Benchmarking of a Linear Convection/Diffusion Problem in MATLAB

Öztürk S. B.

Middle East Technical University, Çankaya, Üniversiteler St., TR

E-mail: berkay.ozturk_01@metu.edu.tr

04.04.2023

Abstract. We present a benchmark project for a second order steady convection/diffusion problem with pure Dirichlet condition, using the finite differences method (FDM). We used a MATLAB code implementing the Thomas Matrix Algorithm to solve the matrix system and compared our results firstly with varying nodal density and secondly with varying ε values. We found that increasing nodal density decreases the numerical error, whereas decreasing ε values increase the overall numerical error in directed difference and exponential fitting schemes, but does not alter the error in the central difference scheme.

Keywords: MATLAB, Convection-Diffusion, PDE, Finite Difference Method, Computational Physics.

1. Introduction:

There are several methods to handle partial differential equation models. Some of which are Finite Element Method(FDM), Finite Element Method (FEM), and Pseudo-Spectral methods. To gain insight from a complex physical system mostly requires more than fundamental knowledge. A mathematical model is first generated that mimics the behavior and the evolution of the system. However, while arriving at a model describing such a system errors start to present. Furthermore, while solving the problem in a numerical environment, computational errors accumulate the overall error. At this point, benchmarking is a crucial path to select the parameters in such a way that the numerical error is minimal. In this project, we present a benchmark problem solving the second order convection/diffusion problem using the FDM, in MATLAB.

2. The Description of the Problem

While solving partial differential equations in computational environments with FDM, it is crucial to get the coefficient matrix of the primary variable. Then, using the known matrix operations, the system of equations is solved and the solution is mapped onto the grid points. In general, the strong form of second order convection/diffusion problem is,

$$-\varepsilon u^{''}_{(x)} - a_{(x)}u^{'}_{(x)} + b_{(x)}u_{(x)} = f_{(x)} \quad in \;\; \Omega \tag{1}$$

Here, $u_{(x)}$ is the primary function of interest, $f_{(x)}$ is the source function, ε is the diffusivity coefficient, $a_{(x)}$ and $b_{(x)}$ are the coefficients depending on the problem. The $\alpha_{(x,y)}$ function may also be thought as the heat conduction coefficient of material recognized from the heat diffusion/conduction problem. The boundary conditions subject to equation (1) are,

$$\begin{aligned} \xi_0 u_{(c)} - \eta_0 \varepsilon u^{'}_{(c)} = \varphi \;\; in \;\; \partial\Omega \\ \xi_1 u_{(d)} - \eta_1 \varepsilon u^{'}_{(d)} = \psi \;\; in \;\; \partial\Omega \end{aligned} \tag{2}$$

where the ξ terms represent the Dirichlet type boundary condition at boundary points (c,d), η terms represent the Neumann type boundary conditions. The mixture of two conditions arise the Robin boundary value problem. In our case, the problem is described by the partial differential system with Dirichlet boundary conditions as follows,

$$\begin{aligned} -\varepsilon u^{''}_{(x)} - x u^{'}_{(x)} + u_{(x)} = (1+\varepsilon\pi^2)\cos(\pi x) + \pi x \sin(\pi x) \;\; in \;\; \Omega \\ u_{(-1)} = -1, \quad u_{(1)} = 1 \quad in \;\; \partial\Omega \end{aligned} \tag{3}$$

Where the problem domain is one dimensional, and in $x \in (-1, 1)$. Compared to the general expression in relation 1, we have $a_{(x)} = x$, $b_{(x)} = 1$, and the source function is as given in the RHS. The known analytical solution to be compared for the benchmark analysis is,

$$u_{(x)} = \cos(\pi x) + x + \frac{x \; erf(x/\sqrt{2\varepsilon}) + \sqrt{2\varepsilon/\pi}\exp(-x^2/2\varepsilon)}{erf(1/\sqrt{2\varepsilon}) + \sqrt{2\varepsilon/\pi}\exp(1/2\varepsilon)} \tag{4}$$

2.1. Mapping the Problem onto the Finite Difference Scheme

In the FDM, a continuous problem is mapped onto the discrete set of points. The derivative operators of any kind are evaluated in those grids, and solved for the primary function of interest. In our case, the discretized version of the continuous problem yields,

$$\begin{aligned} \frac{-\varepsilon}{h^2}\left(\gamma_i - \frac{a_{(x_i)}h}{2\varepsilon}\right)u^h_{i-1} + \left(\frac{2\varepsilon\gamma_i}{h^2} + b_{(x_i)}\right)u^h_i - \frac{\varepsilon}{h^2}\left(\gamma_i + \frac{a_{(x_i)}h}{2\varepsilon}\right)u^h_{i+1} = f_{(x_i)} \\ u_{(x_1)} = -1, \;\; u_{(x_n)} = 1 \end{aligned} \tag{5}$$

where γ is the artificial diffusion factor, and corresponds to different approximation schemes of the differential operators, subscript for the i^{th} variable of the space domain, superscript for the time level h. The different γ factors to the corresponding functions are given below.

$$\begin{aligned} \gamma = 1 \quad &(Central\ Difference) \\ \gamma = 1 + 0.5|R_i| \quad &(Directed\ Difference) \\ \gamma = 0.5R_i \coth(0.5R_i) \quad &(Exponential\ Fitting) \end{aligned} \tag{6}$$

where $R_i = a_{(x_i)}h/\varepsilon \equiv$ *Local Reynold's number.*

To gather the solution for the discrete linear system described in Equation 5, we describe it in a matrix form as follows,

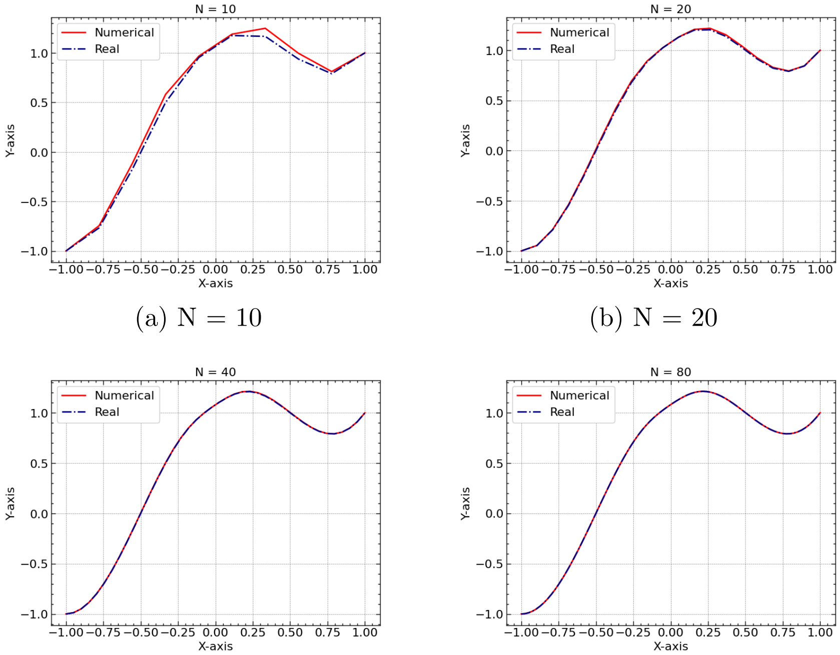
$$\begin{bmatrix} B_1 & -C_1 & 0 & . & . & 0 \\ -A_2 & B_2 & -C_2 & 0 & . & . \\ 0 & -A_3 & B_3 & . & . & . \\ . & 0 & . & . & . & . \\ . & . & . & . & . & 0 \\ . & . & . & . & B_{N-1} & -C_{N-1} \\ 0 & . & . & 0 & -A_N & B_N \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ . \\ . \\ . \\ u_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ . \\ . \\ . \\ f_N \end{bmatrix} \tag{7}$$

Which is a tridiagonal sparse matrix, and has special algorithms to generate and solve in numerical environments, since problems with a high number of nodal elements require a vast amount of memory and computation sources. The coefficients in relation 5 are simplified by the notation set of (A_i, B_i, C_i) in the domain $(2, N-1)$, which are shown in the relation below.

$$\begin{aligned} A_i &= \frac{\varepsilon}{h^2}\left(\gamma_i - \frac{a_{(x_i)}h}{2\varepsilon}\right) \\ B_i &= \left(\frac{2\varepsilon\gamma_i}{h^2} + b_{(x_i)}\right) \\ C_i &= \frac{\varepsilon}{h^2}\left(\gamma_i + \frac{a_{(x_i)}h}{2\varepsilon}\right) \end{aligned} \tag{8}$$

3. Solution in MATLAB

We solved the problem described in section 1, in MATLAB environment. The test configurations are divided into two parts. In the first part, we hold the ε coefficient constant with a value of 0.01, and with a varying nodal point density of N = (10, 20, 40, 80). In the second part, we take the point density constant with a value of N = 40 and apply the simulation with varying ε value of $(1, 10^{-1}, 10^{-2}, 10^{-3})$. In both parts, we applied the conditions for three different γ functions as described in relation (6).



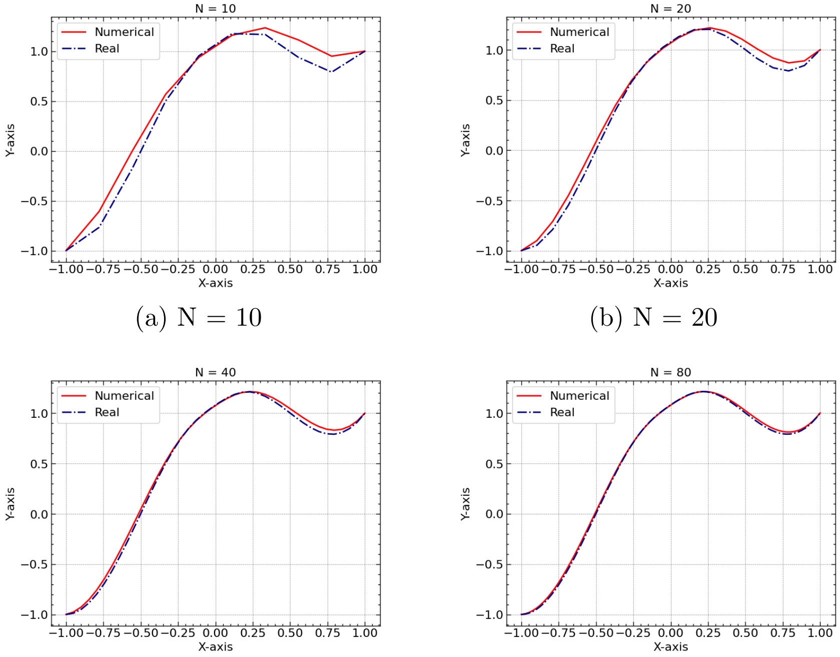
(a) N = 10

(b) N = 20

(c) N = 40

(d) N = 80

Figure 1: Illustration of the ***centered difference scheme*** with varying ***N*** values. Red lines represent the analytical solution, and blue dotted dashed lines represent the numerical solution.



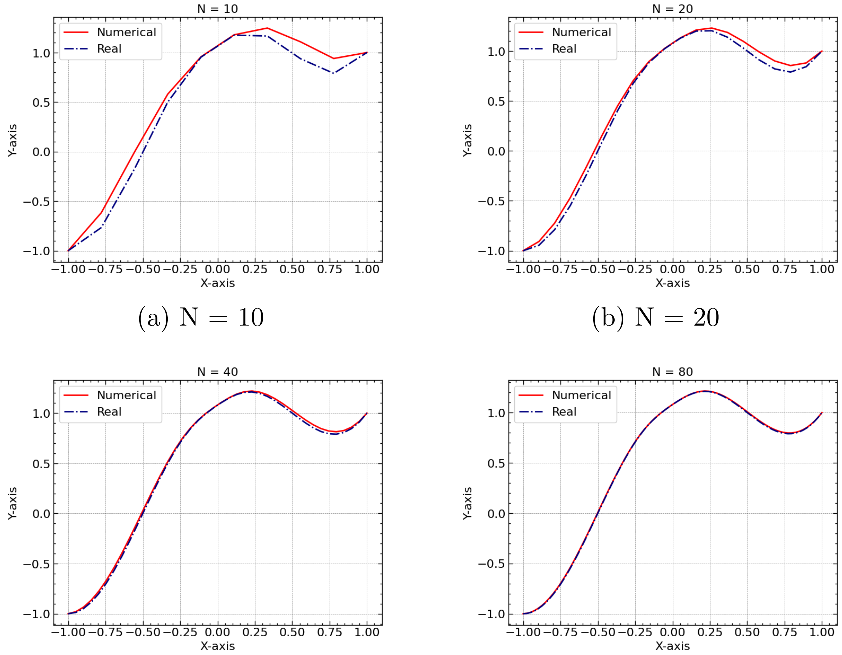
(a) N = 10

(b) N = 20

(c) N = 40

(d) N = 80

Figure 2: Illustration of the ***directed difference scheme*** with varying ***N*** values. Red lines represent the analytical solution, and blue dotted dashed lines represent the numerical solution.



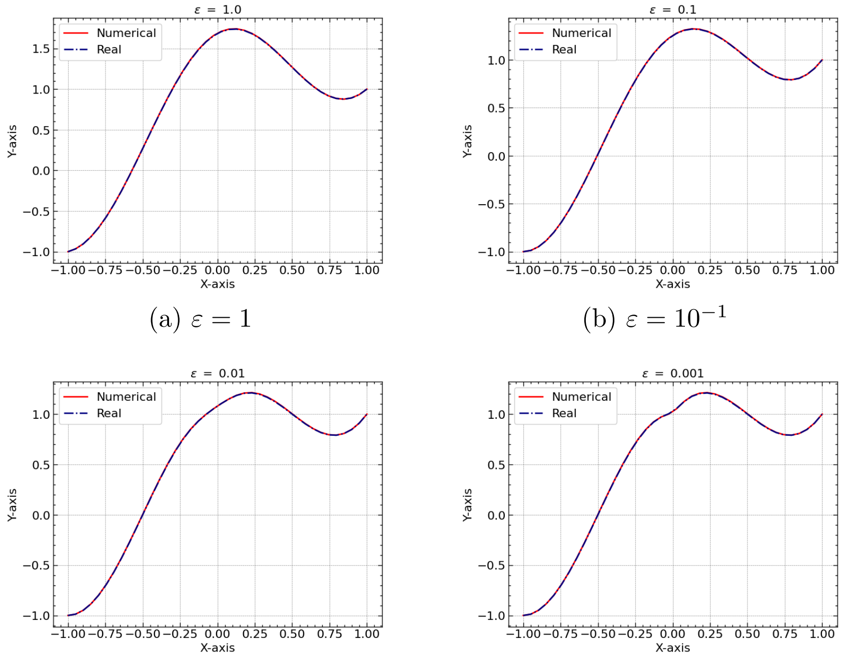
(a) N = 10

(b) N = 20

(c) N = 40

(d) N = 80

Figure 3: Illustration of the ***exponential fitting scheme*** with varying ***N*** values. Red lines represent the analytical solution, and blue dotted dashed lines represent the numerical solution.



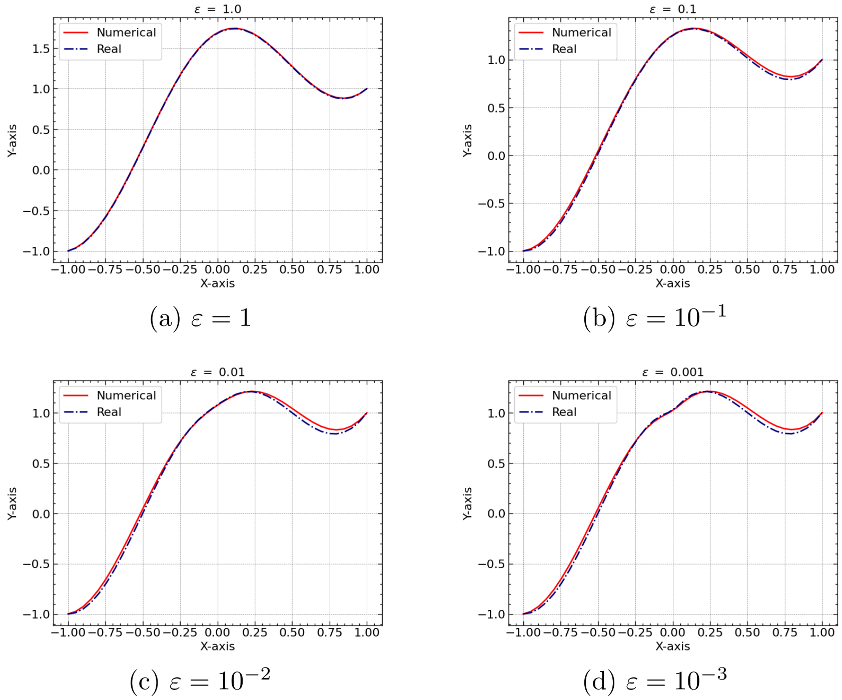
(a) $\varepsilon = 1$

(b) $\varepsilon = 10^{-1}$

(c) $\varepsilon = 10^{-2}$

(d) $\varepsilon = 10^{-3}$

Figure 4: Illustration of the ***central difference scheme*** with varying ε values. Red lines represent the analytical solution, and blue dotted dashed lines represent the numerical solution.



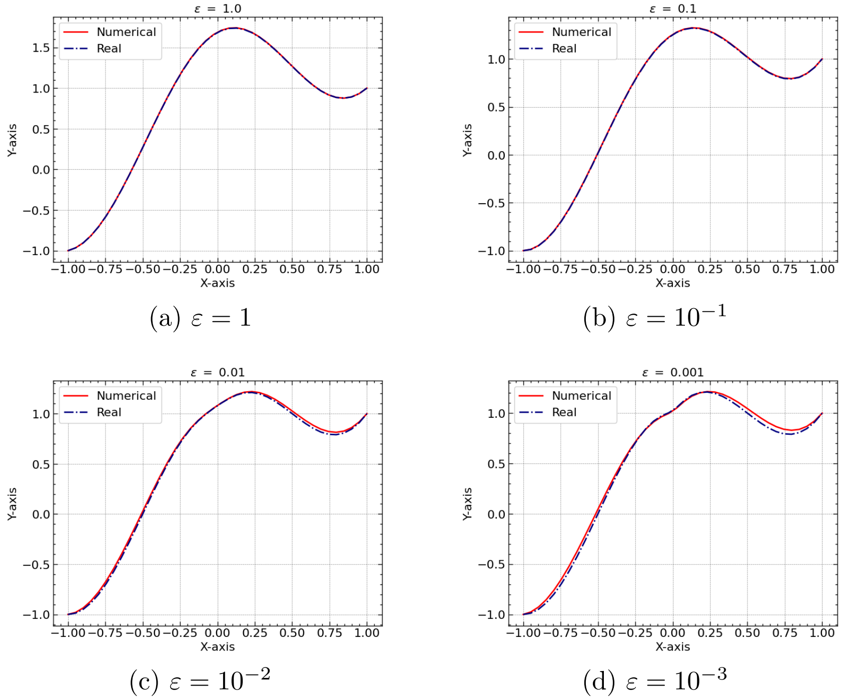
(a) $\varepsilon = 1$

(b) $\varepsilon = 10^{-1}$

(c) $\varepsilon = 10^{-2}$

(d) $\varepsilon = 10^{-3}$

Figure 5: Illustration of the ***directed difference scheme*** with varying ε values. Red lines represent the analytical solution, and blue dotted dashed lines represent the numerical solution.



(a) $\varepsilon = 1$

(b) $\varepsilon = 10^{-1}$

(c) $\varepsilon = 10^{-2}$

(d) $\varepsilon = 10^{-3}$

Figure 6: Illustration of the ***exponential fitting scheme*** with varying ε values. Red lines represent the analytical solution, and blue dotted dashed lines represent the numerical solution.

4. Data and Error Analysis

As seen from the figures, in all central, directed, and exponential schemes, the accuracy increases with increasing N values, however, decreasing ε values affect the error in a more complex way, and there is no single trend for all the difference schemes. To get more insight from the results, we investigated the errors in terms of root mean square error, and maximum norms. The relations we used to calculate the errors are given below. In table 1, the errors are given and illustrated in figure 7.

$$\begin{aligned} \|e\|_2 &= \left[\frac{1}{N^2}\sum_{i=1}^{N}(u_{(x_i)} - u_i)^2\right]^{1/2} \\ \|e\|_\infty &= \max_{1\leq i\leq N} |u_{(x_i)} - u_i| \end{aligned} \tag{9}$$

Table 1: $\|e\|_2$ by varying N values.

N	Central	Directed	Exponential
10	0.0463	0.1104	0.1078
20	0.0100	0.0599	0.0511
40	0.0023	0.0312	0.0195
80	0.0006	0.0159	0.0058

Table 2: $\|e\|_\infty$ by varying N values.

N	Central	Directed	Exponential
10	0.0815	0.1749	0.1704
20	0.0100	0.0971	0.0796
40	0.0023	0.0508	0.0301
80	0.0006	0.0258	0.0089

Table 3: $\|e\|_2$ by varying ε values.

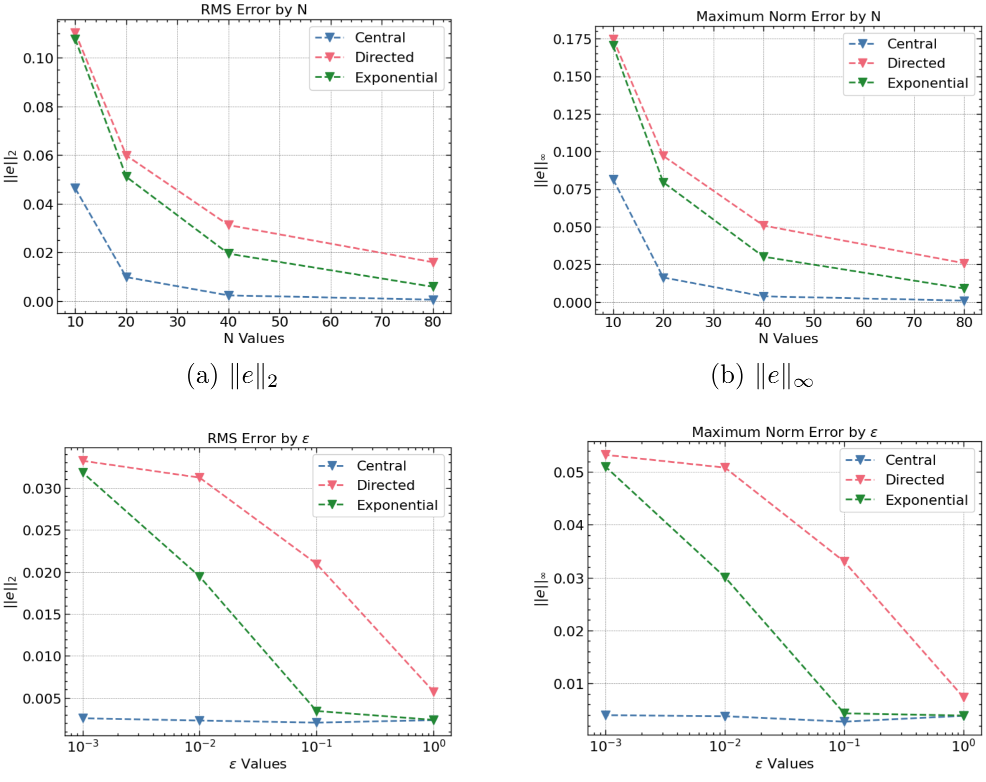
ε	Central	Directed	Exponential
1	0.0024	0.0058	0.0024
10^{-1}	0.0021	0.0210	0.0035
10^{-2}	0.0023	0.0312	0.0195
10^{-3}	0.0026	0.0332	0.0318

Table 4: $\|e\|_\infty$ by varying ε values.

ε	Central	Directed	Exponential
1	0.0038	0.0073	0.0039
10^{-1}	0.0027	0.0330	0.0043
10^{-2}	0.0038	0.0508	0.0301
10^{-3}	0.0039	0.0532	0.0510

Looking at figure 7, increasing N density decreases the error quadratically for the central difference scheme, and linearly for directed and exponential schemes. However, there is a fluctuation in the error for the central scheme, as ε value is changed. This is due to the local violation of the stability criteria given in relation 10, and it is investigated in figure 8 for the three methods. Furthermore, there is an overall performance decrease of the directed and exponential methods as we decreased the ε value.

$$\gamma_i > 0.5|R_i| \tag{10}$$

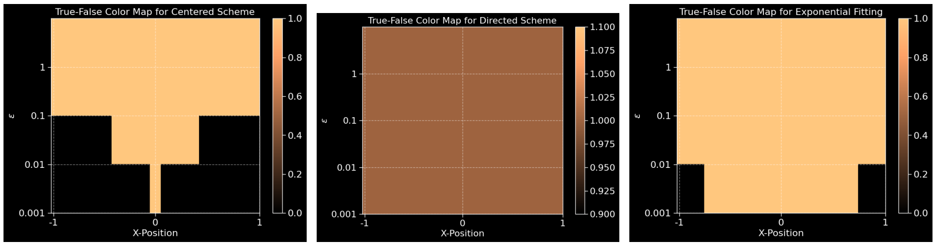


(a) $\|e\|_2$ (b) $\|e\|_\infty$

(c) $\|e\|_2$ (d) $\|e\|_\infty$

Figure 7: Distribution of RMS and normal errors by control parameters. ***(a)*** shows the RMS error, ***(b)*** shows the norm error by changing mesh density. Where logarithmic scaling is applied, ***(c)*** and ***(d)*** show RMS and maximum norm error, respectively.

$$\|L^h(u)^h - (Lu)'h\|_\infty \le C\left(\max_{1\le i\le N-1}|1-\gamma_i| + h^2\right) \tag{11}$$



(a) (b) (c)

Figure 8: True-False map for the criteria given in equation (10), for the three methods. If the condition is met, the value of 1 is displayed, otherwise, the value of 0 is displayed. ***(a)*** shows the Central Difference , ***(b)*** shows the Directed Difference, ***(c)*** shows the Exponential Fitting scheme. X-axis represents the one dimensional problem domain $x \in (-1, -1)$, Y-axis represents the ε values.

5. Conclusion

We simulated the problem described in equation (3) by FDM in MATLAB, using Thomas Matrix Algorithm to solve the linear system shown in equations (5) and (7). We firstly held the ε value constant and increased the grid density, and we found that decreasing the grid spacing increases the accuracy linearly in Directed Scheme and Exponential Scheme, and quadratically in Centered Scheme as expected from equation (11). Secondly, we held the grid density constant and decreased the ε value. We found that decreasing ε may cause local stability violation in Central and Exponential schemes as described in equation (10), and shown in figure 8. However, the Directed Scheme was not affected by the stability criteria, and remained the same for all ε values, as expected. Overall, the Central scheme is determined to be more profitable if the $a_{(x_i)}$ term is isotropic, and stability is not violated. On the other side, if the system is highly anisotropic, the Directed scheme may be preferred due to its inherent stability.