The University of Sheffield

**Development of an Agent-based Model Capturing Cellular**

**Interactions Associated with Heart Attack**

Harrison Paul Cooper

Supervised By:

Dr. Dawn Walker

COM3610

<Date>

This report is submitted in partial fulfilment of the requirement for the degree of MComp Computer Science with a Year in Industry by Harrison P. Cooper

Signed Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations which are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name:  Harrison Paul Cooper

Signature:

Date:

Abstract

Ageing is believed to be the largest contributor to the deterioration of the wall lining the inside of our blood vessels. Ageing is dictated by a series of rules which produce emergent behaviours between cells. This can be modelled with the cells as agents to provide a deeper understanding of cell interactions during healing.

Through thorough reading, several parameters have been found that can form a basis for the

testing and experimentation of the software.

This report goes through the current state of the art in agent based modelling, comparing relevant software and potential modifications that could be made to them for this project.

Acknowledgements

I would like to take the time to give my thanks to Dr Dawn Walker for her continued encouragement and expert advice throughout this challenging project.

Glossary

Eukaryotic Cell: A biological cell with a membrane-bound nucleus

Endothelial Cell: Cells that line blood vessels inner surfaces

In Vitro: Experimentation outside a living organism (in glass)

Quiescence: A state of cellular inactivity

Senescence: Deterioration of functional cellular characteristics

Telomere: A segment of DNA at the end of chromosomes

Pro-atherosclerotic: Pertaining to atherosclerosis, which is when arteries thicken from fatty

deposits.

Contents

Title page i

Signed Declaration ii

Abstract iii

Acknowledgements iv

Glossary v

Contents vi-vii

**1 Introduction 1**

1.1 Background Information 1

1.2 Aims and Objectives 1

1.3 Constraints 1

1.4 Summary of Report 1-2

**2 Literature Review 2**

2.1 The Endothelial Cell Cycle 2

2.2 Ageing 3

2.3 Senescent Cells 3

2.4 Atheroprone Sites 3

2.5 Methods of Modelling 4

2.6 Review of Agent Based Software 4-5

2.7 Cell Migration

2.8 Contact Inhibition and Confluence Detection

**3 Requirements and Analysis 5**

3.1 Aims and Objectives 5

3.1.1 Objectives 5

3.1.2 Parameters 5

3.1.3 Rules 6

3.1.4 Emergent Behaviour

3.2 Areas not Covered 6

3.3 Risk Analysis 6-7

3.4 Evaluation and Testing 8

Testing Strategy?

User Evaluation?

Functional Requirements

Non-functional requirements

Justification of ABM methodology

**4 Design 8** Implementation Selection  
  
 Methods of Choice

Agents

General Cell

Endothelial Cell

Quiescent Cell

Senescent Cell  
  
Environment

… Each type of starting cell has a random xyz in range ij …   
  
Rules

General Cell Rules

Apoptosis

Endothelial Rules

Turnover  
  
Senescence  
  
Migration  
  
Quiescence

Mitosis

Stages

Growth

Quiescent Rules

Turnover

Senescence

Movement  
  
Proliferation

Growth

Senescence

Senescent Rules

Turnover   
  
Movement

Growth

Output

Confluence Detection

Scratch Creation

**5 Implementation and Testing**

5.1 Implementation

5.1.1 EC Mitosis

5.1.2 EC Quiescence

5.1.3 EC Senescence

5.1.4 Quiescent Cell Senescence

5.1.5 Quiescent Cell Differentiation

5.1.6 Senescent Cell Growth

5.1.4 Command Line Interface

5.1.5 Simulation Termination

5.2 Testing

5.2.1 Unit Testing

5.2.2 Verification of ABM System

5.2.3 User Testing

**6 Results and Discussion** 6.1 Completed Software and Documentation  
  
 6.2 User Story Analysis  
  
 6.3 Analysis of Underlying CellABM

6.3.1 Findings

6.3.2 Program Efficiency and Runtime Analysis

6.3.3 Goals Achieved  
  
 6.3.4 Further Work

**7 Conclusions**

**References 14-15**

**1 Introduction**

1.1 Agent Based Models

1.1 Background Information

The cells which line our blood vessels are called Endothelial cells (EC), which form a layer known as the Endothelium. This layer of cells can repair itself after injury, which is essential to good health, however, the repair process becomes slower with age due to an increased number of larger cells which actively hinder the healing.

These cells are generally in a confluent layer, therefore a larger number of cells are no longer dividing, however, when they’re wounded, such as an atheroma, the confluence is broken and the cells leave this phase to continue dividing, repairing the damaged tissue. This process is slower in elderly patients due to the increased number of larger cells, or if the same area is damaged a second time after repair. This is due to scar tissue being less capable of mitosis and repair.

1.2 Aims and Objectives

The main aim of this project is to estimate the affect ageing has on the ability for blood vessels to heal after being scratched. The implications of this project will help professionals further understand the process of wound healing and to provide further insights into the conditions affecting the deadly disease atherosclerosis, which can lead to strokes and heart attacks.

The way the main aim will be implemented requires the development of an agent based model (ABM) to encapsulate the key behaviours associated with ECs, including: cell proliferation, apoptosis, and senescence. This model will record the time taken for the wound to repair itself, and observe any emergent behaviour that takes place through the mitosis and movement of the cells, at varying ages. For the basis of producing a software solution, I will be looking at the benefits different types of modelling possess, such as Cellular Automata (CA) and Agent Based Modelling (ABM). Then, I’ll be building on top of current software frameworks, which already provide basic logic, by giving the agents and environment differing behaviours.

I’ll be observing the difference between elderly and younger cells to see how much, if any, age affects repair time.

This project has ample room for expansion; some of these aims include: modelling the problems associated when the endothelium layer doesn’t sufficiently repair in time, and the effect on endothelium repair after successive tears (allowing significant scar tissue to build up), showing the differences in speed and process of the repair. It would also be beneficial to model a more realistic vessel shape as the blood flow turbulence has a dramatic effect on healing ability.

1.3 Constraints

For the initial version of the system that will be developed, the model will be restricted to a single monolayer of ECs in a 2D plane. There will also be restrictions on the number of cells modelled due to the environment size, and these cells will have no interaction with other tissues.

1.4 Summary of Report

Over the next few pages, I’ll summarise the literature read to date, picking out any data that could be used as parameters, go through the current state of several relevant software and how they can be adjusted to this project. Next, we’ll discuss in detail the aims and objectives, what will not be covered and why, and any experiments or tests that will be carrying out at the end of the project. Finally, I end on a conclusion on what’s been found so far, my achievements to date and a project plan to take through semester 2.

**2 Literature Review**

Our blood vessels inner most wall is called the endothelium and is comprised of ECs. These cells have certain behaviours which lead them, over time, to decrease their rate of healing. This can cause problems as the damaged artery wall allows for fatty material to build up over time. If this builds up too much or ruptures, a blood clot can form blocking the artery; if this artery supplies blood to the heart it causes a heart attack. There are several ways software can be used to model this behaviour to better understand and predict undesirable affects, such as atheroma formation. The way this project tackles modelling is an agent based approach, where each EC is simulated and can move around the model independently.

2.1 The Endothelial Cell Cycle

Firstly, it’s important to fully understand the mechanisms by which our ECs divide and any biological factors that can change its behaviour. ECs are a specific type of Eukaryotic Cell that line our blood vessels. When these cells are healthy, they secrete molecules, such as hormones, into the blood stream to maintain homeostasis [1]. This is vital as it helps fend off disease progression, keeping the individual healthy.

EC’s, like other Eukaryotic Cells undergo several distinct phases during replication as shown in the diagram below, however have another stage they can enter before S Phase.

Figure 2.1: Phases of eukaryotic cell cycle [2]

Stages G1, S and G2 are called Interphase; this is the time when the cell is increasing in size, and the lengths of time in each stage are proportional to their relative lengths. As shown in the figure, during S phase, the DNA is replicated forming a copy of itself which moves onto M phase (mitosis), when the enlarged cell splits into 2 identical daughter cells [3]. The length of time for a normal Eukaryotic Cell to undergo proliferation is around 24 hours, with 1 hour of that being the M phase, therefore 23 hours (96%) of the time is during cell growth and DNA replication, during which time the cell grows to be about twice its size [3].

However, for ECs there is another cycle between the G1 and S phase. This is called the G0 phase and generally known as the quiescence state. This is a state of inactivity, usually induced when EC proliferation is no longer required. If there is a stressor, such as a decrease in external pressure due to the ECs spreading out or moving, the quiescent cell can move out of G0 back into the normal eukaryotic cell cycle [4]. However, if the EC stays in the quiescent state for too long, it’s possible for it to develop into a senescent cell over time where it will never return to the normal cycle [5].

In general, ECs are long, flat cells around 5-10μm in radius and 1-2μm wide [6].

2.2 Ageing

An important factor that contributes to pro-atherosclerotic changes to the endothelium is ageing [7]. The number of times an EC can divide is limited, and once reached the cell goes into growth arrest, known as senescence [8]. This is due to the shortening of the ECs telomeres (the end parts of DNA) by 50-200 base pairs each time the cell proliferates. Once these telomeres are shorter than a critical length, the cell becomes senescent. The number of times a cell can proliferate is known as the Hayflick Limit, and for normal ECs is around 50 [9].

Studies have shown that senescent cells accumulate in tissues with age [http://www.pnas.org/content/pnas/92/20/9363.full.pdf, https://onlinelibrary.wiley.com/doi/10.1111/j.1474-9726.2009.00481.x], and Cellular Senescence in Aging Primates [http://science.sciencemag.org.sheffield.idm.oclc.org/content/311/5765/1257.full] has shown that the number of senescent cells increases exponentially with age, with total cell count reaching >15% senescent in elderly cases. The limitations of this paper include: that the results are from baboons, not humans and so the lifespan is only from 5-30, and the cells were taken from the medial aspect of the arm rather than the endothelium layer. However, this paper is useful in the fact that baboon’s telomeres, like humans shorten with proliferation, and the baboon’s cells also undergo senescence.

2.3 Senescent Cells

It has been noted the senescent ECs have several characteristics which differ them from normal ECs. First of all, they are unable to undergo mitosis and have a turnover rate of around 3 years [7], they become enlarged after entering this state [10] and slow down surrounding ECs. Warboys suggests that senescent ECs could be the main contributor and initiator of atherosclerosis. In vitro, it has been seen that senescence in the ECs increases during a turbulent, disturbed flow, from 1% of EC being senescent using a 13 dynes/cm2 uniform flow compared to just over 2% senescent EC when exposed to a flow fluctuating between +/- 5 dynes/cm2 at 1Hz. It’s also noted that for these two categories, the number of multinucleate cells with a diameter > 100µm increased from 0.5% to 1.5. This increase in number of senescent ECs is believed to be due to an increase turnover rate of ECs at these turbulent atheroprone sites. Meaning that this increased level of proliferation should be considered when developing my senescent cell model. It can also be hypothesised here that in general, over time, more cell proliferation will occur and thus there will be an increase in the total number of senescent cells within the environment.

Another important fact Warboys reveals is that due to the size of the senescent ECs, this has a detrimental effect to the speeds of its neighbouring cells, acting as a blockage, and slowing them down. This can hinder wound healing as it will take longer for healthy mitotic ECs to fill the gap. As mentioned above, there’s is also an increase in the number of senescent cells over time, therefore I expect my model to show that with age, it takes longer for any wounds to heal.

2.4 Atheroprone Sites

Not all ECs within our blood vessel have the same physiological behaviours; this is due to the differing environmental factors within the vessels, discussed above. This leads to parts of our blood vessels under going higher levels of injury than others. In fact, the main disease this project is aimed at further understanding, atherosclerosis, is rather specific, and can be most commonly be found at the bends or branches of arterial trees [11]. These bends and branches are known as atherosusceptible sites, which have enhanced proinflamitory activation, increasing rate of proliferation [11]. These atherosusceptible sites therefore have a higher rate of injury and cell turnover compared to EC at atherprotected sites [12, 13, 14]. Analysis by Chaudhury et al showed that the ECs at Atheroprone sites express proteins that respond to lipopolysaccharides by priming for apoptosis and proliferation [11]. They also state that wherever JNK1 is active is where apoptosis and EC turnover occur in arteries.

I will therefore be looking at branches and bends within my model as they are the areas where there is the highest level of turbulence and concentration of JNK; leading to the greatest injury of the endothelium wall. Which, in turn has the greatest concentration of EC apoptosis and proliferation.

2.5 Methods of Modelling

There are two clear options for modelling the interactions between ECs and senescent cells. Cellular automata (CA) is an orthogonal grid of similar cells that interact with their neighbouring cells. Its advantages are that runtime is extremely quick and it can produce complex macro-scale emergent behaviour of the interacting cells [15]. However, the disadvantages are that due to the orthogonal grid, cells are fixed in place, unable to move; this is very much a simplification of the project as ECs move around on the endothelium to fill gaps and is an important factor for wound healing. Another disadvantage of CA is that it can only model local interaction between neighbouring cells, therefore any change further away from the cell won’t be noticed until it cascades down the subsequent neighbouring cells over several iterations

Another modelling method would be to use equation in continuum modelling. Which primarily uses differential equations to model the population. These differential equations can be used to show predator and prey relationships such as: where the number of predators directly effects the prey and vice-versa. This equations do not model each cell and instead would focus on modelling the cell density over time.

Finally, an Agent Based Model is a dynamic system of interacting agents. This dynamic property is crucial in producing realistic emergent behaviours as it more closely resembles what occurs in nature. The downside is, that due to the free movement of the cells, expensive calculations must be implemented to resolve overlapping and collisions in more accurate systems, introducing scalability issues. However, there are several methods out there for reducing the time taken; Epitheliome, an ABM created by Dr. Dawn Walker [16], embedded their overlap logic as C within their MATLAB code. This is also possible within python [17]. For these reasons, I believe it’s best to complete this project using an Agent Based Model.

2.6 Review of Agent Based Software

There are several existing ABM that have been developed to monitor cellular interactions. The first, Epitheliome, by Dr. Dawn Walker [16] is the most applicable to my research question. It uses an agent based modelling approach to visualise the time taken and movement of endothelial cells into a wound with different levels of Calcium ions in the environment. The underlying logic of Epitheliome is laid out more in [https://www-sciencedirect-com.sheffield.idm.oclc.org/science/article/pii/S030326470400070X#FIG2] It accurately models the contact inhibition of cells and differentiation of endothelial cells to quiescent cells in the G0 phase.

The implementation of the cell cycle is similar to what was discussed in 2.1 with each cell progressing one tick through the cell cycle each iteration. With the duration of S-G2-M phase and G1 phase being slightly different for each cell, imitating the random nature of cells.

The limitations of this approach to my project is the lack of senescent cell differentiation present in the simulations which would act as barriers to the endothelial and quiescent cells during migration, and therefore Epitheliome is unable to monitor the rate of wound healing with age.

So far, I’ve tested two computer programs that use agent based modelling to allow for the type of emergent biological behaviours I’m looking for. The first program is SPARK which is a lightweight and efficient tool for CA. Being so lightweight, Spark is very capable of modelling the number of cells I would require for this project; in fact, it can simulate a grid of 101x101 with 10201 cells in real time. Its programs are written in SPARK-PL which is translated into Java source code, meaning a significant amount of time will be required to learn the new language. Another downside is that being a CA the ECs are embedded into the endothelial matrix (the layer the cells sit on top) and therefore are unable to move around the system, and as explained above, this is a simplification of reality as ECs are constantly moving or shifting on top of the endothelium layer.

The other program is a python based ABM by Marziha Tehrani, a PhD student, called CellABM. It’s used to model the interactions between cancer cells and stem cells. It has several classes which allows the user to easily change the rules of each phase of the cell cycle along with the initial cell parameters, such as size, direction and speed. However at large cell numbers is rather slow and has no capabilities of interacting with the agents during the simulation.

There are three other software frameworks I’ve looked at, but not as in-depth as the two described above; they are: Net Logo, Mason, and Repast.



Below, I have quantitatively summarised the strengths of each software in relation to each other. I use a scoring system between 1 (low) and 5 (high) and multiply that by the weight of each category. This gives a total showing the overall usefulness of the software.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | **Comparison of Software** | | | | |
| **Spark** | **CellABM** | **Net Logo** | **Mason** | **Repast** |
| **Method (CA or ABM)** | 0.1 | 1 | 5 | 5 | 5 | 5 |
| **Contact Resolution** | 0.2 | 1 | 4 | 2 | 2 | 3 |
| **Language** | 0.2 | 2 | 5 | 3 | 3 | 3 |
| **Interaction during simulation (GUI)** | 0.1 | 3 | 1 | 5 | 4 | 5 |
| **Speed** | 0.3 | 4 | 2 | 4 | 4 | 4 |
| **Familiarisation** | 0.1 | 3 | 5 | 1 | 1 | 1 |
| **Total** | 1 | 2.8 | 3.5 | 3.3 | 3.2 | 3.5 |

Table 2.1: Quantifying the differences between possible software

From Table 2.1 CellABM and Repast both score the highest at 3.5 meaning they’re equally suited to this project. However, the defining factors between the two are the graphical user interface (GUI) where Repast scored 5 and CellABM 1, and familiarisation where Repast scored 1 and CellABM 5.

As this project doesn’t require a GUI as there is no interaction with the simulation whilst running, familiarisation is the more important metric and so CellABM is the software of choice.

2.7 Cell Migration

A key element of ECs is their ability to migrate. This EC migration is an important fundamental process to our life, allowing the formation of embryos, organs and tissues. For developed Humans, migration allows for immunosuppression and more importantly to my research question, the migration of ECs into the wound of a damaged blood vessel to restore the vessels integrity [https://link.springer.com/article/10.1007/s00018-014-1678-0].

ECs will migrate in a random manner if there are no external stimuli and will diffuse into the available space [https://www-sciencedirect-com.sheffield.idm.oclc.org/science/article/pii/S0006349570863470] until a confluence is formed. Once the cells have formed the confluence, the bond to each other and the endothelial surface, preventing further migration.

2.8 Contact Inhibition and Confluence Detection

Over time, ECs will migrate into any open space and if possible proliferate to form new ECs. This will continue to occur until the area is filled with ECs and there is no more space for proliferation. When cells come into contact with each other, cell growth is arrested by a process known as Contact Inhibition [https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2780760/], meaning that when a monolayer is formed ECs are no longer able to proliferate. If the ECs are unable to proliferate they eventually differentiate into Quiescent cells where they no longer undergo mitosis.

Confluence Detection occurs when migration and proliferation is no longer possible due to the contact inhibition on the monolayer. At this point, several of the ECs will have differentiated into Quiescent Cells.

**3 Requirements and Analysis**

3.1 Methodology

For the development of the program to discover the effect age has on heart attacks, an Agent Based Model will provide the best results for the user. ABMs model each cell individually with their own parameters, allowing for a more distributed representation of the cells, such as each cell can vary in radius slightly from each other. An ABM also provides a graphical output of how the cells move, allowing us to better understand what’s happening with the emergent behaviour in a more visual way. The ABM approach is better than a continuum approach as in continuum modelling there is no individual agent representation and so approximations may be too significant to produce reliable results. Cellular automata wasn’t chosen as it would incorrectly model the endothelial cells on the environment, not allowing them to migrate into the wound and therefore not answering the research question.

3.2 Aims and Requirements

The main aim of this project is to demonstrate and help professional understand further the affect ageing, and other physiological factors, has on the ability for a wounded area of ECs to repair itself. The main observation will be time taken for the ECs to divide and move into the gap of the wound, once more forming a confluent layer.

To facilitate the main aim, we’ve seen the benefits several current software have, to form the start of the project; however, they lack the correct logic or behaviours that occurs within blood vessels. Below, I outline the functional and non-functional requirements, parameters, and rules that need to be met to produce an accurate and correct model.

3.2.1 Functional Requirements

|  |
| --- |
| **It is critical that the system:** |
| Uses an appropriate time scale for each iteration |
| Creates a wound when a confluence is made |
| Model’s Senescent cells |

Table 3.1: Critical functional requirements

|  |
| --- |
| **It is important that the system:** |
| Produces quiescent cells when proliferation is no longer possible |
| Models quiescent cells differentiating to proliferating cells |
| Models proliferating cells differentiating to senescent cells |
| Tells the user how long it took for wound healing to occur |
| Produces graphs of cell locations each iteration |

Table 3.2: Important functional requirements

|  |
| --- |
| **It is desirable that the system:** |
| Forms a confluence before being wounded |
| Models Senescent Cells as barriers |
| Stops the simulation when second confluence is formed |

Table 3.3: Desirable functional requirements

|  |
| --- |
| **It is optional that the system:** |
| Models senescent cell death |

Table 3.4: Optional functional requirements

3.2.2 Non-functional Requirements

|  |
| --- |
| **It is desirable that the system:** |
| Is simple to run from the command line |
| Is commented well for future development |

3.5: Non-functional requirements

3.2.3 Parameters

These parameters have either been gained from literature review or are an educated guess which will be refined heuristically on the final product.

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Data** | **Source** |
| PC diameter | 10-20μm | Literature Review |
| Senescent cell diameter | < 100μm | Literature Review |
| PC speed | 1μm/min | Educated Guess |
| Senescent cell speed | 0 | Literature Review |
| PC direction | Random | Educated Guess |
| PC growth factor | 2 x during proliferation | Literature Review |
| Cell turnover | 50 times | Literature Review |
| PC turnover time | 24hrs | Literature Review |
| Senescent cell turnover time | 3 days | Literature Review |
| Time period | 6 hours | Educated Guess |

Table 3.6: Values associated with the parameters for the software.

3.2.3 Rules

|  |  |
| --- | --- |
| **Rule Name** | **Behaviour** |
| Mitosis | * Splitting enlarged EC into 2 equal sized half cells. |
| Apoptosis | * When turnover limit reached, enter * Remove cell from environment |
| Quiescence | * When no more proliferation possible, enter * When proliferation possible, exit * No mitosis |
| Senescence | * When cell turnover hit, enter * When cell has been quiescent for long enough, enter * Static * Enter cell growth * No mitosis * Cell turnover = 3 years |
| Collision Correction | * Adjust overlapping cells so they no longer are |
| Cell growth | * Double in size for ECs * Grow up to 10 times in size for senescent cells |

Table 3.7: Check-list of the behaviours each implemented rule should have.

3.2.5 Emergent Behaviours

Emergent behaviours arise through the interaction of the above rules and are not hard-coded, but observed. Some of these behaviours in action include the formation of tissues and organs and the expansion of tumours. For this project, I expect to see an emergent behaviour of wound healing when the blood vessel is damaged, by having the Quiescent cells differentiate back to Proliferating cells (PCs) due to the increased space, and these PCs migrating and proliferating to fill the space; once more forming a monolayer of cells which will differentiate back to Quiescent Cells. Another expected emergent behaviour is the obstruction of migration of PCs from the Senescent cells leading to delayed healing, increasing the chances of forming an atheroma and blood clot, leading to a heart attack.

3.3 Areas not Covered

Either due to time or computational constraints there are a few areas that this project will not be covering. Firstly, due to the lack of understanding the advanced Biology of the inner workings of ECs, I will be unable to implement all the of rules biologists have found that cause cellular senescence.

Another area I will not be covering are the multiple ways the endothelial monolayer gap can be filled during healing. I am only modelling the spreading of adjacent ECs into the gap due to the decrease in pressure caused by the lack of cells pushing back. The other ways the gap can be filled include: hyperplasia of existing endothelial cells and engraftment of circulating endothelial progenitor cells [7].

I am also assuming, that I am modelling ECs from a healthy person with a Hayflick limit of 50, ignoring deficiencies such as Werner syndrome which causes individuals to have a population growth of 53% and total replicative life span of 27% compared to normal cells [18].

I will not be creating a graphical user interface (GUI) for the user to change parameters on the fly in the simulation. All parameters will be set at the beginning of the simulation and shall remain unchanged. To observe the effect of the changing parameters, several simulations must be run with varying initial conditions.

3.4 Risk Analysis

I’ve included all the risks I believe are associated with my project below. I outline the nature of the risk, then give it a likelihood and impact score from 1 – 4, 1 being unlikely / negligible and 4 being very likely / project threatening then provide a mitigation plan to decrease severity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **Likelihood** | | | |
| Very unlikely  1 | Unlikely  2 | Likely  3 | Very Likely  4 |
| **Impact** | Negligible 1 | 1 | 2 | 3 | 4 |
| Low 2 | 2 | 4 | 6 | 8 |
| Significant 3 | 3 | 6 | 9 | 12 |
| Catastrophic 4 | 4 | 8 | 12 | 16 |

Table 3.8: Risk Rating Matrix where Risk Rating = Likelihood x Impact

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Event | Likelihood | Impact | Risk Rating | Mitigation |
| Loss of developers’ code | 1 | 4 | 4 | Backups of the developers’ machine are take daily to an external hard-drive. The code will also be tracked on GitHub. |
| External event prevents progression | 2 | 3 | 6 | Careful project planning implementation of contingency plans if developer starts to fall behind. Some weeks are designed to have less work in case developer needs to catch up. |
| Optimistic project plan | 3 | 3 | 9 | Enough time must be given to the development of the software and is something that shouldn’t be rushed. Adjustment to project plan may be required if developer start to lag. |
| Completion of code hinders completion of dissertation | 2 | 4 | 8 | Enough time will be given to produce several drafts of the final dissertation in the project plan. |
| New functions not working with current software | 2 | 3 | 6 | Ensuring there are no compatibility issues and correct design practices are followed, such as the creation of UML diagrams showing function interaction. |
| Contact resolution scalability not fixed | 3 | 4 | 12 | Review of different software for contact resolution. Decreasing experiment area is a last resort to ensuring a confluence can be modelled. |
| Lack of accurate data | 4 | 3 | 12 | Continual reviewing of papers surrounding the topic for any extra hints. Otherwise a heuristic approach with several simulations should provide accurate results. |
| System too slow for use under standard conditions | 3 | 4 | 12 | Avoid implementation of nested loops, and constantly assess performance. Possibility of running simulation on Iceberg. |
| Requirements change during development | 1 | 3 | 3 | The code will be implemented in an Object Orientated manner, providing modularity of functions with little refactoring. |

Table 3.9: Risk identification, analysis and planned mitigations.

3.5 Evaluation and Testing

Tests will mainly focus on what occurs after the wound has been created. For this to occur, a confluence must be formed. To save time, one simulation can be run at the desired environment size to determine the number of cells the simulation stabilises towards so this can be used as the starting condition of future tests, saving time as confluence formation won’t be simulated. Theoretically, if the environment is 2500μm2 and each proliferating cell can grow up to 10μm in diameter, 2500 cells can fit onto the environment. However, this doesn’t factor the size of senescent cells or the fact cells can be of different sizes.

There are several tests that could be used to measure the success of the project once everything’s completed.

Test 1 would involve the variation of age and the subsequent measurement of change in time for the wound to heal. To vary age, as shown in the literature review, the number of starting senescent cells within the model will change, with younger patients having fewer senescent cells and elderly patients more. This test is paramount as it will be the main evidence used to answer the main aim.

Test 2 involves varying the wound size and observing the time taken for the wound to heal for each age group.

Test 3 involves producing simple test cases on the simulation to show the rule behaviours in a controlled environment where no other rules are acting on the cell. This will show that each rule works on the micro scale and therefore will work when scaled up to macro size.

Test 4 involves qualitative validation of whether the emergent behaviour looks like the predicted behaviour.

# Test 5 involves local sensitivity testing: if a parameter is varied by a small amount, what is the change in the model? This result can then be used as further calibration of the model parameters as feedback [A validation methodology for agent-based simulations]. However, this type of testing has its limitations as it only varies one parameter at a time, whereas the interaction between parameters could be more important.

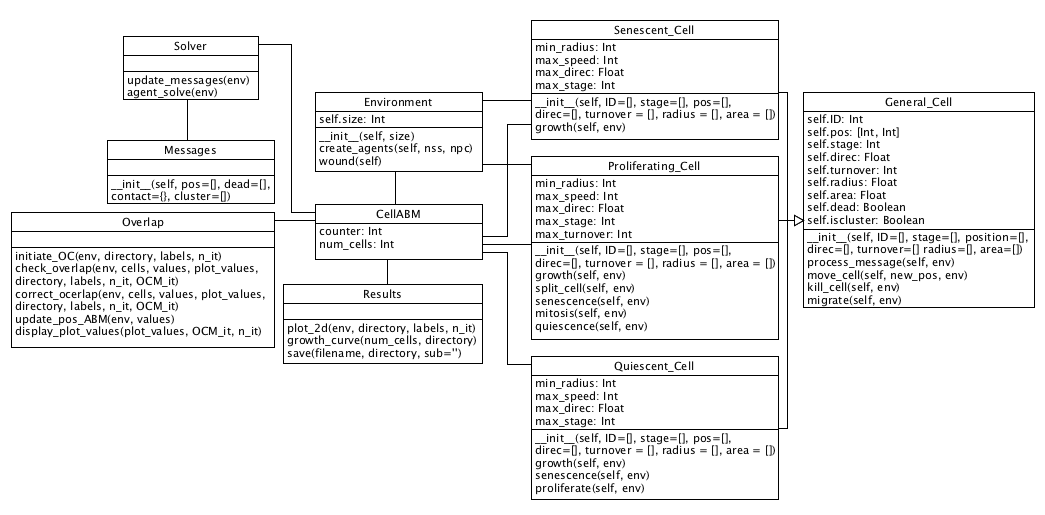
Test 6 would involve performing a statistical test on the time taken for the wound to close whilst varying the percentage of Senescent cells in the model. If the simulation is run enough times, Student’s T test can be used to determine whether the increase in Senescent Cells is significant enough to change the model’s behaviour. However, this test is very rigorous and requires a lot of simulated data and real world experiments to analyse the model and so may not be feasible.

The evaluation of my work will include the results I gather from the tests above and comparing them against current literature showing blood vessel wound healing in vitro.

1. **Design**As seen above, there are several ways of developing an ABM to implement the requirements. In this chapter, we will explore the underling language of the program and how it can be used to model an ABM, then discuss the class diagram and flow charts of how information will flow through the system, finally discussing what simulations will be run to answer the research question.
   1. **An overview of Python and its Class System**Since the implementation will be driven using CellABM, Python is the language of choice for this project. Python is similar to other widely used languages such as Java and JavaScript [https://www.python.org/doc/essays/comparisons/] in that it is an interpreted and an Object Orientated Programming (OOP) language. However, Python has some significant differences that lead it to be syntactically easier to read than Java and it has better code reuse than JavaScript. A Python program is generally 3-5 times smaller than the same program written in Java, thus decreasing development time and reducing the chance of bugs.   
      In Python, data is encapsulated inside objects. These objects can change their own data or interact with other objects. This method of object orientation can be used to represent the different types of cells required in the program.

Python also uses inheritance. This means that instead of writing the same function for several classes, there can be one parent class with the function and other classes can inherited that function from them, reducing repeated code. In the case of CellABM, this means each cell type: Proliferative, Senescent, and Quiescent can all inherit the same apoptosis (cell death) function from an overall general\_cell class.

* 1. **Class Diagrams**This class diagram is intended to show the information flow throughout the program and how the classes communicate with each other. An important feature to note is the general\_cell class acting as a parent class for the three cell types.

Figure 4.1: Class diagram of CellABM

* 1. **Environment**At the beginning of the program, the user will define several key parameters used to initialise the environment. Notably, the size (in micrometres), the number of starting Proliferating Cells and the number of starting Senescent Cells. This allows the user to define cell ratios for differing patient ages in accordance with the research question.

The Environment class creates the starting agents with a random set of parameters taken from a distributed range given, and appends them to a list of starting agents.

The environment will be modelled as a discrete space where agents cannot leave, to preserve computational runtime, and will provide the space for the agents to interact with each other. Cell positions can be mapped into this 2D space using a 2D array of equal size to the user’s definition and giving each cell an [x, y] coordinate.

Although in biology endothelial cells live in a 3D space, they tend not to over-lap one another, thus creating a 2D plane. For this reason, it is believed that little information is lost by modelling in 2D.

* 1. **Theorised Program Flow**Below are the guides that will be followed during the development of the program. They provide the road map of how each class and function interacts with each other, leading to emergent behaviour of the cells. A quick overview of the cellular differentiation is given in Figure 4.5.1, showing how, generally, endothelial cells start out being normal Proliferating cells, then they can either move onto being Quiescent or Senescent. Quiescent cells can revert to Proliferating cells or turn Senescent if they persist long enough. As shown, Senescent cells act as a sink, trapping the cell in that state until the end of the simulation.

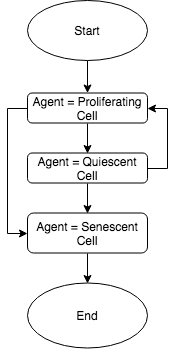
****

Figure 4.2: Cellular differentiation

* + 1. **CellABM**This flow chart shows how the overall main class will run. It will start by taking the parameters from the user, initialising the environment with these parameters and ensuring the initial agents aren’t overlapping. When this is set up, the program will move into an iterative process of solving the agents (allowing to perform their programmed rules), ensuring they aren’t overlapping and then checking the number of quiescent cells in the environment. If the number of quiescent cells is larger than the threshold, the environment simulates the wound and the loop continues. At the end of each iteration, a graph will be plotted showing the location of each agent on the environment.

When the number of quiescent cells passes the threshold for a second time, the simulation is stopped as a confluence will have re-formed, this will also produce a growth curve of the agents over the iterations.

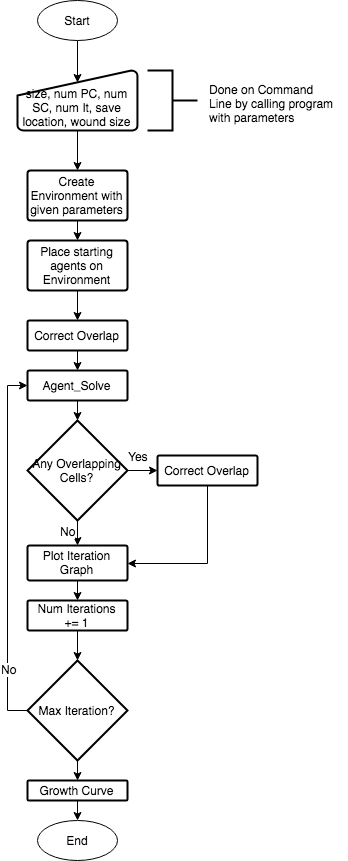


Figure 4.3: CellABM class overview

* + 1. **Cell Differentiation**

A more thorough plan of cell evolution is given below in figure 4.6.2.1. This shows the intended logic behind each of the cell stages, and how the cells will differentiate with the simulation.

The Proliferative cells have both a turnover value and stage value (not shown here). The turnover is the Hayflick Limit mentioned in the Literature review, and once reached, the proliferative cell will differentiate into a senescent cell. Cell stage however, will be used to track what stage in the cell cycle the cell is at and to decide whether the proliferative cell should undergo mitosis that iteration.

The quiescent and senescent cells only have a stage value associated with them. As these cells do not undergo mitosis, there is no need to track what stage of the cell cycle these cells are in and is therefore used as the Hayflick representation.

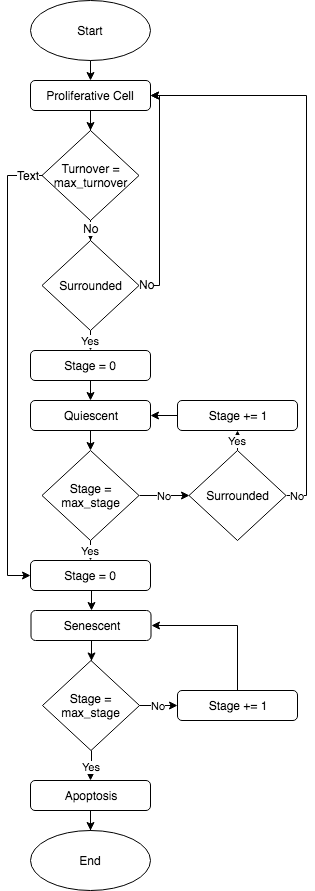
****

Figure 4.4: Cell Differentiation Steps

* + 1. **Agent\_Solve**This flow chart has been created by looking at the current underlying logic for the agent\_solve class in CellABM and including the extra steps required to allow for the new rules and cells the question requires. These steps will be run on every cell in the model.

For Proliferative and Quiescent Cells, it is important to test whether they will become Senescent first as if this is true it shows the cells have passed the Hayflick limit, as seen in chapter 2.2, and their telomere ends have passed their critical length, so the cell must turn Senescent.

Senescent cells are unable to differentiate back to a PC or QC, thus ever iteration they only test to see whether they will under apoptosis.

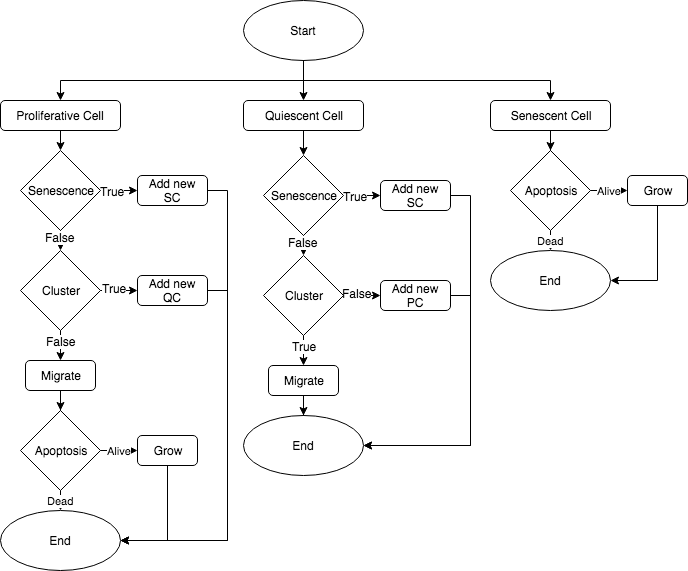


Figure4.5: Overview of agent\_solve class flow

* 1. **Simulations to Run**

As the main objective of this project is to determine the different times taken for a wound to heal whilst varying the person’s age, several simulations will be run with varying percentages of senescent cells in accordance with the primate paper in chapter 2.2, with the time taken being plotted.

ABMs are generally stochastic, and CellABM is no different. The initial placement of cells onto the environment is random, so too is their starting size and stage in the cell cycle. Due to these random variables, several simulations with the same starting parameters must be run to achieve adequate analysis of the model.

Results of the simulations will be compared to an in vitro study of human umbilical vein endothelial cells which have been wounded with p20 pipette (around 400μm) on an area of 1mm by 1mm [https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5154238/pdf/kcam-08-05-969641.pdf].

To show that the implemented rules are working as expected, micro simulations will be run with set parameters and cell stages to ensure the behaviours will work correctly on the macro scale. The simulations will involve a low number of cells, around 1 to 10, and will be simulated for the least amount of time required to observe the desired behaviour.

1. **Implementation and Testing**This section is concerned with the final process involved with implementing the background logic to produce the desired emergent behaviours. It will go through several of the rules outlined in 3.1.3 in detail. Then move onto testing of each of these rules, unit tests, and acceptance testing. 
   1. Implementation

CellABM already had several sections of the program and logic developed; therefore, this chapter will focus on the areas of the program that have been changed or developed to produce the required emergent behaviour and observations.  
CellABM was originally written in Python 2.7 which was released in 2010 but is seen as the legacy version of the language, with Python 3.6 being the supported language of choice for present and future programs [https://wiki.python.org/moin/Python2orPython3]. Thankfully many of the modules from Python 2.7 have been ported over to Python 3.6, such as NumPy which CellABM uses for matrix creation and mathematical functions. This leaves only basic refactoring of the code and changing print statements to functions to make CellABM Python 3.6 compatible. The changes brought in by Python 3.6 are to adjust certain aspects of the old Python program language to be simpler for new programmers to develop, and make it easier to read.

These rules have been created using the logic shown in the design flow charts.

5.1.1 Changes to CellABM

A significant amount of refactoring has taken place to convert the original code into PEP8 [https://www.python.org/dev/peps/pep-0008/] and a significant number of unused parameters have been removed. In addition to these adaptions, a new agent has been introduced to increase the total number to 3. Docstrings have been created for each class and method, allowing future development of the program to be achieved easily.

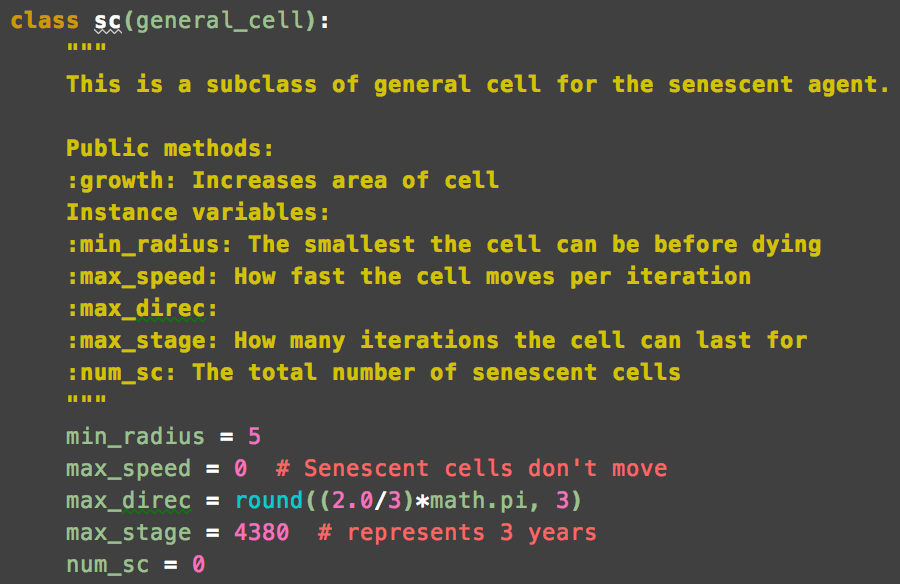
5.1.2 Senescent Agent

5.1.2.1 Class overview

The senescent agent is a subclass of the general cell class allowing for varying parameters to be specific to the senescent cell. As proliferating and quiescent cells can differentiate into senescent cells and they are capable of being 5μm radius this is the minimum radius the senescent cells can be.

These cells are intended to act as blockers slowing down the wound healing, therefore a speed of 0 has been assigned to them, ensuring they don’t migrate around the simulation.

As seen in chapter 2.3 senescent cells can live upwards of three years, therefore as each iteration is six hours, the cells can be in the simulation for a maximum of 4380 iterations. However, it is very unlikely for a simulation to run for this long and is intended to be used alongside the initial creation of senescent cells where they are given a random stage between 1 and 4380.

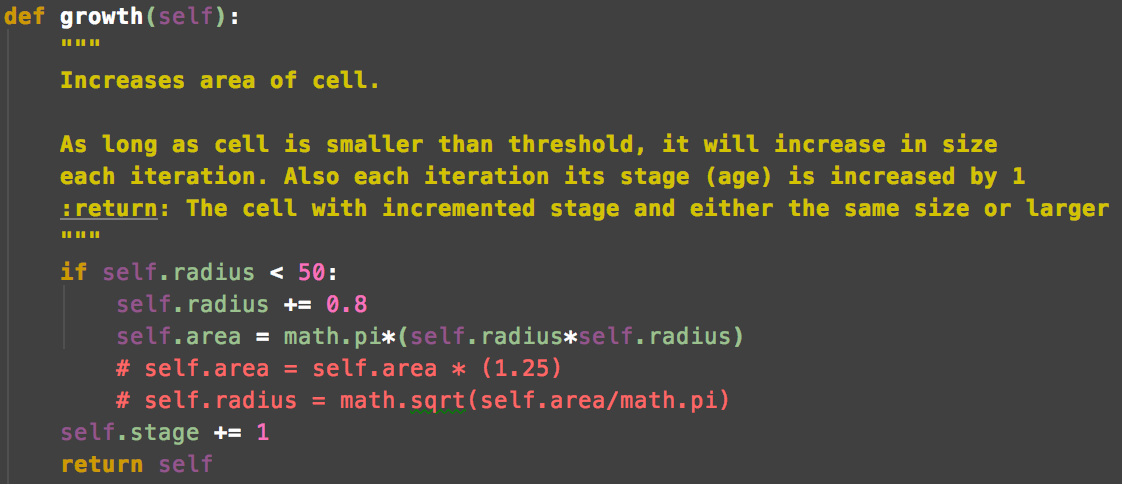


5.1.2.2 Growth

From meeting with my domain expert, Prof. Paul Evans, it was found that senescent cells can, in some cases, grow up to 10 times their original size in the first two weeks, then they stay relatively the same size for the rest of their life. This means they can potentially grow up to 100μm in diameter. The program controls this by calling a growth function each iteration increasing the cells radius by a set amount.

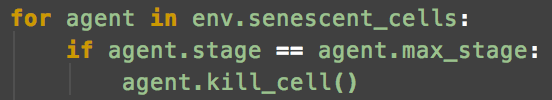
As the senescent cells grow within their first two weeks and each iteration equates to six hours of simulated time, they should reach 100μm within 56 iterations. To achieve this the growth function increases the cells radius by 0.8μm each iteration. However, this on its own has no prevention for the cell to increase over 100μm. To control this a condition is used to ensure only cells that are smaller than 100μm diameter have their radius increased.

This function also increases the age of the cell by 1 each iteration to allow for apoptosis.



5.1.2.3 Apoptosis

When senescent cells have lived for three years, stage = 4380, they undergo apoptosis.



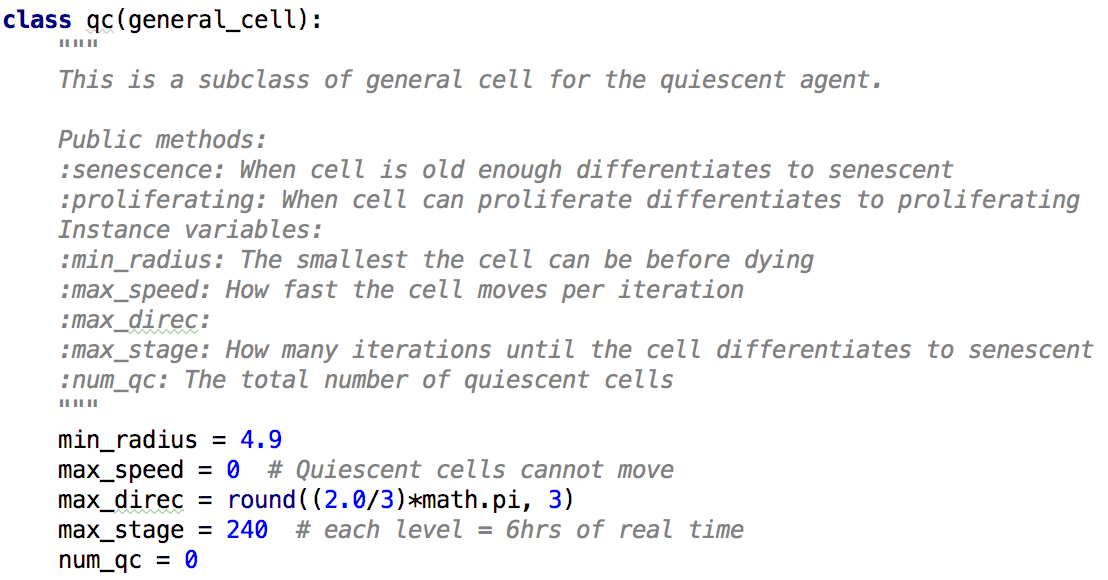
5.1.3 Quiescent Agent

5.1.3.1 Class Overview

The quiescent agent is a subclass of the general cell allowing different parameters to the senescent and proliferating agents. As proliferating cells differentiate into quiescent cells and the smallest a PC can be is 4.9μm in radius, the same is true for the QC.

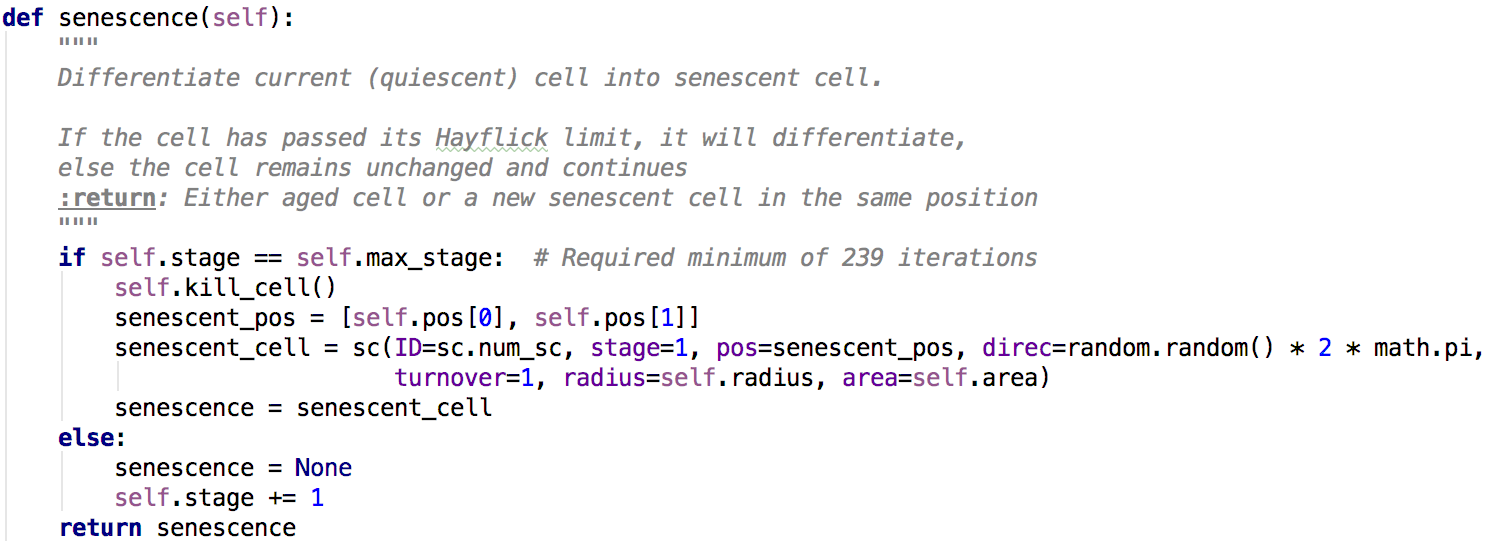
Quiescent cells occur when proliferation is no longer possible, generally when a monolayer has been formed, for this reason the agents have been programmed to have a speed of 0 and so they don’t actively migrate in the simulation.

It has been theorised here that QCs live for around two months before turning senescent. However, the simulation usually isn’t run for this long and new QCs are created with a stage of 1, therefore QC differentiation to senescent cells will rarely be seen.



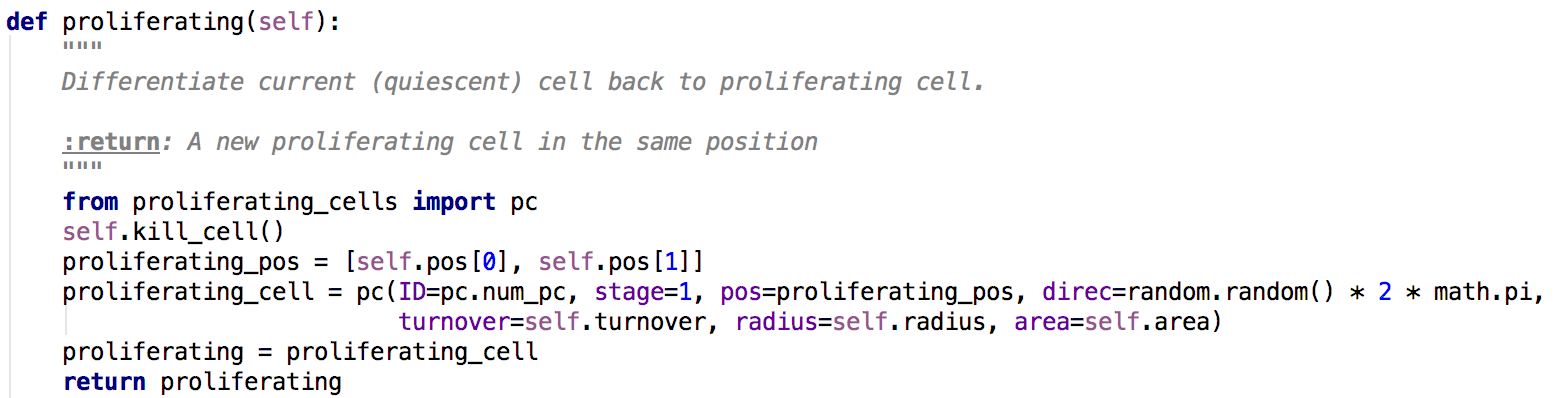
5.1.3.2 Senescence

Quiescent cells (QCs) can differentiate into Senescent Cells (SCs) when they have been in the simulation for long enough. Each iteration the QC is tested to see whether it can differentiate, if it can the current QC is removed from the simulation by killing it and a new SC is created with the original QCs position, radius, and area. If differentiation is not possible the age (stage) of the cell is increased by one.



5.1.3.3 Proliferating

When there is adequate space around the Quiescent Cell (QC) it can differentiate back to a Proliferating Cell (PC) as seen in Figure 4.2. Each iteration the number of cells surrounding the QC is added up and if it is under the user set threshold it is believed that space has freed up around the QC, allowing it to proliferate. The differentiation is made by killing off the QC and creating a new PC with the same: position, turnover, radius, and area of the QC.



5.1.4 Proliferating Agent

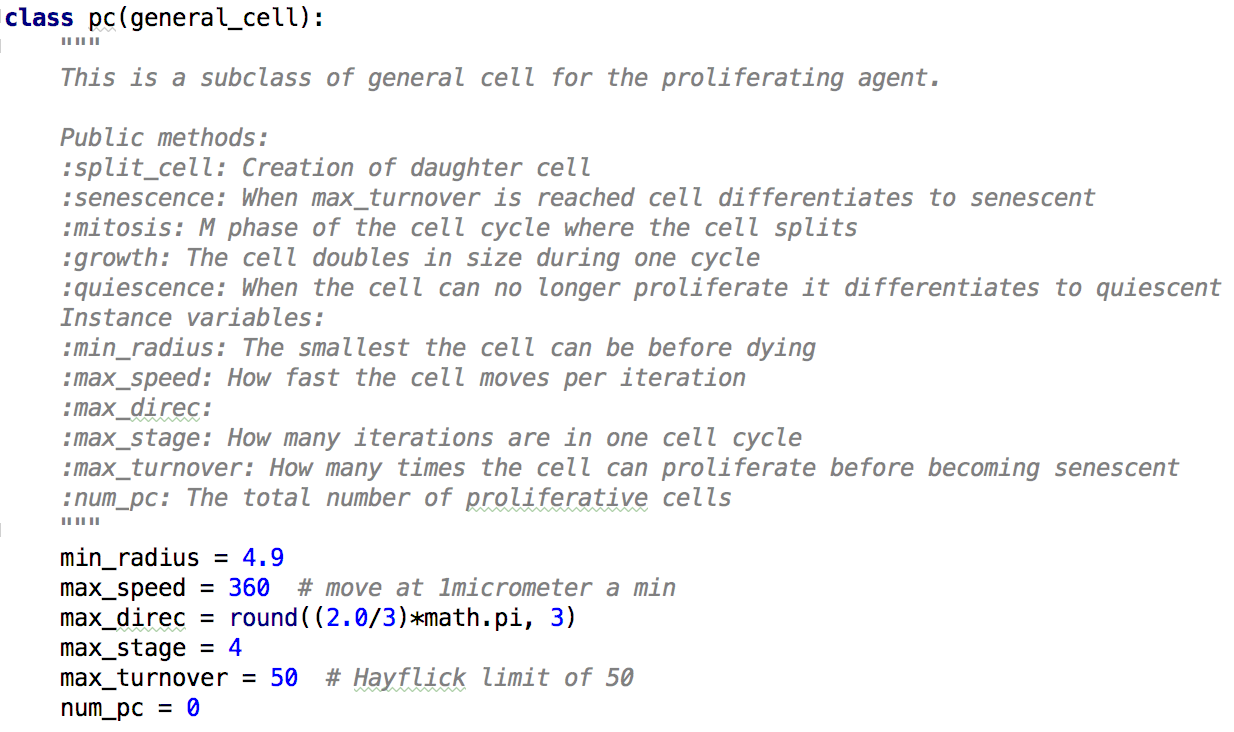
5.1.4.1 Class Overview

The Proliferating Cell (PC) will be the most prevalent agent as it the source agent as seen in Figure 4.2. The PC class is a subclass of the general cell class and extends it by giving the PC specific behaviours. As seen in 2.1 endothelial cells have a radius between 5 and 10μm and so the minimum radius for PCs is set to 4.9. If it was set to 5, there would be an edge case where some newly formed PCs that start out with a radius of 5 will be removed during the apoptosis function.

I have guessed that PCs move at 1 μm per minute, giving them a speed of 360μm for the iteration.

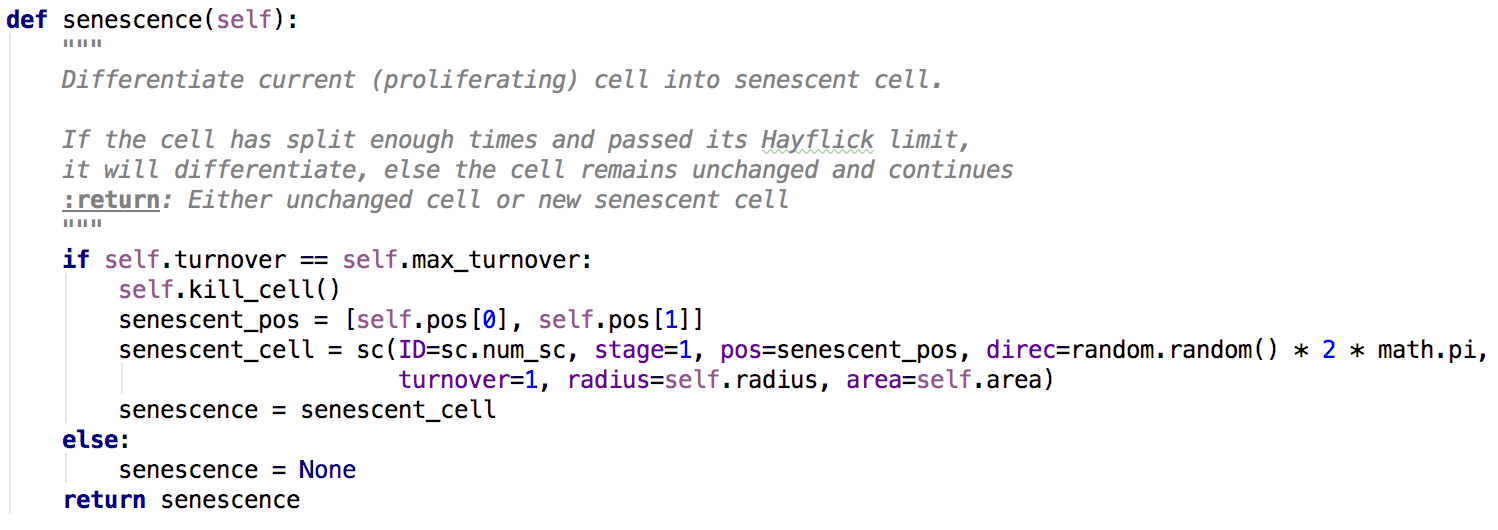
As seen in 2.1 endothelial cells have distinct stages in the cell cycle. This is tracked by assigning a stage to each PC with G1 = 1, S = 2, G2 = 3, and M = 4.

From 2.2 is it seen that each time a cell undergoes mitosis and divided its telomeres shorten, thus after several divisions the telomeres are too short to continue the dividing and the cell turns Senescent, this limit is known as the Hayflick limit and has been shown to be around 50 divisions. Thus the maximum turnover for each PC is set to 50.



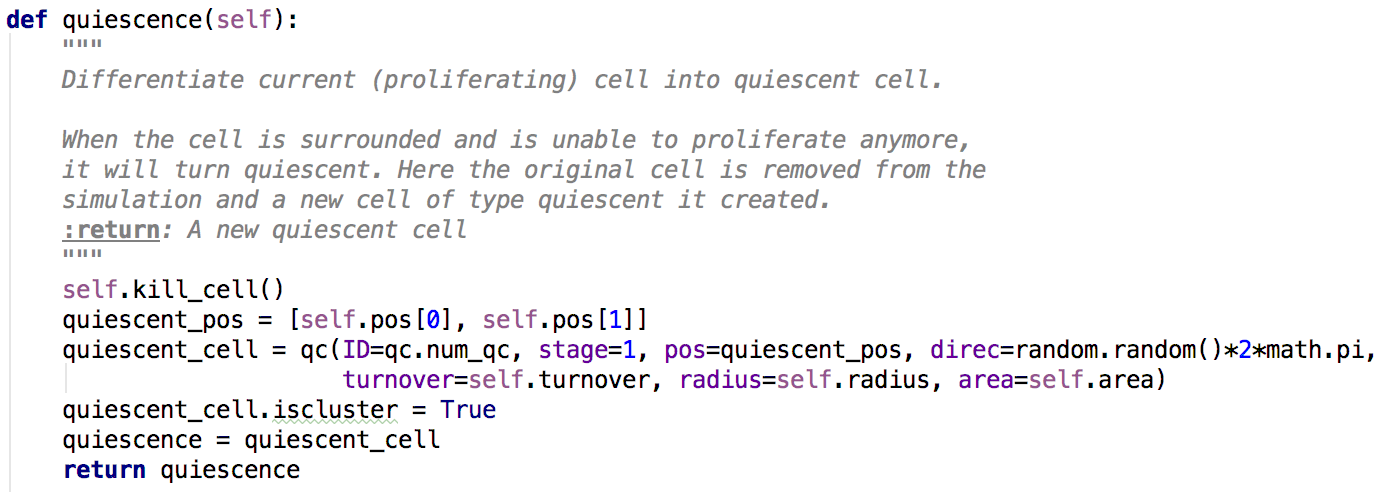
5.1.4.2 Senescence

As mentioned in 2.2 and 5.1.4.1 Proliferating Cells will turn Senescent when they have hit the Hayflick Limit of 50. This differentiation is executed by removing the current PC from the simulation and creating a new SC at the same position and with same radius and area. As the SC agent uses turnover not for counting the number of times it has divided but for how many iterations it has been alive for, the new SC has its turnover set to 1.



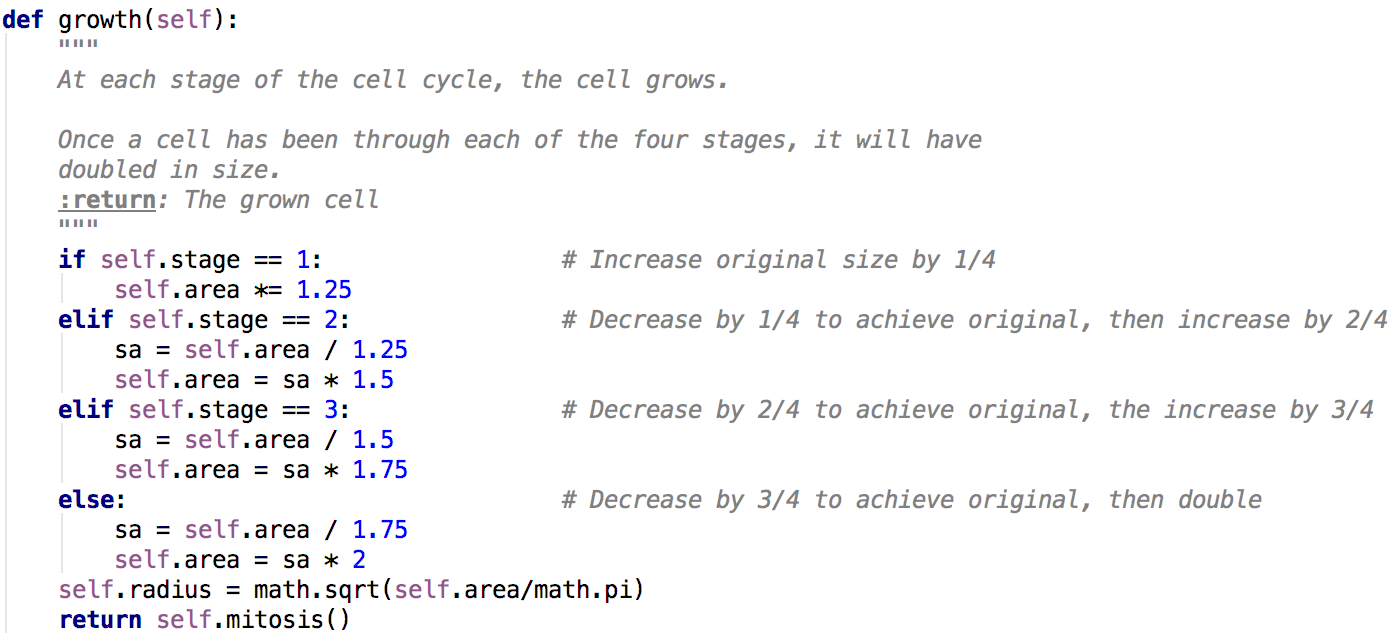
5.1.4.3 Quiescence

As seen in 2.1 Proliferating Cells can enter a special state within the cell cycle known as G0 or the quiescent state. This occurs when the cell is unable to proliferate due to the surrounding pressures from other cells. The detection of number of neighbours is shown in 5.X.X.X and if is past a user defined threshold the cell goes into growth arrest and turns Quiescent. This is achieved by removing the current PC from the simulation and creating a new QC agent in its place with the same: turnover, radius, and area.



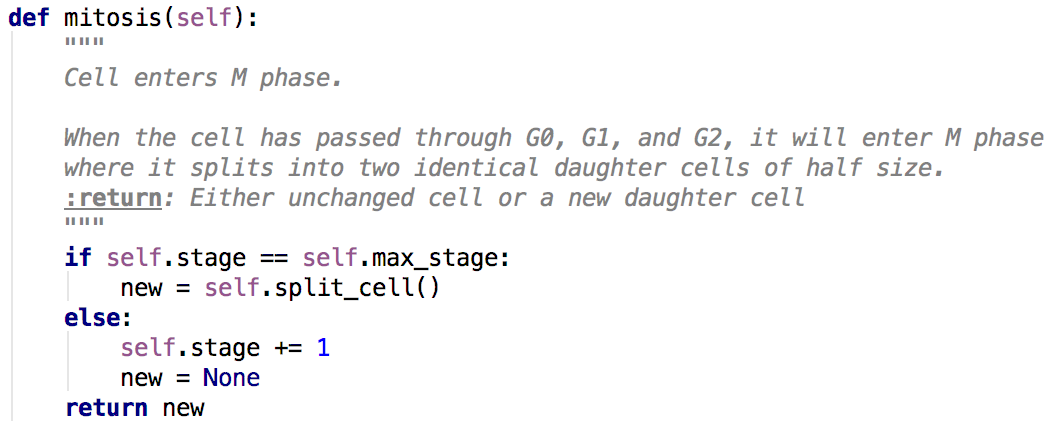
5.1.4.4 Growth

Through one iteration of the cell cycle a proliferative cell doubles in size so it can divide into two equally sized daughter cells during mitosis. Therefore, what could be done is to increase the size of the cell by two times only when it is in stage 4, however this will assume that all growth occurs just before cell division, will make the growth look sporadic, and is an incorrect model of the biology. Therefore, this function has been created to increase the size of the cell by ¼ each stage so that when mitosis comes around it is double the size.



5.1.4.5 Mitosis

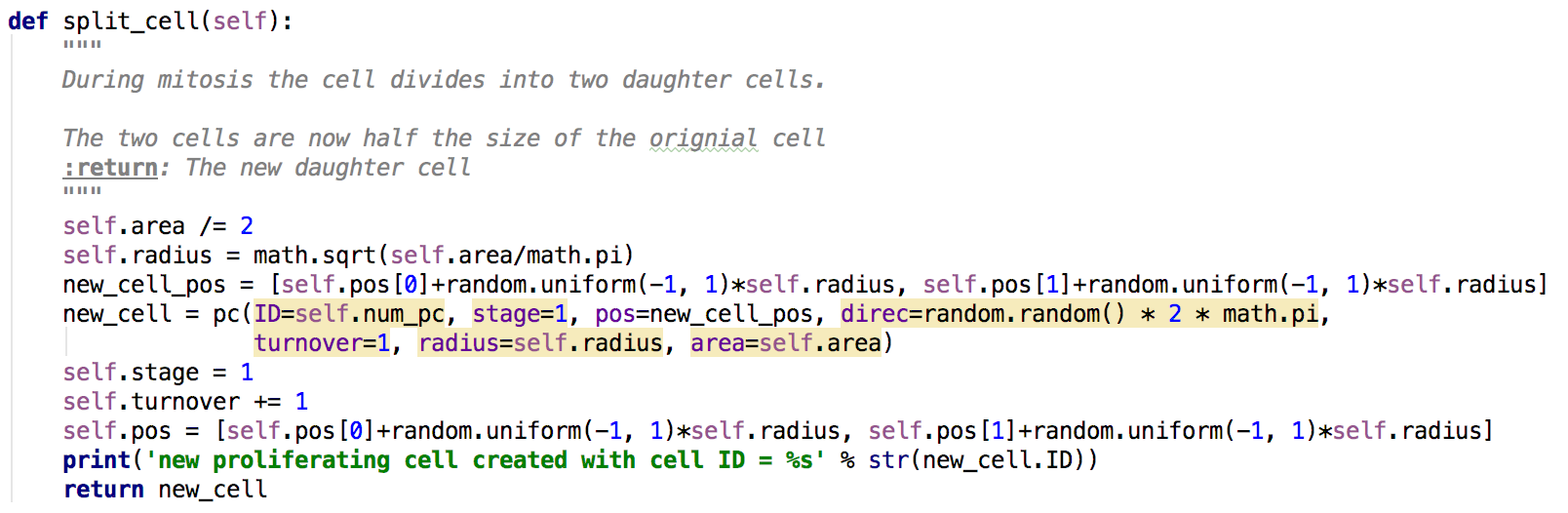
When Proliferative Cells enter M phase of the cell cycle they undergo mitosis. This is where the parent cell replicates and divides into two equally sized daughter cells. This function checks to see if the cell has entered M phased. If true it sends the cell to be split. If false, and the cell is in another stage of the cell cycle the function will increment what stage of the cycle that cell is in by 1. Returning either the two new daughter cells or the original cell further along in the cell cycle.



5.1.4.6 Split Cell

When the cell is undergoing mitosis, it splits into two equally sized daughter cells. This is achieved by reducing the area of the current (parent) cell by two and creating a new proliferative cell next to the current cell with the same area and radius but with a turnover of 1. As the parent cell has divided, its telomeres have shortened and to reflect this the turnover is increased by 1.

After the parent cell has divided it enters G1 phase and this is reflected by setting its stage back to 1.



5.1.5 Agent Solve

A crucial aspect of agent based models is the application of rules (behaviours) each iteration. The agent solve class is used for just that. It is called each iteration from the main CellABM class and it takes the environment, containing the numbers of each type of agent, as its one parameter. The implementation of this class has been adapted from the original to decrease types of environment and has been extended as per Figure 4.5 to include the logic for the new agents.

For each senescent agent, it only checks to see what stage the cell is at. If it has reached its max stage (3 years) the loop will kill the cell.

For each proliferative agent, it starts by testing whether the cell can turn senescent, if false it will test to see if the cell should undergo cellular arrest and enter quiescence, if false it will migrate the agent and then test to see if its smaller than the min radius, killing it if true.

For each quiescent agent, it starts by testing whether the cell can turn senescent, if false it will test to see if the cell can leave cellular arrest and proliferate again. It will then migrate the cell

Each new agent created by agent solve is added to a list of new cells before the next iteration and all agents are check to see if they’re alive, removing them from the environment if false.

5.1.6 Environment

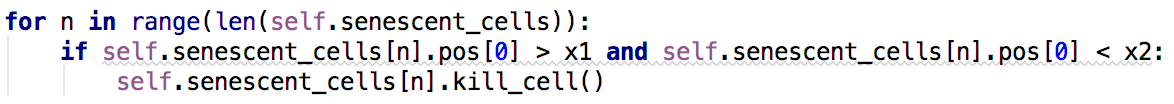
5.1.6.1 Create Agents

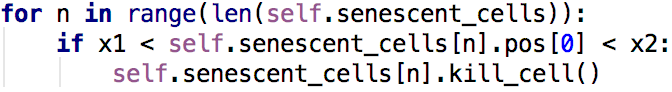
This function has been adapted from the original to include cell stages and to incorporate the new agents. For the user defined number of starting senescent and proliferating cells, the function will create a new cell of that type with stochastic variables within ranges. Quiescent cells have not been implemented in this function as they are an emergent behaviour that occurs when a monolayer has formed.

5.1.6.2 Wound

This function is used when the first confluence is created to form the simulated p20 pipette wound. The wound is created across the whole Y axis but only by a user defined length across the centre of the X axis. The size of the wound can be altered on the command line before the simulation is run. Any cells that are within a x1 and x2 range are removed from the simulation using the .kill\_cell() method.

Special consideration has been given to the creation of chained comparisons with Figure 5.X being the desired implementation over Figure 5.X2.





5.1.7 Overlap Correction

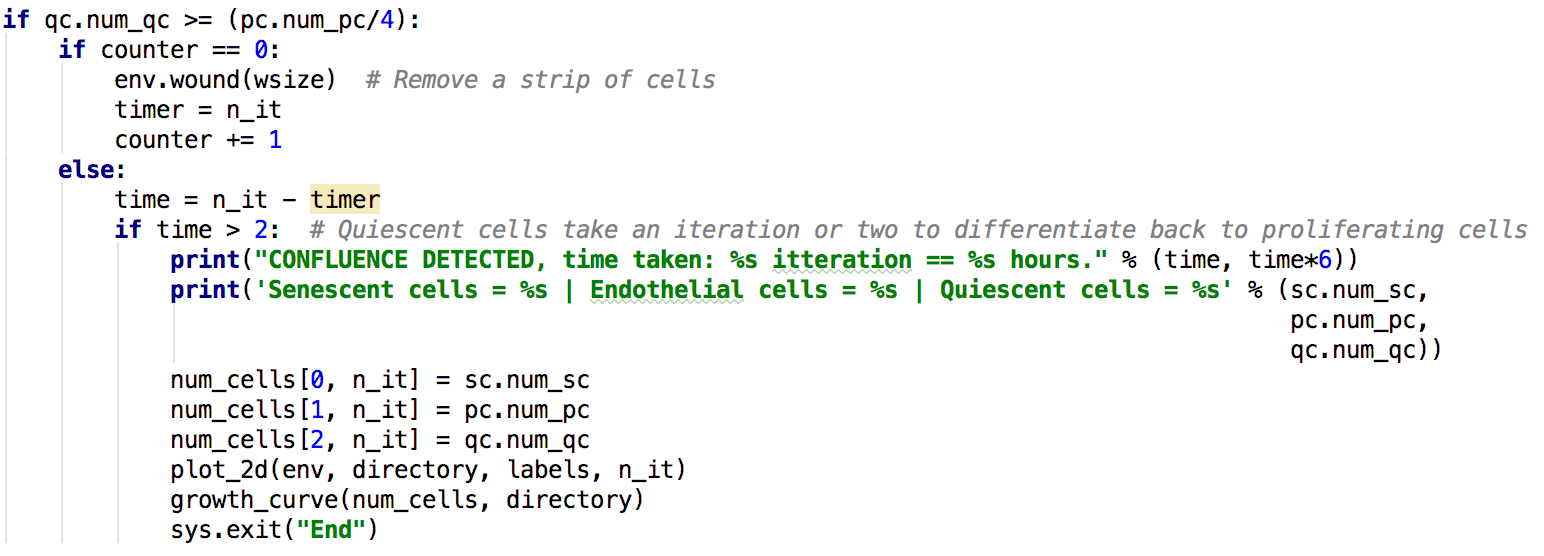
This class has been mainly left as is from the original as it already works off a model of cell correction from literature [Is this true?] It works by using a brute force approach to correct the overlap. First each cell from the environment is added to a list ‘cells’. This list of cells is iterated through in turn with each cell i compared with each other cell j to see if their current position on the environment and the size of each cell causes them to overlap. A list of overlapping cells is then created and passed to the correct overlap function where the cells are assigned new positions to ensure they no longer overlap. However, in moving the cells to a new position can cause them to overlap with another cell and so the process must be repeated until no cells are overlapping. The function update\_radii has been kept from the original code to help prevent the overlap of the cell by slightly adjusting their radius if they’re overlapping.

5.1.8 Confluence Detection

A novel approach has been taken for the detection of a confluence. It works off the emergence behaviour of quiescence which only occurs when the cell density is high enough to prevent proliferation and creation of new cells. When the total number of quiescent cells have passed a threshold, a confluence has formed. Here the threshold has been set to be ¼ the number of proliferative cells as this gave the best results.

After the first confluence, the wound is simulated and it takes an iteration for the quiescent cells to notice the extra space and differentiate back to proliferative cells.

When the second confluence has occurred, the simulation is halted and the total time for the wound to heal is output.



5.1.9 Command Line Interface

The program does not utilise a GUI and therefore all conditions for the simulation must be given at the start on the command line. The conditions that can be changed are: the size of the model, the starting number of senescent cells, the starting number of proliferating cells, the number of iterations to mode, the size of the wound, and the name of the directory to save the output graphs. These conditions are then passed through the program to where they are needed.

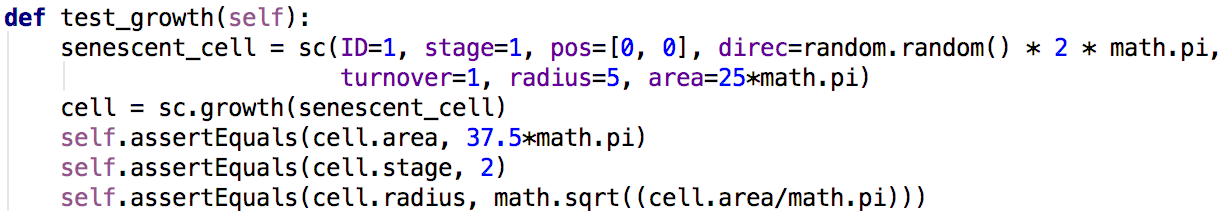
code_images/command_line.png

code_images/command_line_2.png

**5.2 Testing**

5.2.1 Unit Testing

Senescence Growth



**5.2.2 Verification of ABM System**

**5.2.3 User Testing**

**5.2.4 Face Behaviour Testing**

Illustrate “Coding Traps”

* + Nested for loops increase time complexity
* Highlight novel aspects to algorithms
* Done according to scheme within Analysis chapter and follow suitable model
* Functional testing
* User-acceptance testing
* Developed techniques evaluated against standard result set for calibration

1. **Results and Discussion**

All simulations were run on a 2015 15” mac book pro with a 2.8GHz Intel core i7 processor and 16GB 1600MHz DDR3 memory.

* 1. **Simulation Results**
  2. **Program Efficiency and Runtime Analysis**
  3. **User Story Analysis**
  4. **Goals Achieved**
* Degree to which findings support original objectives of project.
  1. **Further Work**
* New areas of investigation
* Statistical validation. Currently not possible as there is not enough data surrounding time taken for epithelial walls to heal when level of Senescence is varied.
* Current part of work not completed
  1. **Completed Software and Documentation**  
     The code developed in chapter 5 can be found on the code repository website GitHub at: https://github.com/HarrisonCooper/dissertation. The documentation on how to run the software can be found in Appendix <>.

**7 Conclusion**

7.1 Conclusion

This survey and analysis started by outlining the biological processes that go on within an EC, allowing us to understand the rules behind the behaviours that cause wound healing to slow down with age. The research then went onto the various methods applicable to this project, finding an agent based approach to be best. A review of suitable software was conducted in table 2.1 showing CellABM and Repast to be equal in applicability. For this reason, during the beginning of semester 2, time will be given to see if the runtime of CellABM can be decreased; if not resorting to Repast is a viable option, although implementation of contact resolution is not desirable due to its complexity.

Several parameters and rules have been sourced to act as guidelines for the development of the program, however due to the uncertainty around the accuracy of this data, a heuristic approach will also be used throughout.

7.2 Progress

So far to date, I’ve managed to convert CellABM from python 2 in to python 3, basic comments to functions with the expectation of full documentation once the full system works and is tested. I’ve run several simulations of Marzihas’ code as shown below, with the area and number of iterations is the same across all tests at 0.1mm2 and 50 respectively. The program was run on a 2015 15” Mac Book Pro with a 2.8 GHz Intel Core i7 and 16GB 1600 MHz DDR3 memory.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Test 1** | **Test 2** | **Test 3** | **Test 4** |
| **Number of Cancer Cells at Start** | 10 | 100 | 0 | 200 |
| **Number of Stem Cells at Start** | 10 | 200 | 500 | 300 |
| **Total Number of Cells at Start** | 20 | 300 | 500 | 500 |
| **Number of Cancer Cells at End** | 18 | 176 | 0 | 368 |
| **Number of Stem Cells at End** | 17 | 304 | 774 | 476 |
| **Total Number of Cells at End** | 35 | 480 | 774 | 844 |
| **Difference in Number of Cells** | 15 | 180 | 274 | 344 |
| **Time Taken (hours)** | 0.017 | 0.502 | 1.308 | 2.332 |

Table 7.1: Time taken to run CellABM with varying starting parameters.

This table shows us a possible drawback to CellABM, which is the computational power required. As shown for Test 4 it took 2.332 hours to model a total of 500 initial cells. Scaling this up to 1mm2 would therefore take a significant amount of time longer to even form a confluence, let alone the wound healing. This is down to the runtime of the overlap.py class, which is used to correct any overlapping of cells caused by mitosis or movement. The nested for loops used to iterate through each cell in turn means the time taken for the program to compute is at least Ο(n2). This is a scalability issue, as when I increase the area of the environment, the overlap class won’t be able to keep up with the increase in number of cells. However, this outcome may not be as adverse as it seems since each cell in CellABM is fixed at 3μm in diameter, whereas the ECs will be around 5-10μm in diameter and the senescent cells will be around 50-100μm. This means that overall, my simulation will have fewer cells than what CellABM models, reducing the scalability risk slightly.

Another downside is that Marzihas code doesn’t implement any cell growth, and each cell is the same diameter as every other cell for the whole simulation. This is a simplification and would not provide a realistic model of ECs.

This program is useful as it automatically outputs a graph showing the growth of each cell type over time, shown below. This can be used in my application to determine the rate of time required for the wound to heal with different starting parameters.

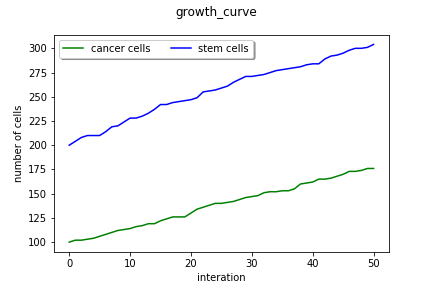
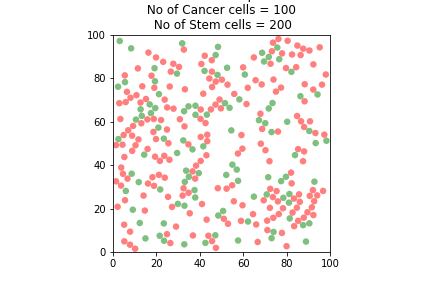
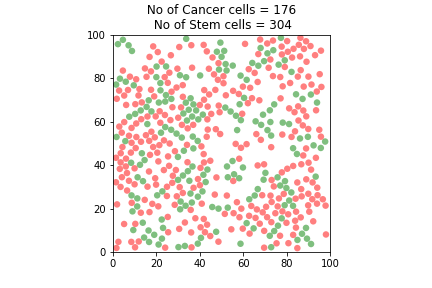


Figure 7.1: Taken from running CellABM with 0.1mm2 area, 100 cancer

cells and 200 stem cells.

CellABM also outputs a 2D and 3D image of environment each iteration, showing the movement of the cells over time, and demonstrate the emergent behaviours of wound healing with age. I can use this to demonstrate my models emergent behaviours such as stopping mitosis and entering quescence once a confluence is formed, visualising the shape and size of the wound, and how the wound is healed.





Figures 7.2 and 7.3: Taken from running CellABM with 0.1mm2 area, 100 cancer cell

and 200 stem cells, showing the initial and final iterations.

I have also been experimenting another software that was mentioned in the literature review, Spark. So far, I’ve managed to get spark running on my machine, gone through the example exercises provided and looked at the documentation [19]. The exercises modelled 10201 cells and their behaviours in real time; the limitation being that the cells are unable to move around, thus decreasing accuracy.

References

[1] Pearson, J. (2000). Normal endothelial cell function. *Lupus*, 9(3), pp.183-188.

[2] Ncbi.nlm.nih.gov. (2017). *Figure 14.1, Phases of the cell cycle - The Cell - NCBI*

*Bookshelf*. [online] Available at:

https://www.ncbi.nlm.nih.gov/books/NBK9876/figure/A2435/?report=objectonly

[Accessed 3 Dec. 2017].

[3] Cooper, G. (2000). *The cell*. Washington, D.C.: ASM Press.

[4] Cell Proliferation. (1991). *Index medicus*, [online] 24(1). Available at:

http://onlinelibrary.wiley.com/journal/10.1111/(ISSN)1365-2184 [Accessed 3 Dec.

2017].

[5] En.wikipedia.org. (2017). *G0 phase*. [online] Available at:

https://en.wikipedia.org/wiki/G0\_phase [Accessed 3 Dec. 2017].

[6] Lab.anhb.uwa.edu.au. (1998). *Blue Histology - more about Endothelial Cells*. [online] Available at: http://www.lab.anhb.uwa.edu.au/mb140/moreabout/endothel.htm [Accessed 27 Nov. 2017].

[7] P.Brandes, R. (2005). Endothelial Aging. *Cardiovascular Research*, [online] 66(2),

pp.286–294. Available at: https://doi.org/10.1016/j.cardiores.2004.12.027 [Accessed 3

Dec. 2017].

[8] Foreman, K. and Tang, J. (2003). Molecular mechanisms of replicative senescence in

endothelial cells. *Experimental Gerontology*, 38(11-12), pp.1251-1257.

[9] Senescence.info. (2017). *Cellular Senescence: The Hayflick Limit and Senescent and*

*Aging Cells*. [online] Available at: http://www.senescence.info/cell\_aging.html [Accessed

3 Dec. 2017].

[10] Warboys, C., de Luca, A., Amini, N., Luong, L., Duckles, H., Hsiao, S., White, A., Biswas, S., Khamis, R., Chong, C., Cheung, W., Sherwin, S., Bennett, M., Gil, J., Mason, J., Haskard, D. and Evans, P. (2014). Disturbed Flow Promotes Endothelial Senescence via a p53-Dependent Pathway. *Arteriosclerosis, Thrombosis, and Vascular Biology*, [online] 34(5), pp.985-995. Available at: http://atvb.ahajournals.org/content/suppl/2014/03/20/ATVBAHA.114.303415.DC1.html [Accessed 26 Nov. 2017].

[11] Chaudhury, H., Zakkar, M., Boyle, J., Cuhlmann, S., van der Heiden, K., Luong, L., Davis, J., Platt, A., Mason, J., Krams, R., Haskard, D., Clark, A. and Evans, P. (2010). c-Jun N-Terminal Kinase Primes Endothelial Cells at Atheroprone Sites for Apoptosis. *Arteriosclerosis, Thrombosis, and Vascular Biology*, [online] 30(3), pp.546-553. Available at: http://atvb.ahajournals.org/cgi/content/full/30/3/546 [Accessed 20 Nov. 2017].

[12] Gerrity, R., Richardson, M., Somer, J., Bell, F. and Schwartz, C. (1977). Endothelial cell morphology in areas of in vivo Evans blue uptake in aorta of young pigs. *Am J Path*, (89), pp.313-335.

[13] Hansson, G., Chao, S., Schwartz, S. and Reidy, M. (1985). Aortic endothelial cell death and replication in normal and lipopolysaccharide-treated rats. *Am J Pathol*, (121), pp.123-127.

[14] Hu, Y., Foteinos, G., Xiao, Q. and Xu, Q. (2008). RAPID ENDOTHELIAL TURNOVER IN ATHEROSCLEROSIS-PRONE AREAS COINCIDES WITH STEM CELL REPAIR IN APOE-DEFICIENT MICE. *Atherosclerosis*, 199(2), p.467.

[15] Pavelka, J., Tel, G. and Bartosek, M. (2000). *SOFSEM'99 - Theory and Practice of*

*Informatics*. New York: Springer.

[16] Walker, D., Hill, G., Wood, S., Smallwood, R. and Southgate, J. (2004). Agent-Based

Computational Modeling of Wounded Epithelial Cell Monolayers. *IEEE Transactions*

*on Nanobioscience*, 3(3), pp.153-163.

[17] Docs.python.org. (2017). *1. Extending Python with C or C++ — Python 3.6.3*

*documentation*. [online] Available at:

https://docs.python.org/3/extending/extending.html [Accessed 3 Dec. 2017].

[18] Salk, D., Bryant, E., Au, K., Hoehn, H. and Martin, G. (1981). Systematic growth

studies, cocultivation, and cell hybridization studies of Werner syndrome cultured skin

fibroblasts. *Human Genetics*, 58(3), pp.310-316.

[19] Pitt.edu. (2017). *SPARK - Simple Platform for Agent-based Representation of Knowledge*.

[online] Available at: http://www.pitt.edu/~cirm/spark/documentation.html [Accessed 4 Dec.

2017].