# Homework 1

Suraj Jaiswal

February 8, 2019

**Abstract**

This document represents a brief report about the solutions for the 1st assignment of the course "Computational Methods in Mechanics". There are three exercise tasks from the book "Programming for Computations - MATLAB/Octave" that needs to be solved. The first task is to find the crossing points between the two graphs. The second task is about linear interpolation.

## 1 Introduction

There are three exercise tasks (exercise-2.9, 2.15, and 2.18) from the book "Programming for Computations - MATLAB/Octave" that needs to be solved. The first task is about finding the crossing points between the functions $f(x) = x$ and $g(x) = x^2$ on the interval $[-4, 4]$. The aim in the first task is to find the approximate values of $x$, by trail and error, where the two graphs cross each other. The second task is about linear interpolation where the measured values of $y_i$ are collected regularly, once every minute, at times $t_i = i$, where $i = 0, 1, ..., N$. The aim in the second task is to find the linearly interpolated value of $y$ at a particular time, $t$, provided by the user.

## 2 Method

For the first task, the interval $[-4, 4]$ is divided into $N$ equal sections. Functions, $f(x) = x$ and $g(x) = x^2$ are evaluated over this interval and stored in an array. For the interval provided, $|f(x) - g(x)| < error$ is evaluated, and the values of $x$ for which this condition holds true are the desired values for the first task.

For the second task, the measured values of $y_i$ are collected regularly, once every minute, at times $t_i = i$. A *userInput.m* function is created to input user value of time at which the interpolated value needs to be calculated. For negative value of time, the function is terminated. Another function, *linear_interpolation_function.m* is created that first finds the left and right index values ($i$ and $i + 1$) for the user input value of time. Using this index, the values $y(i)$ and $y(i + 1)$ are are calculated. Accordingly, the linear interpolation is carried out using the following equation:

$$y = y(i) + (t - t(i))\frac{y(i + 1) - y(i)}{t(i + 1) - t(i)} \tag{1}$$

where, $t$ is the user input value of time and $y$ is the desired interpolated value.

## 3 Results and Discussion

For the first task, it was observed that for 400 equal sections and $error = 0.01$, the two graphs intersect as two points 0.01 and 0.99 as shown in Figure 1. If one increases or decreases the value of $N$, by keeping the $error = 0.01$ as it is, the Matlab code returns unexpected outcomes. If $N = 100$, then the Matlab code does

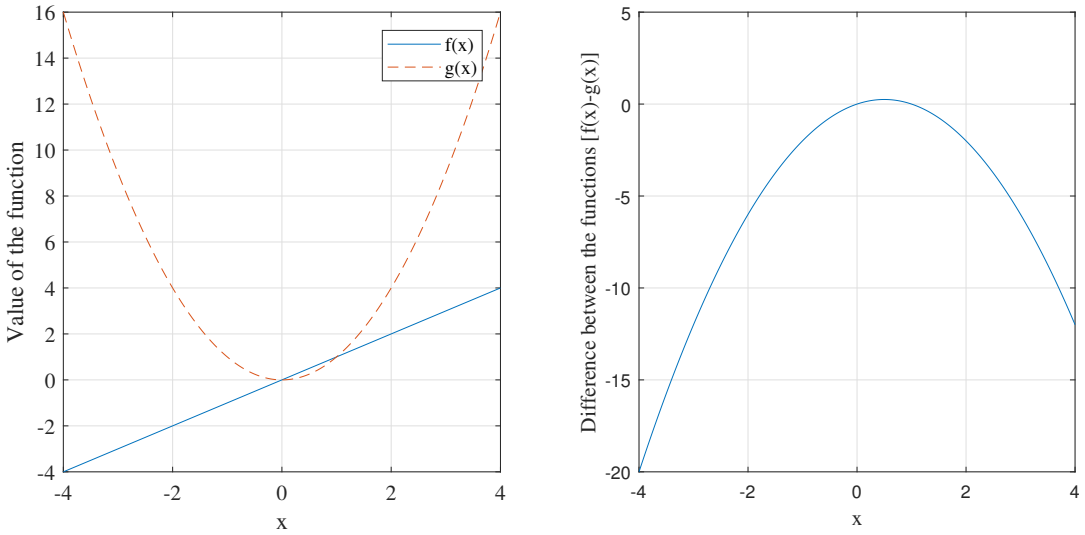


Figure 1: Crossing points of functions $f(x) = x$ and $g(x) = x^2$ (left), and the difference between them (right)

not show any interaction points, and when $N = 1000$, the Matlab code results into four intersection points, -0.00400, 0.00400, 0.99700, and 1.00501. The reason for this is because Matlab provides an approximated value of $x$ and whose evaluation depends on the error value.

For the second task, $y_i = [4.4, 2.0, 11.0, 21.5, 7.5]$ and $t_i = [0, 1, 2, 3, 4]$ min. The interpolated value at $t = 2.5$ min is 16.25 and at $t = 3.1$ min is 20.10.

# 4 Conclusions

The author was able to successfully implement the Matlab code and carry out the desired solution for the first two exercises.