

# Project 1

Micheal Saybolt

Bjon Charlery

March 2, 2017

## Abstract

This project explored solving differential equations, specifically linear second order equations, by using variety of different methods including LU Decomposition, Gauss Forward and Backward Substitution : General algorithm, and Gaussian Substitution : Special Case.

## 1 Introduction

The objective of this project was to solve a differential equation in the form of: (fill in DiffEq here) that can be written into the linear equation ( $\mathbf{Ax} = \mathbf{b}$ ). Since A is a tridiagonal matrix there is an actual or analytical solution that can be made for this particular matrix. The following sections will describe the origins for the algorithms used and their respective results.

## 2 Description

This section will detail the methods used to solve this linear equation system.(Insert equation). It is solved from three methods:

- . LU decomposition
- . Tridiagonal Solver: Gauss Elimination
- . Custom Gauss Elimination

### 2.1 LU Decomposition Solver

This method consists of creating a Lower-Triangular matrix  $\mathbf{L}$  and a Upper-Triangular matrix  $\mathbf{U}$  for factorizing the linear system ( $\mathbf{A} = \mathbf{LU}$ ).

$$\begin{bmatrix} \mathbf{alpha} & \mathbf{beta} \\ \mathbf{gamma} & \mathbf{delta} \end{bmatrix}$$

Once (L) and (U) are found they can be used to solve the following equation: ( $\mathbf{Av} = \mathbf{LUv} = \mathbf{h^2f}$ ) by allowing ( $\mathbf{Ly} = \mathbf{h^2f}$ ) and ( $\mathbf{Uv} = \mathbf{y}$ ) with backwards substitution, in this case that would be required two times.

## 2.2 Tridiagonal Solver: General Gauss Elimination

This method consists of using Gauss elimination due to the matrix (A) only having values along the three diagonals of the matrix.

$$\begin{bmatrix} \textit{alpha} & \textit{beta} \\ \textit{gamma} & \textit{delta} \end{bmatrix}$$

The diagonal of the matrix will be changed using the following formula while the "Upper Diagonal" will not be changed:

$$d_i = b_i - a_i c_{i-1} / d_{i-1}$$

The "Lower Diagonal" will be changed so that all elements below the diagonal are zeros. The vector (f) changes with the following formula:

$$w_i = f_i - a_i w_{i-1} / d_{i-1}$$

With backwards substitution then can be performed after following:

$$v_i = w_{i-1} - c_{i-1} v_i / d_{i-1}$$

$$\text{with } d_1 = b_1, w_1 = f_1, \text{ and } v_n = h^2 w_n / d_n$$

## 2.3 Special Case Solver: Custom Gauss Elimination

Since the matrix we are using has only -1 along the "Upper diagonal" and "Lower Diagonal", -2 along the "Main Diagonal"

(insert a matrix to show)

the value of  $d_i$  can be simplified to  $2 - 1/d_{i-1}$ , the value of  $w_i$  is now simplified to  $f_i + w_{i-1} / d_{i-1}$ , and  $v_i = w_{i-1} + v_i / d_{i-1}$

These calculation require an initial values for  $d_i, w_i$ , and  $v_i$  which are 2,  $f_1$ , and  $h^2 w_n / d_n$ . Even further simplification shows that every value of  $d_i$  is equal to  $(i+1)/i$ , where  $i$  is the same index used for  $d_i$ .

## 3 Results

## 4 Conclusions

From examining all of the methods used to solve the original linear equation we have concluded that the Special Case Solver is the most efficient at finding our solution, followed by the Tridiagonal Solver which is just a generalized case of the previous method. Lastly the least efficient method used to solve the linear equation was the LU Decomposition Solver.