OCR Computer Science H446

NEA Submission May 2023

Ace Harvey

Centre No 20153, Candidate No 4

## Project Overview

My project is a piece of software designed to visualise complicated mathematical systems, such as the Lorenz attractor, and other similarly complex “chaotic” systems.

* . The project should visualise these by rendering high-resolution high-fidelity images which trace a point in the system as time progresses.
* The project could also produce videos of a system as it evolves over time, rather than a single image.
* The project should prioritise quality of image, as well as efficiency.
* The project should also be highly configurable, such as changing scale, colours and rendering algorithms easily.
* Users should be able to enter their own custom governing equations for systems
* Users should be able to save and load configuration files so that they can easily recreate images with slight changes if needed.
* The project should also have a graphical user interface, and could include a small preview window.

Contents

[Project Overview 1](#_Toc114040704)

[Analysis – Computational Methods 3](#_Toc114040705)

[Analysis - Stakeholders 4](#_Toc114040706)

[Analysis – Existing Solutions 5](#_Toc114040707)

[Analysis - Features 6](#_Toc114040708)

[Analysis - Limitations 7](#_Toc114040709)

[Analysis - Requirements 8](#_Toc114040710)

[Analysis – Success Criteria 9](#_Toc114040711)

[Design - Decomposition 10](#_Toc114040712)

[Design – Structure 11](#_Toc114040713)

[Design – Algorithms 12](#_Toc114040714)

[Design – Features 13](#_Toc114040715)

[Design – Key Jawns 14](#_Toc114040716)

[Design – Testing Data 15](#_Toc114040717)

[Design – Further Data 16](#_Toc114040718)

# Analysis – Computational Methods

Abstraction

The project should take a config file as an input, or take input from a UI to change said config file, then output a picture or video in the format selected by the user. However, processing of data into images is a complicated problem, and one that has been solved by many before me. As such, for some elements of output processing (mostly pertaining to encoding raw pixels into filetypes) I will be using existing libraries such as PIL (Python Image Library) and openCV for video encoding. This usage of existing libraries is abstraction, a computational method.

Furthermore, I could use abstraction to hide away the complicated workings of the project, only showing the user the parameters that affect the look of the final image rather than all of the parameters for image rendering which, when adjusted, could cause unexpected results.

Decomposition

The project decomposes into various tasks easily, such as dealing with the mathematical expressions, the image processing and rendering, etc. By using decomposition in this way, I can break down the problem into less complicated problems and solve those individually, which would be easier than tackling the whole problem at once.

Logical nature

The project requires the use of logic in many different aspects – such as validating mathematical expressions, evaluating said mathematical expressions and validating configuration files.

Procedural nature

The project is also highly suited to procedural programming techniques, as the program can be decomposed into smaller programs and then recombined at the end. Doing so would also allow me to reuse different parts of the project in other parts of the project, or potentially in future projects. Building smaller subprograms and combining them allows for a smoother workflow and also allows me to isolate problems or bottlenecks easily.

Input and output.

Presenting output is a problem that leads itself to the use of computational methods as the simplest way to present an output diagram is through an image or video on a computer screen – attempting to do otherwise would be complicated, requiring some form of physical image creation which could be expensive due to the need for physical materials and would be inconvenient if the user made a mistake in input parameters and had to re-create the image again, costing double. As such, I believe a computational approach is best to present the output.

Furthermore, getting input to the project would be difficult without a computational approach as I would like to present the parameters in an easy to use and understand way. This leads itself to using a computational approach as I could easily create a graphical or command line interface that labels and explains various parameters and allows the easy entry of numbers or equations through use of a mouse and/or keyboard.

Saving configurations and outputs

In addition, the project should be able to save and load inputs via the use of configuration files, which is easy to implement via computational methods such as file handling (such as pythons `open` syntax), but would be difficult to do without a computer. Using a computer avoids the need to either remember the desired configuration, or have a physical way of storing or inputting parameters.

Also, the project should be able to save its outputs, which again would be easy to do via file handling on a computer. Other approaches could be inconvenient for the user, for example, saving images using a computer would allow the user to then analyse or process them further using other computer software, in a way that would be difficult if not impossible should a computational approach not be used for saving outputs.

Why is this project suited to a computational approach?

As I have detailed, there are many reasons I believe that make this problem solvable via use of a computational approach; input, output and processing would all be challenging to implement without use of computational methods, and as such I believe that the whole project should be done via a computational approach. Furthermore, the usage of libraries to process raw image and video data would not be possible without the use of a computer, and as such the project is highly suited to a computational approach.

# Analysis – Stakeholders

# Analysis – Existing Solutions

# Analysis - Features

# Analysis - Limitations

# Analysis - Requirements

# Analysis – Success Criteria

# Design - Decomposition

# Design – Structure

# Design – Algorithms

# Design – Features

# Design – Key Jawns

# Design – Testing Data

# Design – Further Data