An Impact of the Small and Large Grid Sizes on Differential Equations

Lubna Naz[1,*], Fahim Raees[2], A.H. Sheikh[3], Asif Ali[4]

[1]Department of Natural Sciences, BNB Women Univesity, Sukkur
[2]Mathematics, Faculty of Information Sciences and Humanities, NED UET Karachi
[3]Department of Mathematics & Statistics, CCSIS , Institute of Business Management, Karachi
[4]Department of Basic Sciences & Related Studies, MUET Jamshoro.
[*]Corresponding author: agshaikh@quest.edu.pk

Abstract

The finite difference technique is oldest numerical method to solve differential equations. Like many differential equations, Helmholtz differential equation which is used to describe many physical phenomena, has long been solved using finite difference method. can be described by Helmholtz Differential equations. The solution of the Helmholtz type differential equations is very important. The information that it belongs together because it tells one coherent story just knowing a little bit about finite differences through to how to solve differential equations an especial technique is used, how to implement finite difference method and the tool which is used as generic enough that will immediately be given a whole new differential equation. The analysis of small to moderate sized presented with the help of a few examples. The improved finite difference method is presented with examples, the method is simple, clear, and short the MatLab code is available, the improved finite difference method is suitable and easy to implement, manually as well as computationally.

Keywords—Finite Difference, Differential equations, large grids.

1 Introduction

In The finite difference approximation method [1], [2], changes differential equations, whether an ordinary differential or partial differential equation into, a linear system of equations, and it does not give a symbolic solution. Many problems related to steady state oscillations (mechanical, acoustical, and thermal), and wave scattering [3] is modelled by the Helmholtz equation [4], [5], [6], [7]. Wave scattering has many applications in physics, engineering, and science. The basic philosophy of the finite difference, what would do is in the governing differential equation, we manipulate the governing differential equation directly. In the finite difference method, we replace the differential notations with some algebraic expressions. The finite difference method is a completely numerical method for solving differential equations and working with discrete functions. Finite differences through to how to solve differential equations, using the special technique will be shown to implement the finite difference method.

ISSN: 2523-0379 (Online), ISSN: 1605-8607 (Print)
DOI: https://doi.org/10.52584/QRJ.1902.09

The most important aspect of this paper is that we will be able to derive a finite difference approximation, and know to know how discretization of the differential equation gives the sparse types matrices this makes beginners and researchers understand the importance of how, and why we change the governing differential equation into a system of linear equations, or sparse matrices. When the talking about finite difference approximations that do imply that we don't have a function a continuous smooth function stored in memory, really only know the function at discrete points, here point f_1, f_2, and f_3, we don't store the function between those points and in fact, we don't know what the function is between those points. Some interpolation to make some good guesses but the point is we don't know what the function is at in-between points. Suppose we have this discrete function, but we want to know the slope at some point along this function in this case let's say want to know the slope at f_2. How we do that if only know the function at discrete points, in this case, will recognize that the derivative is a slope, so we can estimate the at position f_2, by connecting point f_1, and f_3, with a line and we, will say that the slope of that line will approximate the slope the true slope at f_2, and the reason we do that is that our points f_1, and f_3 are symmetrically surrounding the point f_2, and the slope connecting points f_1 and f_3, which is rise over run how much is our rise it is $f_3 - f_1$ and our run that's the span between x_1, and x_3, so it's just $2\Delta x$, where Δx, is the increment just from x_1, and x_2.

$$f'(x_2) = \frac{rise}{run} = \frac{f_3 - f_1}{2\Delta x} \tag{1}$$

Similarly the derivatives $f'(x)$ for at remaining different points x_1, x_3, x_4 and x_5 will be

$$f'(x_1) = \frac{rise}{run} = \frac{f_2 - f_0}{2\Delta x}$$
$$f'(x_3) = \frac{rise}{run} = \frac{f_4 - f_2}{2\Delta x}$$
$$f'(x_4) = \frac{rise}{run} = \frac{f_5 - f_3}{2\Delta x}$$
$$f'(x_5) = \frac{rise}{run} = \frac{f_6 - f_4}{2\Delta x}$$

This technique is difficult when considering several points, which is usual practice to get accurate solution of the differential equations in that case above method will be cumbersome, to be incorporated, we need to change strategy;

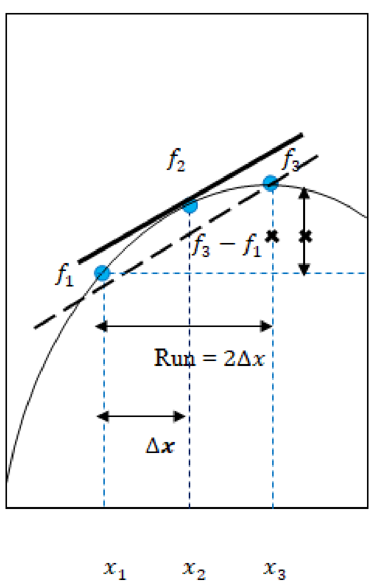

Fig. 1: Centre Difference Technique Presented at a point f_2

2 First Order Derivatives

Using Taylor's theorem, the second order Central Difference Approximations (CDA) to first and second derivatives at the interior mesh point (i,j) are:

$$(\frac{\partial f}{\partial x})_{i,j} = \frac{f_{i+1,j} - f_{i-1,j}}{2h} + O(h^2)$$
$$(\frac{\partial f}{\partial y})_{i,j} = \frac{f_{i,j+1} - f_{i,j-1}}{2k} + O(k^2)$$

Construct a square matrix so that it premultiplies a vector, we get a vector containing the first order partial derivative. We would like to figure out what the square matrix looks when premultiply F, get another column vector that if we multiply would see the derivative of F that is, in this case, the first-order derivative.

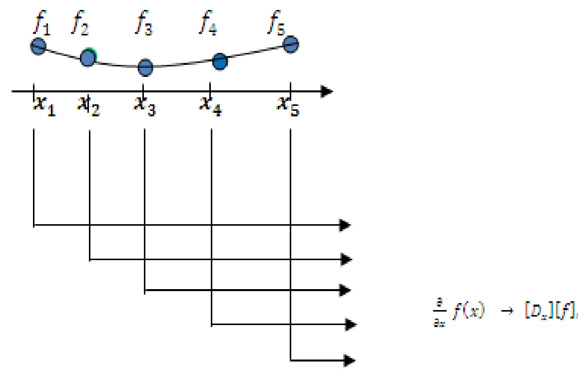

Fig. 2: The derivative of a function goemetricaly presented at any prescribed point

$$\begin{bmatrix} \\ ? \\ \\ \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f5 \end{bmatrix} = \begin{bmatrix} \frac{f_2-f_0}{2\Delta x} \\ \frac{f_3-f_1}{2\Delta x} \\ \frac{f_4-f_2}{2\Delta x} \\ \frac{f_5-f_3}{2\Delta x} \\ \frac{f_6-f_4}{2\Delta x} \end{bmatrix}$$

The approach by writing a large blank matrix equation, where our derivative matrix is what we are calling it, that is what goes in the blank matrix such that we pre-multiply column vector F we get the derivative, here we notice that right hand side is the answer, we have to find out the blank matrix.

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix} = \begin{bmatrix} \frac{f_2-f_0}{2\Delta x} \\ \frac{f_3-f_1}{2\Delta x} \\ \frac{f_4-f_2}{2\Delta x} \\ \frac{f_5-f_3}{2\Delta x} \\ \frac{f_6-f_4}{2\Delta x} \end{bmatrix} \tag{2}$$

Dealing with a finite difference, we have to keep in mind and keep control, the first is the positioning of the points from which the finite-difference approximation is calculated, in this case, the more densely they are packed usually the more accurate, we can estimate our function or one of its derivatives, but of course that's more calculation so there is a typical trade-off, we can improve our accuracy but it requires more calculations which probably requires more memory more simulation time. In finite difference, the next thing which is also extremely important is the knowledge of where we are evaluating our finite difference, and we could choose to evaluate the finite difference. It is also important that, finite differences when we are using them when we are deriving them we need to have a good knowledge of the distribution of points from which we are calculating the finite difference and where that finite difference is calculating the derivative or interpolating the function.

3 Problem & Approximating Derivative Using Finite Differences

Another way to solve the ODE boundary value problems is the finite difference method, where we can use finite difference formulae at evenly spaced grid points to approximate the differential equations. This way, we can transform a differential equation into a system of algebraic equations to solve. In the finite difference method, the derivatives in the differential equation are approximated using the finite difference formulae. We can divide the interval of [a,b] into n equal subintervals of length h as presented.
We consider a differential equation

$$\frac{d^2y}{dx^2} - 3\frac{dy}{dx} + 2y = 0 \;\; 0 \leq x \leq 5 \tag{3}$$

with boundary conditions

$$y(0) = 1, \;\; y'(5) = 5. \tag{4}$$

Using the standard finite difference method, we can write differential Equation (3) as

$$\frac{y(x+\Delta x) - 2y(x) + y(x-\Delta)}{\Delta x^2} - 3\frac{y(x+\Delta x) - y(x-\Delta x)}{\Delta x} + 2y(x) = 0 \tag{5}$$

Conveniently, the finite difference equation can be written in terms of array indices and this will take the form

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2} - 3\frac{y_{i+1-y_{i-1}}2\Delta x}{+}2y_i = 0. \tag{6}$$

This makes it easier to implement for computational purpose;

$$\left(\frac{1}{\Delta x^2} + \frac{3}{2\Delta x}\right) y_(i-1) + \left(2 - \frac{2}{2\Delta x^2}\right) y_i + \left(\frac{1}{\Delta x^2} - \frac{3}{2\Delta x}\right) y_(i+1) = 0. \tag{7}$$

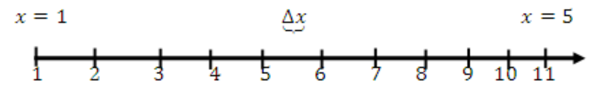

Fig. 3: A 1-D line divided into 10 equal sub-intervals

Now different cases are considered with two different types of boundary conditions and different grid sizes to compare the solutions at different grid sizes.

3.1 Case-I

The grid spacing is denoted by Δx. The size is taken as $N = 11$, which is illustrated in Figure 1. For this choice, Δx can be written as

$$\Delta x = \frac{(x_b - x_a)}{N-1} = \frac{(5-0)}{(11-1)} = 0.5,$$

substituting $\Delta x = 0.5$ into the finite difference equation Equation 7 which gives

$$\left(\frac{1}{0.5^2} + \frac{3}{(2(0.5))}\right) y_(i-1) + \left(2 - \frac{2}{0.5^2}\right) y_i + (\frac{1}{0.5^2} - \frac{3}{(2(0.5)}\Big) y_(i+1) = 0 \tag{8}$$

and

$$7y_(i-1) - 6y_i + y_(i+1) = 0. \tag{9}$$

We have to solve our differential Equation 9, this is a simple differential equation, so we have to write, once in every point on the grid, we have eleven points, which are given

$$\left\{\begin{array}{ll} y_0 = 1 & 7y_3 - 6y_4 + y_5 = 0 \\ 7y_1 - 6y_2 + y_3 = 0| & 7y_4 - 6y_5 + y_6 \\ 7y_2 - 6y_3 + y_4 = 0 & 7y_5 - 6y_6 + y_7 = 0 \end{array}\right\} \tag{10}$$

$$\left\{\begin{array}{ll} 7y_6 - 6y_7 + y_8 = 0 & 7y_9 - 6y_{10} + y_{11} + 0 \\ 7y_7 - 6y_8 + y_9 = 0| & \\ 7y_8 - 6y_9 + y_{10} = 0 & y_{11} = 5 \end{array}\right\} \tag{11}$$

and Matrix form will be

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & -6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & -6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & -6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & -6 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & -6 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 & -6 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 & -6 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & -6 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & -6 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 5 \end{bmatrix} \tag{12}$$

Solving the matrix equation will give the solution

$$(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}) = (1.0, 1.6, 2.5, 3.9, 6.3, 10.0, 15.7, 24.1, 35.1, 41.8, 5.0).$$

4 Improved Finite Difference Scheme

The last one is simpler because we had a simple differential equation it does get more complicated and tedious especially, multiple coupled sets of differential equations or pde to solve, so it should be in a manner, that could implement on MatLab code, working it is rather specific to solve that single equation, this new method is much more generic, and can rapidly solve different differential equations or different sets of differential equations, for the small to moderate size problem, this is an approved way of implementing the finite difference method

$$y(x) = [y], d/(dx)y(x) = [D_x], d^2/(dx^2)y(x) = [D_x^2] \tag{13}$$

Equation 13 takes form;

$$[D_x^2][y] - 3[D_x][y] + 2[y] = 0 \tag{14}$$

The unknown function of x becomes column vector, and everything operating y becomes a square matrix, term by term, and write everything in matrix form, above Equation 14 is the matrix form of an equation, but we want to write in terms of a standard form that is $AX = B$, for that

$$([D_x^2] - 3[D_x] + 2I)[y] = 0. \tag{15}$$

Standard form is

$$[A][y] = 0, \tag{16}$$

where

$$[A] = ([D_x^2] - 3[D_x] + 2I). \tag{17}$$

This is a way to calculate the square matrix A the sum is an algebraic combination of three matrices, with all same size matrices, what we have to do we will change Equation 17 into one line MatLab code i.e.

$$[A] = ([D_x^2] - 3[D_x] + 2I) = DX2 - 3DX + 2I. \tag{18}$$

This equation is so generic, this becomes very simple and rapid to change this matrix for any differential equation, we are solving we would just change it as one line of code, and that's neat. Here in this method, we start direct matrix Equation 18,

$$[A][y] = 0 \Longrightarrow \begin{bmatrix} -6 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & -6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & -6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & -6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & -6 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & -6 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 & -6 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 & -6 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & -6 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & -6 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & -6 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{19}$$

The Equation 19 is matrix equation, and boundary values are not discretized in the matrix, so the matrix is simply a raw matrix equation in standard form. We have to incorporate boundary values, the first thing we want to do is the 1st and last rows, these are the areas where we want to force all the values zero, which means we are throwing out the finite difference equation that had written for the 1st and the 11th point, we just throw them out, and make them all zeros, after that we will go into the same rows, and we put a 1 (one) in the diagonal position, then we will go to the last column which had all zeros will insert our boundary values that is the first point we are forcing value 1, and then the last point forcing a value of 5.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & -6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & -6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & -6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & -6 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & -6 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 & -6 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 & -6 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & -6 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & -6 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 5 \end{bmatrix}. \tag{20}$$

$$(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}) = (1.0, 1.6, 2.5, 3.9, 6.3, 10.0, 15.7, 24.1, 35.1, 41.8, 5.0)$$

4.1 Case-II

In this case, the grid size is chosen as $\Delta x = 0.25$. Solving the Equation(06), so let us solve this using $N = 21$ uniform grid points, and spacing Δx then;

$$\Delta x = \frac{(x_b - x_a)}{(N-1)} = \frac{(5-0)}{(21-1)} = 0.25,$$

substituting $\Delta x = 0.25$ into the finite difference equation Equation(8), which gives

$$\left(\frac{1}{(0.25)^2} + \frac{3}{2(0.25)}\right) y_(i-1) + \left(2 - \frac{2}{(0.25)^2}\right) y_i + \left(\frac{1}{(0.25)^2} - \frac{3}{2(0.25)}\right) y_(i+1) = 0 \tag{21}$$

$$22y_{i-1} - 30y_i + 10y_{i+1} = 0 \tag{22}$$

We have to solve the same differential equation with Equation 21, boundary values so we have simple to write down the finite difference. Equation 22, once in every point on the grid, we have twenty one points, this is the sort of raw standard matrix in which boundary value is not present given.

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
22 & -30 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 22 & -30 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 22 & -30 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 22 & -30 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 22 & -30 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 22 & -30 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 22 & -30 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 & -30 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 & -30 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 & -30 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 & -30 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 & -30 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 & -30 & 10 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 & -30 & 10 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 & -30 & 10 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 & -30 & 10 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 & -30 & 10 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 & -30 & 10 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 & -30 & 10 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \\ y_{16} \\ y_{17} \\ y_{18} \\ y_{19} \\ y_{20} \\ y_{21} \end{bmatrix}
=
\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 5 \end{bmatrix}, \quad (23)$$

4.2 Case-III

In this case, the boundary values y(0)=1 and y(5)=2.5 are chosen in addition to the grid size $\Delta x = 0.125$. This grid-size is illustrated in Figure 4.

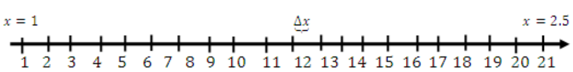

Fig. 4: The 1-D line divided into 20 equal sub-intervals

For solving the Equation 06 on grid using N=21; we have

$$\Delta x = (x_b - x_a)/(N-1) = (2.5-0)/(21-1) = 0.125$$

Substituting $\Delta x = 0.125$ into the finite difference equation Equation(8), which gives

$$(1/(0.125)^2 + 3/(2.(0.125)))y_(i-1) + (2 - 2/(0.125)^2)y_i \\ + (1/(0.125)^2 - 3/(2.(0.125)))y_(i+1) = 0 \quad (24)$$

and

$$38y_(i-1) - 63y_i + 26y_(i+1) = 0 \quad (25)$$

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
38 & -63 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 38 & -63 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 38 & -63 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 38 & -63 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 38 & -63 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 38 & -63 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 38 & -63 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 38 & -63 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 38 & -63 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 38 & -63 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 38 & -63 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 38 & -63 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 38 & -63 & 26 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 38 & -63 & 26 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 38 & -63 & 26 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 38 & -63 & 26 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 38 & -63 & 26 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 38 & -63 & 26 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 38 & -63 & 26 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \\ y_{16} \\ y_{17} \\ y_{18} \\ y_{19} \\ y_{20} \\ y_{21} \end{bmatrix}
=
\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2.5 \end{bmatrix}, \quad (26)$$

4.3 Cale-IV

The different boundary values y(0)= 1 "and" y(1)=1 are considered in addition to the grid size $\Delta x = 0.05$.

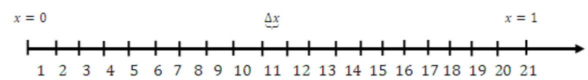

Fig. 5: The 1-D line divided into 20 equal sub-intervals

Solving the Equation 06, so let us solve this using N=21 uniform grid points, and spacing Δx then;

$$\Delta x = (x_b - x_a)/(N-1) = (1-0)/(21-1) = 0.05,$$

substituting $\Delta x = 0.05$ into the finite difference equation Equation(8), which gives

$$(1/(0.05)^2 + 3/(2.(0.05)))y_(i-1) + (2 - 2/(0.05)^2)y_i \\ + (1/(0.05)^2 - 3/(2.(0.05)))y_(i+1) = 0 \quad (27)$$

$$215y_(i-1) - 399y_i + 185y_(i+1) = 0 \quad (28)$$

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
215 & -399 & 185 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 215 & -399 & 185 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 215 & -399 & 185 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 215 & -399 & 185 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 215 & -399 & 185 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 215 & -399 & 185 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 215 & -399 & 185 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 215 & -399 & 185 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 215 & -399 & 185 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 215 & -399 & 185 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 215 & -399 & 185 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 215 & -399 & 185 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 215 & -399 & 185 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 215 & -399 & 185 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 215 & -399 & 185 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 215 & -399 & 185 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 215 & -399 & 185 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 215 & -399 & 185 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 215 & -399 & 185 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \\ y_{16} \\ y_{17} \\ y_{18} \\ y_{19} \\ y_{20} \\ y_{21} \end{bmatrix}
=
\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (29)$$

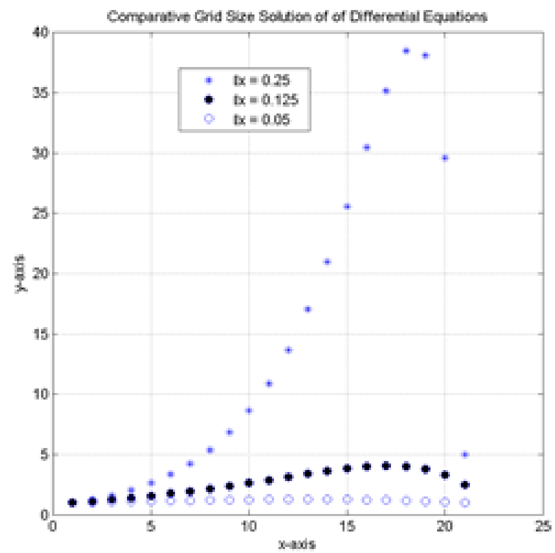

Fig. 6: Comparative between small, moderate and grid sizes

5 Discussion

The solutions obtained from different grid sizes as discussed in different cases in previous section is plotted in Figure 3. Curve of solution when smaller grid size is chosen is closer to exact solution. Naturally, this would have been the case but at cost of more computational time. For these test problems, the matrix equation is solved directly. However, indirect solver can be implemented for more large domain problems.

6 Conclusion

The improved finite difference method is tested, which is specially designed for the hard type of differential equations, and system of differential equation as well pde, in matrix oriented form, and how it is built and assembled, this approach converts the differential equation into MatLab Codes which makes the hard type of differential equation(s), easier to be evaluated as presented. In the last session, the comparison of different grid sizes and different boundary value problems was tested and found that small grid sizes are usually the more accurate, but of course, that needs more calculation, so there is a typical trade-off, we can improve our accuracy but it requires more calculations which probably requires more simulation, more time, and more memory.

References

[1] I. Singer and E. Turkel, “High-order finite difference methods for the Helmholtz equation,” Computer Methods in Applied Mechanics and Engineering, vol. 163, no. 1, Art. no. 1, 1998.

[2] J. M. G-Jordan, S. Rojas, M. F-Villegas, and J. E. Castillo, “A new second order finite difference conservative scheme,” Divulgaciones Mathemticas, vol. 13, no. 1, pp. 107–122, 2005.

[3] S. Operto, J. Virieux, P. Amestoy, J. L’Excellent, L. Giraud, and H. Ali, “3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study,” Geophysics, vol. 72, no. 5, Art. no. 5, 2007.

[4] A. H. Sheikh, “Development of Helmholtz Solver Based on Shifted Laplace Preconditioner and a Multigrid Deflation Technique,” Delft University of Technology, The Netherlands, PhD Thesis, 2014.

[5] A. H. Sheikh, C. Vuik, and D. Lahaye, “Fast iterative solution methods for the Helmholtz equation,” DIAM, TU Delft, pp. 9–11, 2009.

[6] A. Shaikh G, A. H. Sheikh, A. Asif, and S. Zeb, “Critical Review of Preconditioners for Helmholtz Equation and their Spectral Analysis,” Ind. Jour. Sc. Tech, vol. 12, no. 20, May 2019.

[7] A. Bayliss, C. I. Goldstein, and E. Turkel, “On accuracy conditions for the numerical computation of waves,” Journal of Computational Physics, vol. 59, no. 3, pp. 396–404, 1985.

[8] Y. A. Erlangga and R. Nabben, “On a multilevel Krylov Method for the Helmholtz Equation preconditioned by Shifted Laplacian,” Electronic Transactions on Numerical Analysis (ETNA), vol. 31, pp. 403–424, 2008.

[9] A. H. Sheikh, D. Lahaye, L. Garcia Ramos, R. Nabben, and C. Vuik, “Accelerating the Shifted Laplace Preconditioner for the Helmholtz Equation by Multilevel Deflation,” J. Comput. Phys., vol. 322, no. C, pp. 473–490, Oct. 2016.

[10] W. A. Siyal, A. H. Sheikh, K. B. Amur, A. Shaikh G, and R. A. Malookani, “On the Efficiency of Multigrid Solver for Shifted Laplace Equation in a Heterogeneous Medium,” Int. J. Appl. Math, vol. 59, no. 3, pp. 102–114, 2020.