**Team 4**

**Test Plan**

# 1. Introduction

Team 4 will be conducting a comprehensive review of the Plotter Software (Math graphics 2.0.1). The goal of the review is to identify any remaining bugs in the software and fix them. While identifying bugs in the software we will also be making recommendations for improvements and making changes that retain the original functionality but improve the interface and usability of the software.

**Test Plan Objectives**

1. Identify remaining bugs in the software.
2. Report bugs found.
3. Make changes to software to fix bugs.
4. Identify potential improvements in the software.
5. Make changes to the software that improves usability.
6. Report all changes made.

# 2. Scope

1. **Graphical Interface**

The graphical interface will be tested for performance and functionality. The interface will be reviewed and areas for improvement will be identified. Improvements will be made to the GUI that enhance the user experience while still maintaining the functionality of the software.

1. **Renderer**

The code dedicated to rendering the graphs will be reviewed for bugs and potential areas of improvement. The three different rendering methods, Cartesian 2D, Polar 2D and Cartesian 3D will all be tested for bugs and performance capabilities.

1. **Math Calculator**

The math code will be checked for bugs and potential areas of improvement. The Integral button functionality and the Show DF button will be tested. The displayed function field will be tested under varying data sets.

# 3. Test Strategy

The test strategy will be implemented in seven tests. These tests will test the software limitations and capabilities. The testing will identify bugs and areas where improvements can be made. The final stage of the testing process will be to present the changes made to the clients and compare the finished software with the original requirements document.

1. **System Test**

System testing will focus on the behavior of the Software when used under normal usage conditions. Tests will consist of known errors to test error handling as well as user scenarios designed to check how the different components work together.

1. **Performance Test**

Performance testing will measure the systems response time under varying conditions. Apache JMeter will be used to test that all aspects of the software are functioning within the parameters defined by the client.

1. **Automated Test**

Unit tests will be written using TestNG software. These tests will be used through the testing process to verify when changes are made to the software that previously tested code still functions normally.

1. **Stress and Volume Test**

The math code will be tested with numbers that push the boundaries of the software. All math rules will be tested to show correct error messages are thrown when math rules are broken.

1. **Recovery Test**

This testing will break the software in various ways so recovery conditions and be tested and evaluated.

1. **Beta Test**

The software will be made available to real life users to test in daily use scenarios. This will help identify issue that may not have been identified by traditional testing methods.

1. **User Acceptance Test**

Once all testing has been completed and all of the identified bugs have been fixed testing will take place with the clients. The original requirements document will be used to confirm the software meets the client’s expectations.