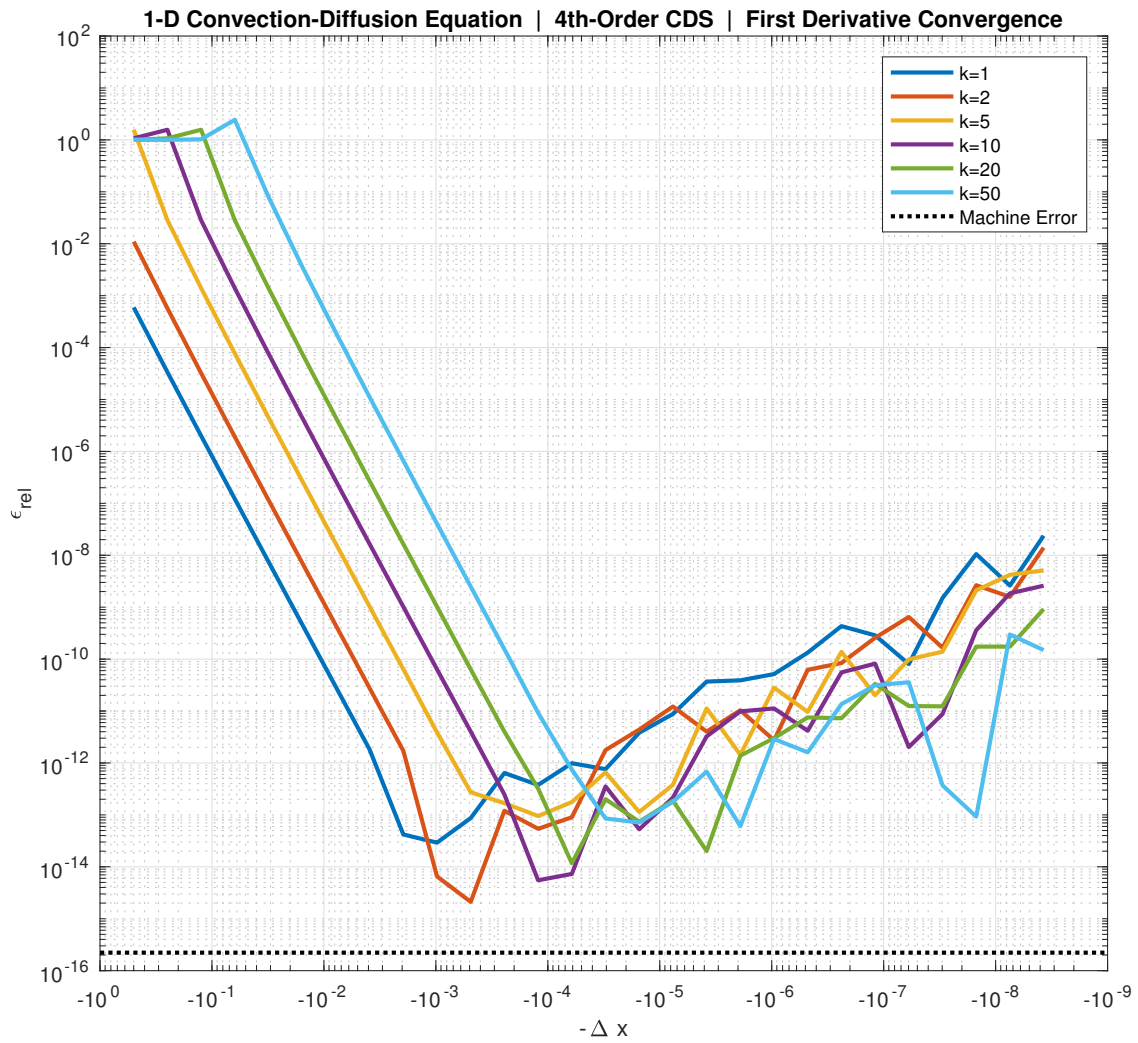


AERO 430 – Numerical Simulation

Exact Solution of the Finite Difference Equations

Ross B. Alexander



Contents

1	Model Problems	4
1.1	1-D Diffusion Equation	4
1.2	1-D Harmonic Wave Equation	4
1.3	1-D Convection-Diffusion Equation	4
1.4	2-D Orthotropic Diffusion Equation	5
2	Analytical Solutions	6
2.1	1-D Diffusion Equation	6
2.2	1-D Harmonic Wave Equation	7
2.3	1-D Convection-Diffusion Equation	8
2.4	2-D Orthotropic Diffusion Equation	9
3	Quantities of Interest	10
3.1	1-D Diffusion Equation	10
3.2	1-D Harmonic Wave Equation	10
3.3	1-D Convection-Diffusion Equation	10
3.4	2-D Orthotropic Diffusion Equation	11
4	Numerical Methods	12
4.1	1-D Diffusion Equation	12
4.1.1	2nd-Order Central Difference Scheme - Exact Finite Difference Method	12
4.1.2	4th-Order Central Difference Scheme - Exact Finite Difference Method	12
4.2	1-D Harmonic Wave Equation	13
4.2.1	2nd-Order Central Difference Scheme - Exact Finite Difference Method	13
4.2.2	4th-Order Central Difference Scheme - Exact Finite Difference Method	13
4.3	1-D Convection-Diffusion Equation	14
4.3.1	2nd-Order Central Difference Scheme - Exact Finite Difference Method	14
4.3.2	4th-Order Central Difference Scheme - Exact Finite Difference Method	14
4.4	2-D Orthotropic Diffusion Equation	15
4.4.1	2nd-Order Central Difference Scheme - Exact Finite Difference Method	15
4.4.2	4th-Order Central Difference Scheme - Exact Finite Difference Method	15
5	Convergence Analysis	17
5.1	Rate of Convergence Derivation	17
5.2	1-D Diffusion Equation	18

5.2.1	2nd-Order Central Difference Scheme	18
5.2.2	4th-Order Central Difference Scheme	19
5.3	1-D Harmonic Wave Equation	20
5.3.1	2nd-Order Central Difference Scheme	20
5.3.2	4th-Order Central Difference Scheme	21
5.4	1-D Convection-Diffusion Equation	22
5.4.1	2nd-Order Central Difference Scheme	22
5.4.2	4th-Order Central Difference Scheme	23
5.5	2-D Orthotropic Diffusion Equation	24
5.5.1	2nd-Order Central Difference Scheme	24
5.5.2	4th-Order Central Difference Scheme	25
6	Discussion	26
A	MATLAB Code	28
A.1	fp_1d_diffusion.m	28
A.2	fp_1d_wave.m	29
A.3	fp_1d_convection.m	30
A.4	fp_2d_diffusion.m	32

1 Model Problems

1.1 1-D Diffusion Equation

The *first* model second-order linear ordinary differential equation boundary-value problem consists of:

- the one-dimensional second-order linear ordinary differential equation:

$$-u''(x) + k^2 u(x) = k^2 x \quad x \in [0, 1] \quad (1.1)$$

- the boundary conditions:

$$u(0) = 0 \quad \text{and} \quad u(1) = 0 \quad (1.2)$$

The physical model of the second-order linear ordinary differential equation boundary-value problem is that of (1) the temperature of a bar for uniaxial heat conduction, and (2) the deflection of a beam for uniaxial deformation with distributed elastic restraint.

1.2 1-D Harmonic Wave Equation

The *second* model second-order linear ordinary differential equation boundary-value problem consists of:

- the one-dimensional second-order linear ordinary differential equation:

$$u''(x) + k^2 u(x) = k^2 x \quad x \in [0, 1] \quad (1.3)$$

- the boundary conditions:

$$u(0) = 0 \quad \text{and} \quad u(1) = 0 \quad (1.4)$$

The physical model of the second-order linear ordinary differential equation boundary-value problem is that of the amplitude of standing waves for uniaxial forced vibration of a bar.

1.3 1-D Convection-Diffusion Equation

The *third* model second-order linear ordinary differential equation boundary-value problem consists of:

- the one-dimensional second-order linear ordinary differential equation:

$$-u''(x) + cu'(x) = 0 \quad x \in [0, 1] \quad (1.5)$$

- the boundary conditions:

$$u(0) = 0 \quad \text{and} \quad u(1) = 1 \quad (1.6)$$

The physical model of the second-order linear ordinary differential equation boundary-value problem is that of the concentration of a flow property that convects and diffuses proportional to the constant c . For example, the convection-diffusion equation could represent the concentration of ink as a function of distance in a quasi-one-dimensional flow.

1.4 2-D Orthotropic Diffusion Equation

The *fourth* model second-order linear ordinary differential equation boundary-value problem consists of:

- the two-dimensional second-order linear partial differential equation:

$$\nabla \cdot (\mathbf{D} \nabla u) = 0 \quad x, y \in [0, 1] \times [0, 1] \quad (1.7)$$

- the orthotropic diffusivity tensor:

$$\mathbf{D} = \begin{bmatrix} k^2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (1.8)$$

- the simplified two-dimensional second-order linear partial differential equation:

$$k^2 \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad x, y \in [0, 1] \times [0, 1] \quad (1.9)$$

- the boundary conditions:

$$\begin{aligned} u(x, 0) &= 0 & u(0, y) &= 0 \\ u(x, 1) &= \sin \pi x & u(1, y) &= 0 \end{aligned} \quad (1.10)$$

The physical model of the orthotropic diffusion equation is that of (1) electrical conduction through a medium that has orthotropic electrical conductivity, (2) magnetism of a material that has orthotropic magnetic permeability, (3) thermal conduction through a material that has orthotropic thermal conductivity, or (4) diffusion through a fluid that has orthotropic diffusivity.

2 Analytical Solutions

2.1 1-D Diffusion Equation

Previously, it was shown that for the one-dimensional second-order linear ordinary differential equation with specified boundary conditions (reproduced below) that $u(x)$ is a solution to the differential equation on $x \in [0, 1]$.

$$-u''(x) + k^2 u(x) = k^2 x \quad x \in [0, 1] \quad (2.1)$$

$$u(0) = 0 \quad \text{and} \quad u(1) = 0 \quad (2.2)$$

$$u(x) = x - \frac{\sinh(kx)}{\sinh(k)} \quad (2.3)$$

A plot of the analytical solution for values of k is depicted in Figure 2.1.1:

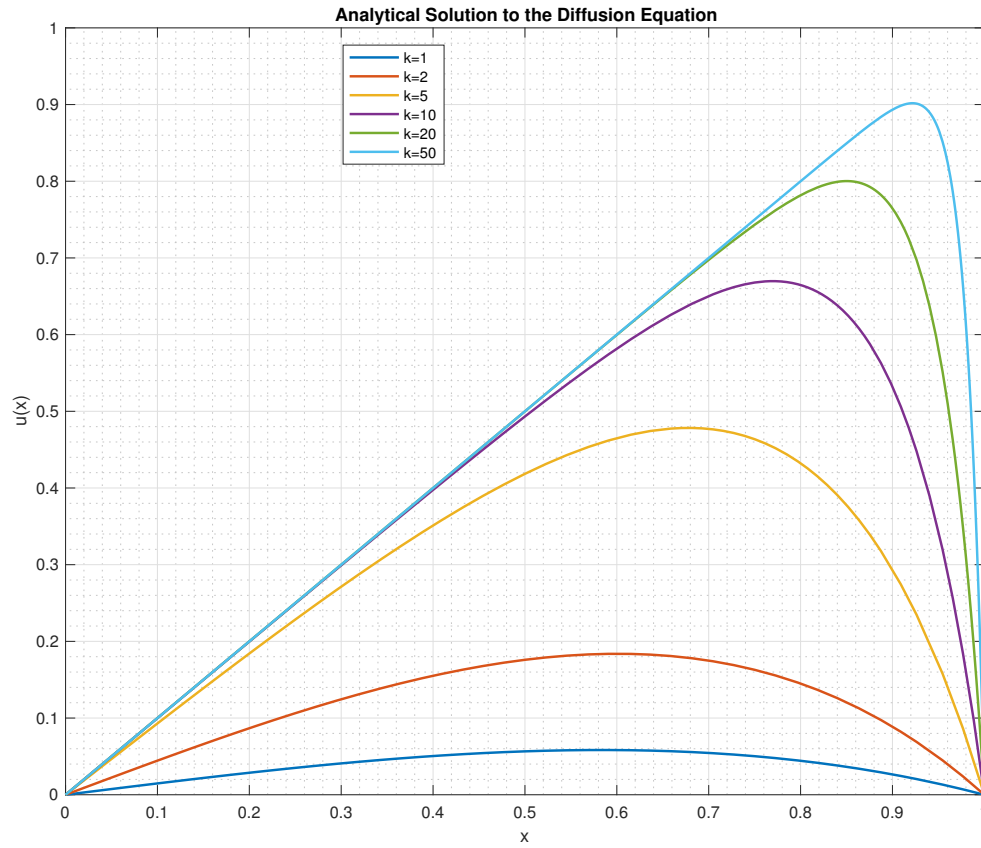


Figure 2.1.1 – Analytical Solution to the Diffusion Equation for Values of k

2.2 1-D Harmonic Wave Equation

Previously, it was shown that for the one-dimensional second-order linear ordinary differential equation with specified boundary conditions (reproduced below) that $u(x)$ is a solution to the differential equation on $x \in [0, 1]$.

$$u''(x) + k^2 u(x) = k^2 x \quad x \in [0, 1] \quad (2.4)$$

$$u(0) = 0 \quad \text{and} \quad u(1) = 0 \quad (2.5)$$

$$u(x) = x - \frac{\sin(kx)}{\sin(k)} \quad (2.6)$$

A plot of the analytical solution for values of k is depicted in Figure 2.2.1:

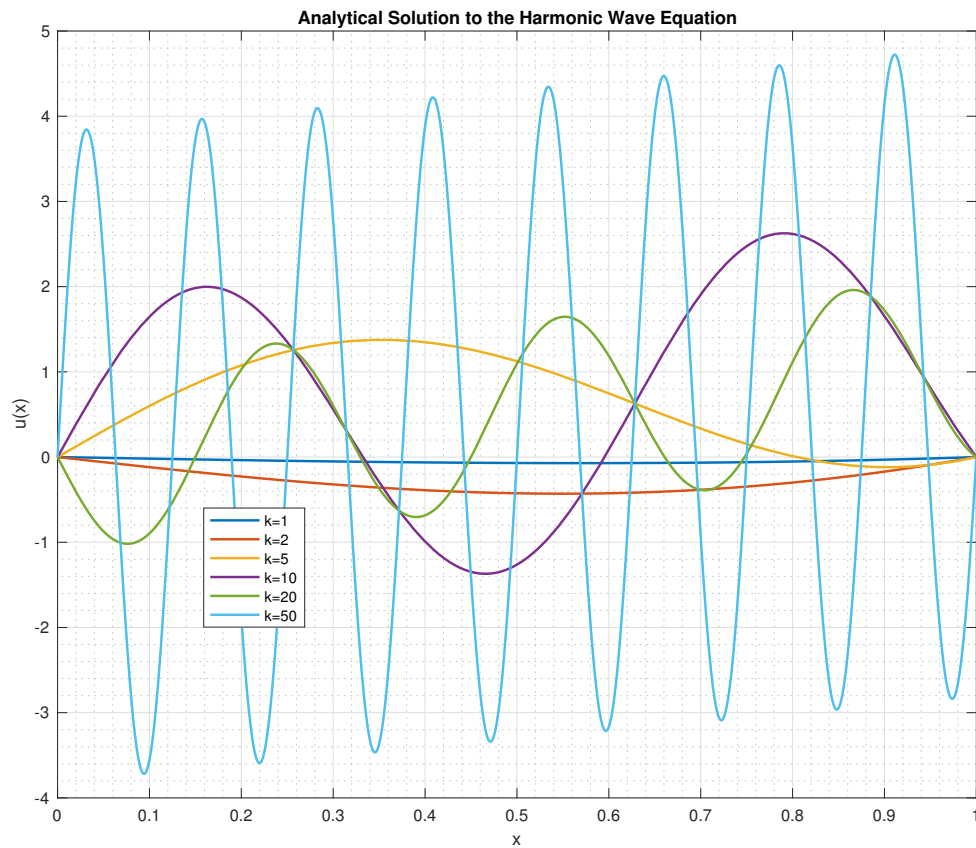


Figure 2.2.1 – Analytical Solution to the Harmonic Wave Equation for Values of k

2.3 1-D Convection-Diffusion Equation

Previously, it was shown that for the one-dimensional second-order linear ordinary differential equation with specified boundary conditions (reproduced below), that $u(x)$ is a solution to the differential equation on $x \in [0, 1]$.

$$-u''(x) + cu'(x) = 0 \quad x \in [0, 1] \quad (2.7)$$

$$u(0) = 0 \quad \text{and} \quad u(1) = 1 \quad (2.8)$$

$$u(x) = \frac{e^{cx} - 1}{e^c - 1} \quad (2.9)$$

A plot of the analytical solution for values of c is depicted in Figure 2.3.1:

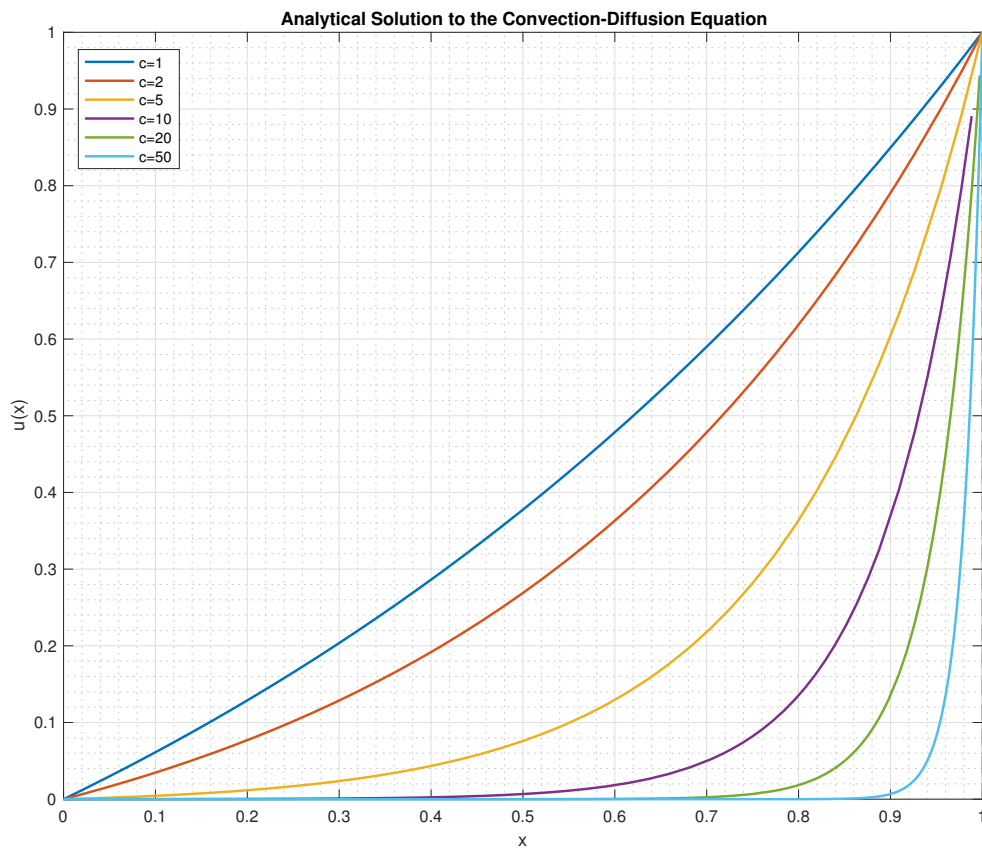


Figure 2.3.1 – Analytical Solution to the Convection-Diffusion Equation for Values of c

2.4 2-D Orthotropic Diffusion Equation

Previously, it was shown that for the two-dimensional second-order linear ordinary differential equation with specified boundary conditions (reproduced below), that $u(x, y)$ is a solution to the differential equation on $(x, y) \in [0, 1] \times [0, 1]$.

$$k^2 \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad x, y \in [0, 1] \times [0, 1] \quad (2.10)$$

$$\begin{aligned} u(x, 0) &= 0 & u(0, y) &= 0 \\ u(x, 1) &= \sin(\pi x) & u(1, y) &= 0 \end{aligned} \quad (2.11)$$

$$u(x, y) = \frac{\sin(\pi x) \sinh(k\pi y)}{\sinh(k\pi)} \quad (2.12)$$

A plot of the analytical solution for values of $k \in \{1, 10\}$ is depicted in Figures 2.4.1 and 2.4.2:

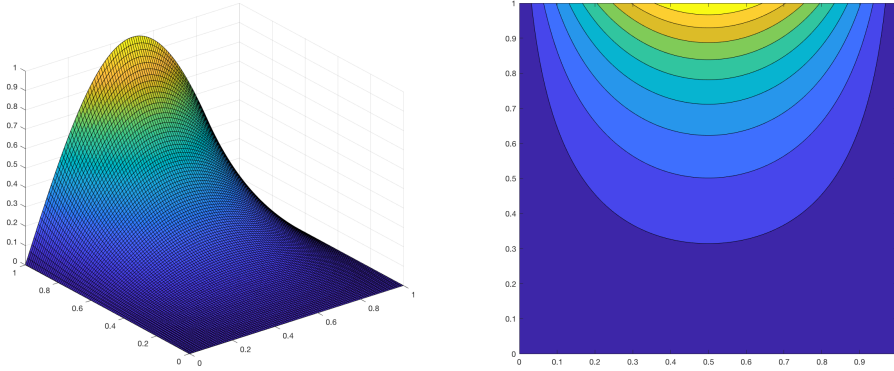


Figure 2.4.1 – Analytical Solution to the 2D Orthotropic Laplacian with $k = 1$

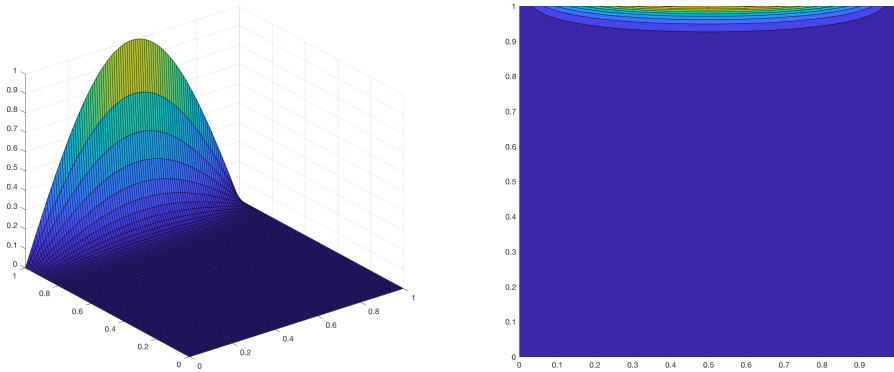


Figure 2.4.2 – Analytical Solution to the 2D Orthotropic Laplacian with $k = 10$

3 Quantities of Interest

3.1 1-D Diffusion Equation

The quantity of interest for the one-dimensional diffusion equation is the derivative at the right boundary, $u'(1)$. Previously, it was shown that the analytical solution of this derivative is given by:

$$u'(1) = 1 - k \coth(k) \quad (3.1)$$

A table of the exact quantity of interest for the tested values of k is included below (for $k \in \{20, 50\}$, error is less than machine epsilon, ϵ):

Table 3.1.1 – Analytical Solution to the Quantity of Interest for the 1-D Diffusion Equation for Values of k

k	$u'(1)$
1	-0.313035
2	-1.074629
5	-4.000454
10	-9.0000000412
20	-19.000000000000000
50	-49.000000000000000

3.2 1-D Harmonic Wave Equation

The quantity of interest for the one-dimensional harmonic wave equation is the derivative at the right boundary, $u'(1)$. Previously, it was shown that the analytical solution of this derivative is given by:

$$u'(1) = 1 - k \cot(k) \quad (3.2)$$

A table of the exact quantity of interest for the tested values of k is included below:

Table 3.2.1 – Analytical Solution to the Quantity of Interest for the 1-D Harmonic Wave Equation for Values of k

k	$u'(1)$
1	0.3579
2	1.9153
5	2.4790
10	-14.4235
20	-7.9399
50	184.8907

3.3 1-D Convection-Diffusion Equation

The quantity of interest for the one-dimensional harmonic wave equation is the derivative at the right boundary, $u'(1)$. Previously, it was shown that the analytical solution of this derivative is given by:

$$u'(1) = \frac{ce^c}{e^c - 1} \quad (3.3)$$

A table of the exact quantity of interest for the tested values of c is included below (for $c = 50$, error is less than machine epsilon, ϵ):

Table 3.3.1 – Analytical Solution to the Quantity of Interest for the 1-D Convection-Diffusion Equation for Values of c

c	$u'(1)$
1	1.58198
2	2.31304
5	5.03392
10	10.00045
20	20.00000004
50	50.00000000

3.4 2-D Orthotropic Diffusion Equation

The quantity of interest for the one-dimensional harmonic wave equation is the first partial derivative with respect to y at the top central boundary, $u_y(\frac{1}{2}, 1)$. Previously, it was shown that the analytical solution of this derivative is given by:

$$u_y(\frac{1}{2}, 1) = k\pi \coth(k\pi) \quad (3.4)$$

A table of the exact quantity of interest for the tested values of k is included below:

Table 3.4.1 – Analytical Solution to the Quantity of Interest for the 2-D Orthotropic Diffusion Equation for Values of k

k	$u_y(\frac{1}{2}, 1)$
1	3.1533
2	6.2832
5	15.7080
10	31.4159
20	62.8319
50	157.0796

4 Numerical Methods

4.1 1-D Diffusion Equation

4.1.1 2nd-Order Central Difference Scheme - Exact Finite Difference Method

Previously, it was shown that for the one-dimensional diffusion equation, the second-order discretization is:

$$(-1) u_{i-1} + (2 + k^2 \Delta x^2) u_i + (-1) u_{i+1} = (k^2 \Delta x^2) x_i \quad (4.1)$$

The solution of the diffusion equation was given earlier by:

$$u(x) = x - \frac{\sinh(kx)}{\sinh(k)} \quad (4.2)$$

Replacing k with \bar{k} to represent the quasi-analytical finite difference coefficient, we arrive at:

$$u(x) = x - \frac{\sinh(\bar{k}x)}{\sinh(\bar{k})} \quad (4.3)$$

Substituting this formula into the discretization and simplifying yields:

$$\bar{k} = \frac{1}{\Delta x} \cosh^{-1} \left(1 + \frac{k^2 \Delta x^2}{2} \right) \quad (4.4)$$

4.1.2 4th-Order Central Difference Scheme - Exact Finite Difference Method

Previously, it was shown that for the one-dimensional diffusion equation, the fourth-order discretization is:

$$\left(-1 + \frac{k^2 \Delta x^2}{12} \right) u_{i-1} + \left(2 + \frac{10k^2 \Delta x^2}{12} \right) u_i + \left(-1 + \frac{k^2 \Delta x^2}{12} \right) u_{i+1} = (k^2 \Delta x^2) x_i \quad (4.5)$$

The solution of the diffusion equation was given earlier by:

$$u(x) = x - \frac{\sinh(kx)}{\sinh(k)} \quad (4.6)$$

Replacing k with \bar{k} to represent the quasi-analytical finite difference coefficient, we arrive at:

$$u(x) = x - \frac{\sinh(\bar{k}x)}{\sinh(\bar{k})} \quad (4.7)$$

Substituting this formula into the discretization and simplifying yields:

$$\bar{k} = \frac{1}{\Delta x} \cosh^{-1} \left(\frac{2 + \frac{10k^2 \Delta x^2}{12}}{-2 \left(-1 + \frac{k^2 \Delta x^2}{12} \right)} \right) \quad (4.8)$$

4.2 1-D Harmonic Wave Equation

4.2.1 2nd-Order Central Difference Scheme - Exact Finite Difference Method

Previously, it was shown that for the one-dimensional harmonic wave equation, the second-order discretization is:

$$(1) u_{i-1} + (-2 + k^2 \Delta x^2) u_i + (1) u_{i+1} = (k^2 \Delta x^2) x_i \quad (4.9)$$

The solution of the harmonic wave equation was given earlier by:

$$u(x) = x - \frac{\sin(kx)}{\sin(k)} \quad (4.10)$$

Replacing k with \bar{k} to represent the quasi-analytical finite difference coefficient, we arrive at:

$$u(x) = x - \frac{\sin(\bar{k}x)}{\sin(\bar{k})} \quad (4.11)$$

Substituting this formula into the discretization and simplifying yields:

$$\bar{k} = \frac{1}{\Delta x} \cosh^{-1} \left(1 - \frac{k^2 \Delta x^2}{2} \right) \quad (4.12)$$

4.2.2 4th-Order Central Difference Scheme - Exact Finite Difference Method

Previously, it was shown that for the one-dimensional harmonic wave equation, the fourth-order discretization is:

$$\left(1 + \frac{k^2 \Delta x^2}{12} \right) u_{i-1} + \left(-2 + \frac{10k^2 \Delta x^2}{12} \right) u_i + \left(1 + \frac{k^2 \Delta x^2}{12} \right) u_{i+1} = (k^2 \Delta x^2) x_i \quad (4.13)$$

The solution of the harmonic wave equation was given earlier by:

$$u(x) = x - \frac{\sin(kx)}{\sin(k)} \quad (4.14)$$

Replacing k with \bar{k} to represent the quasi-analytical finite difference coefficient, we arrive at:

$$u(x) = x - \frac{\sin(\bar{k}x)}{\sin(\bar{k})} \quad (4.15)$$

Substituting this formula into the discretization and simplifying yields:

$$\bar{k} = \frac{1}{\Delta x} \cosh^{-1} \left(\frac{-2 + \frac{10k^2 \Delta x^2}{12}}{-2 \left(1 + \frac{k^2 \Delta x^2}{12} \right)} \right) \quad (4.16)$$

4.3 1-D Convection-Diffusion Equation

4.3.1 2nd-Order Central Difference Scheme - Exact Finite Difference Method

Previously, it was shown that for the one-dimensional convection-diffusion equation, the second-order discretization is:

$$\left(-1 - \frac{c\Delta x}{2}\right)u_{i-1} + (2)u_i + \left(-1 + \frac{c\Delta x}{2}\right)u_{i+1} = 0 \quad (4.17)$$

The solution of the convection-diffusion equation was given earlier by (here we exchange c with k for consistency):

$$u(x) = \frac{e^{kx} - 1}{e^k - 1} \quad (4.18)$$

Replacing k with \bar{k} to represent the quasi-analytical finite difference coefficient, we arrive at:

$$u(x) = \frac{e^{\bar{k}x} - 1}{e^{\bar{k}} - 1} \quad (4.19)$$

Substituting this formula into the discretization and simplifying yields:

$$\bar{k} = \frac{1}{\Delta x} \log \left(\frac{-1 - \frac{k\Delta x^2}{2}}{-1 + \frac{k\Delta x^2}{2}} \right) \quad (4.20)$$

4.3.2 4th-Order Central Difference Scheme - Exact Finite Difference Method

Previously, it was shown that for the one-dimensional convection-diffusion equation, the fourth-order discretization is:

$$\left(-1 - \frac{c\Delta x}{2} - \frac{c^2\Delta x^2}{12}\right)u_{i-1} + \left(2 + \frac{c^2\Delta x^2}{6}\right)u_i + \left(-1 + \frac{c\Delta x}{2} - \frac{c^2\Delta x^2}{12}\right)u_{i+1} = 0 \quad (4.21)$$

The solution of the convection-diffusion equation was given earlier by (here we exchange c with k for consistency):

$$u(x) = \frac{e^{kx} - 1}{e^k - 1} \quad (4.22)$$

Replacing k with \bar{k} to represent the quasi-analytical finite difference coefficient, we arrive at:

$$u(x) = \frac{e^{\bar{k}x} - 1}{e^{\bar{k}} - 1} \quad (4.23)$$

Substituting this formula into the discretization and simplifying yields:

$$\bar{k} = \frac{1}{\Delta x} \log \left(\frac{1 + \frac{k\Delta x}{2} + \frac{k^2\Delta x^2}{12}}{1 - \frac{k\Delta x}{2} + \frac{k^2\Delta x^2}{12}} \right) \quad (4.24)$$

4.4 2-D Orthotropic Diffusion Equation

4.4.1 2nd-Order Central Difference Scheme - Exact Finite Difference Method

Previously, it was shown that for the two-dimensional orthotropic diffusion equation, the second-order discretization on a uniform mesh ($\Delta x = \Delta y$) is the 5-point stencil in x and y :

$$\begin{array}{ccccc} & & u_{i,j+1} & & \\ k^2 u_{i-1,j} & & -2(k^2 + 1)u_{i,j} & & k^2 u_{i+1,j} \\ & & u_{i,j-1} & & \end{array}$$

The solution of the orthotropic diffusion equation was given earlier by:

$$u(x, y) = \frac{\sin(\pi x) \sinh(k\pi y)}{\sinh(k\pi)} \quad (4.25)$$

Replacing k with \bar{k} to represent the quasi-analytical finite difference coefficient, we arrive at:

$$u(x, y) = \frac{\sin(\pi x) \sinh(\bar{k}\pi y)}{\sinh(\bar{k}\pi)} \quad (4.26)$$

Substituting this formula into the discretization and simplifying yields:

$$\bar{k} = \frac{1}{\pi \Delta x} \cosh^{-1} (k^2 + 1 - k^2 \cos(\pi \Delta x)) \quad (4.27)$$

4.4.2 4th-Order Central Difference Scheme - Exact Finite Difference Method

Previously, it was shown that for the two-dimensional orthotropic diffusion equation, the second-order discretization on a uniform mesh ($\Delta x = \Delta y$) is the 9-point stencil in x and y :

$$\begin{array}{ccccc} \frac{1}{12}(k^2 + 1)u_{i-1,j+1} & [1 - \frac{2}{12}(k^2 + 1)]u_{i,j+1} & \frac{1}{12}(k^2 + 1)u_{i+1,j+1} \\ [k^2 - \frac{2}{12}(k^2 + 1)]u_{i-1,j} & [(-2 + \frac{4}{12})(k^2 + 1)]u_{i,j} & [k^2 - \frac{2}{12}(k^2 + 1)]u_{i+1,j} \\ \frac{1}{12}(k^2 + 1)u_{i-1,j-1} & [1 - \frac{2}{12}(k^2 + 1)]u_{i,j-1} & \frac{1}{12}(k^2 + 1)u_{i+1,j-1} \end{array}$$

We will use a simplification of the coefficients in the stencil for brevity:

$$\begin{array}{ccccc} \alpha & \beta & \alpha \\ \gamma & \delta & \gamma \\ \alpha & \beta & \alpha \end{array}$$

The solution of the orthotropic diffusion equation was given earlier by:

$$u(x, y) = \frac{\sin(\pi x) \sinh(k\pi y)}{\sinh(k\pi)} \quad (4.28)$$

Replacing k with \bar{k} to represent the quasi-analytical finite difference coefficient, we arrive at:

$$u(x, y) = \frac{\sin(\pi x) \sinh(\bar{k}\pi y)}{\sinh(\bar{k}\pi)} \quad (4.29)$$

Substituting this formula into the discretization and simplifying yields:

$$\bar{k} = \frac{1}{\pi \Delta x} \cosh^{-1} \left(\frac{-2\gamma \cos(\pi \Delta x) - \delta}{4\alpha \cos(\pi \Delta x) + 2\beta} \right) \quad (4.30)$$

Note: This equation works identically for the second-order stencil as long as the generalized coefficients are used.

5 Convergence Analysis

5.1 Rate of Convergence Derivation

Let the error for a particular mesh size Δx be $E(\Delta x)$:

$$E(\Delta x) = C(\Delta x)^\beta \quad (5.1)$$

Then for a smaller mesh size $\frac{\Delta x}{2}$ we have:

$$E\left(\frac{\Delta x}{2}\right) = C\left(\frac{\Delta x}{2}\right)^\beta \quad (5.2)$$

Dividing the error at each mesh size and taking the logarithm:

$$\frac{E(\Delta x)}{E\left(\frac{\Delta x}{2}\right)} = \frac{C(\Delta x)^\beta}{C\left(\frac{\Delta x}{2}\right)^\beta} = 2^\beta \quad (5.3)$$

$$\log \left[\frac{E(\Delta x)}{E\left(\frac{\Delta x}{2}\right)} \right] = \log(2^\beta) \quad (5.4)$$

$$\log \left[\frac{E(\Delta x)}{E\left(\frac{\Delta x}{2}\right)} \right] = \beta \log(2) \quad (5.5)$$

Rearranging for β and simplifying:

$$\beta = \frac{1}{\log(2)} \left[\log(E(\Delta x)) - \log\left(E\left(\frac{\Delta x}{2}\right)\right) \right] \quad (5.6)$$

Denoting $E_{\Delta x}^* = \log(E(\Delta x))$:

$$\beta = \frac{\mathbf{E}_{\Delta x}^* - \mathbf{E}_{\frac{\Delta x}{2}}^*}{\log(2)} \quad (5.7)$$

5.2 1-D Diffusion Equation

5.2.1 2nd-Order Central Difference Scheme

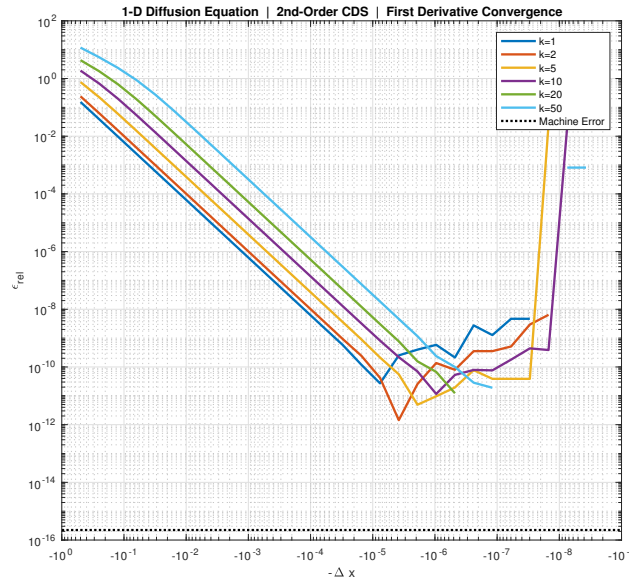


Figure 5.2.1 – Rate of Convergence of $u'(1)$ for 2nd-Order CDS for the 1-D Diffusion Equation for Several Values of k Using Exact FDM

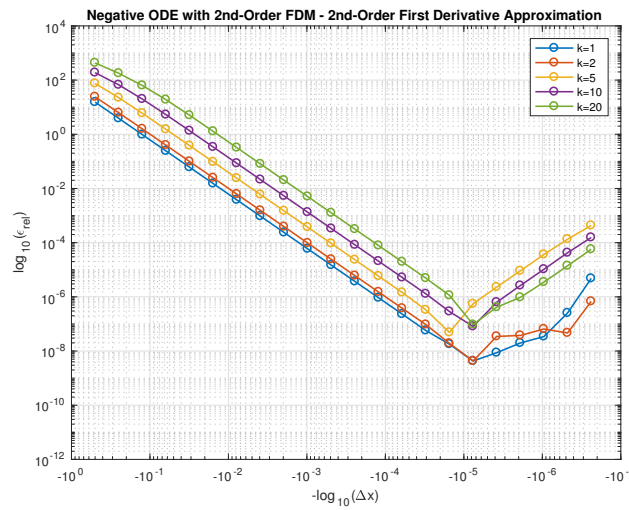


Figure 5.2.2 – Rate of Convergence of $u'(1)$ for 2nd-Order CDS for the 1-D Diffusion Equation for Several Values of k Using Algebraic System FDM

5.2.2 4th-Order Central Difference Scheme

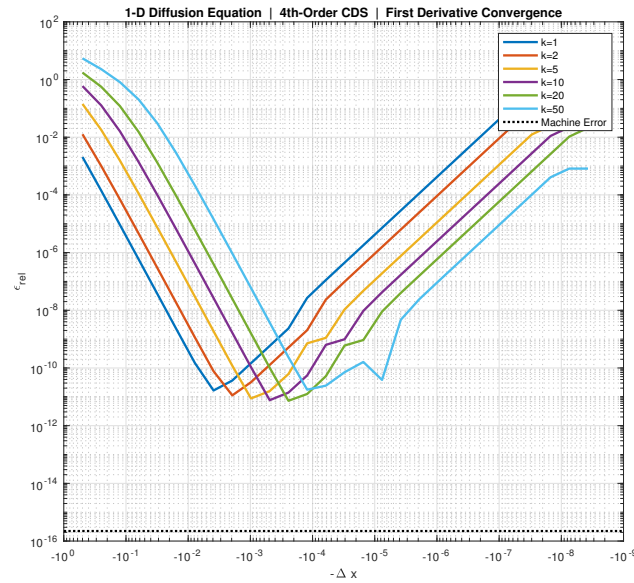


Figure 5.2.3 – Rate of Convergence of $u'(1)$ for 4th-Order CDS for the 1-D Diffusion Equation for Several Values of k Using Exact FDM

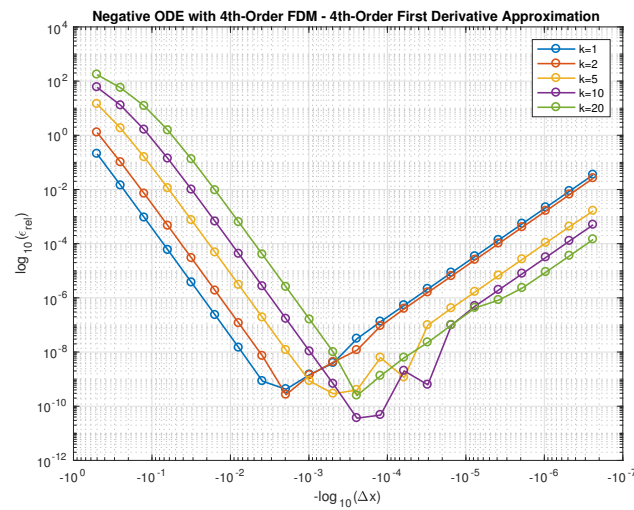


Figure 5.2.4 – Rate of Convergence of $u'(1)$ for 4th-Order CDS for the 1-D Diffusion Equation for Several Values of k Using Algebraic System FDM

5.3 1-D Harmonic Wave Equation

5.3.1 2nd-Order Central Difference Scheme

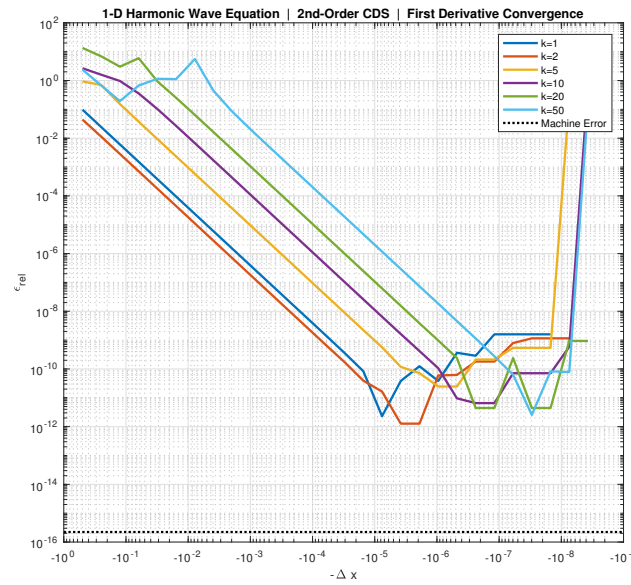


Figure 5.3.1 – Rate of Convergence of $u'(1)$ for 2nd-Order CDS for the 1-D Harmonic Wave Equation for Several Values of k Using Exact FDM

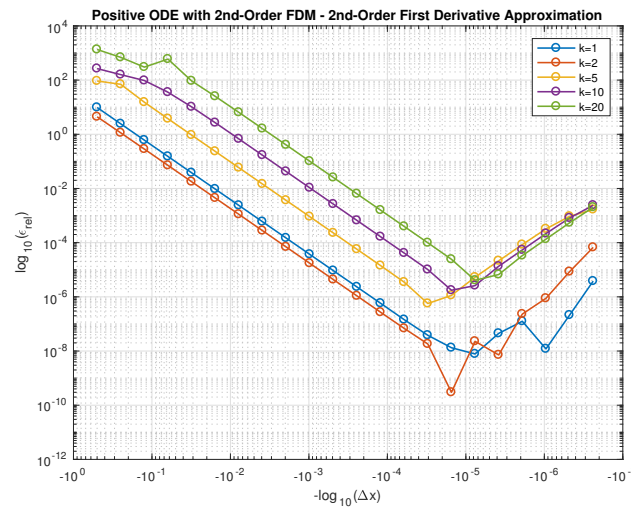


Figure 5.3.2 – Rate of Convergence of $u'(1)$ for 2nd-Order CDS for the 1-D Harmonic Wave Equation for Several Values of k Using Algebraic System FDM

5.3.2 4th-Order Central Difference Scheme

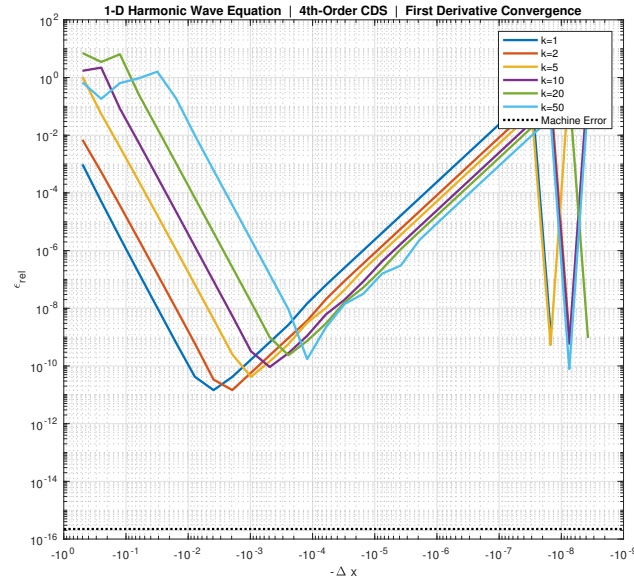


Figure 5.3.3 – Rate of Convergence of $u'(1)$ for 4th-Order CDS for the 1-D Harmonic Wave Equation for Several Values of k Using Exact FDM

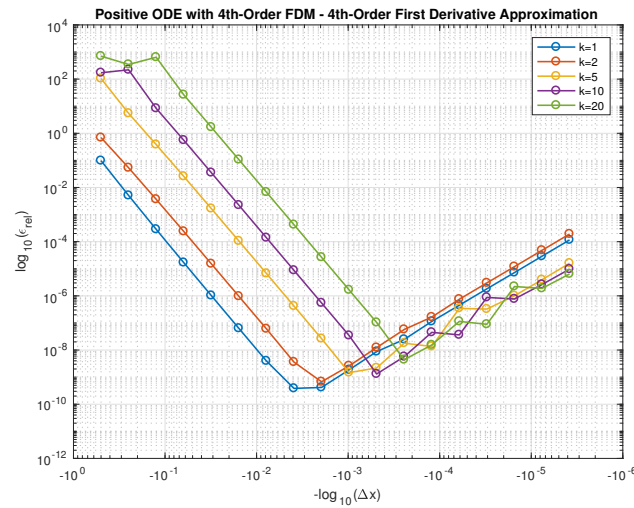


Figure 5.3.4 – Rate of Convergence of $u'(1)$ for 4th-Order CDS for the 1-D Harmonic Wave Equation for Several Values of k Using Algebraic System FDM

5.4 1-D Convection-Diffusion Equation

5.4.1 2nd-Order Central Difference Scheme

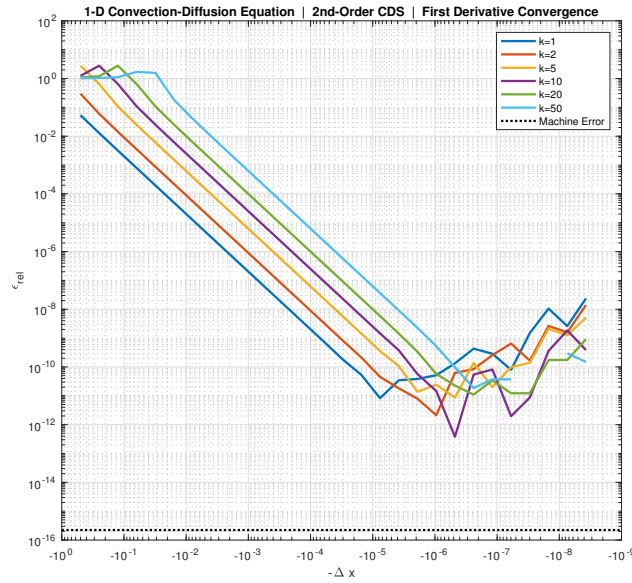


Figure 5.4.1 – Rate of Convergence of $u'(1)$ for 2nd-Order CDS for the 1-D Convection-Diffusion Equation for Several Values of k Using Exact FDM

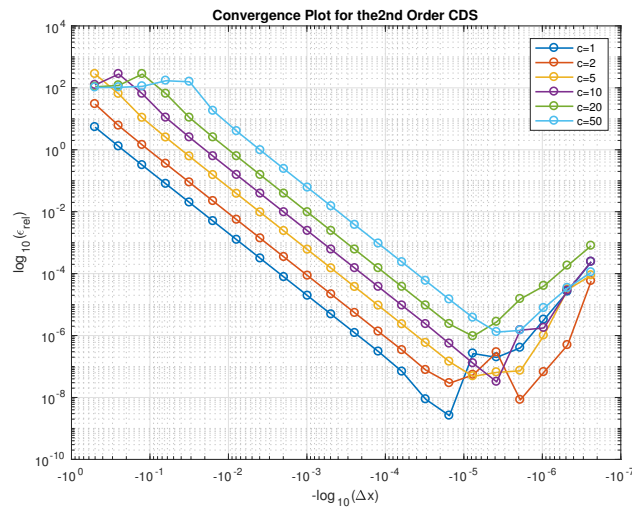


Figure 5.4.2 – Rate of Convergence of $u'(1)$ for 2nd-Order CDS for the 1-D Convection-Diffusion Equation for Several Values of k Using Algebraic System FDM

5.4.2 4th-Order Central Difference Scheme

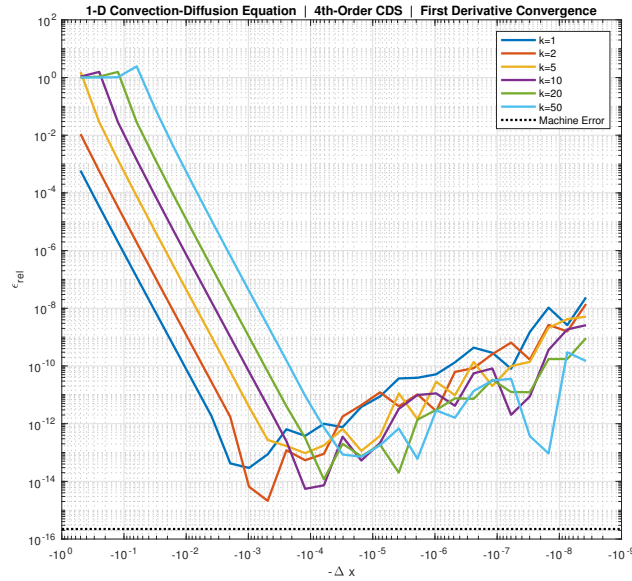


Figure 5.4.3 – Rate of Convergence of $u'(1)$ for 4th-Order CDS for the 1-D Convection-Diffusion Equation for Several Values of k Using Exact FDM

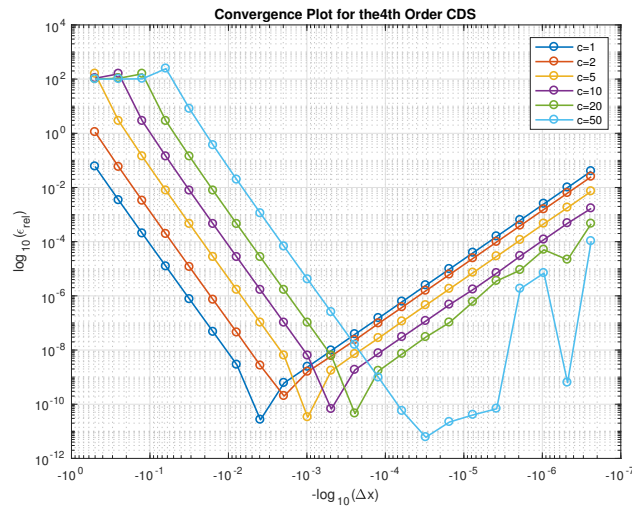


Figure 5.4.4 – Rate of Convergence of $u'(1)$ for 4th-Order CDS for the 1-D Convection-Diffusion Equation for Several Values of k Using Algebraic System FDM

5.5 2-D Orthotropic Diffusion Equation

5.5.1 2nd-Order Central Difference Scheme

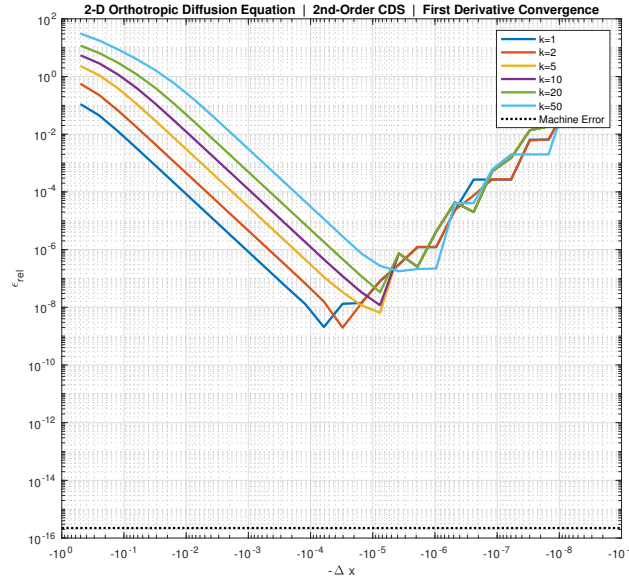


Figure 5.5.1 – Rate of Convergence of $u_y(\frac{1}{2}, 1)$ for 2nd-Order CDS for the 2-D Orthotropic Diffusion Equation for Several Values of k Using Exact FDM

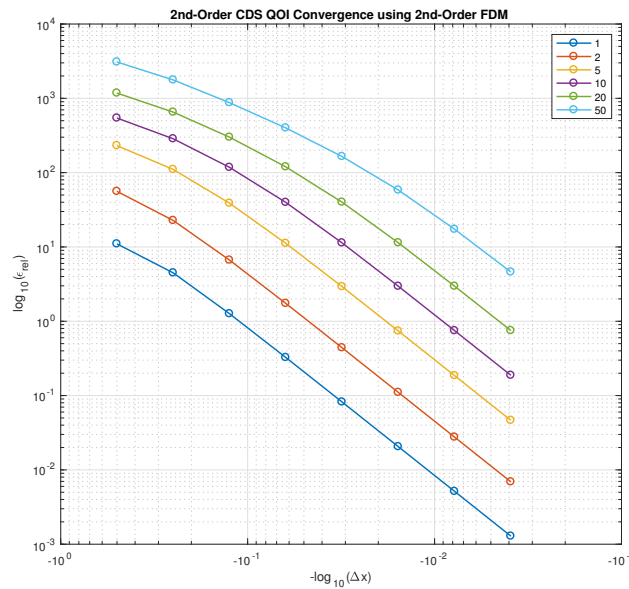


Figure 5.5.2 – Rate of Convergence of $u_y(\frac{1}{2}, 1)$ for 2nd-Order CDS for the 2-D Orthotropic Diffusion Equation for Several Values of k Using Algebraic System FDM

5.5.2 4th-Order Central Difference Scheme

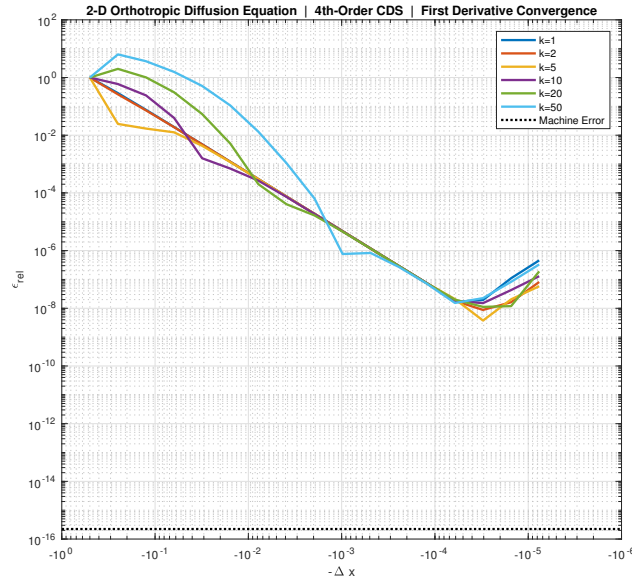


Figure 5.5.3 – Rate of Convergence of $u_y(\frac{1}{2}, 1)$ for 4th-Order CDS for the 2-D Orthotropic Diffusion Equation for Several Values of k Using Exact FDM

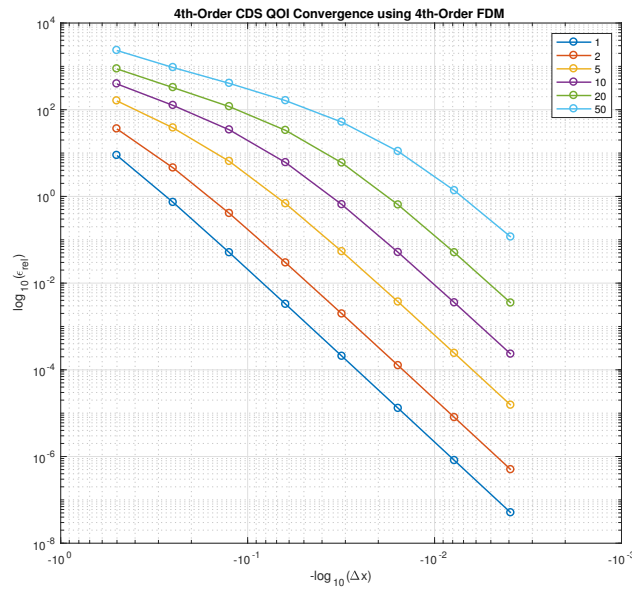


Figure 5.5.4 – Rate of Convergence of $u_y(\frac{1}{2}, 1)$ for 4th-Order CDS for the 2-D Orthotropic Diffusion Equation for Several Values of k Using Algebraic System FDM

6 Discussion

In all cases, it is easy to see that the exact finite difference method solution approach is much simpler than the linear algebraic solution approach. The latter requires the simultaneous solution of all grid points (in 2-D and 3-D this realization becomes much more computationally intensive), while the former allows the calculation of each grid point individually, once the quasi-analytical coefficient is determined. In general, this means that the exact finite difference method approach allows for significantly faster computation times, smaller computational memory allocation, and reduced roundoff error.

If a solution in the exact finite difference method exists for a particular differential equation, then variants also exist in similar form, as shown with the 2-D orthotropic diffusion equation.

For the 1-D diffusion equation, meshes were computed down to $\Delta x = 2^{-28}$. The error in the 2nd-order scheme is roughly lower than that computed with the algebraic form of the finite difference equations, but the solution was able to be computed for much smaller mesh sizes. The error in the 4th-order scheme is roughly lower than that computed with the algebraic form of the finite difference equations, but again the solution was able to be computed for much smaller mesh sizes. For both methods, the asymptotic rate of convergence was strictly the order of the method ($\beta = 2, \beta = 4$).

For the 1-D harmonic wave equation, meshes were computed down to $\Delta x = 2^{-28}$. The error in the 2nd-order scheme is much lower than that computed with the algebraic form of the finite difference equations and approaches 10^{-12} . The error in the 4th-order scheme is roughly lower than that computed with the algebraic form of the finite difference equations. In both cases, the solution was able to be computed for much smaller mesh sizes. For both methods, the asymptotic rate of convergence was strictly the order of the method ($\beta = 2, \beta = 4$).

For the 1-D convection-diffusion equation, meshes were computed down to $\Delta x = 2^{-28}$. The error in the 2nd-order scheme is much lower than that computed with the algebraic form of the finite difference equations and approaches 10^{-12} . The error in the 4th-order scheme is much lower than that computed with the algebraic form of the finite difference equations and approaches 10^{-15} – near machine epsilon, ϵ ! In both cases, the solution was able to be computed for much smaller mesh sizes. For both methods, the asymptotic rate of convergence was strictly the order of the method ($\beta = 2, \beta = 4$).

For the 2-D orthotropic diffusion equation, meshes were computed down to $\Delta x = 2^{-28}$ for the 2nd-order scheme and $\Delta x = 2^{-17}$. The error in the 2nd-order scheme advances significantly below that computed with the algebraic form of the finite difference equations and approaches 10^{-9} . With maximum computing power for the algebraic form of the finite difference equations, computations could only be computed down to $\Delta x = 2^{-8}$. The error in the 4th-order scheme is roughly lower than that computed with the algebraic form of the finite difference equations and approaches 10^{-9} . For both methods, the asymptotic rate of convergence was strictly the order of the method ($\beta = 2, \beta = 4$).

In general, the difference between the convergence of the equations is due in large part to their form and stability. The 1-D diffusion equation and the 1-D harmonic wave equation have largely different convergence properties – one solution is hyperbolic trigonometric and the other solution is purely trigonometric. Additionally, the 1-D convection-diffusion equation implemented with a central difference scheme is unstable for small mesh sizes, but extremely stable once the Peclet condition is satisfied.

There are significant changes from the 1-D equations to the 2-D orthotropic diffusion equation. It is seen that for the second-order method, the derivative can be extracted *explicitly*, but for the fourth-order method, the derivative must be extracted *implicitly*. This shows some of the limitations of this method. The aim of the exact FDM is to avoid the extensive computation resources of the algebraic solution method, and the higher-order method makes this increasingly difficult. Regardless, the results were still able to be extended to much smaller mesh sizes with minimal computational effort.

In general, convergence rates using the exact finite difference method were equivalent to the convergence rates using the algebraic finite difference method, but were achieved at a much lower cost.

A MATLAB Code

A.1 fp_1d_diffusion.m

```
clear all; close all; clc

meshOrder = 1:28;
meshDx = (1/2).^meshOrder;

order = 2;

ufdm = sparse([]);
uexact = sparse([]);

index = 0;

for k = [1 2 5 10 20 50]

    index = index + 1;

    for dx = meshDx

        clear ufdm

        if order == 2

            kbar = 1 / dx * acosh(1 + k^2*dx^2/2);

            %           x = 0:dx:1;
            %           ufdm(-log(dx)/log(2), 1:1/dx+1) = x - sinh(kbar*x)/sinh(kbar);
            %           uxfdm = -ufdm(-log(dx)/log(2), end-1) / dx - dx * k^2 / 2;

            ufdm = 1-dx - sinh(kbar*(1-dx))/sinh(kbar);
            uxfdm = -ufdm / dx - dx * k^2 / 2;

            uxexact = 1 - k*coth(k);

        elseif order == 4

            kbar = 1 / dx * acosh((2+10*k^2*dx^2/12) / (-2*(-1+k^2*dx^2/12)));

            %           x = 0:dx:1;
            %           ufdm(-log(dx)/log(2), 1:1/dx+1) = x - sinh(kbar*x)/sinh(kbar);
            %           uxfdm = 1 / (1 + dx^2 * k^2 / 6) * (-ufdm(-log(dx)/log(2), end-1) / dx - dx * k^2 / 2 + ...
            %                   dx^2 * k^2 / 6 - dx^3 * k^4 / 24);

            ufdm = 1-dx - sinh(kbar*(1-dx))/sinh(kbar);
            uxfdm = 1 / (1 + dx^2 * k^2 / 6) * (-ufdm / dx - dx * k^2 / 2 + ...
            dx^2 * k^2 / 6 - dx^3 * k^4 / 24);

            uxexact = 1 - k*coth(k);

        end

        err(-log(dx)/log(2)) = abs((uxfdm - uxexact) / uxexact);

    end

    uxerr(index, :) = err;

end
```

```

end

figure
hold on
set(gcf, 'Position', [1 1 624 550])

for i = 1:index
plot(-meshDx, uxerr(i, :), 'linewidth', 2);
end
plot([-10^0 -10^-9], [eps eps], 'k:', 'linewidth', 2)
legend('k=1', 'k=2', 'k=5', 'k=10', 'k=20', 'k=50', 'Machine Error')
xlabel('\Delta x'); ylabel('\epsilon_{rel}')
title('1-D Diffusion Equation | First Derivative Convergence')

set(gca, 'xscale', 'log')
set(gca, 'yscale', 'log')
grid on; grid minor;
box on;

```

A.2 fp_1d_wave.m

```

clear all; close all; clc

meshOrder = 1:28;
meshDx = (1/2).^meshOrder;

order = 2;

ufdm = sparse([]);
uexact = sparse([]);

index = 0;

for k = [1 2 5 10 20 50]

index = index + 1;

for dx = meshDx

clear ufdm

if order == 2

kbar = 1 / dx * acos(1 - k^2*dx^2/2);

%           x = 0:dx:1;
%           ufdm(-log(dx)/log(2), 1:1/dx+1) = x - sin(kbar*x)/sin(kbar);
%           uxfdm = -ufdm(-log(dx)/log(2), end-1) / dx + dx * k^2 / 2;

ufdm = 1-dx - sin(kbar*(1-dx))/sin(kbar);
uxfdm = -ufdm / dx + dx * k^2 / 2;

uxexact = 1 - k*cot(k);

elseif order == 4

kbar = 1 / dx * acos((-2+10*k^2*dx^2/12) / (-2*(1+k^2*dx^2/12)));

```

```

%           x = 0:dx:1;
%           ufdm(-log(dx)/log(2), 1:1/dx+1) = x - sin(kbar*x)/sin(kbar);
%           uxfdm = 1 / (1 - dx^2 * k^2 / 6) * (-ufdm(-log(dx)/log(2), end-1) / dx + dx * k^2 / 2 - ...
%               dx^2 * k^2 / 6 - dx^3 * k^4 / 24);

ufdm = 1-dx - sin(kbar*(1-dx))/sin(kbar);
uxfdm = 1 / (1 - dx^2 * k^2 / 6) * (-ufdm / dx + dx * k^2 / 2 - ...
dx^2 * k^2 / 6 - dx^3 * k^4 / 24);

uxexact = 1 - k*cot(k);

end

err(-log(dx)/log(2)) = abs((uxfdm - uxexact) / uxexact);

end

uxerr(index, :) = err;

end

figure
hold on
set(gcf, 'Position', [1 1 624 550])

for i = 1:index
plot(-meshDx, uxerr(i, :), 'linewidth', 2);
end
plot([-10^0 -10^-9], [eps eps], 'k:', 'linewidth', 2)
legend('k=1', 'k=2', 'k=5', 'k=10', 'k=20', 'k=50', 'Machine Error')
xlabel('-\Delta x'); ylabel('\epsilon_{rel}')
title('1-D Harmonic Wave Equation | First Derivative Convergence')

set(gca, 'xscale', 'log')
set(gca, 'yscale', 'log')
grid on; grid minor;
box on;

```

A.3 fp_1d_convection.m

```

clear all; close all; clc

meshOrder = 1:28;
meshDx = (1/2).^meshOrder;

order = 4;

ufdm = sparse([]);
uexact = sparse([]);

index = 0;

for k = [1 2 5 10 20 50]

index = index + 1;

for dx = meshDx

```

```

clear ufdm

if order == 2

kbar = 1 / dx * log((-2-k*dx)/(-2+k*dx));

%           x = 0:dx:1;
%           ufdm(-log(dx)/log(2), 1:1/dx+1) = (exp(kbar*x) - 1) / (exp(kbar) - 1);
%           uxfdm = (1 - k*dx/2)^-1 * (1 - ufdm(-log(dx)/log(2), end-1)) / dx;

ufdm = (exp(kbar*(1-dx)) - 1) / (exp(kbar) - 1);
uxfdm = (1 - k*dx/2)^-1 * (1 - ufdm) / dx;

uxexact = k*exp(k) / (exp(k) - 1);

elseif order == 4

kbar = 1 / dx * log((-1 - k*dx/2 - k^2*dx^2/12)/(-1 + k*dx/2 - k^2*dx^2/12));

%           x = 0:dx:1;
%           ufdm(-log(dx)/log(2), 1:1/dx+1) = (exp(kbar*x) - 1) / (exp(kbar) - 1);
%           uxfdm = (1 - k*dx/2 + k^2*dx^2/6 - k^3*dx^3/24)^-1 * ...
%           (1 - ufdm(-log(dx)/log(2), end-1)) / dx;

ufdm = (exp(kbar*(1-dx)) - 1) / (exp(kbar) - 1);
uxfdm = (1 - k*dx/2 + k^2*dx^2/6 - k^3*dx^3/24)^-1 * ...
(1 - ufdm) / dx;

uxexact = k*exp(k) / (exp(k) - 1);

end

err(-log(dx)/log(2)) = abs((uxfdm - uxexact) / uxexact);

end

uxerr(index, :) = err;

end

figure
hold on
set(gcf, 'Position', [1 1 624 550])

for i = 1:index
plot(-meshDx, uxerr(i, :), 'linewidth', 2);
end
plot([-10^0 -10^-9], [eps eps], 'k:', 'linewidth', 2)
legend('k=1', 'k=2', 'k=5', 'k=10', 'k=20', 'k=50', 'Machine Error')
xlabel('-\Delta x'); ylabel('\epsilon_{rel}')
title('1-D Convection-Diffusion Equation | First Derivative Convergence')

set(gca, 'xscale', 'log')
set(gca, 'yscale', 'log')
grid on; grid minor;
box on;

```

A.4 fp_2d_diffusion.m

```
clear all; close all; clc

meshOrder    = 1:28;
meshDx        = (1/2).^meshOrder;

order = 2;

ufdm = sparse([]);
uexact = sparse([]);

index = 0;

for k = [1 2 5 10 20 50]

    index = index + 1;

    for dx = meshDx

        if order == 2

            kbar = 1/(pi*dx) * acosh(k^2+1-k^2*cos(pi*dx));

            ufdmcenter = sin(pi*1/2)*sinh(kbar*pi*1)/sinh(kbar*pi);
            ufdmright = sin(pi*(1/2+dx))*sinh(kbar*pi*1)/sinh(kbar*pi);
            ufdmleft = sin(pi*(1/2-dx))*sinh(kbar*pi*1)/sinh(kbar*pi);
            ufdmlower = sin(pi*1/2)*sinh(kbar*pi*(1-dx))/sinh(kbar*pi);

            uyfdm = 1 / dx * ((1+k^2)*ufdmcenter - ufdmlower - k^2/2*ufdmleft - k^2/2*ufdmright);
            uyexact = k*pi*coth(k*pi);

        elseif order == 4

            alpha = 1/12*(k^2+1);
            beta = 1-2/12*(k^2+1);
            gamma = k^2-2/12*(k^2+1);
            delta = (-2+4/12)*(k^2+1);

            kbar = 1/(pi*dx) * acosh((-2*gamma*cos(pi*dx)-delta)/(4*alpha*cos(pi*dx)+2*beta));

            n = 1/dx;

            alpha = -k^2/6;
            beta = 1 + k^2/3;
            C = sparse(gallery('tridiag', n+1, alpha, beta, alpha));
            C(1, 1) = 1;
            C(1, 2) = 0;
            C(end, end-1) = 0;
            C(end, end) = 1;

            d = sparse(n+1, 1);

            for i = 2:n
                ufdmcenter = sin(pi*i*dx)*sinh(kbar*pi*1)/sinh(kbar*pi);
                ufdmlower = sin(pi*i*dx)*sinh(kbar*pi*(1-dx))/sinh(kbar*pi);
                d(i) = ufdmcenter/dx - ufdmlower/dx + (k^2*pi^2*dx)/2*ufdmcenter + (k^4*pi^4*dx^3)/24*ufdmcenter;
            end

            uy = C\d;
```



```

uyfdm = uy(n/2+1);
uyexact = k*pi*coth(k*pi);

end

err(-log(dx)/log(2)) = abs((uyfdm - uyexact) / uyexact);

end

uxerr(index, :) = err;

end

figure
hold on
set(gcf, 'Position', [1 1 624 550])

for i = 1:index
plot(-meshDx, uxerr(i, :), 'linewidth', 2);
end
plot([-10^0 -10^-9], [eps eps], 'k:', 'linewidth', 2)
legend('k=1', 'k=2', 'k=5', 'k=10', 'k=20', 'k=50', 'Machine Error')
xlabel('-\Delta x'); ylabel('\epsilon_{rel}')
title('2-D Orthotropic Diffusion Equation | First Derivative Convergence')

set(gca, 'xscale', 'log')
set(gca, 'yscale', 'log')
grid on; grid minor;
box on;

```