

# Computer Science NEA 2025

Nathan Tatkowski



INVICTA GRAMMAR  
SCHOOL

Invicta Grammar School

Centre Number:

Candidate Number:

## Contents

<b>1</b>	<b>Analysis</b>	<b>3</b>
1.1	Problem Identification . . . . .	3
1.2	Identification of why this problem is solvable by computational methods . . . . .	3
1.3	Description of the Current System . . . . .	3
1.4	Identification of Stakeholders . . . . .	3
1.5	Identification of User Needs and Acceptable Limitations . . . . .	4
1.6	Existing Solutions . . . . .	4
1.6.1	MATLAB . . . . .	4
1.6.2	Analytic Methods . . . . .	4
1.6.3	Kerbal Space Program . . . . .	4
1.7	Success Criteria of the Proposed System . . . . .	5
1.8	Data Source(s) . . . . .	5
1.9	Volumetrics - Data Volumes . . . . .	5
1.10	Analysis Data Dictionary . . . . .	5
1.11	Data Flow Diagrams for Existing and Proposed System . . . . .	5
1.12	Justification of Chosen Solution . . . . .	5
1.13	Hardware and Software . . . . .	5
1.14	Entity-Relationship Models . . . . .	5
1.15	Identification of Objects and Object Analysis Diagrams . . . . .	5
<b>2</b>	<b>Design</b>	<b>5</b>
2.1	Overall System Design . . . . .	5
2.2	Description of Modular Structure of System . . . . .	5
2.3	Definition of Data Requirements . . . . .	5
2.4	Identification of Appropriate Storage Media . . . . .	5
2.5	Identification of Processes and Suitable Algorithms for Data Transformation and Completion of the Solution . . . . .	5
2.6	Sample of Algorithms . . . . .	5
2.7	User Interface Design Rationale and Usability Features . . . . .	5
2.8	Security and Integrity of Data . . . . .	5
2.9	System Security (Access Control) . . . . .	5
2.10	Overall Test Strategy . . . . .	5
<b>3</b>	<b>Implementation</b>	<b>5</b>
3.1	Annotated Listing of the Program(s) . . . . .	5
3.2	Annotated "Design Views" showing details of application-generated forms, reports, queries, buttons, cross tabulations, etc. . . . .	5
3.3	Procedure and Variable List . . . . .	5
3.4	Testing to inform development - Testing at each stage . . . . .	5
3.5	Re-Testing . . . . .	5
<b>4</b>	<b>Testing</b>	<b>5</b>
4.1	Test Plan . . . . .	5
4.2	Test Data . . . . .	5
4.3	Areas to Test . . . . .	5
4.4	Tables . . . . .	5
4.5	Justification of Data Selection . . . . .	5
4.6	Evidence of Testing . . . . .	5
4.7	Program Listing . . . . .	5

<b>5</b>	<b>Evaluation</b>	<b>5</b>
5.1	Comparison of Performance of your System against Success Criteria . . . . .	5
5.2	Analysis of User Feedback . . . . .	5
5.3	Evaluation of Usability Features . . . . .	5
5.4	Maintenance Issues and Limitations of the Product . . . . .	5
5.5	Improvements and Possible Extensions . . . . .	5

# 1 Analysis

## 1.1 Problem Identification

Often times during research it is important and very helpful to be able to visualise the events that are being analysed. When working with models in two dimensions, it is easy enough to be able to draw out an accurate diagram, even if it may be tedious. However, 2D models are nowhere near as applicable or useful as considering events in three dimensions, stemming from the fact that the world we live in is three-dimensional. Something that would aid intuition and help in problem-solving would be a way to have events modelled quickly, accurately, and clearly, given a set of initial conditions.

## 1.2 Identification of why this problem is solvable by computational methods

The key requirements stated above (accuracy, haste, and clarity) lend themselves very well to using computational methods. Computers are able to make calculations orders of magnitude faster than by hand or by analogue machine, and to a virtually arbitrary degree of accuracy. Many modern central processing units (CPUs) are also able capable of making use of concurrent processing, further increasing the advantage that a computer would have over a human. Graphical processing units (GPUs) are specifically designed for parallel processing, making them especially useful for graphics, which would allow for high quality renders for the user to be able to see. Any data that you would need to consider can be displayed in a clear and user-friendly fashion, making it highly customisable to fit the individual persons needs and for many attributes to be studied at the same time.

## 1.3 Description of the Current System

Without using computer simulations, the usual process is to produce a handful of equations by hand that would model the attributes of an object, commonly the path it takes in three-dimensional space. This has the benefit of giving exact values and equations that are very useful when trying to understand the underlying reasons for an event happening. For example when considering a pendulum, it is clear from the equation

$$T = 2\pi\sqrt{\frac{L}{g}}$$

that the period of the pendulum  $T$  does not depend on the mass of the object doing the swinging. However, when running computer simulations, such relationships may not be as obvious, and as computers aren't able to analytically solve problems (i.e. through the use of rigorous mathematics), this is a drawback that I will have to consider.

## 1.4 Identification of Stakeholders

After considering the problem I identified the following groups that could use a solution to this problem, as well as having useful insight on how a program like this should function.

- **University Students** often have to deal with complex systems and a way of visualising them would be very beneficial. I have been able to contact a student at the University of Aberdeen doing a masters in electrical and mechanical engineering. Their name is Hugo, and they are 21-years-old.
- **A-Level Students**, specifically students taking physics, would be able to greatly further their understanding of core concepts and be able to explore new ideas on their own. I have been able to communicate with Daniel, a year 12 physics student, about being a stakeholder for this project.

- **Teachers** of A-Level and below could make great use of simulation software in order to make learning much easier with models and demonstrations that are clear and easy to understand. I have been able to contact [person here] about being a stakeholder for this project, who teaches physics at Invicta Grammar School.

## 1.5 Identification of User Needs and Acceptable Limitations

- **Stuff goes here**

## 1.6 Existing Solutions

### 1.6.1 MATLAB

MATLAB here

### 1.6.2 Analytic Methods

Analytic methods here

### 1.6.3 Kerbal Space Program

Kerbal Space Program here

## 1.7 Success Criteria of the Proposed System

## 1.8 Data Source(s)

## 1.9 Volumetrics - Data Volumes

## 1.10 Analysis Data Dictionary

## 1.11 Data Flow Diagrams for Existing and Proposed System

## 1.12 Justification of Chosen Solution

## 1.13 Hardware and Software

## 1.14 Entity-Relationship Models

## 1.15 Identification of Objects and Object Analysis Diagrams

# 2 Design

## 2.1 Overall System Design

## 2.2 Description of Modular Structure of System

## 2.3 Definition of Data Requirements

## 2.4 Identification of Appropriate Storage Media

## 2.5 Identification of Processes and Suitable Algorithms for Data Transformation and Completion of the Solution

## 2.6 Sample of Algorithms

## 2.7 User Interface Design Rationale and Usability Features

## 2.8 Security and Integrity of Data

## 2.9 System Security (Access Control)

## 2.10 Overall Test Strategy

# 3 Implementation

## 3.1 Annotated Listing of the Program(s)

## 3.2 Annotated "Design Views" showing details of application-generated forms, reports, queries, buttons, cross tabulations, etc.

## 3.3 Procedure and Variable List

## 3.4 Testing to inform development - Testing at each stage

## 3.5 Re-Testing

# 4 Testing

## 4.1 Test Plan

## 4.2 Test Data

## 4.3 Areas to Test

## 4.4 Tables

## 4.5 Justification of Data Selection

## 4.6 Evidence of Testing

## 4.7 Program Listing