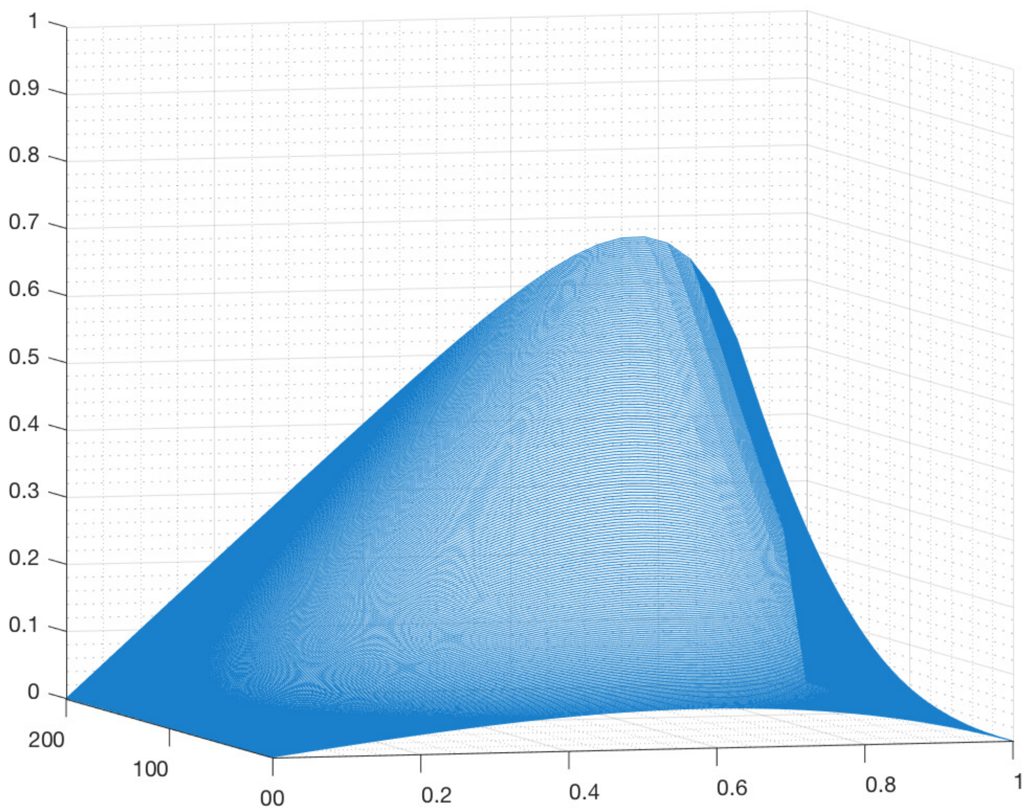
AERO 430 – Numerical Simulation

Second-Order Linear Ordinary Differential Equation Boundary-Value Problem

Ross B. Alexander



Contents

1 Model Problem

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

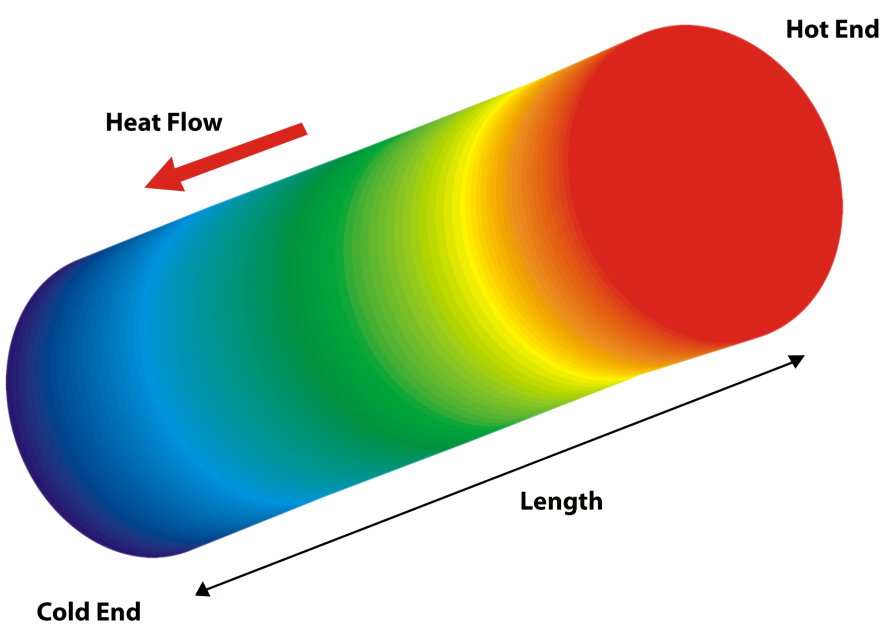
- the second-order linear ordinary differential equation:

$$\pm u''(x) + k^2 u(x) = k^2 x \qquad x \in (0,1) \tag{1.1}$$

- the boundary conditions:

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

The model second-order linear ordinary differential equation is given with a plus-or-minus sign, as the results of the solution of each second-order linear ordinary differential equation are similar. The physical model of the positive case is that of the amplitude of standing waves for uniaxial forced vibration of a bar. The physical model for the negative case 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.



2 Analytical Solution

2.1 Positive ODE

The following equation is the positive second-order linear ordinary differential equation (ODE).

$$u''(x) + k^2 u(x) = k^2 x \tag{2.1}$$

2.1.1 Homogeneous Solution

Let the homogeneous solution to the positive ODE be defined as $u_h(x)$. Then, $u_h(x)$ must satisfy the following homogeneous ODE.

$$u_h''(x) + k^2 u_h(x) = 0 \tag{2.2}$$

The solution of the homogeneous ODE is assumed to be of the form:

$$u_h(x) = e^{\lambda x} \tag{2.3}$$

Taking the second-derivative of $u_h(x)$, substituting the second-derivative into the homogeneous ODE, and reducing the equation yields the **characteristic equation**.

$$u_h''(x) = \lambda^2 e^{\lambda x} \tag{2.4}$$

$$\lambda^2 e^{\lambda x} + k^2 e^{\lambda x} = 0 \tag{2.5}$$

$$\boldsymbol{\lambda^2 + k^2 = 0} \tag{2.6}$$

Solving for λ yields:

$$\lambda = \pm ik \tag{2.7}$$

The homogenous solution $u_h(x)$ is then:

$$u_h(x) = \alpha e^{ikx} + \beta e^{-ikx} \tag{2.8}$$

Making a transformation with the following relations, a more sophistocated solution can be developed:

$$\gamma = \frac{\alpha + \beta}{2} \quad \text{and} \quad \delta = i\frac{\alpha - \beta}{2} \tag{2.9}$$

$$u_h(x) = \gamma \frac{e^{ikx} + e^{-ikx}}{2} + \delta \frac{e^{ikx} - e^{-ikx}}{2i} \tag{2.10}$$

$$\mathbf{u_h(x) = \gamma \cos(kx) + \delta \sin(kx)} \tag{2.11}$$

2.1.2 Particular Solution

Let the particular solution to the positive ODE be defined as $u_p(x)$. Then, $u_p(x)$ must satisfy the ODE:

$$u_p''(x) + k^2 u_p(x) = k^2 x \tag{2.12}$$

The second-derivative of $u_p(x)$, $u_p''(x)$, is assumed to be zero, and thus yields the particular solution $u_p(x)$:

$$k^2 u_p(x) = k^2 x \tag{2.13}$$

$$\mathbf{u_p(x) = x} \tag{2.14}$$

2.1.3 Boundary Conditions

Given that $u_h(x)$ is a solution to the homogeneous ODE and $u_p(x)$ is a solution to the ODE, then the combination of $u_h(x)$ and $u_p(x)$ is also a solution to the ODE.

$$u(x) = u_h(x) + u_p(x) \tag{2.15}$$

$$u(x) = \gamma cos(kx) + \delta sin(kx) + x \tag{2.16}$$

The boundary conditions for the model problem are:

$$u(0) = 0 \quad \text{and} \quad u(1) = 0 \tag{2.17}$$

Applying the first boundary condition, $u(0) = 0$, we get that $\gamma = 0$:

$$u(0) = 0 = \gamma cos(0) + \delta sin(0) + 0 \tag{2.18}$$

$$\gamma = 0 \tag{2.19}$$

Applying the second boundary condition, $u(1) = 0$, we get that $\delta = \frac{-1}{sin(k)}$:

$$u(1) = 0 = \delta sin(k) + 1 \tag{2.20}$$

$$\delta = \frac{-1}{sin(k)} \tag{2.21}$$

2.1.4 Analytical Solution

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

$$u''(x) + k^2 u(x) = k^2 x \qquad x \in (0, 1) \tag{2.22}$$

$$u(0) = 0 \quad \text{and} \quad u(1) = 0 \tag{2.23}$$

$$\mathbf{u(x) = x - \frac{sin(kx)}{sin(k)}} \tag{2.24}$$

2.2 Negative ODE

The following equation is the negative second-order linear ordinary differential equation (ODE).

$$-u''(x) + k^2 u(x) = k^2 x \tag{2.25}$$

2.2.1 Homogeneous Solution

Let the homogeneous solution to the negative ODE be defined as $u_h(x)$. Then, $u_h(x)$ must satisfy the following homogeneous ODE.

$$-u_h''(x) + k^2 u_h(x) = 0 \tag{2.26}$$

The solution of the homogeneous ODE is assumed to be of the form:

$$u_h(x) = e^{\lambda x} \tag{2.27}$$

Taking the second-derivative of $u_h(x)$, substituting the second-derivative into the homogeneous ODE, and reducing the equation yields the **characteristic equation**.

$$u_h'' = \lambda^2 e^{\lambda x} \tag{2.28}$$

$$-\lambda^2 e^{\lambda x} + k^2 e^{\lambda x} = 0 \tag{2.29}$$

$$\boldsymbol{-\lambda^2 + k^2 = 0} \tag{2.30}$$

Solving for λ yields:

$$\lambda = \pm k \tag{2.31}$$

The homogenous solution $u_h(x)$ is then:

$$u_h(x) = \alpha e^{kx} + \beta e^{-kx} \tag{2.32}$$

By making a transformation with the following relations, a more sophistocated solution can be developed:

$$\gamma = \frac{\alpha + \beta}{2} \quad \text{and} \quad \delta = \frac{\alpha - \beta}{2} \tag{2.33}$$

$$u_h(x) = \gamma \frac{e^{kx} + e^{-kx}}{2} + \delta \frac{e^{kx} - e^{-kx}}{2} \tag{2.34}$$

$$\mathbf{u_h(x)} = \gamma \mathbf{cosh(kx)} + \delta \mathbf{sinh(kx)} \tag{2.35}$$

2.2.2 Particular Solution

Let the particular solution to the negative ODE be defined as $u_p(x)$. Then, $u_p(x)$ must satisfy the ODE:

$$-u_p''(x) + k^2 u_p(x) = k^2 x \tag{2.36}$$

The second-derivative of $u_p(x)$, $u_p''(x)$, is assumed to be zero, and thus yields the particular solution $u_p(x)$:

$$k^2 u_p(x) = k^2 x \tag{2.37}$$

$$\mathbf{u_p(x) = x} \tag{2.38}$$

2.2.3 Boundary Conditions

Given that $u_h(x)$ is a solution to the homogeneous ODE and $u_p(x)$ is a solution to the ODE, then the combination of $u_h(x)$ and $u_p(x)$ is also a solution to the ODE.

$$u(x) = u_h(x) + u_p(x) \tag{2.39}$$

$$u(x) = \gamma cosh(kx) + \delta sinh(kx) + x \tag{2.40}$$

The boundary conditions for the model problem are:

$$u(0) = 0 \quad \text{and} \quad u(1) = 0 \tag{2.41}$$

Applying the first boundary condition, $u(0) = 0$, we get that $\gamma = 0$:

$$u(0) = 0 = \gamma cosh(0) + \delta sinh(0) + 0 \tag{2.42}$$

$$\gamma = 0 \tag{2.43}$$

Applying the second boundary condition, $u(1) = 0$, we get that $\delta = \frac{-1}{sinh(k)}$:

$$u(1) = 0 = \delta sinh(k) + 1 \tag{2.44}$$

$$\delta = \frac{-1}{sinh(k)} \tag{2.45}$$

2.2.4 Analytical Solution

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

$$-u''(x) + k^2 u(x) = k^2 x \qquad x \in (0, 1) \tag{2.46}$$

$$u(0) = 0 \quad \text{and} \quad u(1) = 0 \tag{2.47}$$

$$\mathbf{u(x) = x - \frac{sinh(kx)}{sinh(k)}} \tag{2.48}$$

3 Numerical Methods

3.1 Second-Order Second-Derivative Finite Difference Method

3.1.1 Derivation

Developing the Taylor series for $u(x)$ in the vicinity of $x = i$:

$$u_{i-1} = u_i - \Delta x u_i' + \frac{\Delta x^2}{2} u_i'' - \frac{\Delta x^3}{6} u_i^{(3)} + \frac{\Delta x^4}{24} u_i^{(4)} + \mathcal{O}(\Delta x^5) \tag{3.1}$$

$$u_{i+1} = u_i + \Delta x u_i' + \frac{\Delta x^2}{2} u_i'' + \frac{\Delta x^3}{6} u_i^{(3)} + \frac{\Delta x^4}{24} u_i^{(4)} + \mathcal{O}(\Delta x^5) \tag{3.2}$$

Adding the Taylor series for u_{i-1} and u_{i+1} and canceling terms:

$$u_{i+1} + u_{i-1} = 2u_i + \Delta x^2 u_i'' + \mathcal{O}(\Delta x^4) \tag{3.3}$$

Rearranging terms to solve for u_i'':

$$u_i'' = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} + \mathcal{O}(\Delta x^2) \tag{3.4}$$

From this specific second-derivative formulation using the finite difference method, the approximation can be observed to be second-order ($\mathcal{O}(\Delta x^2)$).

3.1.2 Results – Positive ODE

Figure 3.1.1 – Positive ODE – 2nd-Order FDM for $k = 1$

Figure 3.1.2 – Positive ODE – 2nd-Order FDM for $k = 2$

Figure 3.1.3 – Positive ODE – 2nd-Order FDM for $k = 5$

Figure 3.1.4 – Positive ODE – 2nd-Order FDM for $k = 10$

Figure 3.1.5 – Positive ODE – 2nd-Order FDM for $k = 20$

3.1.3 Discussion – Positive ODE

For the positive ODE, the analytical solution oscillates and generally increases in the amplitude of the oscillations as k increases. Like expected, as mesh size is decreased, the approximation of the solution to the model problem approaches the analytical solution. At high values of k ($k \geq 20$), the mesh size $\Delta x = (1/2)^6$ begins to be insufficient to resolve the solution as the number of complete cycles on the domain increases. This insufficiency appears to propagate for higher values of k as overall resolution is poorer for all mesh sizes. It is likely that a fourth-order approximation of the second-derivative term would yield better agreement with the analytical solution.

3.1.4 Results – Negative ODE

Figure 3.1.6 – Negative ODE – 2nd-Order FDM for $k = 1$

Figure 3.1.7 – Negative ODE – 2nd-Order FDM for $k = 2$

Figure 3.1.8 – Negative ODE – 2nd-Order FDM for $k = 5$

Figure 3.1.9 – Negative ODE – 2nd-Order FDM for $k = 10$

Figure 3.1.10 – Negative ODE – 2nd-Order FDM for $k = 20$

3.1.5 Discussion – Negative ODE

For the negative ODE, the analytical solution asymptotically approaches the line $y = x$ as k increases. Like expected, as mesh size is decreased, the approximation of the solution to the model problem approaches the analytical solution. Unlike the positive ODE, at high values of k ($k \geq 20$), the mesh size $\Delta x = (1/2)^6$ is sufficient to resolve the solution as difference between different values of k is increasingly negligible. It is likely that a fourth-order approximation of the second-derivative term would yield better agreement with and quicker convergence to the analytical solution.

4 Convergence Analysis

4.1 First-Order First-Derivative Finite Difference Method

4.1.1 Derivation

Developing the Taylor series for $u(x)$ in the vicinity of $x = 1$:

$$u_{N-1} = u_N - \Delta x u'_N + \frac{\Delta x^2}{2} u''_N + \mathcal{O}(\Delta x^3) \tag{4.1}$$

Rearranging terms to solve for u'_N:

$$u'_N = \frac{u_N - u_{N-1}}{\Delta x} + \mathcal{O}(\Delta x) \tag{4.2}$$

Switching to a compact notation where $u_N = u_N$, $u_{N-1} = u_{N-1}$, etc.:

$$u'_N = \frac{u_N - u_{N-1}}{\Delta x} + \mathcal{O}(\Delta x) \tag{4.3}$$

Applying the boundary condition $u(1) = u_N = 0$:

$$u'_N = \frac{-u_{N-1}}{\Delta x} + \mathcal{O}(\Delta x) \tag{4.4}$$

From this specific first-derivative formulation at the boundary $x = 1$ using the finite difference method, the approximation can be observed to be first-order ($\mathcal{O}(\Delta x)$).

4.1.2 Results

Figure 4.1.1 – Positive ODE – 2nd-Order FDM with 1st-Order First-Derivative Approximation

Figure 4.1.2 – Negative ODE – 2nd-Order FDM with 1st-Order First-Derivative Approximation

4.1.3 Discussion

The positive ODE appears to be less stable than the negative ODE – a conclusion drawn from the significant deviation from an ideal rate of convergence. This is likely because the solution to the positive ODE can contain numerous oscillations, while the solution to the negative ODE is a single smooth oscillation for every solution. Therefore, the well-behavedness of an ODE solution is a factor in the rate of convergence of a particular finite difference method.

As the above figures indicate, the logarithm of the error decreases roughly at a rate of equal to the negative logarithm of the mesh size. Thus, that the approximation used is first-order accurate and has a rate of convergence of 1.

4.2 Second-Order First-Derivative Finite Difference Method

4.2.1 Derivation

Developing the Taylor series for $u(x)$ in the vicinity of $x = 1$:

$$u_{N-1} = u_N - \Delta x u'_N + \frac{\Delta x^2}{2} u''_N + \mathcal{O}(\Delta x^3) \tag{4.5}$$

Rearranging terms to solve for u'_N, but leaving the second-derivative term:

$$u'_N = \frac{u_N - u_{N-1}}{\Delta x} + \frac{\Delta x}{2} u''_N + \mathcal{O}(\Delta x^2) \tag{4.6}$$

Returning to the differential equation, rearranging for the second derivative, and evaluating the differential equation at $x = 1$ with corresponding boundary condition $u(1) = u_N = 0$:

$$\pm u''(x) + k^2 u(x) = k^2 x \tag{4.7}$$

$$u''_N = \pm k^2 (1 - u_N) \tag{4.8}$$

$$u''_N = \pm k^2 \tag{4.9}$$

This equation yields the exact sign correspondence with the sign of the ODE.

Substituting Equation 4.9 into Equation 4.6

$$u'_N = \frac{u_N - u_{N-1}}{\Delta x} \pm \frac{k^2 \Delta x}{2} + \mathcal{O}(\Delta x^2) \tag{4.10}$$

Applying the boundary condition $u(1) = u_N = 0$:

$$u'_N = \frac{-u_{N-1}}{\Delta x} \pm \frac{k^2 \Delta x}{2} + \mathcal{O}(\Delta x^2) \tag{4.11}$$

From this specific first-derivative formulation at the boundary $x = 1$ using the finite difference method, the approximation can be observed to be second-order ($\mathcal{O}(\Delta x^2)$).

4.2.2 Results

Figure 4.2.1 – Positive ODE – 2nd-Order FDM with 2nd-Order First-Derivative Approximation

Figure 4.2.2 – Negative ODE – 2nd-Order FDM with 2nd-Order First-Derivative Approximation

4.2.3 Discussion

Like earlier, the positive ODE appears to be less stable than the negative ODE – a conclusion drawn from the deviation from an ideal rate of convergence. This is likely because the solution to the positive ODE can contain numerous oscillations, while the solution to the negative ODE is a single smooth oscillation for every solution. Therefore, the well-behavedness of an ODE solution is a factor in the rate of convergence of a particular finite difference method.

As the above figures indicate, the logarithm of the error decreases roughly at a rate of 2 times the negative logarithm of the mesh size. Thus, that the approximation used is second-order accurate and has a rate of convergence of 2. It is likely that if a fourth-order finite difference method is used, a fourth-order first-derivative scheme would be needed in order to see the appropriate rate of convergence.

5 MATLAB Code

```
clear all; close all; clc

%% Initial Conditions

ode.type = 'Positive';
ode.order = 4;

dudx.order = 2;

mesh.order = 1:6;
mesh.dx = 0.5.^mesh.order;

rowID = 0;

%% Boundary Value Problem Solution

for k = [1 2 5 10 20]

    figure
    xlabel('x');     ylabel('u(x)');
    grid on;         grid minor;
    box on;          hold on;

    if ode.order == 2
        titleString = strcat(ode.type, ' ODE with 2nd-Order FDM for k=', num2str(k));
    elseif ode.order == 4
        titleString = strcat(ode.type, ' ODE with 4th-Order FDM for k=', num2str(k));
    end

    title(titleString)

    rowID = rowID + 1;
    colID = 0;

    for dx = mesh.dx

        nx = 1 / dx + 1;
        x = linspace(0, 1, nx);

        A = zeros(nx, nx);
        b = zeros(nx, 1);

        if strcmpi(ode.type, 'positive') && ode.order == 2
            alpha = 1 / k^2 / dx^2;
            beta = -2 / k^2 / dx^2 + 1;
        elseif strcmpi(ode.type, 'positive') && ode.order == 4
            alpha = 1 / k^2 / dx^2 + 1/12;
            beta = -2 / k^2 / dx^2 + 10/12;
        elseif strcmpi(ode.type, 'negative') && ode.order == 2
            alpha = -1 / k^2 / dx^2;
            beta = 2 / k^2 / dx^2 + 1;
        elseif strcmpi(ode.type, 'negative') && ode.order == 4
            alpha = -1 / k^2 / dx^2 + 1/12;
            beta = 2 / k^2 / dx^2 + 10/12;
        end

        for i = 1:nx
```

```
            if i == 1
                A(i, i) = 1;
                b(i) = 0;
            elseif i == nx
                A(i, i) = 1;
                b(i) = 0;
            else
                A(i, i-1) = alpha;
                A(i, i) = beta;
                A(i, i+1) = alpha;
                b(i) = x(i);
            end
        end

        u = A\b;

        plot(x, u, 'linewidth', 1)

        colID = colID + 1;

        if strcmpi(ode.type, 'positive') && dudx.order == 1
            dudx.fdm(rowID, colID) = - u(end-1) / dx;
        elseif strcmpi(ode.type, 'positive') && dudx.order == 2
            dudx.fdm(rowID, colID) = - u(end-1) / dx + dx * k^2 / 2;
        elseif strcmpi(ode.type, 'negative') && dudx.order == 1
            dudx.fdm(rowID, colID) = - u(end-1) / dx;
        elseif strcmpi(ode.type, 'negative') && dudx.order == 2
            dudx.fdm(rowID, colID) = - u(end-1) / dx - dx * k^2 / 2;
        end

        if strcmpi(ode.type, 'positive')
            dudx.exact(rowID, colID) = 1 - k * cos(k) / sin(k);
        elseif strcmpi(ode.type, 'negative')
            dudx.exact(rowID, colID) = 1 - k * cosh(k) / sinh(k);
        end

    end

    if strcmpi(ode.type, 'positive')
        fplot(@(x) x-sin(k*x)/sin(k), [0 1], '-k', 'linewidth', 1.5)
    elseif strcmpi(ode.type, 'negative')
        fplot(@(x) x-sinh(k*x)/sinh(k), [0 1], '-k', 'linewidth', 1.5)
    end

    legend('\Deltax = (1/2)^1', '\Deltax = (1/2)^2', '\Deltax = (1/2)^3', '\Deltax = (1/2)^4', ...
    '\Deltax = (1/2)^5', '\Deltax = (1/2)^6', 'Analytical Solution', ...
    'location', 'best')

    figureString = strcat(lower(ode.type), '_ode_order_', num2str(ode.order), '_k_', num2str(k));

    %saveas(gcf, figureString, 'epsc')

end

%% Convergence Analysis

figure
xlabel('-log_{10}(\Deltax)');   ylabel('log_{10}(\epsilon_{rel})');
grid on;                        grid minor;
box on;                         hold on;

logRelError = log10(abs(dudx.exact-dudx.fdm) ./ abs(dudx.exact) * 100);
```

```
for kID = 1:5
    plot(-log10(mesh.dx), logRelError(kID, :), '-o', 'linewidth', 1.25);
end

if ode.order == 2 && dudx.order == 1
    titleString = strcat(ode.type, ' ODE with 2nd-Order FDM - 1st-Order First Derivative
                    Approximation');
elseif ode.order == 2 && dudx.order == 2
    titleString = strcat(ode.type, ' ODE with 2nd-Order FDM - 2nd-Order First Derivative
                    Approximation');
elseif ode.order == 4 && dudx.order == 1
    titleString = strcat(ode.type, ' ODE with 4th-Order FDM - 1st-Order First Derivative
                    Approximation');
elseif ode.order == 4 && dudx.order == 2
    titleString = strcat(ode.type, ' ODE with 4th-Order FDM - 2nd-Order First Derivative
                    Approximation');
end

title(titleString)

legend('k=1', 'k=2', 'k=5', 'k=10', 'k=20')

figureString = strcat(lower(ode.type), '_ode_order_', num2str(ode.order), '_fd_order_',
                num2str(dudx.order));

%saveas(gcf, figureString, 'epsc')
```