Erw



(NHCS, n.d.)

Diane Young

S46969909

Final report

BIOE3001: Quantitative Methods in Biomedical Engineering

Table of Contents

[Model Formulation 2](#_Toc148943818)

[Introduction 2](#_Toc148943819)

[Model Formulation 3](#_Toc148943820)

[Parameter Identification 4](#_Toc148943821)

[Model Implementation 6](#_Toc148943822)

[Formulation 6](#_Toc148943823)

[Model Assessment 7](#_Toc148943824)

[Data Usage 7](#_Toc148943825)

[Parameter Estimation 8](#_Toc148943826)

[Parameter Sensitivity 8](#_Toc148943827)

[Accuracy, Complexity, Future 9](#_Toc148943828)

[Bibliography 10](#_Toc148943829)

[Appendix 13](#_Toc148943830)

[Appendix 1: Km/Kmax values 13](#_Toc148943831)

[Appendix 2: Patient Data 13](#_Toc148943832)

[Appendix 3: Healthy Data 13](#_Toc148943833)

[Appendix 4: Patients Filtered Graphs 14](#_Toc148943834)

[Appendix 5: Averages of Patients Graphs 15](#_Toc148943835)

[Appendix 6: LDL vs Statin Graph Parameterised 16](#_Toc148943836)

[Appendix 7: Model Formulations 16](#_Toc148943837)

[Appendix 8: Patient Data Python (Filtered) 25](#_Toc148943838)

[Appendix 9: Patient Data Python Averages 28](#_Toc148943839)

[Appendix 10: Healthy Data Python 31](#_Toc148943840)

[Appendix 11: Patient Parameterisation Python 38](#_Toc148943841)

[Appendix 12: Parameter Sensitivity Python Script 47](#_Toc148943842)

# Model Formulation

## Introduction

Atherosclerosis, a chronic inflammatory disease that affects the arteries, is the underlying cause of 50% of all deaths in Western society (Pahwa & Jialal, 2023). Characterised by the accumulation of fatty deposits, cholesterol, cellular waste products, calcium and other substances within the walls of the arteries, creating plaques (Victor Chang Cardiac Research Institute, n.d.), which gradually narrow and harden the arteries, reducing blood flow (Mayo Clinic, n.d.).

Atherosclerosis begins with endothelial injury; triggered by factors such as high blood pressure or cholesterol (John Hopkins Medicine, n.d.). This injury allows low-density lipoprotein (LDL) cholesterol to enter the artery, provoking inflammation and becoming oxidised. Oxidised LDL is taken up by macrophages, turning them into foam cells, a hallmark feature of early atherosclerotic lesions (Peramaiyan Rajendran, 2019). Over time there is smooth muscle cell proliferation, extracellular matrix deposition, calcification, plaque rupture and finally thrombosis. Over time, smooth muscle cells in the artery wall begin to proliferate and migrate to the site of the injury. These cells contribute to the growth of the atherosclerotic plaque. As the plaque matures, it contains not only foam cells and smooth muscle cells but also deposits of extracellular matrix components like collagen and proteoglycans. Calcium deposits may accumulate within the plaque, making it harder and more rigid (Cleveland Clinic, n.d.).

Atherosclerotic plaques can become unstable and prone to rupture. When a plaque ruptures, it releases cholesterol and tissue factors, triggering the formation of blood clots (thrombosis) within the artery. If a clot blocks blood flow, it can lead to a heart attack or stroke. Wall shear stress (WSS) plays an important role in the process, with increasing WSS, the risk of thrombosis or rupture will also increase (Katharina Urschel, 2021).

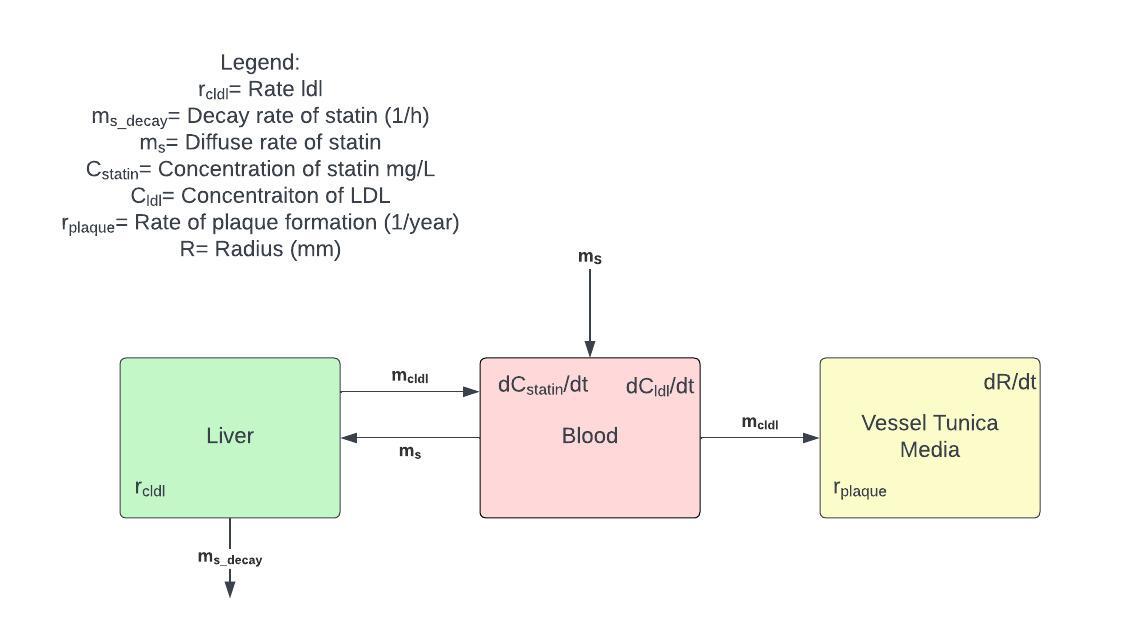
Treatment options for atherosclerosis include diet and lifestyle changes, to lower LDL cholesterol, improve blood pressure and maintain a healthy weight. Surgical options for treating atherosclerosis, include angioplasty to open narrowed arteries or coronary artery bypass grafting to bypass blocked or narrowed arteries (News Medical and Life Sciences, n.d.). Pharmacologically atherosclerosis is treated using medications that lower the LDL to reduce risks of blood clots. The focus of this paper will be on statins. HMG-CoA reductase is an enzyme that is responsible for converting HMG-CoA into Mevalonate, which is a precursor in the cholesterol synthesis pathway (Talreja, Kerndt, & Cassagnol., 2023). Statins are among the initial medications used to treat atherosclerosis, effectively lowering LDL cholesterol, through several mechanisms. By inhibiting HMG-CoA reductase which plays a critical role in cholesterol synthesis, increasing LDL receptor expression enhancing the body’s ability to remove LDL cholesterol, and modulating inflammatory responses. This medication is prescribed to people who typically have an LDL concentration above 1900 mmol/m3, that diet and exercise cannot reduce. If the person has had a previous stroke, heart attack or peripheral artery disease (British Heart Foundation, n.d.). As statin is the most common medication used for treating atherosclerosis (Mayo Clinic, 2023) it is important to understand the mechanisms behind it.

Figure 1: Balance Volume Diagram

## Model Formulation

A summary of these mechanisms that will be explored in this paper can be seen in the balance volume diagram shown in Figure 1.

The mechanism that was explored to model atherosclerosis was the viscosity affecting the wall shear stress. From there the following equations were made to explain these terms in terms of what was able to be measured in Table 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Equation # | Equation | Explanation | Reference | Fits into Equation |
| 1 |  | Overall wall shear stress based on the viscosity equation | (Weirui Lei, 2023) | - |
| 2 |  | Change in velocity based on change in radius | (Myriam Cilla, 2014) | 1 |
| 3 |  | Change of radius based on time | (Dorota Formanowicz, A Control-Theoretic Model of Atherosclerosis, 2019) | 1,2 |
| 4 |  | Viscosity based on shear stress equation | (Mohammed Ameenuddin, 2019) | 1 |
| 5 |  | Equations to be inputted based on haematocrit, to be inputted into the final equation | 4 |
| 6 |  | Haematocrit based on LDL concentration | (Michael B. Fessler, 2013) | 5 |
| 7 |  | Shear rate based on radius | (Centre for Industrial Rheology, n.d.) | 4 |

Table 1: Equations Used Throughout Model

By using data from (Dorota Formanowicz, A Control-Theoretic Model of Atherosclerosis, 2019) the dR/dt was found to be the equation above. As this model was not modelling LDL and is only a simulation this would be less relevant.

C is the carrying capacity which determines the health status of the patient, and the maximum plaque size when they pass away (Myriam Cilla, 2014). With the treated LDL, this carrying capacity would be approximately 30% less than usual as statins affect the LDL by approximately 30% (Jimyon Kim, A Population Pharmacokinetic–Pharmacodynamic Model for Simvastatin that Predicts Low-Density Lipoprotein-Cholesterol Reduction in Patients with Primary Hyperlipidaemia, 2011).

When the statin first enters the body it will first have to go through the circulatory system to reach the liver where it will be used (ms).

Equation 8, fits into Equation 5:

Equation 9:

As simvastatin is a medication that affects an enzyme and an enzyme uses Michaelis-Menten reaction kinetics, the effect of statin on LDL (dLDL) will be affected by these reaction kinetics.

Equation 10, fits into Equation 5,8,9

Based off sources such as (Jimyon Kim, A Population Pharmacokinetic–Pharmacodynamic Model for Simvastatin that Predicts Low-Density Lipoprotein-Cholesterol Reduction in Patients with Primary Hyperlipidaemia, 2011)this is what was found to be the equation, with the peak of simvastatin medication being before:

Equation 12, fits into equation 9:

This equation was used to describe the statins' effect on the LDL concentration as it does not affect it directly only indirectly.

## Parameter Identification

Table 1 describes all of the parameters that are to be used and some explanations of the accuracy and usefulness of each parameter:

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Name of Parameter** | **Values/ Explanation** | **Reference** |
| CcLDL | Concentration of LDL | <1800 mmol/m3 (Healthy) | (CSIRO, n.d.) |
| >2000 mmol/m3 (Unhealthy) |
| Uo | Velocity of Tibial artery | 0.4 m/sec (Typically) | (Jeffrey D. Crawford MD, 2015) |
| rcldl | Normal LDL production rate | 1800 mmol/m^3 | (CSIRO, n.d.) |
| kstatin | Rate of statin effectiveness | 20-30% | (Study of the Effectiveness of Additional Reductions in Cholesterol and Homocysteine (SEARCH) Collaborative Group, 2011) |
| rdecay | The rate of statin being used is not shown in dldl | 0.5 1/h | Approximated |
| λ | Half-life of simvastatin | 5 hours | (Marilisa Bove, 2017) |
| r | The normal radius of an anterior tibial artery | 5 mm during the systolic phase | (Alaa Eddin Nwilati, 2019) |
| x | From literature (Initial thickness) | Healthy=0.04mm | (Dorota Formanowicz, A Control-Theoretic Model of Atherosclerosis, 2019) |
| Unhealthy=0.05mm |
| c | From literature (Rate of growth) | Healthy= 1mm/year |
| Unhealthy=1.8mm/year |
| Km | Michaelis-menten rate | 2 mmol/L.s | Trial and Error |
| Kmax | Michaelis-menten maximum rate | 0.02 mmol/L.s | Trial and Error |
| Cs | Concentration of statin | 0-5 ng/mL dependent on time within the cycle | (Jimyon Kim, A Population Pharmacokinetic–Pharmacodynamic Model for Simvastatin that Predicts Low-Density Lipoprotein-Cholesterol Reduction in Patients with Primary Hyperlipidaemia, 2011) |
| Time (h) | Time in hours | Statin is taken every 24 hours, effects of statin can be seen within a few weeks | (NHS, n.d.) |
| Time (y) | Time in years | For other variables that cannot be seen throughout a few weeks (change in radius, WSS, etc.), statin dosage is reevaluated every year |  |
| K1 | Rate before peak | 2.43 ng/mL/h | Approximated from (Jimyon Kim, A Population Pharmacokinetic–Pharmacodynamic Model for Simvastatin that Predicts Low-Density Lipoprotein-Cholesterol Reduction in Patients with Primary Hyperlipidaemia, 2011) |
| K2 | Rate after peak 1 | 4.985 ng/mL/h |
| K3 | Rate after peak 2 | 0.13 ng/mL/h |

Table 1: Parameter Values and Explanations When Needed

There was no reliable Km or Kmax for the effectiveness of statin, which was implemented and educated guessed, through trial and error in the Python model to determine the approximate efficacy rate of statin on LDL. This was also put through Excel to double-check the results and this can be found in Appendix 1.

# Model Implementation

## Formulation

The LDL concentration was modelled over two different time scales. The first way it was modelled in weeks along with the concentration of statin, is to show statin effect on the LDL concentration only in a couple of weeks, of which the mechanisms of LDL and statin affect it.

This statin concentration is based on the model in (Jimyon Kim, A Population Pharmacokinetic–Pharmacodynamic Model for Simvastatin that Predicts Low-Density Lipoprotein-Cholesterol Reduction in Patients with Primary Hyperlipidaemia, 2011). Although the patient dosage was 40mg, only about 5-10% of it can be seen in the model and this is because this is measured in terms of blood concentration. The simvastatin that is measured is when it is in the absorption or the systemic circulation phase (Davis, 2014). Most LDL being able to be detected during the systemic circulation phase where it goes and affects the different parts of the body that it needs to affect. Simvastatin has a moderate to high systemic circulation rate than other statin medication types (Marianne K. DeGorter, 2013). For the LDL concentration, realistically this wouldn’t be as linear as it is seen, however, based on the calculations collected this is what arose in Figure 2:

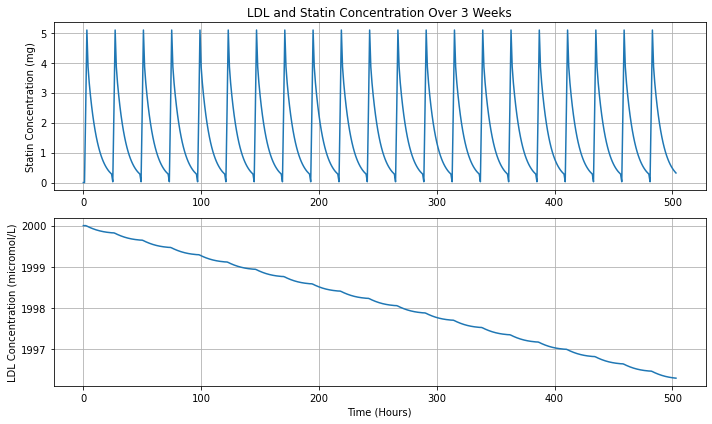


Figure 2: Statin and LDL vs Time (Weeks)

LDL was also modelled in years, as the change of radius and WSS would only be able to be seen in long-term statin usage. This data and the equation that was made for it were based on literature (Reeskamp, 2021), with the treated patient being evaluated every year to ensure that the dosage was being effectively evaluated.

A graph with different colored lines

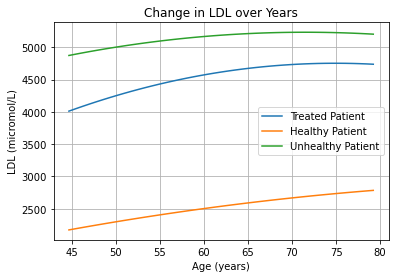
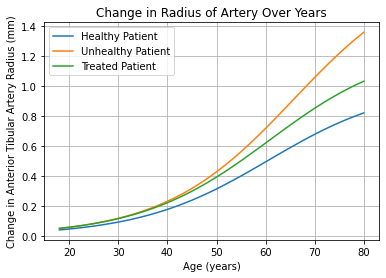
Description automatically generated

Figure 5: WSS vs Time (Years)

Figure 3: Radius of Artery vs Time

Figure 4: LDL vs Time (Years)

As it is hard to see the change in radius over a couple of weeks, the change in radius was modelled in years. This was based on the equation formulated in (Dorota Formanowicz, A Control-Theoretic Model of Atherosclerosis, 2019). LDL affects the thickness of the artery according to the same statin effect that the long-term LDL would have.

As WSS is also hard to see over only a couple of weeks it was modelled in years. This is based on both the long-term model for change in the artery and change in LDL over the years. Modelling this is crucial because it reveals the fundamental mechanisms underlying LDL and highlights the immense danger it poses.

All of the Python code for this section can be found in Appendix 7.

# Model Assessment

## Data Usage

The patient data was initially separated for each individual, allowing clear identification of each patient’s dataset. Following this, high and low-pass filters were used to process the data (Figure 6 and Appendix 8). However, these filtering techniques did not yield any significant improvements or changes in the dataset. Subsequently, the R-R interval was calculated for each patient using Python code, despite this effort this data was not used in the model, however, the results can be found in Appendix 8. Due to the R-R not being used, the average systolic and diastolic values for each patient were evaluated and compared with the corresponding values from the healthy data set which can be found in Figure 7, Appendix 9 and 10. unfortunately, during the conversion of the patient data into flowrates, issues were encountered that made some values appear incoherent, such as negative velocities, the summary of this data can be found in Appendix 2 and 3.

As the data might have been interfered with by noise or other external factors, a high pass and low pass filter was then applied to the data:

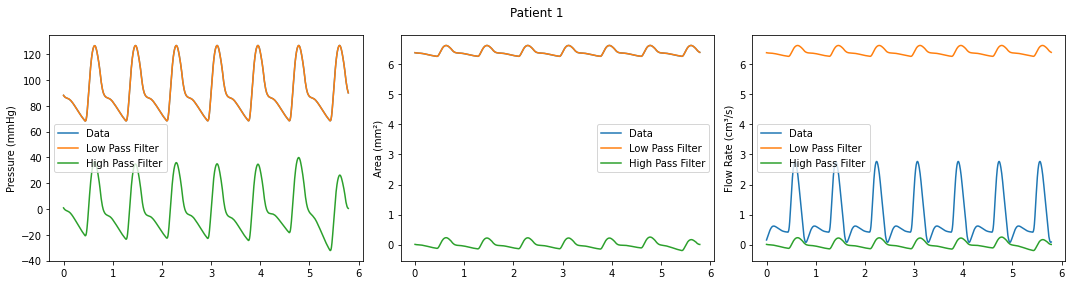


Figure 6: Applying Low Pass and High Pass Filters on Patient 1 Data

A graph with blue lines and orange dots

Description automatically generated

Figure 7: Maximum and Minimum Points (x) on Patient 1 Data

It was determined that Patients 3,4 and 5 were atherosclerotic, with Patient 2 being within healthy ranges and Patient 1 having a high blood pressure but exhibiting no evidence of atherosclerotic plaque growth. Patient 1 should be evaluated, and lifestyle changes should be made to ensure that they do not become atherosclerotic. Patients 3, 4 and 5 were determined to have significant atherosclerotic plaque growth, with patient 1 exhibiting warning signs of potentially developing atherosclerosis in the future.

Patient 3 data was determined to be the patient to be modelled as they are significantly ill. Due to only having the area, pressure and flow rate of this patient limited data could be achieved.

## Parameter Estimation

By assuming that the artery measured was a perfect circle, the radius of the patient was able to be found by using the area (Appendix 2). Then, by using the average radius of the artery (5mm) the change in radius over time was found. It was corrected to the correct units to be implemented into the model that was created, the calculation can be found in Appendix 11. The LDL was taken based on literature where the average LDL was found for the patient who was 45 years old. This patient model is ‘Parameterised’ and modelled against the ‘Model’, which is the treated model from the Model Implementation: Model Formulation section on page 6, the parameters in the original model that can be parameterised with the patient data can be seen in Figures 8 and 9:

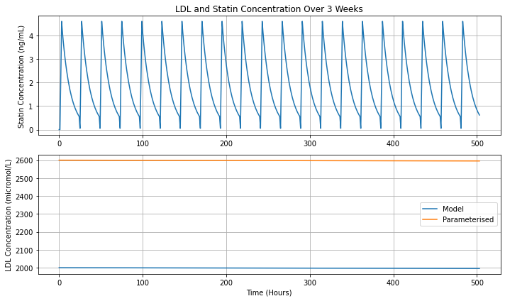
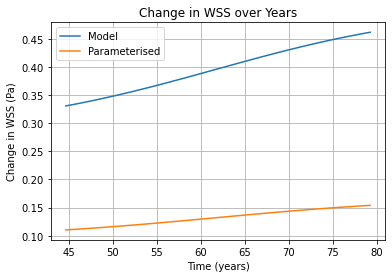
 

Figure 9: Stain and LDL vs Time (All Models)

Figure 8: WSS vs Time (All Models)

Realistically these are only initial values and there is no telling how the person will react or their lifestyle habits apart from the data that is given, which is very limited. The LDL and change in artery have not changed due to only being given initial values, so their model will follow the same model that was presented in the Model Implementation Section. Although in Figure 9 it appears that the LDL is changing, it is and the dynamics can be seen in Appendix 6.

## Parameter Sensitivity

A sensitivity analysis was performed on the statin concentration parameters to assess the change in model fit (correlation coefficient). These were evaluated for ±1%, ±5% and ±20%, for k1,k2 and k3 of Equation 9 presented in ‘Model Formulations’. Figures 10, 11 and 12 show the results of this sensitivity analysis. Appendix 12 is the Python script for these plots. The sensitivity analysis can be seen below:

A graph of data and data

Description automatically generated with medium confidenceA graph of data and numbers

Description automatically generated with medium confidenceA graph of data and data

Description automatically generated with medium confidence

Figure 10: Before Peak Parameter Sensitivity Analysis

Figure 11: After Peak 1 Parameter Sensitivity Analysis

Figure 12: After Peak 2 Parameter Sensitivity Analysis

Table 2, shows the results of performing this analysis on parameters and includes the correlation coefficient values of the sensitivity analysis, with the original correlation coefficient of the current model being 0.97:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Before Parameter 1** | **After Parameter 1** | **After Parameter 2** |
| **20%** | **+** | 0.95 | 0.21 | 0.77 |
| **-** | 0.87 | 0.59 | 0.04 |
| **5%** | **+** | 0.97 | 0.91 | 0.96 |
| **-** | 0.96 | 0.92 | 0.89 |
| **1%** | **+** | 0.98 | 0.97 | 0.97 |
| **-** | 0.95 | 0.96 | 0.96 |

Table 2: Correlation Coefficient Values

By analysing the sensitivity analysis based on the model and the provided simvastatin data, it becomes evident that each parameter has a significant influence on different aspects of the model. There are some notable parameters (After Peak Parameter 1 +20%, After Peak Parameter 2 -20%) which, when the sensitivity analysis was applied looked inappropriate, the limitation of this model is that only a certain range of values of which this equation will work. To address this limitation, the development of more comprehensive statin models with a broader range of parameters is recommended. For instance, when adjusting before peak parameter +1% it increases the model fit of the data with a correlation coefficient of 0.98.

## Accuracy, Complexity, Future

The balance volume diagram, describing the species and kinetics application to the problem, has undergone significant simplification. Studies such as (Wenrui Hao, 2014) exhibit an already more complex model than the one that was presented.

One of the central mechanisms proposed, involving the impact of viscosity on WSS is evident in Figures 5 and 8. Recognising this mechanism’s influence is vital, as it underpins the patient’s overall physiological health. As WSS is imperative to understand whether a patient artery is at risk of thrombosis and rupture.

Concerning parameterisation, this model did fit significantly well as the correlation coefficient was 0.97 to the statin data provided. Sensitivity analysis suggests this can be improved, which can be explored later. With fair and increasing accuracy with the parameter before peaking, this model also had some limitations with some values becoming nonsensical when the sensitivity analysis was applied, implying a limited range that this parameter would suit. A more robust or complex model would suit this better In fitting the patient’s data into the model several assumptions needed to be made, due to the unavailability of key data points, such as LDL concentration.

The statin data used for the sensitivity analysis was based on an average age 24-year-old male with average weight. These assumptions may not fully reflect the circumstances of the patients with different age groups and body compositions. Realistically statin concentration in blood would still be affected by other factors such as eating and fluctuations due to diet, exercise and the age of the patient as well as these are known to affect the statin concentration in blood (Corn G, 2023) (S de Andrés, 2004). implementing these factors into the sensitivity analysis would benefit this study Further refinement Is possible to achieve a more precise representation of statin absorption into the body. This entails accounting for the conversion of statin into simvastatin acid, which is the active form responsible for lowering LDL, which is a model that can be presented in (Janthima Methaneethorn, 2014). Additionally, integrating data on LDL’s entry into the tunica intima to form foam cells, a process that is not included in our model but is available in the literature, such as in (Weirui Lei, 2023) and should be considered as further adds complexity and accuracy to the model.

In conclusion, while this model provides a foundational understanding of statins’ impact on LDL, there is room for increased complexity and precision. Further exploration and refinement of our model are recommended to enhance its accuracy and practicality.

# Bibliography

Alaa Eddin Nwilati, D. A. (2019). FJVIS 28. True Anterior Tibial Artery Aneurysm With Rupture and Compartment Syndrome. *Journal of Vascular Surgery*, 198-199.

British Heart Foundation. (n.d.). *Statins: Common questions answered*. Retrieved from British Heart Foundation: https://www.bhf.org.uk/informationsupport/heart-matters-magazine/medical/drug-cabinet/statins#:~:text=Statins%20are%20prescribed%20to%20people,are%20worried%20about%20side%20effects.

Centre for Industrial Rheology. (n.d.). *Viscosity testing for process design*. Retrieved from Centre for Industrial Rheology: https://www.rheologylab.com/articles/viscosity-testing-for-process-design/

Cleveland Clinic. (n.d.). *Coronary Artery Calcification*. Retrieved from Cleveland Clinic: https://my.clevelandclinic.org/health/diseases/22953-coronary-artery-calcification

Corn G, M. M. (2023). Association Between Age and LDL-C Response to Statins. *American College of Cardiology*.

CSIRO. (n.d.). *Cholesterol Facts*. Retrieved from CSIRO: https://www.csiro.au/en/research/health-medical/nutrition/cholesterol-facts

Davis, A. (2014). *A Medicine's Life Inside the Body*. Retrieved from National Institute of General Medical Sciences: https://www.nigms.nih.gov/education/Inside-Life-Science/Pages/A-Medicines-Life-Inside-the-Body.aspx#:~:text=Most%20often%2C%20the%20bloodstream%20is,could%20be%20a%20side%20effect.

Dorota Formanowicz, J. B. (2019). A Control-Theoretic Model of Atherosclerosis. *International Journal of Molecular Sciences*, 785.

Dorota Formanowicz, J. B. (2019). A Control-Theoretic Model of Atherosclerosis. *International Journal of Molecular Sciences*, 785.

Janthima Methaneethorn, P. K. (2014). Pharmacokinetic modeling of simvastatin, nelfinavir and their interaction in humans. *Annu Int Conf IEEE Eng Med Biol Soc.*, 15-18.

Jeffrey D. Crawford MD, N. G. (2015). Characterization of tibial velocities by duplex ultrasound in severe peripheral arterial disease and controls. *Journal of Vascular Surgery*, 17-29.

Jimyon Kim, B.-J. A.-S.-S. (2011). A Population Pharmacokinetic–Pharmacodynamic Model for Simvastatin that Predicts Low-Density Lipoprotein-Cholesterol Reduction in Patients with Primary Hyperlipidaemia. *Basic & Clinical Pharmacology & Toxicology*, 156-163. Retrieved from Basic & Clinical Pharmacology & Toxicology.

John Hopkins Medicine. (n.d.). *Atherosclerosis*. Retrieved from Health: https://www.hopkinsmedicine.org/health/conditions-and-diseases/atherosclerosis

Katharina Urschel, M. T. (2021). Investigation of Wall Shear Stress in Cardiovascular Research and in Clinical Practice—From Bench to Bedside. *International Journal of Molecular Sciences*.

Marianne K. DeGorter, R. G.-H.-Q. (2013). Clinical and Pharmacogenetic Predictors of Circulating Atorvastatin and Rosuvastatin Concentration in Routine Clinical Care. *Circulation: Genomic and Precision Medicine*, 400-408.

Marilisa Bove, F. F. (2017). Pharmacokinetic drug evaluation of ezetimibe + simvastatin for the treatment of hypercholesterolemia. *Expert Opinion on Drug Metabolism and Toxicology*, 1099-1104.

Mayo Clinic. (2023). *Arteriosclerosis / atherosclerosis*. Retrieved from Mayo Clinic: https://www.mayoclinic.org/diseases-conditions/arteriosclerosis-atherosclerosis/diagnosis-treatment/drc-20350575#:~:text=Statins%20and%20other%20cholesterol%20drugs.&text=Statins%20are%20commonly%20used%20to,fibrates%20and%20bile%20acid%20sequestrants.

Mayo Clinic. (n.d.). *Arteriosclerosis/Atherosclerosis*. Retrieved from Mayo Clinic: https://www.mayoclinic.org/diseases-conditions/arteriosclerosis-atherosclerosis/symptoms-causes/syc-20350569

Michael B. Fessler, K. R. (2013). Relationship between serum cholesterol and indices of erythrocytes and platelets in the US population. *Journal of Lipid Research*, 3177-3188.

Mikhail A. Panteleev, N. K. (2021). Wall shear rates in human and mouse arteries: Standardization of hemodynamics for in vitro blood flow assays: Communication from the ISTH SSC subcommittee on biorheology. *Journal of Thrombosis and Haemostasis*.

Mohammed Ameenuddin, M. A. (2019). Effects of shear-dependent viscosity and hematocrit on blood flow. *Applied Mathematics and Computation*, 299-311.

Myriam Cilla, E. P. (2014). Mathematical modelling of atheroma plaque formation and development in coronary arteries. *The Royal Society Publishing*, 90.

News Medical and Life Sciences. (n.d.). *Surgeries and Procedures for Atherosclerosis*. Retrieved from News Medical and Life Sciences: https://www.news-medical.net/health/Surgeries-and-Procedures-for-Atherosclerosis.aspx#:~:text=Surgeries%20and%20procedures%20for%20atherosclerosis%20include%20Angioplasty%2C%20Stent%20placement%2C%20Coronary,cardiac%20angioplasty%20or%20atherectomy%20devi

NHCS. (n.d.). *Atherosclerosis (Plaque Buildup): How It Can Lead to a Heart Attack*. Retrieved from HealthXchange.org: https://www.healthxchange.sg/heart-lungs/heart-attack/atherosclerosis-plaque-heart-attack

NHS. (n.d.). *Statins*. Retrieved from NHS Infrom: https://www.nhsinform.scot/tests-and-treatments/medicines-and-medical-aids/types-of-medicine/statins/#:~:text=Statins%20come%20as%20tablets%20that,level%20within%20a%20few%20weeks.

Pahwa, R., & Jialal, I. (2023). Atherosclerosis. *StatPearls*.

Peramaiyan Rajendran, 1. T. (2019). The Vascular Endothelium and Human Diseases. *International Journal of Biological Sciences*, 1057-1069.

Reeskamp, N. S. (2021). From evidence to practice: development of web-based. *Neth Heart J\*, 441-450.

S de Andrés, A. L. (2004). Interactions between foodstuffs and statins. *Nutr Hosp.*, 195-201.

Schachter, M. (2005). Chemical, pharmacokinetic and pharmacodynamic properties of statins: an update. *Fundamental Clinical Pharmacology*, 117-25.

Study of the Effectiveness of Additional Reductions in Cholesterol and Homocysteine (SEARCH) Collaborative Group. (2011). Intensive lowering of LDL cholesterol with 80 mg versus 20 mg simvastatin daily in 12 064 survivors of myocardial infarction: a double-blind randomised trial. *Lancet*, 126.

Talreja, O., Kerndt, C. C., & Cassagnol., M. (2023). Simvastatin. *StatPearls*. Retrieved from StatPearls.

Victor Chang Cardiac Research Institute. (n.d.). *Atherosclerosis*. Retrieved from Victor Change Cardiac Research Institute: https://www.victorchang.edu.au/heart-disease/atherosclerosis#:~:text=Atherosclerosis%20is%20the%20hardening%20and,affect%20people%20in%20different%20ways.

Weirui Lei, J. H. (2023). Mathematical Modelling of the Effects of Statins pn the Growth of Necrotic Core in Atherosclerotic Plaque. *Mathematical Modelling of Natural Phenomena*.

Wenrui Hao, A. F. (2014). The LDL-HDL Profile Determines the Risk of Atherosclerosis: A Mathematical Model. *PLOSSONE*.

# Appendix

Appendix 1: Km/Kmax values

## Appendix 2: Patient Data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Patient 1 | Patient 2 | Patient 3 | Patient 4 | Patient 5 |
| Pressure | Systolic | 126.82 | 75.05 | 134.64 | 118.21 | 74.65 |
| Diastolic | 68.40 | 35.82 | 77.38 | 42.57 | 35.59 |
| Area | Systolic | 6.63 | 6.31 | 4.54 | 3.59 | 3.50 |
| Diastolic | 6.27 | 6.07 | 4.33 | 3.49 | 3.39 |
| Flowrate | Systolic | 2.77 | 1.81 | 2.40 | 1.90 | 1.26 |
| Diastolic | 0.08 | -0.13 | 0.34 | 0.29 | 0.17 |
| R-R Interval |  | 0.83 | 1.124 | 0.83 | 1.124 | 1.124 |

Area to radius for patient 3

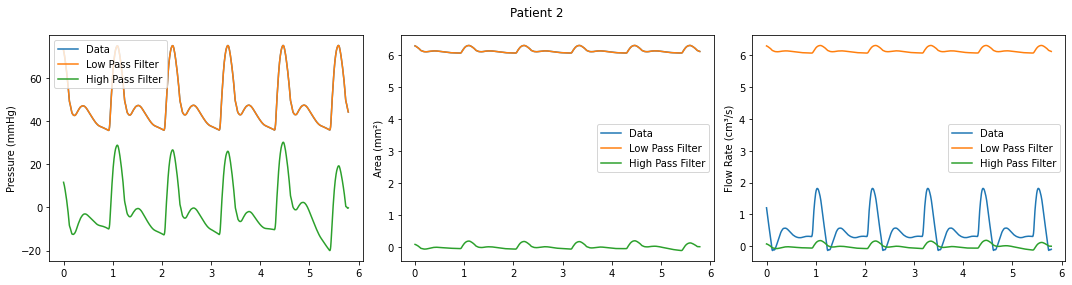
Average Area= pi\*r2

4.4.35=pi \*r2 -> r=1.19mm

## Appendix 3: Healthy Data

|  |  |  |  |
| --- | --- | --- | --- |
| Age | Parameter | Systolic/Diastolic | Value |
| 20s | Area (mm2) | Systolic | 6.79 |
| Diastolic | 6.11 |
| Pressure (mmHg) | Systolic | 118.14 |
| Diastolic | 65.73 |
| Flowrate (cm3/s) | Systolic | 4.67 |
| Diastolic | 0.53 |
| 30s | Area (mm2) | Systolic | 6.75 |
| Diastolic | 6.12 |
| Pressure (mmHg) | Systolic | 122.48 |
| Diastolic | 67.58 |
| Flowrate (cm3/s) | Systolic | 4.64 |
| Diastolic | 0.60 |
| 40s | Area (mm2) | Systolic | 6.72 |
| Diastolic | 6.12 |
| Pressure (mmHg) | Systolic | 128.78 |
| Diastolic | 67.96 |
| Flowrate (cm3/s) | Systolic | 4.59 |
| Diastolic | 0.64 |
| 50s | Area (mm2) | Systolic | 6.67 |
| Diastolic | 6.12 |
| Pressure (mmHg) | Systolic | 132.09 |
| Diastolic | 65.88 |
| Flowrate (cm3/s) | Systolic | 4.45 |
| Diastolic | 0.66 |
| 60s | Area (mm2) | Systolic | 6.63 |
| Diastolic | 6.12 |
| Pressure (mmHg) | Systolic | 134.25 |
| Diastolic | 63.2 |
| Flow rate (cm3/s) | Systolic | 4.25 |
| Diastolic | 0.66 |

## Appendix 4: Patients Filtered Graphs



A graph with green and orange lines

Description automatically generated

A graph with green and orange lines

Description automatically generated

A graph with green and orange lines

Description automatically generated

## Appendix 5: Averages of Patients Graphs

A graph of a graph

Description automatically generated with medium confidence

A graph of blue lines

Description automatically generated

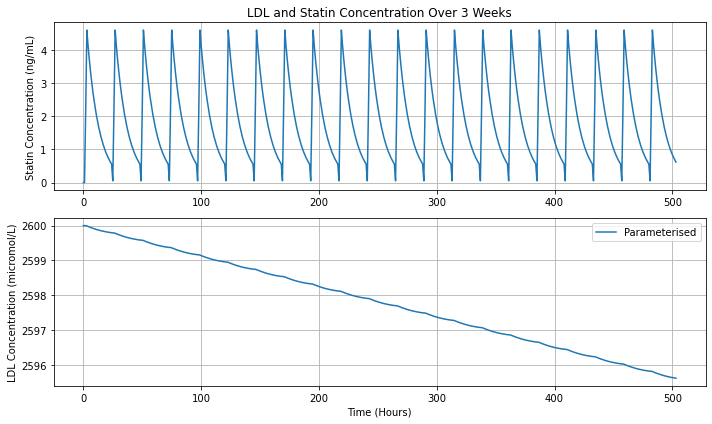
A graph of a graph

Description automatically generated with medium confidence

A graph of a graph

Description automatically generated with medium confidence

## Appendix 6: LDL vs Statin Graph Parameterised



## Appendix 7: Model Formulations

"""

BIOE3001: Quantitative Methods in Biomedical Engineering

Model Implementation

Author: Diane Young

"""

#Importing necessary libraries

import numpy as np

import matplotlib.pyplot as plt

#FUNCTIONS

#Caclulating change in ldl based on statin dosage

def calculate\_dldl(statin\_dosage):

#All possible dosages of statins

dosage\_values = [0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80]

#Empty array to store dldl values at each point

dldl\_values = []

#For each time frame dosage values are to be evaluated, it changes ldl based on the Michaelis-Menten Equation

for dosage in dosage\_values:

dldl = (K\_max \* dosage/10) / (K\_m + dosage/10)

dldl\_values.append(dldl)

return dldl

#Finding change in radius over change in time using differential from literature

def drdt(x, c):

#This equation was found through literature

return 0.06 \* x \* (1 - x / c)

#Function to determine WSS

def WSS(cldl,R):

#Initialising needed arrays

hem=np.zeros(80)

inf=np.zeros(80)

naught=np.zeros(80)

sr=np.zeros(80)

vis=np.zeros(80)

dUdr=np.zeros(80)

WSS=np.zeros(80)

for t in range(0,79):

#Determining viscosity based on ldl concentration

#Hematocrit based on ldl

hem[t]=0.08\*cldl[t]-0.0736

#Other parameters based on hematocrit

inf[t]= 0.0045+0.02\*hem[t]

naught[t]=-0.13+1.12\*hem[t]

#Given value

thin=0.06

#Velocity

u = 0.42 #m/s

#Average Radius

r= 0.003 #m

#Shear rate

sr[t]= 15.5\*(u/R[t])

#Finding viscosity

vis[t]=inf[t]+naught[t]\*(1+np.log(1+thin\*sr[t])/(1+thin\*sr[t]))

#dU/dR Equation

dUdr[t]=-2\*u\*((2\*r)/R[t])

#Final WSS Equation

WSS[t]=vis[t]\*dUdr[t]

return WSS

#TIME COURSES

#Time across 3 weeks

hours = np.linspace(0, 504, 504)

#This will be modelled from 18 years to 80 years (Assuming statin dosage starts at 18)

age = np.linspace(18, 80, 80)

#INITALISING ARRAYS

#Concentration of Statin

C\_statin = np.zeros(504)

#Concentration of LDL

C\_ldl = np.zeros(504)

#The effect statin has on LDL

statin\_effect = np.zeros(504)

ldl\_treated = np.zeros(80)

#Change in statin concentation

decay\_term=np.empty(504)

#OR= Original Radius of the person

OR = [3] \* 80

#PARAMETERS

#For weeks dosage the statin dosage is specified

dosage\_statin = 30 # mg

#Parameters for statin (Michalis Menten Reaction)

K\_max = 0.02 #mmol/L.s

K\_m = 2 #mmol/L.s

#Values for radius calculations

#Healthy Case

#Rate of growth

c\_h = 1 #mm/year

#Initial Thickness

init\_h = 0.04 #mm

#Unhealthy Case

#Rate of growth

c\_u = 1.8 #mm/year

#Initial thickness

init\_u = 0.05 #mm

# Half-life and decay rate of simvastatin

half\_life = 5 #hour

decay\_rate = 7.5 #mg/hr

#Equations for different ldl patients (mmol/L)

#Healthy Values

k1\_hea= 0.0002

k2\_hea= 0.0425

k3\_hea= 0.6731

#Unhealthy Values

k1\_unhea= 0.0005

k2\_unhea= 0.0715

k3\_unhea= 2.6762

#Simvastatin values

half\_life=5 #hours

rate\_decay=0.5 #1/h

statin\_effectiveness=30 #%

statin\_concentration = 0 #mg

#30% of ldl is elimated during treated which makes the rate 30% less

treated\_affect= c\_u\*0.3 #mm/year

statin\_dosage=10

#EQUATIONS

# Healthy Patient Equation (Years)

ldl\_healthy = -k1\_hea \* age\*\*2 + k2\_hea \* age + k3\_hea

# Unhealthy Patient Equation (Years)

ldl\_unhealthy = -k1\_unhea \* age\*\*2 + k2\_unhea \* age + k3\_unhea

diffuse\_rate\_before\_peak=2.43 #mg/h

#INITIAL CONDITIONS

#Setting inital values

C\_ldl=np.ones(504)\*2000 #micrommol/L

C\_statin[0] = 0 #mg

ldl\_treated[0] = 3.8 #mmol/L (Same as the unhealthy case)

#These are lists because of the way that the statins dosage is max at 80

healthy = [init\_h]

unhealthy = [init\_u]

treated = [init\_u]

#Calculating statin and ldl over 3 week treatment period

for t in range(0, (504 - 1)):

# Every 24 hours statin medication is taken

time\_within\_cycle = t % 24

# Michaelis-Menten Equation for Describing Statins Effect

statin\_effect[t] = (K\_max \* C\_statin[t]) / (K\_m + C\_statin[t])

#Decay term for statin

decay\_term[t] = C\_statin[t] \* np.exp(-rate\_decay \* (1 / half\_life))

diffuse\_rate\_after\_peak= 4.985 \* np.exp(-0.13\*time\_within\_cycle)

#Up until statin concentration peaks

if time\_within\_cycle < 3:

# Time before statin peaks

statin\_concentration = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

else:

#Time after peak

statin\_concentration = diffuse\_rate\_after\_peak - decay\_term[t]

# Increase the statin concentration by the calculated amount

C\_statin[t + 1] = max(0, C\_statin[t] + statin\_concentration)

# Change in LDL is only affected by how much statin affects it

dC\_ldl = statin\_effect[t]

# Calculate the new concentration

C\_ldl[t + 1] = max(0, C\_ldl[t] - dC\_ldl)

#Radius over time in years

for t in range(0, 79):

#Next healthy term is the previous one + change based on differential

next\_h = healthy[-1] + drdt(healthy[-1], c\_h)

#Append it to the healthy radius change list

healthy.append(next\_h)

#Next unhealthy term is the previous one + change based on differential

next\_u = unhealthy[-1] + drdt(unhealthy[-1], c\_u)

#Append it to the unhealthy radius change list

unhealthy.append(next\_u)

#Next treated term is the previous one + change based on differental (rate is affected by treated affect)

next\_t = treated[-1] + drdt(treated[-1], c\_u-treated\_affect)

#Append it to the treated radius change list

treated.append(next\_t)

#Calculating change in ldl concentration by statin dosage

for t in range(0, 79):

# Calculate dldl based on current statin dosage

dldl = calculate\_dldl(statin\_dosage)

# Update ldl\_treated using the equation and dldl

ldl\_treated[t + 1] = -k1\_unhea \* t\*\*2 + k2\_unhea \* t + k3\_unhea - dldl \* statin\_effectiveness

# Check if cholesterol level exceeds 2 and increase statin dosage by 5 (evaluated every year)

if ldl\_treated[t + 1] > 2:

statin\_dosage += 5

#To be able to find the difference in

differences\_healthy = [a - b for a, b in zip(OR, healthy)]

differences\_unhealthy = [a - b for a, b in zip(OR, unhealthy)]

differences\_treated = [a - b for a, b in zip(OR, treated)]

# Calculate WSS for each scenario using the respective LDL concentrations

WSS\_healthy = WSS(ldl\_healthy/1000, differences\_healthy)

WSS\_unhealthy = WSS(ldl\_unhealthy/1000, differences\_unhealthy)

WSS\_treated = WSS(ldl\_treated/1000, differences\_treated)

#PLOTTING

#Age vs Thickness of Artery Graph

plt.figure()

plt.plot(age, healthy, label='Healthy Patient')

plt.plot(age, unhealthy, label='Unhealthy Patient')

plt.plot(age, treated, label='Treated Patient')

plt.xlabel('Age (years)')

plt.ylabel('Change in Anterior Tibular Artery Radius (mm)')

plt.title('Change in Radius of Artery Over Years')

plt.legend()

plt.grid(True)

#Hours vs LDL/Statin Concentration Graph (Weeks)

plt.figure(figsize=(10, 6))

plt.subplot(211)

plt.title('LDL and Statin Concentration Over 3 Weeks')

plt.plot(range(0, 504), C\_statin)

plt.ylabel('Statin Concentration (ng/mL)')

plt.grid(True)

plt.subplot(212)

plt.plot(range(0, 504), C\_ldl)

plt.ylabel('LDL Concentration (micromol/L)')

plt.xlabel('Time (Hours)')

plt.grid(True)

plt.tight\_layout()

#Age vs. WSS Graph (Years)

plt.figure()

plt.plot(age[:79], WSS\_treated[:79]\*1000, label='Treated Patient')

plt.plot(age[:79], WSS\_healthy[:79]\*1000, label='Healthy Patient')

plt.plot(age[:79], WSS\_unhealthy[:79]\*1000, label='Unhealthy Patient')

plt.legend()

plt.grid()

plt.xlabel('Time (years)')

plt.ylabel('Change in WSS (Pa)')

plt.title('Change in WSS over Years')

#Age vs. LDL Graph (Years)

plt.figure()

plt.plot(age[:79], ldl\_treated[:79]\*1000, label='Treated Patient')

plt.plot(age[:79], ldl\_healthy[:79]\*1000, label='Healthy Patient')

plt.plot(age[:79], ldl\_unhealthy[:79]\*1000, label='Unhealthy Patient')

plt.legend()

plt.grid()

plt.xlabel('Age (years)')

plt.ylabel('LDL (micromol/L)')

plt.title('Change in LDL over Years')

## Appendix 8: Patient Data Python (Filtered)

"""

BIOE3001

Filtering and R-R Intervals of Patients

Author: Diane Young

"""

import numpy as np

import matplotlib.pyplot as plt

import scipy.signal as sig\_sci

#Functions are from BIOE3001: Lecture 9

#Low-Pass Filter

def butter\_lowpass\_filter(data, cutoff, fs):

order=4

nyq\_f=fs/2

normal\_cutoff = cutoff / nyq\_f

b, a = sig\_sci.butter(order, normal\_cutoff, btype='low', analog=False) # Get the filter coefficients

y = sig\_sci.filtfilt(b, a, data) # Apply a linear phase filter with the specified co-efficients

return y

#High Pass Filter

def butter\_highpass\_filter(data, cutoff, fs):

order=4

nyq\_f=fs/2

normal\_cutoff = cutoff / nyq\_f

b, a = sig\_sci.butter(order, normal\_cutoff, btype='high', analog=False) # Get the filter coefficients

y = sig\_sci.filtfilt(b, a, data) # Apply a linear phase filter with the specified co-efficients

return y

# Paths for each patient

data\_paths = [

r'C:\Users\UQ\Desktop\BIOE3001\Project\Part C\Wilemet\_YourPatientsToAnalyse\_MultiPulseData\25yo\_pt745\_anttibial.csv',

r'C:\Users\UQ\Desktop\BIOE3001\Project\Part C\Wilemet\_YourPatientsToAnalyse\_MultiPulseData\35yo\_pt345\_anttibial.csv',

r'C:\Users\UQ\Desktop\BIOE3001\Project\Part C\Wilemet\_YourPatientsToAnalyse\_MultiPulseData\45yo\_pt423\_anttibial.csv',

r'C:\Users\UQ\Desktop\BIOE3001\Project\Part C\Wilemet\_YourPatientsToAnalyse\_MultiPulseData\55yo\_pt126\_anttibial.csv',

r'C:\Users\UQ\Desktop\BIOE3001\Project\Part C\Wilemet\_YourPatientsToAnalyse\_MultiPulseData\65yo\_pt445\_anttibial.csv'

]

#Through trial and error fs was found

fs=400

nyq\_f=fs/2

# Create subplots for each patient

for patient, data\_path in enumerate(data\_paths):

data = np.genfromtxt(data\_path, delimiter=',')

#Time is the first column

time = data[0, :]

#Figure

plt.figure(figsize=(15, 4))

# Pressure (mmHg)

plt.subplot(1, 3, 1)

#Vectors for BP, Area and Flow

BP\_Vect= data[1, :] / 133.322

Area\_Vect= data[3, :]\*10\*\*6

Flow\_Vect= data[2, :] \* 1e6

#Applying filters

filtered\_pressure2= butter\_highpass\_filter(BP\_Vect,0.1,nyq\_f)

filtered\_pressure1 = butter\_lowpass\_filter(BP\_Vect,80, nyq\_f)

filtered\_area1= butter\_lowpass\_filter(Area\_Vect,80, nyq\_f)

filtered\_area2= butter\_highpass\_filter(Area\_Vect,0.1,nyq\_f)

filtered\_flow1= butter\_lowpass\_filter(Flow\_Vect,80, nyq\_f)

filtered\_flow2= butter\_highpass\_filter(Flow\_Vect,0.1,nyq\_f)

#Plotting Pressure(mmHg)

peaks,\_=sig\_sci.find\_peaks(BP\_Vect, prominence=(30,None))

plt.plot(time, BP\_Vect, label='Data')

plt.plot(time,filtered\_pressure1, label='Low Pass Filter')

plt.plot(time,filtered\_pressure2, label='High Pass Filter')

plt.ylabel('Pressure (mmHg)')

plt.suptitle(f'Patient {patient + 1}')

plt.legend()

# Plotting Area (mm2)

plt.subplot(1, 3, 2)

plt.plot(time, data[3, :]\*10\*\*6, label='Data')

plt.plot(time,filtered\_area1, label='Low Pass Filter')

plt.plot(time,filtered\_area2, label='High Pass Filter')

plt.legend()

plt.ylabel('Area (mm²)')

# Plotting Flow Rate (mL/s)

plt.subplot(1, 3, 3)

plt.plot(time, data[2, :] \* 1e6, label= 'Data') # Convert m³/s to cm³/s

plt.plot(time,filtered\_area1, label='Low Pass Filter')

plt.plot(time,filtered\_area2, label='High Pass Filter')

plt.ylabel('Flow Rate (cm³/s)')

plt.legend()

plt.tight\_layout()

plt.show()

#Finding R-R intervals manually

positive\_peak\_times = time[peaks]

print(f"Positive Peak Times for Patient {patient + 1}:")

print(positive\_peak\_times)

## Appendix 9: Patient Data Python Averages

"""

BIOE3001: Averages

Author: Diane Young

"""

import numpy as np

import matplotlib.pyplot as plt

import scipy.signal as sig\_sci

import statistics as stats

#Data Paths

data\_paths = [

r'C:\Users\UQ\Desktop\BIOE3001\Project\Part C\Wilemet\_YourPatientsToAnalyse\_MultiPulseData\25yo\_pt745\_anttibial.csv',

r'C:\Users\UQ\Desktop\BIOE3001\Project\Part C\Wilemet\_YourPatientsToAnalyse\_MultiPulseData\35yo\_pt345\_anttibial.csv',

r'C:\Users\UQ\Desktop\BIOE3001\Project\Part C\Wilemet\_YourPatientsToAnalyse\_MultiPulseData\45yo\_pt423\_anttibial.csv',

r'C:\Users\UQ\Desktop\BIOE3001\Project\Part C\Wilemet\_YourPatientsToAnalyse\_MultiPulseData\55yo\_pt126\_anttibial.csv',

r'C:\Users\UQ\Desktop\BIOE3001\Project\Part C\Wilemet\_YourPatientsToAnalyse\_MultiPulseData\65yo\_pt445\_anttibial.csv'

]

#Most of this code was taken off BIOE:3001 Lecture 9

for patient, data\_path in enumerate(data\_paths):

#Bringing in the data

data = np.genfromtxt(data\_path, delimiter=',')

time = data[0, :]

plt.figure(figsize=(15, 4))

#Pressure (mmHg)

BP\_Vect = data[1, :] / 133.322

peaks, \_ = sig\_sci.find\_peaks(BP\_Vect, prominence=(30, None))

neg\_peaks, \_ = sig\_sci.find\_peaks(-BP\_Vect, prominence=(30, None))

plt.subplot(1, 3, 1)

plt.plot(time, BP\_Vect)

plt.plot(time[peaks], BP\_Vect[peaks], "x")

plt.plot(time[neg\_peaks], BP\_Vect[neg\_peaks], "x")

plt.ylabel('Pressure (mmHg)')

plt.suptitle(f'Patient {patient + 1}')

#Area (mm2)

A\_Vect = data[3, :] \* 10\*\*6

plt.subplot(1, 3, 2)

plt.plot(time, A\_Vect)

plt.ylabel('Area (mm²')

A\_peaks, \_ = sig\_sci.find\_peaks(A\_Vect, prominence=(0.1, None))

A\_neg\_peaks, \_ = sig\_sci.find\_peaks(-A\_Vect, prominence=(0.1, None))

plt.plot(time[A\_peaks], A\_Vect[A\_peaks], "x")

plt.plot(time[A\_neg\_peaks], A\_Vect[A\_neg\_peaks], "x")

#Flow (cm3/s)

FL\_Vect = data[2, :] \* 1e6 #Ensuring the right units

plt.subplot(1, 3, 3)

plt.plot(time, FL\_Vect)

plt.ylabel('Flow Rate (cm³/s)')

F\_peaks, \_ = sig\_sci.find\_peaks(FL\_Vect, prominence=(0.5, None))

F\_neg\_peaks, \_ = sig\_sci.find\_peaks(-FL\_Vect, prominence=(0.5, None))

plt.plot(time[F\_peaks], FL\_Vect[F\_peaks], "x")

plt.plot(time[F\_neg\_peaks], FL\_Vect[F\_neg\_peaks], "x")

#Show Plot

plt.tight\_layout()

plt.show()

# Calculating the mean systolic and diastolic values for pressure, area, and flow rate

Mean\_Systolic\_Pressure = stats.mean(BP\_Vect[peaks])

Mean\_Diastolic\_Pressure = stats.mean(BP\_Vect[neg\_peaks])

Mean\_Systolic\_Area = stats.mean(A\_Vect[A\_peaks])

Mean\_Diastolic\_Area = stats.mean(A\_Vect[A\_neg\_peaks])

Mean\_Systolic\_Flow = stats.mean(FL\_Vect[F\_peaks])

Mean\_Diastolic\_Flow = stats.mean(FL\_Vect[F\_neg\_peaks])

#Print the average for analysis

print(f"Systolic Pressure for Patient {patient + 1}: {Mean\_Systolic\_Pressure}")

print(f"Diastolic Pressure for Patient {patient + 1}: {Mean\_Diastolic\_Pressure}")

print(f"Systolic Area for Patient {patient + 1}: {Mean\_Systolic\_Area}")

print(f"Diastolic Area for Patient {patient + 1}: {Mean\_Diastolic\_Area}")

print(f"Systolic Flow Rate for Patient {patient + 1}: {Mean\_Systolic\_Flow}")

print(f"Diastolic Flow Rate for Patient {patient + 1}: {Mean\_Diastolic\_Flow}")

## Appendix 10: Healthy Data Python

# -\*- coding: utf-8 -\*-

"""

Created on Wed Oct 18 14:24:38 2023

Healthy Data Python

author: Diane Young

"""

import numpy as np

import matplotlib.pyplot as plt

def find\_highest\_lowest(data, label):

#Finding the systolic and diastolic maximums for each healthy average

highest\_point = np.max(data)

lowest\_point = np.min(data)

#Printing for analysis

print(f"Highest point for {label}: {highest\_point}")

print(f"Lowest point for {label}: {lowest\_point}")

#Copied from Appendix in Data Set

HealthyPWs = np.zeros((19,488,4))

HealthyPWs[:,:,0] = np.genfromtxt(r'C:\Users\UQ\Desktop\BIOE3001\Project\Part C\Charlton\_KnownAgeAverages\_SinglePulseData\PWs\_AntTibial\_A.csv', delimiter=',')

HealthyPWs[:,:,1] = np.genfromtxt(r'C:\Users\UQ\Desktop\BIOE3001\Project\Part C\Charlton\_KnownAgeAverages\_SinglePulseData\PWs\_AntTibial\_P.csv', delimiter=',')

HealthyPWs[:,:,2] = np.genfromtxt(r'C:\Users\UQ\Desktop\BIOE3001\Project\Part C\Charlton\_KnownAgeAverages\_SinglePulseData\PWs\_AntTibial\_PPG.csv', delimiter=',')

HealthyPWs[:,:,3] = np.genfromtxt(r'C:\Users\UQ\Desktop\BIOE3001\Project\Part C\Charlton\_KnownAgeAverages\_SinglePulseData\PWs\_AntTibial\_U.csv', delimiter=',')

Twenties = HealthyPWs[1:4, 1:381, :]

Thirties = HealthyPWs[4:7, 1:381, :]

Fourties = HealthyPWs[7:10, 1:381, :]

Fifties = HealthyPWs[10:13, 1:381, :]

Sixties = HealthyPWs[13:16, 1:381, :]

Seventies = HealthyPWs[16:19, 1:381, :]

#Determining the flow in the correct units (cm3/s)

Twenties\_flow = Twenties[:, :, 0] \* Twenties[:, :, 3] \* 1e6

Twenties\_flow = Twenties[:, :, 0] \* Twenties[:, :, 3] \* 1e6

Thirties\_flow = Thirties[:, :, 0] \* Thirties[:, :, 3] \* 1e6

Fourties\_flow = Fourties[:, :, 0] \* Fourties[:, :, 3] \* 1e6

Fifties\_flow = Fifties[:, :, 0] \* Fifties[:, :, 3] \* 1e6

Sixties\_flow = Sixties[:, :, 0] \* Sixties[:, :, 3] \* 1e6

Seventies\_flow = Seventies[:, :, 0] \* Seventies[:, :, 3] \* 1e6

#20s

plt.figure()

plt.suptitle('20s')

# Artery Area

plt.subplot(2, 3, 1)

mean = np.nanmean(Twenties[:, :, 0], axis=0) \* 10 \*\* 6

lowersd = mean - np.nanstd(Twenties[:, :, 0], axis=0) \* 10 \*\* 6

highersd = mean + np.nanstd(Twenties[:, :, 0], axis=0) \* 10 \*\* 6

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Artery Area')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Artery Area (mm^2)')

# Pressure

plt.subplot(2, 3, 2)

mean = np.nanmean(Twenties[:, :, 1], axis=0)

lowersd = mean - np.nanstd(Twenties[:, :, 1], axis=0)

highersd = mean + np.nanstd(Twenties[:, :, 1], axis=0)

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Pressure')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Pressure (mmHg')

# Flow Rate

plt.subplot(2, 3, 3)

mean = np.nanmean(Twenties\_flow, axis=0)

lowersd = mean - np.nanstd(Twenties\_flow, axis=0)

highersd = mean + np.nanstd(Twenties\_flow, axis=0)

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Flow Rate')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Flow Rate (cm^3/s)')

#30s

plt.figure()

plt.suptitle('30s')

# Artery Area

plt.subplot(2, 3, 1)

mean = np.nanmean(Twenties[:, :, 0], axis=0) \* 10 \*\* 6

lowersd = mean - np.nanstd(Thirties[:, :, 0], axis=0) \* 10 \*\* 6

highersd = mean + np.nanstd(Thirties[:, :, 0], axis=0) \* 10 \*\* 6

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Artery Area')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Artery Area (mm^2)')

# Pressure

plt.subplot(2, 3, 2)

mean = np.nanmean(Twenties[:, :, 1], axis=0)

lowersd = mean - np.nanstd(Thirties[:, :, 1], axis=0)

highersd = mean + np.nanstd(Thirties[:, :, 1], axis=0)

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Pressure')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Pressure (mmHg')

# Flow Rate

plt.subplot(2, 3, 3)

mean = np.nanmean(Twenties\_flow, axis=0)

lowersd = mean - np.nanstd(Thirties\_flow, axis=0)

highersd = mean + np.nanstd(Thirties\_flow, axis=0)

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Flow Rate')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Flow Rate (cm^3/s)')

#40s

plt.figure()

plt.suptitle('40s')

# Artery Area

plt.subplot(2, 3, 1)

mean = np.nanmean(Fourties[:, :, 0], axis=0) \* 10 \*\* 6

lowersd = mean - np.nanstd(Fourties[:, :, 0], axis=0) \* 10 \*\* 6

highersd = mean + np.nanstd(Fourties[:, :, 0], axis=0) \* 10 \*\* 6

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Artery Area')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Artery Area (mm^2)')

# Pressure

plt.subplot(2, 3, 2)

mean = np.nanmean(Twenties[:, :, 1], axis=0)

lowersd = mean - np.nanstd(Fourties[:, :, 1], axis=0)

highersd = mean + np.nanstd(Fourties[:, :, 1], axis=0)

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Pressure')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Pressure (mmHg')

# Flow Rate

plt.subplot(2, 3, 3)

mean = np.nanmean(Fourties\_flow, axis=0)

lowersd = mean - np.nanstd(Fourties\_flow, axis=0)

highersd = mean + np.nanstd(Fourties\_flow, axis=0)

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Flow Rate')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Flow Rate (cm^3/s)')

#50s

plt.figure()

plt.suptitle('50s')

# Artery Area

plt.subplot(2, 3, 1)

mean = np.nanmean(Fifties[:, :, 0], axis=0) \* 10 \*\* 6

lowersd = mean - np.nanstd(Fifties[:, :, 0], axis=0) \* 10 \*\* 6

highersd = mean + np.nanstd(Fifties[:, :, 0], axis=0) \* 10 \*\* 6

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Artery Area')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Artery Area (mm^2)')

# Pressure

plt.subplot(2, 3, 2)

mean = np.nanmean(Twenties[:, :, 1], axis=0)

lowersd = mean - np.nanstd(Fifties[:, :, 1], axis=0)

highersd = mean + np.nanstd(Fifties[:, :, 1], axis=0)

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Pressure')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Pressure (mmHg')

# Flow Rate

plt.subplot(2, 3, 3)

mean = np.nanmean(Fourties\_flow, axis=0)

lowersd = mean - np.nanstd(Fifties\_flow, axis=0)

highersd = mean + np.nanstd(Fifties\_flow, axis=0)

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Flow Rate')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Flow Rate (cm^3/s)')

#60s

plt.figure()

plt.suptitle('60s')

# Artery Area

plt.subplot(2, 3, 1)

mean = np.nanmean(Sixties[:, :, 0], axis=0) \* 10 \*\* 6

lowersd = mean - np.nanstd(Sixties[:, :, 0], axis=0) \* 10 \*\* 6

highersd = mean + np.nanstd(Sixties[:, :, 0], axis=0) \* 10 \*\* 6

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Artery Area')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Artery Area (mm^2)')

# Pressure

plt.subplot(2, 3, 2)

mean = np.nanmean(Sixties[:, :, 1], axis=0)

lowersd = mean - np.nanstd(Sixties[:, :, 1], axis=0)

highersd = mean + np.nanstd(Sixties[:, :, 1], axis=0)

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Pressure')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Pressure (mmHg')

# Flow Rate

plt.subplot(2, 3, 3)

mean = np.nanmean(Fourties\_flow, axis=0)

lowersd = mean - np.nanstd(Sixties\_flow, axis=0)

highersd = mean + np.nanstd(Sixties\_flow, axis=0)

plt.plot(np.linspace(0, 380, 380) / 500, mean, color='b', label='Flow Rate')

plt.fill\_between(np.linspace(0, 380, 380) / 500, lowersd, highersd, color='b', alpha=0.15)

plt.ylabel('Flow Rate (cm^3/s)')

#Finding systolic and diastolic values and printing for analysis

find\_highest\_lowest(Twenties[:, :, 0], "Artery Area 20s")

find\_highest\_lowest(Twenties[:, :, 1], "Pressure 20s")

find\_highest\_lowest(Twenties\_flow, "Flowrate 20s")

find\_highest\_lowest(Thirties[:, :, 0], "Artery Area 30s")

find\_highest\_lowest(Thirties[:, :, 1], "Pressure 30s")

find\_highest\_lowest(Thirties\_flow, "Flowrate 30s")

find\_highest\_lowest(Fourties[:, :, 0], "Artery Area 40s")

find\_highest\_lowest(Fourties[:, :, 1], "Pressure 40s")

find\_highest\_lowest(Fourties\_flow, "Flowrate 40s")

find\_highest\_lowest(Fifties[:, :, 0], "Artery Area 50s")

find\_highest\_lowest(Fifties[:, :, 1], "Pressure 50s")

find\_highest\_lowest(Fifties\_flow, "Flowrate 50s")

find\_highest\_lowest(Sixties[:, :, 0], "Artery Area 60s")

find\_highest\_lowest(Sixties[:, :, 1], "Pressure 60s")

find\_highest\_lowest(Sixties\_flow, "Flowrate 60s")

plt.show()

## Appendix 11: Patient Parameterisation Python

## # -\*- coding: utf-8 -\*-

## """

## Created on Mon Oct 23 09:00:52 2023

## @author: UQ

## """

## """

## BIOE3001: Quantitative Methods in Biomedical Engineering

## Patient Parameterisation

## Author: Diane Young

## """

## #Importing necessary libraries

## import numpy as np

## import matplotlib.pyplot as plt

## #FUNCTIONS

## #Caclulating change in ldl based on statin dosage

## def calculate\_dldl(statin\_dosage):

## #All possible dosages of statins

## dosage\_values = [0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80]

## #Empty array to store dldl values at each point

## dldl\_values = []

## #For each time frame dosage values are to be evaluated

## for dosage in dosage\_values:

## dldl = (K\_max \* dosage/10) / (K\_m + dosage/10)

## dldl\_values.append(dldl)

## 

## return dldl

## #PATIENT INITIAL DATA USED HERE

## def drdt(x, c): #From literature

## return 0.06 \* x \* (1 - x / c)

## def WSS(cldl,R, u, r):

## hem=np.zeros(80)

## inf=np.zeros(80)

## naught=np.zeros(80)

## sr=np.zeros(80)

## vis=np.zeros(80)

## dUdr=np.zeros(80)

## WSS=np.zeros(80)

## 

## for t in range(0,79):

## hem[t]=0.08\*cldl[t]-0.0736

## inf[t]= 0.0045+0.02\*hem[t]

## naught[t]=-0.13+1.12\*hem[t]

## thin=0.06

## #PATIENT VALUE FOR u AND r

## u = 0.31 #m/s

## sr[t]= 15.5\*(u/R[t])

## vis[t]=inf[t]+naught[t]\*(1+np.log(1+thin\*sr[t])/(1+thin\*sr[t]))

## dUdr[t]=-2\*u\*((2\*r)/R[t])

## WSS[t]=vis[t]\*dUdr[t]

## return WSS

## #TIME COURSES

## #Time across 3 weeks

## hours = np.linspace(0, 504, 504)

## #Statin is taken every 24 hours

## Take\_statin = np.arange(0, 504, 24)

## #This will be modelled from 18 years to 80 years (Assuming statin dosage starts at 18)

## age = np.linspace(18, 80, 80)

## C\_statin = np.zeros(504)

## C\_ldl = np.zeros(504)

## statin\_effect = np.zeros(504)

## ldl\_treated = np.ones(80)\*2600

## time\_since\_last\_dosage=np.zeros(504)

## WSS\_healthy = np.empty(80)

## WSS\_unhealthy = np.empty(80)

## WSS\_treated = np.empty(80)

## #OR= Original Radius of the person

## OR = [3] \* 80 #mm

## #PARAMETERS

## #For weeks dosage the statin dosage is specified

## dosage\_statin = 30 # mg

## #Parameters for statin (Michalis Menten Reaction)

## K\_max = 0.02 #mmol/L.s

## K\_m = 2 #mmol/L.s

## #Values for radius calculations

## #Healthy Case

## #Rate of growth

## c\_h = 1 #mm/year

## #Initial Thickness

## init\_h = 0.04 #mm

## #Unhealthy Case

## #Rate of growth

## c\_u = 1.8 #mm/year

## #Initial thickness

## init\_u = 0.05 #mm

## #30% of ldl is elimated during treated which makes the rate 30% less

## treated\_affect= c\_u\*0.3 #mm/year

## #Starting dosage of statin

## statin\_dosage = 10 #mg

## # EQUATIONS

## # Half-life and decay rate of simvastatin

## half\_life = 5 #hour

## decay\_rate = 7.5 #mg/hr

## #Equations for different ldl patients (mmol/L)

## #Healthy Values

## k1\_hea= 0.0002

## k2\_hea= 0.0425

## k3\_hea= 0.6731

## #Unhealthy Values

## k1\_unhea= 0.0005

## k2\_unhea= 0.0715

## k3\_unhea= 2.6762

## #Simvastatin values

## half\_life=5 #hours

## rate\_decay=0.5 #1/h

## statin\_effectiveness=30 #%

## # Healthy Patient Equation

## ldl\_healthy = -k1\_hea \* age\*\*2 + k2\_hea \* age + k3\_hea

## # Unhealthy Patient Equation

## ldl\_unhealthy = -k1\_unhea \* age\*\*2 + k2\_unhea \* age + k3\_unhea

## diffuse\_rate\_before\_peak=2.2

## #Velocity guess and patient data

## u\_guess= 0.42 #m/s

## u\_patient= 0.31 #m/s

## #Radius guess and patient data

## r\_guess=0.003 #m

## r\_patient=0.001 #m

## #INITIAL CONDITIONS

## #Setting inital values

## #Cldl guess and average patient data

## C\_ldl\_guess=np.ones(504)\*2000 #micromol/L

## C\_ldl\_patient=np.ones(504)\*2600 #micrommol/L

## C\_statin[0] = 0 #mg

## #These are lists because of the way that the statins dosage is max at 80

## healthy = [init\_h]

## unhealthy = [init\_u]

## treated = [0.08]

## dC\_statin=np.empty(504)

## statin\_concentration = 0

## decay\_term=np.empty(504)

## for t in range(0, (504 - 1)):

## # Calculate the time within the 24-hour cycle

## time\_within\_cycle = t % 24

## 

## # Michaelis-Menten Equation for Describing Statins Effect

## statin\_effect[t] = (K\_max \* C\_statin[t]) / (K\_m + C\_statin[t])

## 

## decay\_term[t] = C\_statin[t] \* np.exp(-rate\_decay \* (1 / half\_life))

## diffuse\_rate\_after\_peak= 5 \* np.exp(-0.1\*time\_within\_cycle)

## #Up until statin concentration peaks

## if time\_within\_cycle < 3:

## # Increasing portion of equation

## statin\_concentration = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

## else:

## statin\_concentration = diffuse\_rate\_after\_peak - decay\_term[t]

## 

## # Increase the statin concentration by the calculated amount

## C\_statin[t + 1] = max(0, C\_statin[t] + statin\_concentration)

## 

## # Change in LDL is only affected by how much statin affects it

## dC\_ldl = statin\_effect[t]

## # Calculate the new concentration

## C\_ldl\_guess[t + 1] = max(0, C\_ldl\_guess[t] - dC\_ldl)

## C\_ldl\_patient[t + 1] = max(0, C\_ldl\_patient[t] - dC\_ldl)

## #Radius over time in years

## #PATIENT INITIAL DATA USED HERE

## for t in range(0, 79):

## #Next healthy term is the previous one + change based on differential

## next\_h = healthy[-1] + drdt(healthy[-1], c\_h)

## #Append it to the healthy radius change list

## healthy.append(next\_h)

## #Next unhealthy term is the previous one + change based on differential

## next\_u = unhealthy[-1] + drdt(unhealthy[-1], c\_u)

## #Append it to the unhealthy radius change list

## unhealthy.append(next\_u)

## 

## #Next treated term is the previous one + change based on differental (rate is affected by treated affect)

## next\_t = treated[-1] + drdt(treated[-1], c\_u-treated\_affect)

## #Append it to the treated radius change list

## treated.append(next\_t)

## 

## #Calculating change in ldl concentration by statin dosage

## 

## for t in range(0, 79):

## # Calculate dldl based on current statin dosage

## dldl = calculate\_dldl(statin\_dosage)

## 

## # Update ldl\_treated using the equation and dldl

## ldl\_treated[t + 1] = -k1\_unhea \* t\*\*2 + k2\_unhea \* t + k3\_unhea - dldl \* statin\_effectiveness

## # Check if cholesterol level exceeds 2 and increase statin dosage by 5

## if ldl\_treated[t + 1] > 2:

## statin\_dosage += 5

## # Calculate the differences between OR and healthy using list comprehension

## differences\_healthy = [a - b for a, b in zip(OR, healthy)]

## differences\_unhealthy = [a - b for a, b in zip(OR, unhealthy)]

## differences\_treated = [a - b for a, b in zip(OR, treated)]

## # Calculate WSS for each scenario using the respective LDL concentrations

## WSS\_treated\_guess = WSS(ldl\_treated/1000, differences\_treated, u\_guess, r\_guess)

## WSS\_treated\_patient = WSS(ldl\_treated/1000, differences\_treated, u\_patient, r\_patient)

## #Plotting

## #Age vs Thickness of Artery Graph

## """

## plt.figure()

## plt.plot(age[34:], treated[34:], label='Treated Patient')

## plt.xlabel('Age (years)')

## plt.ylabel('Change in Anterior Tibular Artery Radius (mm)')

## plt.title('Change in Radius of Artery Over Years')

## plt.legend()

## plt.grid(True)

## """

## #Hours vs LDL/Statin Concentration Graph (Weeks)

## plt.figure(figsize=(10, 6))

## plt.subplot(211)

## plt.title('LDL and Statin Concentration Over 3 Weeks')

## plt.plot(range(0, 504), C\_statin)

## plt.ylabel('Statin Concentration (ng/mL)')

## plt.grid(True)

## plt.subplot(212)

## plt.plot(range(0, 504), C\_ldl\_guess,label='Model')

## plt.plot(range(0, 504), C\_ldl\_patient, label='Parameterised')

## plt.legend()

## plt.ylabel('LDL Concentration (micromol/L)')

## plt.xlabel('Time (Hours)')

## plt.grid(True)

## plt.tight\_layout()

## #Age vs. WSS Graph (Years)

## plt.figure()

## plt.plot(age[34:79], WSS\_treated\_guess[34:79]\*1000, label='Model')

## plt.plot(age[34:79], WSS\_treated\_patient[34:79]\*1000, label='Parameterised')

## plt.legend()

## plt.grid()

## plt.xlabel('Time (years)')

## plt.ylabel('Change in WSS (Pa)')

## plt.title('Change in WSS over Years')

## #Age vs. LDL Graph (Years)

## """

## plt.figure()

## plt.plot(age[34:79], ldl\_treated[34:79]\*1000)

## plt.legend()

## plt.grid()

## plt.xlabel('Age (years)')

## plt.ylabel('LDL (micromol/L)')

## plt.title('Change in LDL over Years')

## """Appendix 12: Parameter Sensitivity Python Script

"""

BIOE3001: Quantitative Methods in Biomedical Engineering

Parameter Sensitivity

Author: Diane Young

"""

#Importing necessary libraries

import numpy as np

import matplotlib.pyplot as plt

C\_statin = np.zeros(504)

C\_ldl = np.zeros(504)

statin\_effect = np.zeros(504)

ldl\_treated = np.zeros(80)

C\_statin1 = np.zeros(504)

C\_statin2 = np.zeros(504)

C\_statin3 = np.zeros(504)

C\_statin4 = np.zeros(504)

C\_statin5 = np.zeros(504)

C\_statin6 = np.zeros(504)

C\_statin7 = np.zeros(504)

C\_statin8 = np.zeros(504)

C\_statin9 = np.zeros(504)

C\_statin10 = np.zeros(504)

C\_statin11 = np.zeros(504)

C\_statin12 = np.zeros(504)

C\_statin13 = np.zeros(504)

C\_statin14 = np.zeros(504)

C\_statin15 = np.zeros(504)

C\_statin16 = np.zeros(504)

C\_statin17 = np.zeros(504)

C\_statin18 = np.zeros(504)

C\_statin19 = np.zeros(504)

C\_statin20 = np.zeros(504)

#PARAMETERS

#Parameters for statin (Michalis Menten Reaction)

K\_max = 0.02 #mmol/L.s

K\_m = 2 #mmol/L.s

# EQUATIONS

# Half-life and decay rate of simvastatin

half\_life = 5 #hour

decay\_rate = 7.5 #mg/hr

#Simvastatin values

half\_life=5

rate\_decay=0.1

statin\_effectiveness=30 #%

diffuse\_rate\_before\_peak=2

diffuse\_rate\_before\_peak\_20plus= diffuse\_rate\_before\_peak+ 0.2\*diffuse\_rate\_before\_peak

diffuse\_rate\_before\_peak\_20minus= diffuse\_rate\_before\_peak- 0.2\*diffuse\_rate\_before\_peak

diffuse\_rate\_before\_peak\_5plus= diffuse\_rate\_before\_peak+ 0.05\*diffuse\_rate\_before\_peak

diffuse\_rate\_before\_peak\_5minus= diffuse\_rate\_before\_peak- 0.05\*diffuse\_rate\_before\_peak

diffuse\_rate\_before\_peak\_1plus= diffuse\_rate\_before\_peak+ 0.01\*diffuse\_rate\_before\_peak

diffuse\_rate\_before\_peak\_1minus= diffuse\_rate\_before\_peak- 0.01\*diffuse\_rate\_before\_peak

k\_da1= 4.985

k\_da1\_20plus= k\_da1+0.2\*k\_da1

k\_da1\_20minus=k\_da1-0.2\*k\_da1

k\_da1\_5plus= k\_da1+0.05\*k\_da1

k\_da1\_5minus=k\_da1-0.05\*k\_da1

k\_da1\_1plus= k\_da1+0.01\*k\_da1

k\_da1\_1minus=k\_da1-0.01\*k\_da1

k\_da2=0.13

k\_da2\_20plus= k\_da2+0.2\*k\_da2

k\_da2\_20minus= k\_da2-0.2\*k\_da2

k\_da2\_5plus= k\_da2+0.05\*k\_da2

k\_da2\_5minus= k\_da2-0.05\*k\_da2

k\_da2\_1plus= k\_da2+0.01\*k\_da2

k\_da2\_1minus= k\_da2-0.01\*k\_da2

#INITIAL CONDITIONS

#Setting inital values

C\_ldl=np.ones(504)\*2000 #micrommol/L

C\_statin[0] = 0 #mg

ldl\_treated[0] = 3.8 #mmol/L (Same as the unhealthy case)

#These are lists because of the way that the statins dosage is max at 80

dC\_statin=np.empty(504)

statin\_concentration = 0

decay\_term=np.empty(504)

for t in range(0, (504 - 1)):

# Calculate the time within the 24-hour cycle

time\_within\_cycle = t % 24

# Michaelis-Menten Equation for Describing Statins Effect

statin\_effect[t] = (K\_max \* C\_statin[t]) / (K\_m + C\_statin[t])

#decay rate of statin

decay\_term[t] = C\_statin[t] \* np.exp(-rate\_decay \* (1 / half\_life))

diffuse\_rate\_after\_peak= k\_da1 \* np.exp(-k\_da2\*time\_within\_cycle)

diffuse\_rate\_after\_peak1= k\_da1\_20plus \* np.exp(-k\_da2\*time\_within\_cycle)

diffuse\_rate\_after\_peak2= k\_da1\_20minus \* np.exp(-k\_da2\*time\_within\_cycle)

diffuse\_rate\_after\_peak3= k\_da1 \* np.exp(-k\_da2\_20plus\*time\_within\_cycle)

diffuse\_rate\_after\_peak4= k\_da1 \* np.exp(-k\_da2\_20minus\*time\_within\_cycle)

diffuse\_rate\_after\_peak20= k\_da1 \* np.exp(-k\_da2\*time\_within\_cycle)

diffuse\_rate\_after\_peak21= k\_da1 \* np.exp(-k\_da2\*time\_within\_cycle)

diffuse\_rate\_after\_peak22= k\_da1 \* np.exp(-k\_da2\*time\_within\_cycle)

diffuse\_rate\_after\_peak23= k\_da1 \* np.exp(-k\_da2\*time\_within\_cycle)

diffuse\_rate\_after\_peak31= k\_da1\_5plus \* np.exp(-k\_da2\*time\_within\_cycle)

diffuse\_rate\_after\_peak41= k\_da1\_5minus \* np.exp(-k\_da2\*time\_within\_cycle)

diffuse\_rate\_after\_peak32= k\_da1\_1plus \* np.exp(-k\_da2\*time\_within\_cycle)

diffuse\_rate\_after\_peak42= k\_da1\_1minus \* np.exp(-k\_da2\*time\_within\_cycle)

diffuse\_rate\_after\_peak7= k\_da1 \* np.exp(-k\_da2\_5plus\*time\_within\_cycle)

diffuse\_rate\_after\_peak8= k\_da1 \* np.exp(-k\_da2\_5minus\*time\_within\_cycle)

diffuse\_rate\_after\_peak9= k\_da1 \* np.exp(-k\_da2\_1plus\*time\_within\_cycle)

diffuse\_rate\_after\_peak10= k\_da1 \* np.exp(-k\_da2\_1minus\*time\_within\_cycle)

#Up until statin concentration peaks

if time\_within\_cycle < 3:

statin\_concentration = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

statin\_concentration1 = diffuse\_rate\_before\_peak\_20plus \* time\_within\_cycle - decay\_term[t]

statin\_concentration2 = diffuse\_rate\_before\_peak\_20minus \* time\_within\_cycle - decay\_term[t]

statin\_concentration3 = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

statin\_concentration4 = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

statin\_concentration5 = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

statin\_concentration6 = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

statin\_concentration20 = diffuse\_rate\_before\_peak\_5plus \* time\_within\_cycle - decay\_term[t]

statin\_concentration21 = diffuse\_rate\_before\_peak\_5minus \* time\_within\_cycle - decay\_term[t]

statin\_concentration31 = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

statin\_concentration41 = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

statin\_concentration32 = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

statin\_concentration42 = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

statin\_concentration22 = diffuse\_rate\_before\_peak\_1plus \* time\_within\_cycle - decay\_term[t]

statin\_concentration23 = diffuse\_rate\_before\_peak\_1minus \* time\_within\_cycle - decay\_term[t]

statin\_concentration7 = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

statin\_concentration8 = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

statin\_concentration9 = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

statin\_concentration10 = diffuse\_rate\_before\_peak \* time\_within\_cycle - decay\_term[t]

else:

statin\_concentration = diffuse\_rate\_after\_peak - decay\_term[t]

statin\_concentration1 = diffuse\_rate\_after\_peak - decay\_term[t]

statin\_concentration2 = diffuse\_rate\_after\_peak - decay\_term[t]

statin\_concentration3 = diffuse\_rate\_after\_peak1 - decay\_term[t]

statin\_concentration4 = diffuse\_rate\_after\_peak2 - decay\_term[t]

statin\_concentration5 = diffuse\_rate\_after\_peak3 - decay\_term[t]

statin\_concentration6 = diffuse\_rate\_after\_peak4 - decay\_term[t]

statin\_concentration20 = diffuse\_rate\_after\_peak20 - decay\_term[t]

statin\_concentration21 = diffuse\_rate\_after\_peak21 - decay\_term[t]

statin\_concentration22 = diffuse\_rate\_after\_peak22 - decay\_term[t]

statin\_concentration23 = diffuse\_rate\_after\_peak23 - decay\_term[t]

statin\_concentration31 = diffuse\_rate\_after\_peak31 - decay\_term[t]

statin\_concentration41 = diffuse\_rate\_after\_peak41 - decay\_term[t]

statin\_concentration32 = diffuse\_rate\_after\_peak32 - decay\_term[t]

statin\_concentration42 = diffuse\_rate\_after\_peak42 - decay\_term[t]

statin\_concentration7 = diffuse\_rate\_after\_peak7 - decay\_term[t]

statin\_concentration8 = diffuse\_rate\_after\_peak8 - decay\_term[t]

statin\_concentration9 = diffuse\_rate\_after\_peak9 - decay\_term[t]

statin\_concentration10 = diffuse\_rate\_after\_peak10 - decay\_term[t]

#Increase the statin concentration by the calculated amount

C\_statin[t + 1] = max(0, C\_statin[t] + statin\_concentration)

C\_statin1[t + 1] = max(0, C\_statin1[t] + statin\_concentration1)

C\_statin2[t + 1] = max(0, C\_statin2[t] + statin\_concentration2)

C\_statin3[t + 1] = max(0, C\_statin3[t] + statin\_concentration3)

C\_statin4[t + 1] = max(0, C\_statin4[t] + statin\_concentration4)

C\_statin5[t + 1] = max(0, C\_statin5[t] + statin\_concentration5)

C\_statin6[t + 1] = max(0, C\_statin6[t] + statin\_concentration6)

C\_statin7[t + 1] = max(0, C\_statin7[t] + statin\_concentration20)

C\_statin8[t + 1] = max(0, C\_statin8[t] + statin\_concentration21)

C\_statin9[t + 1] = max(0, C\_statin7[t] + statin\_concentration22)

C\_statin10[t + 1] = max(0, C\_statin8[t] + statin\_concentration23)

C\_statin11[t + 1] = max(0, C\_statin11[t] + statin\_concentration31)

C\_statin12[t + 1] = max(0, C\_statin12[t] + statin\_concentration41)

C\_statin13[t + 1] = max(0, C\_statin13[t] + statin\_concentration32)

C\_statin14[t + 1] = max(0, C\_statin14[t] + statin\_concentration42)

C\_statin15[t + 1] = max(0, C\_statin15[t] + statin\_concentration7)

C\_statin16[t + 1] = max(0, C\_statin16[t] + statin\_concentration8)

C\_statin17[t + 1] = max(0, C\_statin17[t] + statin\_concentration9)

C\_statin18[t + 1] = max(0, C\_statin18[t] + statin\_concentration10)

# Change in LDL is only affected by how much statin affects it

dC\_ldl = statin\_effect[t]

# Calculate the new concentratioNo documentn

C\_ldl[t + 1] = max(0, C\_ldl[t] - dC\_ldl)

#Statin data given in Appendix

time\_data = [0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 5, 6, 8, 10, 12, 24]

statin\_concentration\_data = [0, 2.3, 4.1, 4.65, 4.9, 4.55, 4.2, 3.8, 3.2, 2.4, 2, 1.45, 1.05, 0.75, 0.3]

#Calculating correlation coefficients of the model

def correlation\_coefficient(equation, model):

covariance = np.cov(equation, model)

correlation = covariance[0, 1] / (np.std(equation) \* np.std(model))

return correlation

#Correlation Coefficients

#20%

correlation\_model = correlation\_coefficient(C\_statin[1:16], statin\_concentration\_data[:15])

correlation\_model\_plus20 = correlation\_coefficient(C\_statin1[1:16], statin\_concentration\_data[:15])

correlation\_model\_minus20 = correlation\_coefficient(C\_statin2[1:16], statin\_concentration\_data[:15])

correlation\_model\_after\_peak1\_plus20 = correlation\_coefficient(C\_statin3[1:16], statin\_concentration\_data[:15])

correlation\_model\_after\_peak1\_minus20 = correlation\_coefficient(C\_statin4[1:16], statin\_concentration\_data[:15])

correlation\_model\_after\_peak2\_plus20 = correlation\_coefficient(C\_statin5[1:16], statin\_concentration\_data[:15])

correlation\_model\_after\_peak2\_minus20 = correlation\_coefficient(C\_statin6[1:16], statin\_concentration\_data[:15])

#5%

correlation\_model\_plus5 = correlation\_coefficient(C\_statin7[1:16], statin\_concentration\_data[:15])

correlation\_model\_minus5 = correlation\_coefficient(C\_statin8[1:16], statin\_concentration\_data[:15])

correlation\_model\_after\_peak1\_plus5 = correlation\_coefficient(C\_statin11[1:16], statin\_concentration\_data[:15])

correlation\_model\_after\_peak1\_minus5 = correlation\_coefficient(C\_statin12[1:16], statin\_concentration\_data[:15])

correlation\_model\_after\_peak2\_plus5 = correlation\_coefficient(C\_statin15[1:16], statin\_concentration\_data[:15])

correlation\_model\_after\_peak2\_minus5 = correlation\_coefficient(C\_statin16[1:16], statin\_concentration\_data[:15])

#1%

correlation\_model\_plus1 = correlation\_coefficient(C\_statin9[1:16], statin\_concentration\_data[:15])

correlation\_model\_minus1 = correlation\_coefficient(C\_statin10[1:16], statin\_concentration\_data[:15])

correlation\_model\_after\_peak1\_plus1 = correlation\_coefficient(C\_statin13[1:16], statin\_concentration\_data[:15])

correlation\_model\_after\_peak1\_minus1 = correlation\_coefficient(C\_statin14[1:16], statin\_concentration\_data[:15])

correlation\_model\_after\_peak2\_plus1 = correlation\_coefficient(C\_statin17[1:16], statin\_concentration\_data[:15])

correlation\_model\_after\_peak2\_minus1 = correlation\_coefficient(C\_statin18[1:16], statin\_concentration\_data[:15])

# Print correlation coefficients

print(f"Correlation Coefficient (Model): {correlation\_model:.2f}")

print(f"Correlation Coefficient (+20% Model, before peak parameter): {correlation\_model\_plus20:.2f}")

print(f"Correlation Coefficient (-20% Model, before peak parameter): {correlation\_model\_minus20:.2f}")

print(f"Correlation Coefficient (+20% Model, after peak parameter 1): {correlation\_model\_after\_peak1\_plus20:.2f}")

print(f"Correlation Coefficient (-20% Model, after peak parameter 1): {correlation\_model\_after\_peak1\_minus20:.2f}")

print(f"Correlation Coefficient (+20% Model, after peak parameter 2): {correlation\_model\_after\_peak2\_plus20:.2f}")

print(f"Correlation Coefficient (-20% Model, after peak parameter 2): {correlation\_model\_after\_peak2\_minus20:.2f}")

print(f"Correlation Coefficient (+5% Model, before peak parameter): {correlation\_model\_plus5:.2f}")

print(f"Correlation Coefficient (-5% Model, before peak parameter): {correlation\_model\_minus5:.2f}")

print(f"Correlation Coefficient (+5% Model, after peak parameter 1): {correlation\_model\_after\_peak1\_plus5:.2f}")

print(f"Correlation Coefficient (-5% Model, after peak parameter 1): {correlation\_model\_after\_peak1\_minus5:.2f}")

print(f"Correlation Coefficient (+5% Model, after peak parameter 2): {correlation\_model\_after\_peak2\_plus5:.2f}")

print(f"Correlation Coefficient (-5% Model, after peak parameter 2): {correlation\_model\_after\_peak2\_minus5:.2f}")

print(f"Correlation Coefficient (+1% Model, before peak parameter): {correlation\_model\_plus1:.2f}")

print(f"Correlation Coefficient (-1% Model, before peak parameter): {correlation\_model\_minus1:.2f}")

print(f"Correlation Coefficient (+1% Model, after peak parameter 1): {correlation\_model\_after\_peak1\_plus1:.2f}")

print(f"Correlation Coefficient (-1% Model, after peak parameter 1): {correlation\_model\_after\_peak1\_minus1:.2f}")

print(f"Correlation Coefficient (+1% Model, after peak parameter 2): {correlation\_model\_after\_peak2\_plus1:.2f}")

print(f"Correlation Coefficient (-1% Model, after peak parameter 2): {correlation\_model\_after\_peak2\_minus1:.2f}")

#Plotting Before Peak Paramter

plt.title('Before Peak Parameter')

plt.plot()

plt.plot(range(0, 23), C\_statin[1:24], label='Model',marker='x')

plt.plot(range(0, 23), C\_statin1[1:24], label='+20% Model')

plt.plot(range(0, 23), C\_statin2[1:24], label='-20% Model')

plt.plot(range(0, 23), C\_statin7[1:24], label='+5% Model')

plt.plot(range(0, 23), C\_statin8[1:24], label='-5% Model')

plt.plot(range(0, 23), C\_statin9[1:24], label='+1% Model')

plt.plot(range(0, 23), C\_statin10[1:24], label='-1% Model')

plt.ylabel('Statin Concentration (mg)')

plt.xlabel('Time(hrs)')

plt.plot(time\_data, statin\_concentration\_data, label='Data', color='red', marker='o', linestyle='dashed')

plt.legend()

plt.grid(True)

plt.show()

#Plotting After Peak Parameter 1

plt.plot()

plt.title('After Peak Parameter 1')

plt.plot(range(0, 23), C\_statin[1:24], label='Model',marker='x')

plt.plot(range(0, 23), C\_statin3[1:24], label='+20% Model')

plt.plot(range(0, 23), C\_statin4[1:24], label='-20% Model')

plt.plot(range(0, 23), C\_statin11[1:24], label='+5% Model')

plt.plot(range(0, 23), C\_statin12[1:24], label='-5% Model')

plt.plot(range(0, 23), C\_statin13[1:24], label='+1% Model')

plt.plot(range(0, 23), C\_statin14[1:24], label='-1% Model')

plt.ylabel('Statin Concentration (mg)')

plt.xlabel('Time(hrs)')

plt.plot(time\_data, statin\_concentration\_data, label='Data', color='red', marker='o', linestyle='dashed')

plt.legend()

plt.grid(True)

plt.show()

#Plotting After Peak Parameter 2

plt.plot()

plt.title('After Peak Parameter 2')

plt.plot(range(0, 23), C\_statin[1:24], label='Model',marker='x')

plt.plot(range(0, 23), C\_statin5[1:24], label='+20% Model')

plt.plot(range(0, 23), C\_statin6[1:24], label='-20% Model')

plt.plot(range(0, 23), C\_statin15[1:24], label='+5% Model,')

plt.plot(range(0, 23), C\_statin16[1:24], label='-5% Model')

plt.plot(range(0, 23), C\_statin17[1:24], label='+1% Model')

plt.plot(range(0, 23), C\_statin18[1:24], label='-1% Model')

plt.plot(time\_data, statin\_concentration\_data, label='Data', color='red', marker='o', linestyle='dashed')

plt.ylabel('Statin Concentration (mg)')

plt.xlabel('Time(hrs)')

plt.legend()

plt.grid(True)

plt.show()