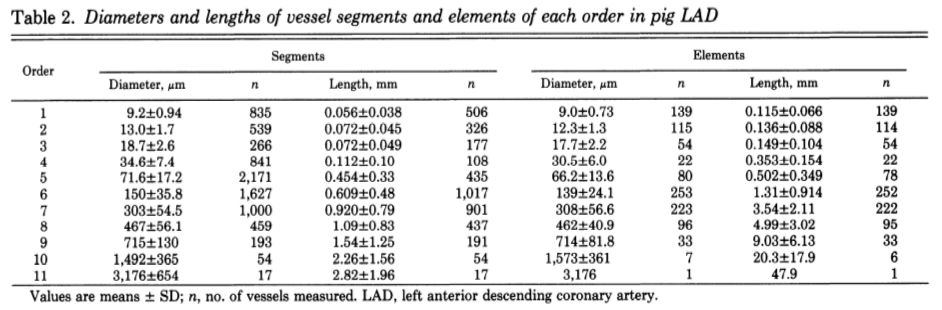
Y. Zhou, G. K. (1999). On the design of the coronary arterial tree: a generalization of Murray's law. *Pysics in medicine and biology*, 2929.

## Appendix

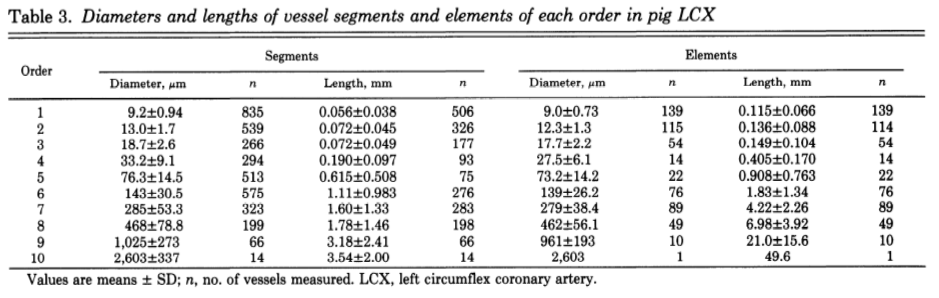
### Appendix 1.1 Diameters and lengths of vessel segments and elements in each order of vessels in pig RCA (G S Kassab, 1993)



### Appendix 1.2 Diameters and lengths of vessel segments and elements in each order of vessels in pig LAD (G S Kassab, 1993)



### Appendix 1.3 Diameters and lengths of vessel segments and elements in each order of vessels in pig RCX (G S Kassab, 1993)



### Appendix 1.4 Patient data set 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Proximal x** | **Proximal y** | **Distal x** | **Distal y** | **Parent** | **Daughter 1** | **Daughter 2** | **Flow Rate** |
| 1 | 1 | 2 | 1.1 | 1.6 | 0 | 2 | 3 | 11 |
| 2 | 1.1 | 1.6 | 1.2 | 1.4 | 1 | 6 | 7 | 8 |
| 3 | 1.1 | 1.6 | 0.85 | 1.4 | 1 | 5 | 4 | 3 |
| 4 | 0.85 | 1.4 | 0.87 | 1.35 | 3 | 0 | 0 | 1 |
| 5 | 0.85 | 1.4 | 0.6 | 1.2 | 3 | 19 | 18 | 2 |
| 6 | 1.2 | 1.4 | 1.3 | 1.15 | 2 | 20 | 21 | 2 |
| 7 | 1.2 | 1.4 | 1.21 | 1.2 | 2 | 8 | 9 | 6 |
| 8 | 1.21 | 1.2 | 1.1 | 1.1 | 7 | 0 | 0 | 1 |
| 9 | 1.21 | 1.2 | 1.15 | 0.8 | 7 | 10 | 11 | 5 |
| 10 | 1.15 | 0.8 | 1.25 | 0.7 | 9 | 0 | 0 | 1 |
| 11 | 1.15 | 0.8 | 1.2 | 0.5 | 9 | 12 | 13 | 4 |
| 12 | 1.2 | 0.5 | 1.1 | 0.4 | 11 | 14 | 15 | 2 |
| 13 | 1.2 | 0.5 | 1.3 | 0.4 | 11 | 16 | 17 | 2 |
| 14 | 1.1 | 0.4 | 1.11 | 0.2 | 12 | 0 | 0 | 1 |
| 15 | 1.1 | 0.4 | 0.9 | 0.2 | 12 | 0 | 0 | 1 |
| 16 | 1.3 | 0.4 | 1.5 | 0.4 | 13 | 0 | 0 | 1 |
| 17 | 1.3 | 0.4 | 1.5 | 0.2 | 13 | 0 | 0 | 1 |
| 18 | 0.6 | 1.2 | 0.55 | 1.1 | 5 | 0 | 0 | 1 |
| 19 | 0.6 | 1.2 | 0.65 | 1.1 | 5 | 0 | 0 | 1 |
| 20 | 1.3 | 1.15 | 1.5 | 1.05 | 6 | 0 | 0 | 1 |

### Appendix 1.5 Patient data set 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Proximal x** | **Proximal y** | **Distal x** | **Distal y** | **Parent** | **Daughter 1** | **Daughter 2** | **Flow Rate** |
| 1 | 1 | 2 | 0.9 | 1.6 | 0 | 2 | 3 | 11 |
| 2 | 0.9 | 1.6 | 0.8 | 1.4 | 1 | 6 | 7 | 8 |
| 3 | 0.9 | 1.6 | 1.15 | 1.4 | 1 | 5 | 4 | 3 |
| 4 | 1.15 | 1.4 | 1.13 | 1.35 | 3 | 0 | 0 | 1 |
| 5 | 1.15 | 1.4 | 1.4 | 1.2 | 3 | 19 | 18 | 2 |
| 6 | 0.8 | 1.4 | 0.7 | 1.15 | 2 | 20 | 21 | 2 |
| 7 | 0.8 | 1.4 | 0.79 | 1.2 | 2 | 8 | 9 | 6 |
| 8 | 0.79 | 1.2 | 0.9 | 1.1 | 7 | 0 | 0 | 1 |
| 9 | 0.79 | 1.2 | 0.85 | 0.8 | 7 | 10 | 11 | 5 |
| 10 | 0.85 | 0.8 | 0.75 | 0.7 | 9 | 0 | 0 | 1 |
| 11 | 0.85 | 0.8 | 0.8 | 0.5 | 9 | 12 | 13 | 4 |
| 12 | 0.8 | 0.5 | 0.9 | 0.4 | 11 | 14 | 15 | 2 |
| 13 | 0.8 | 0.5 | 0.7 | 0.4 | 11 | 16 | 17 | 2 |
| 14 | 0.9 | 0.4 | 0.89 | 0.2 | 12 | 0 | 0 | 1 |
| 15 | 0.9 | 0.4 | 1.1 | 0.2 | 12 | 0 | 0 | 1 |
| 16 | 0.7 | 0.4 | 0.5 | 0.4 | 13 | 0 | 0 | 1 |
| 17 | 0.7 | 0.4 | 0.5 | 0.2 | 13 | 0 | 0 | 1 |
| 18 | 1.4 | 1.2 | 1.45 | 1.1 | 5 | 0 | 0 | 1 |
| 19 | 1.4 | 1.2 | 1.35 | 1.1 | 5 | 0 | 0 | 1 |
| 20 | 0.7 | 1.15 | 0.5 | 1.05 | 6 | 0 | 0 | 1 |
| 21 | 0.7 | 1.15 | 0.55 | 1.15 | 6 | 0 | 0 | 1 |

### Appendix 1.6 Patient data set 3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Proximal x** | **Proximal y** | **Distal x** | **Distal y** | **Parent** | **Daughter 1** | **Daughter 2** | **Flow Rate** |
| 1 | 1 | 2 | 0.9 | 1.8 | 0 | 2 | 3 | 11 |
| 2 | 0.9 | 1.8 | 0.7 | 1.6 | 1 | 4 | 5 | 6 |
| 3 | 0.9 | 1.8 | 1.3 | 1.6 | 1 | 14 | 15 | 5 |
| 4 | 0.7 | 1.6 | 0.6 | 1.5 | 2 | 6 | 7 | 2 |
| 5 | 0.7 | 1.6 | 0.65 | 1.2 | 2 | 8 | 9 | 5 |
| 6 | 0.6 | 1.5 | 0.55 | 1.65 | 4 | 0 | 0 | 1 |
| 7 | 0.6 | 1.5 | 0.5 | 1.45 | 4 | 0 | 0 | 1 |
| 8 | 0.65 | 1.2 | 0.75 | 1 | 5 | 0 | 0 | 1 |
| 9 | 0.65 | 1.2 | 0.6 | 0.8 | 5 | 11 | 10 | 3 |
| 10 | 0.6 | 0.8 | 0.5 | 0.6 | 9 | 0 | 0 | 1 |
| 11 | 0.6 | 0.8 | 0.7 | 0.6 | 9 | 12 | 13 | 2 |
| 12 | 0.7 | 0.6 | 0.75 | 0.55 | 11 | 0 | 0 | 1 |
| 13 | 0.7 | 0.6 | 0.65 | 0.35 | 11 | 0 | 0 | 1 |
| 14 | 1.3 | 1.6 | 1.45 | 1.65 | 3 | 0 | 0 | 1 |
| 15 | 1.3 | 1.6 | 1.4 | 1.2 | 3 | 16 | 17 | 4 |
| 16 | 1.4 | 1.2 | 1.6 | 0.9 | 15 | 0 | 0 | 1 |
| 17 | 1.4 | 1.2 | 1.4 | 0.9 | 15 | 18 | 19 | 3 |
| 18 | 1.4 | 0.9 | 1 | 0.6 | 17 | 0 | 0 | 1 |
| 19 | 1.4 | 0.9 | 1.5 | 0.5 | 17 | 20 | 21 | 2 |
| 20 | 1.5 | 0.5 | 1.6 | 0.4 | 19 | 0 | 0 | 1 |
| 21 | 1.5 | 0.5 | 1.45 | 0.4 | 19 | 0 | 0 | 1 |

### Appendix 1.7 Patient data set 4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Proximal x** | **Proximal y** | **Distal x** | **Distal y** | **Parent** | **Daughter 1** | **Daughter 2** | **Flow Rate** |
| 1 | 1 | 2 | 1.1 | 1.8 | 0 | 2 | 3 | 11 |
| 2 | 1.1 | 1.8 | 1.3 | 1.6 | 1 | 4 | 5 | 6 |
| 3 | 1.1 | 1.8 | 0.7 | 1.6 | 1 | 14 | 15 | 5 |
| 4 | 1.3 | 1.6 | 1.4 | 1.5 | 2 | 6 | 7 | 2 |
| 5 | 1.3 | 1.6 | 1.35 | 1.2 | 2 | 8 | 9 | 5 |
| 6 | 1.4 | 1.5 | 1.45 | 1.65 | 4 | 0 | 0 | 1 |
| 7 | 1.4 | 1.5 | 1.5 | 1.45 | 4 | 0 | 0 | 1 |
| 8 | 1.35 | 1.2 | 1.25 | 1 | 5 | 0 | 0 | 1 |
| 9 | 1.35 | 1.2 | 1.4 | 0.8 | 5 | 11 | 10 | 3 |
| 10 | 1.4 | 0.8 | 1.5 | 0.6 | 9 | 0 | 0 | 1 |
| 11 | 1.4 | 0.8 | 1.3 | 0.6 | 9 | 12 | 13 | 2 |
| 12 | 1.3 | 0.6 | 1.25 | 0.55 | 11 | 0 | 0 | 1 |
| 13 | 1.3 | 0.6 | 1.35 | 0.35 | 11 | 0 | 0 | 1 |
| 14 | 0.7 | 1.6 | 0.55 | 1.65 | 3 | 0 | 0 | 1 |
| 15 | 0.7 | 1.6 | 0.6 | 1.2 | 3 | 16 | 17 | 4 |
| 16 | 0.6 | 1.2 | 0.4 | 0.9 | 15 | 0 | 0 | 1 |
| 17 | 0.6 | 1.2 | 0.6 | 0.9 | 15 | 18 | 19 | 3 |
| 18 | 0.6 | 0.9 | 1 | 0.6 | 17 | 0 | 0 | 1 |
| 19 | 0.6 | 0.9 | 0.5 | 0.5 | 17 | 20 | 21 | 2 |
| 20 | 0.5 | 0.5 | 0.4 | 0.4 | 19 | 0 | 0 | 1 |
| 21 | 0.5 | 0.5 | 0.55 | 0.4 | 19 | 0 | 0 | 1 |

### Appendix 1.8 Main Tree Generation Script

clear

tic

% Set Starting Conditions

MaxPoints =21; % Sets the number of points generated

MaxSegment = (MaxPoints\*2) + 1; % Total Number of segments

MaxPoints = MaxPoints+1; % For Caluclating r

Aperf = pi()\*0.05^2; % Area To be filled

load TestDataPatient3 % Loads Patient Data

Seg = TestDataPatient3; % Assigns data to Seg structure

Points = 200; % Random Points created for each iteration

BiPoints = 200; % Number of bifurcation Points Tested

u = 3.6\*10^-3; % Blood Viscosity

p = ((1.33\*10^4)-(7.98\*10^3)); % Blood Pressure

Qtot = 5\*10^-6; %8.33\*10^-6; % Total Flow Rate to LAD

Q = Qtot/ (MaxPoints+1); % Set the flow for each terminal end

K = 21; % Start after the initial 21 Segments

while K <= MaxSegment-2 % -2 as it is K + 2

P = (K-1)/2 + 2; % P is number of points currently

r = CircBound(Aperf,P,MaxPoints); % Calculate r of current Circle

Seg(:,1:4) = Seg(:,1:4)\*r; % Scale Segments

dthresh = sqrt(pi()\*(r^2)/P); % Set Distance threshold

dcrit = 0; % Reset Critical Distance

N = 1; % Start Counter for Points

RandPoint = r\*2\*rand(Points,2); % Set random points

% Continue to loop until a satisfactory point is found

while dcrit < dthresh

C = sqrt((RandPoint(N,1)-r)^2 + (RandPoint(N,2)-r)^2);% Check it lies within circle

if C < r

[Seg,dcrit] = pointcheck(Seg,RandPoint,K,N,dcrit);

end

N = N+1;

if N >= Points % If no random point passes

N = 1; % Reset Count

dthresh = dthresh\*0.9; % Lower dthresh

end

end

% The new Point has been found and is now going to be connected

% Loop through every possible segment connection

% M is the connection currently being tested

% Set a high Vmin to begin with to reset it if no point is found Vmin

% will still be 1000 at the end

Vmin = 1000;

for M = 1:K

Int = 0; % Set it to not initially intersect

% If it is connecting to the original Segments it may only

% bifurcate within a restricted range

if M <= 21

minx = min([Seg(M,1);Seg(M,3)]);

maxx = max([Seg(M,1);Seg(M,3)]);

miny = min([Seg(M,2);Seg(M,4)]);

maxy = max([Seg(M,2);Seg(M,4)]);

% Murrays law is given a large leway for these due to the

% restriction in bifurcation points

r1murrayscale = 1.5;

r2murrayscale = 0.5;

else

% Regular Murrays Law leniency

r1murrayscale = 1.2;

r2murrayscale = 0.8;

% Finds the area for the new bifurcation points

minx = min([Seg(M,1);Seg(M,3);Seg(K+2,3)]);

maxx = max([Seg(M,1);Seg(M,3);Seg(K+2,3)]);

miny = min([Seg(M,2);Seg(M,4);Seg(K+2,4)]);

maxy = max([Seg(M,2);Seg(M,4);Seg(K+2,4)]);

end

% Generate Random Bifurcation Points within allocated area

RandBiPoint = rand(BiPoints,2);

RandBiPoint(:,1) = minx + (maxx - minx)\*RandBiPoint(:,1); % Set random points

RandBiPoint(:,2) = miny + (maxy - miny)\*RandBiPoint(:,2);

% Loop through bifurcation points finding the optimal within

% constraints

for F = 1:BiPoints

% Check it lies within circle

C = sqrt((RandBiPoint(F,1)-r)^2 + (RandBiPoint(F,2)-r)^2);

if C < r

% Calculate l and r to find total volume of connection

l1 = sqrt((RandBiPoint(F,1)-Seg(M,1))^2 + (RandBiPoint(F,2)-Seg(M,2))^2);

l2 = sqrt((RandBiPoint(F,1)-Seg(M,3))^2 + (RandBiPoint(F,2)-Seg(M,4))^2);

l3 = sqrt((RandBiPoint(F,1)-Seg(K+2,3))^2 + (RandBiPoint(F,2)-Seg(K+2,4))^2);

% Calculates using Poiseuille

r1 = ((Seg(M,8)+1)\*Q\*l1\*8\*u/(p\*pi()))^(1/4);

r2 = (Seg(M,8)\*Q\*l2\*8\*u/(p\*pi()))^(1/4);

r3 = (Q\*l3\*8\*u/(p\*pi()))^(1/4);

Vnew = pi()\*((l1\*r1^2)+(l2\*r2^2)+(l3\*r3^2));

% If this is a smaller volume than previously

% found to be optimal

if Vnew <= Vmin

% Find the angle of the bifurcation

a = sqrt((Seg(M,3)-Seg(K+2,3))^2 + (Seg(M,4)-Seg(K+2,4))^2);

b = sqrt((RandBiPoint(F,1)-Seg(M,3))^2 + (RandBiPoint(F,2)-Seg(M,4))^2);

c = sqrt((RandBiPoint(F,1)-Seg(K+2,3))^2 + (RandBiPoint(F,2)-Seg(K+2,4))^2);

Angle = acosd((b^2 + c^2 - a^2)/(2\*b\*c));

if Angle <= 80

% Find the angle between the bifuraction and parent

a = sqrt((Seg(M,1)-Seg(K+2,3))^2 + (Seg(M,2)-Seg(K+2,4))^2);

b = sqrt((RandBiPoint(F,1)-Seg(M,1))^2 + (RandBiPoint(F,2)-Seg(M,2))^2);

c = sqrt((RandBiPoint(F,1)-Seg(K+2,3))^2 + (RandBiPoint(F,2)-Seg(K+2,4))^2);

Angle = acosd((b^2 + c^2 - a^2)/(2\*b\*c));

if Angle >= 130

% Find the angle between the bifuraction and parent

a = sqrt((Seg(M,1)-Seg(M,3))^2 + (Seg(M,2)-Seg(M,4))^2);

b = sqrt((RandBiPoint(F,1)-Seg(M,1))^2 + (RandBiPoint(F,2)-Seg(M,2))^2);

c = sqrt((RandBiPoint(F,1)-Seg(M,3))^2 + (RandBiPoint(F,2)-Seg(M,4))^2);

Angle = acosd((b^2 + c^2 - a^2)/(2\*b\*c));

if Angle >= 130

% Check the new bifurcation falls under Murray's law

% and do not grow in radius

r1murray = (r2^3 + r3^3)^(1/3)\*r1murrayscale;

r2murray = (r2^3 + r3^3)^(1/3)\*r2murrayscale;

rdiff = abs(r1 - r1murray);

if r1 <= r1murray && r1>= r2murray && r1 > r2 && r1 > r3

% Checks the parent Segment and its bifurcation obey Murray's law

ParSeg = Seg(M,5);

% As long as it is not segment 1

if ParSeg ~= 0

% Finds Segment Data

DauSeg1 = Seg(ParSeg,6);

DauSeg2 = Seg(ParSeg,7);

QDauSeg1 = Seg(DauSeg1,8);

QDauSeg2 = Seg(DauSeg2,8);

l1 = sqrt((Seg(ParSeg,1)-Seg(ParSeg,3))^2 + (Seg(ParSeg,2)-Seg(ParSeg,4))^2);

l2 = sqrt((Seg(DauSeg1,1)-Seg(DauSeg1,3))^2 + (Seg(DauSeg1,2)-Seg(DauSeg1,4))^2);

l3 = sqrt((Seg(DauSeg2,1)-Seg(DauSeg2,3))^2 + (Seg(DauSeg2,2)-Seg(DauSeg2,4))^2);

% If it is the new segment being tested the flow increases

if DauSeg1 == M

QDauSeg1 = QDauSeg1+1;

l2 = sqrt((Seg(DauSeg1,1)-RandBiPoint(F,1))^2 + (Seg(DauSeg1,2)-RandBiPoint(F,2))^2);

end

if DauSeg2 == M

QDauSeg2 = QDauSeg2+1;

l3 = sqrt((Seg(DauSeg2,1)-RandBiPoint(F,1))^2 + (Seg(DauSeg2,2)-RandBiPoint(F,2))^2);

end

% Calculate r and Murrays law

r1 = ((Seg(ParSeg,8)+1)\*Q\*l1\*8\*u/(p\*pi()))^(1/4);

r2 = (QDauSeg1\*Q\*l2\*8\*u/(p\*pi()))^(1/4);

r3 = (QDauSeg2\*Q\*l3\*8\*u/(p\*pi()))^(1/4);

r1murray = (r2^3 + r3^3)^(1/3)\*r1murrayscale;

r2murray = (r2^3 + r3^3)^(1/3)\*r2murrayscale;

rdiff = abs(r1 - r1murray);

end

if r1 <= r1murray && r1>= r2murray

if r1 > r2 && r1> r3

% Check the other Segment and its daughters for Murray's

DauSeg1 = Seg(M,6);

DauSeg2 = Seg(M,7);

% Provded it is not terminal

if DauSeg1 ~=0

l2 = sqrt((Seg(DauSeg1,1)-Seg(DauSeg1,3))^2 + (Seg(DauSeg1,2)-Seg(DauSeg1,4))^2);

l3 = sqrt((Seg(DauSeg2,1)-Seg(DauSeg2,3))^2 + (Seg(DauSeg2,2)-Seg(DauSeg2,4))^2);

l1 = sqrt((RandBiPoint(F,1)-Seg(M,3))^2 + (RandBiPoint(F,2)-Seg(M,4))^2);

r1 = ((Seg(M,8)+1)\*Q\*l1\*8\*u/(p\*pi()))^(1/4);

r2 = (Seg(DauSeg1,8)\*Q\*l2\*8\*u/(p\*pi()))^(1/4);

r3 = (Seg(DauSeg2,8)\*Q\*l3\*8\*u/(p\*pi()))^(1/4);

r1murray = (r2^3 + r3^3)^(1/3)\*r1murrayscale;

r2murray = (r2^3 + r3^3)^(1/3)\*r2murrayscale;

rdiff = abs(r1 - r1murray);

end

if r1 <= r1murray && r1>= r2murray

if r1 >r2 %&& r1 >r3

% Intersection Check

for Z = 1:K

% Skip the intersection check for the point it connects to

% Checks if it intersects with any of the previous segments

if M ~= Z && Int ==0

% This checks the new daughter segment

line1 = [Seg(Z,1), Seg(Z,2); Seg(Z,3), Seg(Z,4)];

line2 = [RandBiPoint(F,1), RandBiPoint(F,2); Seg(K+2,3), Seg(K+2,4)];

Int = LineIntersectDau(Int,line1,line2);

% This checks the old daughter segments

line2(2,1)= Seg(M,3);

line2(2,2) = Seg(M,4);

Int = LineIntersectDau(Int,line1,line2);

% This tests the parent segment

if Seg(Z,1) ~= Seg(M,1)

if Seg(Z,3) ~= Seg(M,1)

Int = LineIntersectPar(Seg(Z,:),Int,RandBiPoint(F,:),Seg(M,1),Seg(M,2));

end

end

end

end

if Int == 0

Vmin = Vnew;

NewBiffx = RandBiPoint(F,1);

NewBiffy = RandBiPoint(F,2);

L = M;

end

end

end

end

end

end

end

end

end

end

end

end

% This is the end of the for finding a bifurcation point loop

end

Int =0;

% This saves all the new data necessary

if Vmin ~= 1000 && Int ==0

% Sets the points for the new segments

Seg(K+1,1) = NewBiffx;

Seg(K+1,2) = NewBiffy;

Seg(K+1,3) = Seg(L,3);

Seg(K+1,4) = Seg(L,4);

% Changes the parent segments end point

Seg(L,3) = NewBiffx;

Seg(L,4) = NewBiffy;

% Changes the starting point of the segment 3,4 already saved

Seg(K+2,1) = NewBiffx;

Seg(K+2,2) = NewBiffy;

% Assigns the parent to it

Seg(K+1,5) = L;

Seg(K+2,5) = L;

% Assigns parents to old daughters

Daughter1 = Seg(L,6);

Daughter2 = Seg(L,7);

% K + 1 always connecting segment so becomes new parent

if Daughter1 >= 1

Seg(Daughter1,5) = K+1;

Seg(Daughter2,5) = K+1;

end

% Assign the daughter to the new connecting segment

Seg(K+1,6) = Seg(L,6);

Seg(K+1,7) = Seg(L,7);

% Assign the new daughters to the parent

Seg(L,6) = K+1;

Seg(L,7) = K+2;

% Assigns flow to the new point and the connection point

Seg(K+2,8) = 1;

Seg(K+1,8) = Seg(L,8);

V = 1;

% Recursively Increases flow up the tree to the root

while Seg(L,5) >=1

Seg(L,8) = Seg(L,8) + 1;

L = Seg(L,5);

end

% Root's flow always increased by 1 and is not affected by previous

Seg(1,8) = Seg(1,8) + 1;

% Calculates the radius of the new segments

l = sqrt((Seg(K,1)-Seg(K,3))^2 + (Seg(K,2)-Seg(K,4))^2);

dp = p/25;

Seg(K,9)= ((Seg(K,8)\*Q\*l\*8\*u/(p\*pi()))^(1/4));

l = sqrt((Seg(K+1,1)-Seg(K+1,3))^2 + (Seg(K+1,2)-Seg(K+1,4))^2);

Seg(K+1,9)= ((Seg(K+1,8)\*Q\*l\*8\*u/(p\*pi()))^(1/4));

K = K +2; % This is left unsupressed to give the progress

end

Seg(:,1:4) = Seg(:,1:4)/r; % Rescaled back to neutral

NewBiffx= 0;

NewBiffy = 0;

end

% For some reason it is one extra at the end

Seg(1,8) = Seg(1,8) - 1;

%Scale Tree to Final size

P = (K-1)/2 + 2; % P is number of points created

r = CircBound(Aperf,P,MaxPoints); % Calculate r of current Circle

Seg(:,1:4) = Seg(:,1:4)\*r;

rsize = r;

for I = 1:K

V = 0;

T = I;

while Seg(T,5) >=1

T = Seg(T,5);

V = V+1;

end

Seg(I,11) = V;

end

MaxBif = max(Seg(:,11));

% Calculates r for every segment based on the length and the flow rate

% through it

VTot = 0;

for E = 1:MaxSegment

l = sqrt((Seg(E,1)-Seg(E,3))^2 + (Seg(E,2)-Seg(E,4))^2);

Seg(E,9) = (Seg(E,8)\*Q\*l\*8\*u/((p/MaxBif)\*pi()))^(1/4);

VTot = VTot+ (pi()\*(l\*Seg(E,9)^2));

end

% Scales the radius for the plot so that its line width is not x10^-5

Seg(:,10) = Seg(:,9)\*10000;

% Plots visual representation of vessels

figure(1)

title('Generated Coronary Vessel Tree')

for E = 1:MaxSegment

x1 = [Seg(E,1);Seg(E,3)];

x2 = [Seg(E,2);Seg(E,4)];

r = Seg(E,10);

plot(x1,x2,'black','Linewidth',r)

hold on

end

h = circle(rsize,rsize,rsize); % Plot circle boundary

% Plots the average radius against the number of bifuractions down the tree

for I = 1:K

V = 0;

T = I;

while Seg(T,5) >=1

T = Seg(T,5);

V = V+1;

end

Seg(I,11) = V;

end

MaxBif = max(Seg(:,11));

figure(2)

title('Average Segment Diameter Along Vessel Tree')

xlabel('Bifurcation level')

ylabel('Average Segment diameter (mm)')

for N = 0:MaxBif

k = Seg(:,11);

Location = find(k==N);

Radius = 0;

for K = 1:size(Location)

Radius = Radius + Seg(Location(K),9);

end

Radius = Radius/K;

RadiusPlot(N+1,1)= N;

RadiusPlot(N+1,2) = Radius;

end

RadiusPlot(1,1) = 0;

RadiusPlot(1,2) = Seg(1,9);

RadiusPlot(:,2) = RadiusPlot(:,2)\*2000;

plot(RadiusPlot(:,1),RadiusPlot(:,2));

VTot

RadiusPlot(:,2)

toc

Compuationtime = toc;

### Appendix 1.9 Strahler order Script

% First load the data to be processed

% This While loop creates the strahler order for each segment

% Initialise the loop

Seg(:,12) = 0;

Increase = 1;

N = 0;

MaxSegment = 8001;

% While the Strahler order is still increasing

while Increase == 1

Increase = 0;

k = Seg(:,12); % k is the Current strahler orders of all segments

Location = find(k==N); % Find all segments of the current order

% Loop through all segments of this order

for K = 1:size(Location)

if Seg(Location(K),5) ~= 0

%find the parent and daughter segment ID's

Parent = Seg(Location(K),5);

Daughter1 = Seg(Parent,6);

Daughter2 = Seg(Parent,7);

% If both Daughters orders are the same increase the parents by 1

if Seg(Daughter1,12) == Seg(Daughter2,12)

Seg(Parent,12) = N +1;

Increase = 1;

else

% Otherwise set the parent to be the higher of the daughters

if Seg(Daughter1,12) >= Seg(Daughter2,12)

Seg(Parent,12) = Seg(Daughter1,12);

else

Seg(Parent,12) = Seg(Daughter2,12);

end

end

end

end

N= N+1;

end

% The root is always the highest order and is not set within this due to

% the parent of it being 0 causing an error

Seg(1,12) = N-1;

% The diamater based method is repeated 5 times which is more than enough

% for convergences stated in Kassab's method

for F = 1:5

% maxBif is the highest strahler order taken from the bifurcation level

% previously

MaxBif = max(Seg(:,12));

% This finds the mean and standard deviation of each order

for N = 0:MaxBif

k = Seg(:,12);

Location = find(k==N);

Radius = 0;

for K = 1:size(Location)

Radius = Radius + Seg(Location(K),9);

deviationarray(K) = Seg(Location(K),9);

end

Radius = Radius/K;

RadiusPlot(N+1,1)= N;

RadiusPlot(N+1,2) = Radius;

RadiusPlot(N+1,3) = std(deviationarray);

end

% Set the diameter boundaries of each order

for N = 0:MaxBif-1

RadiusPlot(N+1,4) = (RadiusPlot(N+2,2) + RadiusPlot(N+2,3) + RadiusPlot(N+1,2) - RadiusPlot(N+1,3))/2;

end

% Set the Highest strahler order boundary to be just below the root to

% restrict it to be exclusively the root segment

RadiusPlot(MaxBif,4)=RadiusPlot(MaxBif,4)\*0.99;

% Loop through each segment and set its new diamter based strahler

% order

for K = 1:MaxSegment

for I = 1:MaxBif

if Seg(K,9) > RadiusPlot(I,4);

Seg(K,12)= I;

end

end

end

end

% Find the mean and standard deviations of each order for the plot this

% could be removed by repeating the loop one more time without completing

% the end section

MaxBif = max(Seg(:,12));

for N = 0:MaxBif

k = Seg(:,12);

Location = find(k==N);

Radius = 0;

for K = 1:size(Location)

Radius = Radius + Seg(Location(K),9);

deviationarray(K) = Seg(Location(K),9);

Size = size(Location);

RadiusPlot(N+1,4) = Size(1);

end

Radius = Radius/K;

RadiusPlot(N+1,1)= N;

RadiusPlot(N+1,2) = Radius;

RadiusPlot(N+1,3) = std(deviationarray);

end

% This finds the legnths mean and standard deviation at each order

for N = 0:MaxBif

k = Seg(:,12);

Location = find(k==N);

Length = 0;

for K = 1:size(Location)

Length = Length + sqrt((Seg(Location(K),1)-Seg(Location(K),3))^2 + (Seg(Location(K),2)-Seg(Location(K),4))^2);

deviationarray(K) = sqrt((Seg(Location(K),1)-Seg(Location(K),3))^2 + (Seg(Location(K),2)-Seg(Location(K),4))^2);

Size = size(Location);

RadiusPlot(N+1,4) = Size(1);

end

Length = Length/K;

RadiusPlot(N+1,1)= N;

RadiusPlot(N+1,5) = Length;

RadiusPlot(N+1,6) = std(deviationarray);

end

% Increase the order to begin at the arteriole level

RadiusPlot(:,1) = RadiusPlot(:,1) + 5

% scale the radius to be diamater in millimeters

RadiusPlot(:,2) = RadiusPlot(:,2)\*2000;

% Various code for plotting data

plot(RadiusPlot(:,1),RadiusPlot(:,5))

plot(PointsRange(:,1),PointsRange(:,2))

hold on

plot(PointsRange(:,1),PointsRange(:,3),'k')

plot(PointsRange(:,1),PointsRange(:,4),'k')

plot(KasssabLADMorph(:,1),KasssabLADMorph(:,2),'b')

plot(KasssabLADMorph(:,1),KasssabLADMorph(:,3),'b')

plot(KasssabLADMorph(:,1),KasssabLADMorph(:,4),'b')

title('Comparison with Morphometric data')

xlabel('Strahler Order')

ylabel('Average Segment diameter (mm)')

legend('Model','Schreiners Model')

### Appendix 1.10 Function pointcheck

function [Seg,dcrit] = pointcheck(Seg,RandPoint,K,N,dcrit)

% Returns Dcrit calculated if it is smaller than previous dcrits based on

% distance from end and centre points

% Reset Dmin

dmin = 1000000;

for I = 1:K

% Calculate distance between ends and centre

t(1) = abs(sqrt((RandPoint(N,1) - Seg(I,3))^2 + (RandPoint(N,2) - Seg(I,4))^2));

t(2) = abs(sqrt((RandPoint(N,1) - Seg(I,1))^2 + (RandPoint(N,2) - Seg(I,2))^2));

% Find halfway point

distx = (Seg(I,1)+Seg(I,3))/2;

disty = (Seg(I,2)+Seg(I,4))/2;

t(3) = abs(sqrt((RandPoint(N,1) - distx)^2 + (RandPoint(N,2) - disty)^2));

distx = Seg(I,1) - (Seg(I,1)-Seg(I,3))/4;

disty = Seg(I,2) - (Seg(I,2)-Seg(I,4))/4;

t(4) = abs(sqrt((RandPoint(N,1) - distx)^2 + (RandPoint(N,2) - disty)^2));

distx = Seg(I,1) - (Seg(I,1)+Seg(I,3))\*3/4;

disty = Seg(I,2) - (Seg(I,2)+Seg(I,4))\*3/4;

t(5) = abs(sqrt((RandPoint(N,1) - distx)^2 + (RandPoint(N,2) - disty)^2));

% Find the smallest distance from all segments

if min(t) < dmin

dmin = min(t);

end

end

% If this distance is greater than the previous largest distance update the

% value

if dmin > dcrit

dcrit = dmin;

Seg(K+2,3) = RandPoint(N,1);

Seg(K+2,4) = RandPoint(N,2);

end

end

### Appendix 1.11 Function LineIntersectPar

function [Int] = LineIntersectPar(Seg,Int,RandBiPoint,X1,Y1)

% Int is set to 1 if line intersects, if line does not intersect int stays

% the same

% Checks it intersects with the line

x=[Seg(1) Seg(3) RandBiPoint(1) X1];

y=[Seg(2) Seg(4) RandBiPoint(2) Y1];

dt1=det([1,1,1;x(1),x(2),x(3);y(1),y(2),y(3)])\*det([1,1,1;x(1),x(2),x(4);y(1),y(2),y(4)]);

dt2=det([1,1,1;x(1),x(3),x(4);y(1),y(3),y(4)])\*det([1,1,1;x(2),x(3),x(4);y(2),y(3),y(4)]);

if(dt1<=0 & dt2<=0)

Int = 1; %If lines intesect

end

end

### Appendix 1.12 Function LineIntersectDau

function [Int] = LineIntersectDau(Int,line1,line2)

% Int is set to 1 if line intersects, if line does not intersect int stays

% the same

% Checks it intersects with the line

%line1 = [Seg(Z,1), Seg(Z,2); Seg(Z,3), Seg(Z,4)]

%line2 = [RandBiPoint(1), RandBiPoint(2); X1, Y1]

slope = @(line) (line(2,2) - line(1,2))/(line(2,1) - line(1,1));

m1 = slope(line1);

m2 = slope(line2);

intercept = @(line,m) line(1,2) - m\*line(1,1);

b1 = intercept(line1,m1);

b2 = intercept(line2,m2);

xintersect = (b2-b1)/(m1-m2);

yintersect = m1\*xintersect + b1;

isPointInside = @(xint,myline) ...

(xint >= myline(1,1) && xint <= myline(2,1)) || ...

(xint >= myline(2,1) && xint <= myline(1,1));

inside = isPointInside(xintersect,line1) && ...

isPointInside(xintersect,line2);

if inside == 1 && line1(1) ~= line2(1) && line1(3) ~= line2(3)

Int = 1;

end

end

### Appendix 1.13 Function CircBound

function [r] = CircBound(Aperf,P,MaxPoint)

%Provides the radius of the circle currently

r = sqrt(Aperf\*P/(pi()\*MaxPoint));

end

### Appendix 1.14 Function Circle

function h = circle(x,y,r)

hold on

th = 0:pi/50:2\*pi;

xunit = r \* cos(th) + x;

yunit = r \* sin(th) + y;

h = plot(xunit, yunit);

hold off