CS 138 - Numerical Analysis II

Problem Set 2: Lecture Problems + Feedback

Carmelo Ellezandro Atienza
CS 138 WFW
2021-08090

Arturo Miguel Saquilayan
CS 138 WFY
2021-04603

1. Level 1-1

1.1. Solutions to the Second-Order ODE

1.1.1. General Solution to the Second-Order ODE

The given second-order ODE is

$$y'' = -3$$

To find the general solution, we integrate the second-order ODE to get the first derivative y'.

$$y' = \int y'' \, \mathrm{d}x = \int -3 \, \mathrm{d}x = -3x + C_1$$

where C_1 is the constant of integration.

We integrate the first derivative again to get y.

$$y = \int y' \, \mathrm{d}x = \int (-3x + C_1) \, \mathrm{d}x = -\frac{3}{2}x^2 + C_1 x + C_2$$

Thus, the general solution to the second-order ODE is:

$$y(x) = -\frac{3}{2}x^2 + C_1 x + C_2$$

1.1.2. Specific Solution to the Second-Order ODE

We are given two boundary conditions:
1. $y(0) = 1$
2. $y(2) = 1.5$

Using $y(0) = 1$, we substitute $x = 0$ and $y(0) = 1$ into the general solution:

$$1 = -\frac{3}{2}(0)^2 + C_1(0) + C_2$$

$$C_2 = 1$$

Then, using $y(2) = 1.5$, we now substitute $x = 2$ and $y(2) = 1.5$ into the general solution:

$$1.5 = -\frac{3}{2}(2)^2 + C_1(2) + 1$$

$$1.5 = -\frac{3}{2}(4) + 2C_1$$

$$1.5 = -6 + 2C_1 + 1$$

$$1.5 = -5 + 2C_1$$

$$2C_1 = 6.5$$

$$C_1 = 3.25$$

Finally, substituting $C_1 = 3.25$ and $C_2 = 1$ into the general solution, we get the specific solution:

$$y(x) = -\frac{3}{2}x^2 + 3.25x + 1$$

1.2. Shooting Method

1.2.1. Second-order ODE as a system of first-order ODEs

Given the second-order ODE:

$$y'' = -3$$

We introduce a new variable v to reduce the order. Let:

$$v(x) = y'(x)$$

Then:

$$v'(x) = y''(x) = -3$$

Now, the second-order ODE becomes a system of two first-order ODEs:

$$y'(x) = v(x)$$
$$v'(x) = -3$$

1.2.2. Using Explicit Euler to solve the system of first-order ODEs

We have the following system of first-order ODEs:

$$y'(x) = v(x)$$
$$v'(x) = -3$$

Given:

- $y(0) = 1$
- $v(0) = y'(0) = 0$

We attempt to estimate $y(2)$ using Euler's method with $h = 0.5$.

Euler's method updates the values of y and v at each step using:

$$y_{n+1} = y_n + h \times v_n$$
$$v_{n+1} = v_n + h \times (-3)$$

We calculate the values for $x = 0, 0.5, 1.0, 1.5$, and 2.0.

At $x = 0$,

- $y_0 = 1$
- $v_0 = 0$

$$y_1 = y_0 + h \times v_0 = 1 + 0.5 \times 0 = 1$$
$$v_1 = v_0 + h \times (-3) = 0 + 0.5 \times (-3) = -1.5$$

At $x = 0.5$,

- $y_1 = 1$
- $v_1 = -1.5$

$$y_2 = y_1 + h \times v_1 = 1 + 0.5 \times (-1.5) = 1 - 0.75 = 0.25$$
$$v_2 = v_1 + h \times (-3) = -1.5 + 0.5 \times (-3) = -1.5 - 1.5 = -3$$

At $x = 1.0$,

- $y_2 = 0.25$

- $v_2 = -3$

$$y_3 = y_2 + h \times v_2 = 0.25 + 0.5 \times (-3) = 0.25 - 1.5 = -1.25$$
$$v_3 = v_2 + h \times (-3) = -3 + 0.5 \times (-3) = -3 - 1.5 = -4.5$$

At $x = 1.5$,

- $y_3 = -1.25$
- $v_3 = -4.5$

$$y_4 = y_3 + h \times v_3 = -1.25 + 0.5 \times (-4.5) = -1.25 - 2.25 = -3.5$$
$$v_4 = v_3 + h \times (-3) = -4.5 + 0.5 \times (-3) = -4.5 - 1.5 = -6$$

Then, the approximate height of Red bird at $x = 2$ using Explicit Euler and with a step size of $h = 0.5$ is:

$$y(2) \approx -3.5$$

We repeat the steps of Euler's method, this time with $y'(0) = 2$. Now, the conditions are:

- $y(0) = 1$
- $v(0) = y'(0) = 2$

We keep the same update equations and compute the values for $x = 0, 0.5, 1.0, 1.5$, and 2.0.

At $x = 0$,

- $y_0 = 1$
- $v_0 = 2$

$$y_1 = y_0 + h \times v_0 = 1 + 0.5 \times 2 = 1 + 1 = 2$$
$$v_1 = v_0 + h \times (-3) = 2 + 0.5 \times (-3) = 2 - 1.5 = 0.5$$

At $x = 0.5$,

- $y_1 = 2$
- $v_1 = 0.5$

$$y_2 = y_1 + h \times v_1 = 2 + 0.5 \times 0.5 = 2 + 0.25 = 2.25$$
$$v_2 = v_1 + h \times (-3) = 0.5 + 0.5 \times (-3) = 0.5 - 1.5 = -1$$

At $x = 1.0$,

- $y_2 = 2.25$
- $v_2 = -1$

$$y_3 = y_2 + h \times v_2 = 2.25 + 0.5 \times (-1) = 2.25 - 0.5 = 1.75$$
$$v_3 = v_2 + h \times (-3) = -1 + 0.5 \times (-3) = -1 - 1.5 = -2.5$$

At $x = 1.5$,

- $y_3 = 1.75$
- $v_3 = -2.5$

$$y_4 = y_3 + h \times v_3 = 1.75 + 0.5 \times (-2.5) = 1.75 - 1.25 = 0.5$$
$$v_4 = v_3 + h \times (-3) = -2.5 + 0.5 \times (-3) = -2.5 - 1.5 = -4$$

Then, the approximate height of Red Bird at $x = 2$ using Explicit Euler with $y'(0) = 2$ and a step size of $h = 0.5$ is:

$$y(2) \approx 0.5$$

1.2.3. Satisfying the boundary value problem

We attempt to find the value of $y'(0)$ that satisfies the boundary condition $y(2) = 1.5$ by applying the Shooting method. We are given the second-order ODE:

$$y'' = -3$$

with boundary conditions:

- $y(0) = 1$
- $y(2) = 1.5$

The goal is to find the value of $y'(0) = v(0)$ such that the solution satisfies $y(2) = 1.5$.

From the previous Euler calculations, we already have two initial guesses for $y'(0)$:

1. $y'(0) = 0$, which gave $y(2) \approx -3.5$
2. $y'(0) = 2$, which gave $y(2) \approx 0.5$

We apply linear interpolation to estimate the correct value of $y'(0)$, since the relationship between $y(2)$ and $y'(0)$ is approximately linear (the ODE is simple and linear).

The general formula for linear interpolation is:

$$y'(0) = y'(0)_1 + \left(\frac{y(2)_{\text{target}} - y(2)_1}{y(2)_2 - y(2)_1} \right) \times (y'(0)_2 - y'(0)_1)$$

Where:

- $y'(0)_1 = 0, y(2)_1 = -3.5$
- $y'(0)_2 = 2, y(2)_2 = 0.5$
- $y(2)_{\text{target}} = 1.5$

Substituting the values:

$$y'(0) = 0 + \left(\frac{1.5 - (-3.5)}{0.5 - (-3.5)} \right) \times (2 - 0)$$

$$y'(0) = 0 + \left(\frac{1.5 + 3.5}{0.5 + 3.5} \right) \times 2 = \left(\frac{5}{4} \right) \times 2 = 2.5$$

Therefore, the estimated value of $y'(0)$ that satisfies the boundary condition is 2.5.

1.2.4. Approximating height using RK4

Again, we have the following system of first-order ODEs:

$$y'(x) = v(x)$$

$$v'(x) = -3$$

Given the boundary value solution:

- $y(0) = 1$
- $v(0) = y'(0) = 2.5$

We use RK4 to estimate $y(1)$, with a step size of $h = 0.5$, moving from $x = 0$ to $x = 1$ in two steps. The RK4 method computes the next values of y and v as follows:

For $y(x)$:

$$k_1^y = h \times v(x_n)$$
$$k_2^y = h \times \left(v(x_n) + \frac{k_1^v}{2}\right)$$
$$k_3^y = h \times \left(v(x_n) + \frac{k_2^v}{2}\right)$$
$$k_4^y = h \times (v(x_n) + k_3^v)$$

For $v(x)$:

$$k_1^v = h \times (-3)$$
$$k_2^v = h \times (-3)$$
$$k_3^v = h \times (-3)$$
$$k_4^v = h \times (-3)$$

The updates for y and v are then:

$$y_{n+1} = y_n + \frac{1}{6}(k_1^y + 2k_2^y + 2k_3^y + k_4^y)$$
$$v_{n+1} = v_n + \frac{1}{6}(k_1^v + 2k_2^v + 2k_3^v + k_4^v)$$

At $x = 0$,

- $y_0 = 1$
- $v_0 = 2.5$

For y:

$$k_1^y = h \times v(0) = 0.5 \times 2.5 = 1.25$$
$$k_2^y = h \times \left(2.5 - \frac{1.5}{2}\right) = 0.5 \times (2.5 - 0.75) = 0.5 \times 1.75 = 0.875$$
$$k_3^y = h \times \left(2.5 - \frac{1.5}{2}\right) = 0.875$$
$$k_4^y = h \times (2.5 - 1.5) = 0.5 \times 1 = 0.5$$

For v:

$$k_1^v = h \times (-3) = 0.5 \times (-3) = -1.5$$
$$k_2^v = h \times (-3) = -1.5$$
$$k_3^v = h \times (-3) = -1.5$$
$$k_4^v = h \times (-3) = -1.5$$

$$y(0.5) = y(0) + \frac{1}{6}(k_1^y + 2k_2^y + 2k_3^y + k_4^y)$$
$$y(0.5) = 1 + \frac{1}{6}(1.25 + 2(0.875) + 2(0.875) + 0.5)$$
$$y(0.5) = 1 + \frac{1}{6}(1.25 + 1.75 + 1.75 + 0.5) = 1 + \frac{1}{6}(5.25) = 1 + 0.875 = 1.875$$

$$v(0.5) = v(0) + \frac{1}{6}(k_1^v + 2k_2^v + 2k_3^v + k_4^v)$$

$$v(0.5) = 2.5 + \frac{1}{6}(-1.5 + 2(-1.5) + 2(-1.5) + (-1.5))$$

$$v(0.5) = 2.5 + \frac{1}{6}(-1.5 - 3 - 3 - 1.5) = 2.5 + \frac{1}{6}(-9) = 2.5 - 1.5 = 1$$

At $x = 0.5$,

- $y(0.5) = 1.875$
- $v(0.5) = 1$

For y:

$$k_1^y = h \times v(0.5) = 0.5 \times 1 = 0.5$$

$$k_2^y = h \times \left(1 + -\frac{1.5}{2}\right) = 0.5 \times 0.25 = 0.125$$

$$k_3^y = 0.125$$

$$k_4^y = h \times (1 - 1.5) = 0.5 \times (-0.5) = -0.25$$

For v:

$$k_1^v = h \times (-3) = -1.5$$

$$k_2^v = -1.5$$

$$k_3^v = -1.5$$

$$k_4^v = -1.5$$

$$y(1) = y(0.5) + \frac{1}{6}(k_1^y + 2k_2^y + 2k_3^y + k_4^y)$$

$$y(1) = 1.875 + \frac{1}{6}(0.5 + 2(0.125) + 2(0.125) + (-0.25))$$

$$y(1) = 1.875 + \frac{1}{6}(0.5 + 2(0.125) + 2(0.125) + (-0.25))$$

$$y(1) = 1.875 + \frac{1}{6}(0.5 + 0.25 + 0.25 - 0.25) = 1.875 + \frac{1}{6}(0.75) = 1.875 + 0.125 = 2$$

$$v(1) = v(0.5) + \frac{1}{6}(-1.5 + 2(-1.5) + 2(-1.5) + (-1.5))$$

$$v(1) = 1 + \frac{1}{6}(-1.5 - 3 - 3 - 1.5) = 1 + \frac{1}{6}(-9) = 1 - 1.5 = -0.5$$

Thus, the approximate height of Red Bird at $x = 1$ using the RK4 method with a step size of $h = 0.5$ is:

$$y(1) \approx 2$$

1.3. Finite Difference Method

1.3.1. System of Linear Equations Setup

To estimate Red Bird's trajectory using the Finite Difference Method, we must discretize the second-order ODE $y'' = -3$ with step size $h = 0.5$ and set up a system of linear equations to solve for the intermediate points $y(0.5), y(1)$, and $y(1.5)$.

We approximate the second derivative $y''(x)$ using the central difference formula with second-order accuracy:

$$y''(x_i) \approx \frac{y(x_{i+1} - 2y(x_i) + y(x_{i-1}))}{h^2}$$

Thus, the finite difference approximation for $y''(x_i) = -3$ becomes:

$$\frac{y(x_{i+1} - 2y(x_i) + y(x_{i-1}))}{h^2} = -3$$

Given the boundary conditions:

- $y(0) = 1$
- $y(2) = 1.5$

We solve for the intermediate points $y(0.5), y(1), y(1.5)$ at the mesh points $x = 0.5, 1, 1.5$, with step size $h = 0.5$. Setting up the finite difference equations for each mesh point $x = 0.5, 1, 1.5$:

At $x = 0.5$:

$$\frac{y(1) - 2y(0.5) + y(0)}{h^2} = -3$$
$$4(y(1) - 2y(0.5) + y(0)) = -3$$
$$y(1) - 2y(0.5) + 1 = -0.75$$
$$y(1) - 2y(0.5) = -1.75$$

At $x = 1$:

$$\frac{y(1.5) - 2y(1) + y(0.5)}{h^2} = -3$$
$$4(y(1.5) - 2y(1) + y(0.5)) = -3$$
$$y(1.5) - 2y(1) + y(0.5) = -0.75$$

At $x = 1.5$:

$$\frac{y(2) - 2y(1.5) + y(1)}{h^2} = -3$$
$$4(y(2) - 2y(1.5) + y(1)) = -3$$
$$1.5 - 2y(1.5) + y(1) = -0.75$$
$$y(1) - 2y(1.5) = -2.25$$

We now have three equations for the unknowns $y(0.5), y(1), y(1.5)$:

1. $y(1) - 2y(0.5) = -1.75$
2. $y(1.5) - 2y(1) + y(0.5) = -0.75$
3. $y(1) - 2y(1.5) = -2.25$

We write this as a system of linear equations:

$$-2y(0.5) + y(1) = -1.75$$
$$y(0.5) - 2y(1) + y(1.5) + = -0.75$$
$$y(1) - 2y(1.5) = -2.25$$

1.3.2. Solving the System of Linear Equations

We use Gaussian Elimination to solve the system of linear equations. First, we express this system in matrix form as:

$$\begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} y(0.5) \\ y(1) \\ y(1.5) \end{bmatrix} = \begin{bmatrix} -1.75 \\ -0.75 \\ -2.25 \end{bmatrix}$$

We combine the coefficient matrix with the constant matrix on the right side as the augmented matrix:

$$\left[\begin{array}{ccc|c} -2 & 1 & 0 & 1.75 \\ 1 & -2 & 1 & 0.75 \\ 0 & 1 & -2 & -2.25 \end{array}\right]$$

Reducing the augmented matrix to Row-Echelon Form,

$$\left[\begin{array}{ccc|c} -2 & 1 & 0 & 1.75 \\ 1 & -2 & 1 & 0.75 \\ 0 & 1 & -2 & -2.25 \end{array}\right] \overset{R_2 \to R_2 + \frac{1}{2}R_1}{\Longrightarrow} \left[\begin{array}{ccc|c} -2 & 1 & 0 & 1.75 \\ 0 & -\frac{3}{2} & 1 & -1.625 \\ 0 & 1 & -2 & -2.25 \end{array}\right] \overset{R_3 \to R_3 + \frac{2}{3}R_2}{\Longrightarrow} \left[\begin{array}{ccc|c} -2 & 1 & 0 & 1.75 \\ 0 & -\frac{3}{2} & 1 & -1.625 \\ 0 & 0 & -\frac{4}{3} & -3.3333 \end{array}\right]$$

We then perform back substitution to solve for $y(0.5), y(1)$, and $y(1.5)$.

From the third row:

$$-\frac{4}{3}y(1.5) = -3.3333$$
$$y(1.5) = \frac{-3.3333}{-\frac{4}{3}} = 2.5$$

From the second row:

$$-\frac{3}{2}y(1) + y(1.5) = -1.625$$
$$-\frac{3}{2}y(1) + 2.5 = -1.625$$
$$-\frac{3}{2}y(1) = -1.625 - 2.5 = -4.125$$
$$y(1) = \frac{-4.125}{-\frac{3}{2}} = 2.75$$

From the first row:

$$-2y(0.5) + y(1) = -1.75$$
$$-2y(0.5) + 2.75 = -1.75$$
$$-2y(0.5) = -1.75 - 2.75 = -4.5$$
$$y(0.5) = \frac{-4.5}{-2} = 2.25$$

Thus, the solution to the system of equations is:

$$y(0.5) = 2.25$$
$$y(1) = 2.75$$
$$y(1.5) = 2.5$$

1.3.3. Approximating the computing limits

To estimate how many mesh points the "modern computer" can compute in a second using the Finite Difference Method, we consider the number of operations required per mesh point.

For each mesh point, we solve a system of equations like the one solved earlier. This involves:

1. Applying the finite difference approximation to the second-order ODE, which requires evaluating surrounding points.
2. Solving the system of linear equations, which involves matrix operations.

For a large system of N mesh points, solving it using Gaussian elimination typically requires around $O(N^3)$. Since 10^8 operations can be performed in a second, and Gaussian elimination on a matrix of size N requires approximately $O(N^3)$ operations, we estimate N as follows:

The total number of operations for solving the system is approximately:

$$N^3 \approx 10^8$$
$$N \approx (10^8)^{\frac{1}{3}} = 10^{\frac{8}{3}} \approx 464$$

Thus, a modern computer can approximately handle 464 mesh points in a second when using the Finite Difference Method with a full Gaussian elimination process.

2. Blue Bird Bonanza

2.1. Monomial Basis Function Setup

We have the differential equation and the basis function

$$y'' = xy$$
$$y \approx w_1 + w_2 x + w_3 x^2$$

with $y(0) = y(2) = 1$ as our boundary conditions.

At our first boundary condition,

$$y(0) = y_0 = 1$$
$$y_0 = w_1 + w_2(0) + w_3(0)^2 = 1$$
$$\Rightarrow w_1 = 1$$

At our second boundary condition,

$$y(2) = y_2 = 1$$
$$y_2 = w_1 + w_2(2) + w_3(2)^2 = 1$$
$$\Rightarrow w_1 + 2w_2 + 4w_3 = 1$$

At the collocation point $x = 1$,

$$y(1) = y_1 = w_1 + w_2(1) + w_3(1)^2$$
$$y_1 = w_1 + w_2 + w_3$$

We then obtain the second derivative of our basis function,

$$y = w_1 + w_2 x + w_3 x^2$$
$$y' = w_2 + 2w_3 x$$
$$y'' = 2w_3$$

Then substituting the values to our differential equation at $x = 1$,

$$y_1'' = xy_1$$
$$2w_3 = 1(w_1 + w_2 + w_3)$$
$$-w_1 - w_2 + w_3 = 0$$

Forming our system of linear equations,

$$\boxed{\begin{cases} w_1 = 1 \\ -w_1 - w_2 + w_3 = 0 \\ w_1 + 2w_2 + 4w_3 = 1 \end{cases}} \Rightarrow \boxed{\begin{bmatrix} 1 & 0 & 0 \\ -1 & -1 & 1 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}}$$

2.2. Definitely My Favorite Direct Method

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 2 & 4 & 1 \end{array}\right] \overset{\substack{R_1+R_2\to R_2 \\ -R_1+R_3\to R_3}}{\Longrightarrow} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 1 \\ 0 & 2 & 4 & 0 \end{array}\right] \overset{2R_2+R_3\to R_3}{\Longrightarrow} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 1 \\ 0 & 0 & 6 & 2 \end{array}\right]$$

$$\begin{cases} w_1 = 1 \\ -w_2 + w_3 = 1 \Rightarrow \\ 6w_3 = 2 \end{cases} \boxed{w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{2}{3} \\ \frac{1}{3} \end{bmatrix}}$$

2.3. Gaussian Basis Function Setup

We have the same differential equation, but a different basis function,

$$y'' = xy$$
$$y(x) \approx w_1\gamma_{x_1}(x) + w_2\gamma_{x_2}(x) + w_3\gamma_{x_3}(x)$$

where $\gamma_{x_i}(r) = \gamma(|x_i - r|)$ such that $\gamma(r) = e^{-(2r)^2} = e^{-4r^2}$.

Let $x_1 = 0, x_2 = 1, x_3 = 2$, which are our chosen points for the collocation method.

At the first boundary condition,

$$y(0) = 1$$
$$w_1\gamma_{x_1}(0) + w_2\gamma_{x_2}(0) + w_3\gamma_{x_3}(0) = 1$$
$$w_1\gamma(|0-0|) + w_2\gamma(|1-0|) + w_3\gamma(|2-0|) = 1$$
$$w_1\gamma(0) + w_2\gamma(1) + w_3\gamma(2) = 1$$
$$w_1 + (e^{-4})w_2 + (e^{-16})w_3 = 1$$

At the second boundary condition,

$$y(2) = 1$$
$$w_1\gamma_{x_1}(2) + w_2\gamma_{x_2}(2) + w_3\gamma_{x_3}(2) = 1$$
$$w_1\gamma(2) + w_2\gamma(1) + w_3\gamma(0) = 1$$
$$(e^{-16})w_1 + (e^{-4})w_2 + w_3 = 1$$

At the collocation point $x = 1$, we first obtain the second derivative of our basis function,

$$\gamma(r) = e^{-4r^2}$$
$$\gamma'(r) = -8re^{-4r^2}$$
$$\gamma''(r) = (-8)\Big(e^{-4r^2} + r\Big(-8re^{-4r^2}\Big)\Big)$$
$$\Rightarrow \gamma''(r) = e^{-4r^2}(64r^2 - 8)$$

Then we evaluate $y''(x)$ at $x = 1$,

$$y''(1) = w_1\gamma''_{x_1}(1) + w_2\gamma''_{x_2}(1) + w_3\gamma''_{x_3}(1)$$
$$y''(1) = w_1\gamma''(1) + w_2\gamma''(0) + w_3\gamma''(1)$$
$$y''(1) = (56e^{-4})w_1 - 8w_2 + (56e^{-4})w_3$$

Substituting to our differential function at $x = 1$,

$$y'' = xy$$
$$y''(1) = 1\Big(w_1\gamma_{x_1}(1) + w_2\gamma_{x_2}(1) + w_3\gamma_{x_3}(1)\Big)$$
$$y''(1) = w_1\gamma(1) + w_2\gamma(0) + w_3\gamma(1)$$
$$(56e^{-4})w_1 - 8w_2 + (56e^{-4})w_3 = (e^{-4})w_1 + w_2 + (e^{-4})w_3$$
$$\Rightarrow (55e^{-4})w_1 - 9w_2 + (55e^{-4}) = 0$$

Forming our system of linear equations,

$$\begin{cases} w_1 + (e^{-4})w_2 + (e^{-16})w_3 = 1 \\ (55e^{-4})w_1 - 9w_2 + (55e^{-4}) = 0 \\ (e^{-16})w_1 + (e^{-4})w_2 + w_3 = 1 \end{cases} \Rightarrow \begin{bmatrix} 1 & e^{-4} & e^{-16} \\ 55e^{-4} & -9 & 55e^{-4} \\ e^{-16} & e^{-4} & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

2.4. Definitely My Favorite Iterative Method

Let our initial guess be $w^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$. We form our Gauss-Seidel iteration scheme,

$$\begin{cases} w_1^{(k)} = -(e^{-4})w_2^{(k-1)} - (e^{-16})w_3^{(k-1)} + 1 \\ w_2^{(k)} = \left(\frac{55}{9}e^{-4}\right)w_1^{(k)} + \left(\frac{55}{9}e^{-4}\right)w_3^{(k-1)} \\ w_3^{(k)} = -(e^{-16})w_1^{(k)} - (e^{-4})w_2^{(k)} + 1 \end{cases}$$

We can solve this using good ol' Python.

```
import math
def first_equation(w_2, w_3):
    return -(w_2*math.e**-4) - (w_3*math.e**-16) + 1
def second_equation(w_1, w_3):
    return (55/9)*(math.e**-4)*(w_1 + w_3)
def third_equation(w_1, w_2):
    return -(w_1*math.e**-16) - (w_2*math.e**-4) + 1
def gauss_seidel(iterations):
    initial_guess = [1, 1, 1]
    w_1, w_2, w_3 = initial_guess
    for i in range(iterations):
        w_1 = first_equation(w_2, w_3)
        w_2 = second_equation(w_1, w_3)
        w_3 = third_equation(w_1, w_2)
        print(f'Iteration {i+1}')
        print(f'w_1 = {w_1}')
        print(f'w_2 = {w_2}')
        print(f'w_3 = {w_3}')
        print("--------------------")
    return [w_1, w_2, w_3]
```

After three iterations, we have the following results,

	$w^{(0)}$	$w^{(1)}$	$w^{(2)}$	$w^{(3)}$
w_1	1	0.9816842486	0.9959373373	0.9959164465
w_2	1	0.2218077467	0.2229483500	0.2229436733
w_3	1	0.9959373389	0.9959165321	0.9959165321

Thus our solution is approximated by,

$$\boxed{w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0.9959164465 \\ 0.2229436733 \\ 0.9959165321 \end{bmatrix}}$$

2.5. Nontrivial Eigenvalues

For the given eigenvalue problem

$$(D^2 + 4D + 3)y = y'' + 4y' + 3y = -\lambda y$$

with the Neumann boundary conditions $y'(0) = y'(1) = 0$, we first simplify the differential equation,

$$y'' + 4y' + (3 + \lambda)y = 0$$

Thus yielding the auxiliary equation,

$$m^2 + 4m + 3 + \lambda = 0$$

Solving the roots of the auxiliary equation,

$$m = \frac{-4 \pm \sqrt{16 - 4(1)(3 + \lambda)}}{2}$$
$$m = -2 \pm \sqrt{4 - 3 - \lambda}$$
$$m = -2 \pm \sqrt{1 - \lambda}$$

We split our approach into cases.

2.5.1. Case 1 (Positive): $1 - \lambda > 0$
First, we obtain the derivative of y for this case,

$$y = c_1 e^{m_1 x} + c_2 e^{m_2 x}$$
$$y' = c_1 m_1 e^{m_1 x} + c_2 m_2 e^{m_2 x}$$

We then evaluate at the boundary conditions,

$$y'(0) = c_1 m_1 e^0 + c_2 m_2 e^0 = 0$$
$$y'(0) = c_1 m_1 + c_2 m_2 = 0$$
$$\Rightarrow c_2 m_2 = -c_1 m_1$$

$$y'(1) = c_1 m_1 e^{m_1} + c_2 m_2 e^{m_2} = 0$$
$$y'(1) = c_1 m_1 e^{m_1} - c_1 m_1 e^{m_2} = 0$$
$$\Rightarrow c_1 m_1 (e^{m_1} - e^{m_2}) = 0$$

Since $m_1 \neq m_2 \Rightarrow (e^{m_1} - e^{m_2}) \neq 0$ and $m_1 \neq 0$, then $c_1 = 0 \Rightarrow c_2 = 0$. Hence, a trivial solution.

2.5.2. Case 2 (Zero): $1 - \lambda = 0$
Obtaining the derivative of y for this case,

$$y = c_1 e^{mx} + c_2 x e^{mx}$$
$$y' = c_1 m e^{mx} + c_2 e^{mx}(1 + xm)$$

We then evaluate at the boundary conditions,

$$y'(0) = c_1 m e^0 + c_2(1+0) = 0$$
$$y'(0) = c_1 m + c_2 = 0$$
$$\Rightarrow c_2 = -c_1 m$$

$$y'(1) = c_1 m e^m + c_2 e^m (1+m) = 0$$
$$y'(1) = c_1 m e^m - c_1 m e^m (1+m) = 0$$
$$y'(1) = c_1 m e^m - c_1 m e^m - c_1 m^2 e^m = 0$$
$$\Rightarrow c_1 m^2 e^m = 0$$

Since $m^2 \neq 0$ and e^m is a nonzero constant, then $c_1 = 0 \Rightarrow c_2 = 0$. Hence, a trivial solution.

2.5.3. Case 3 (Negative): $1 - \lambda < 0$
We can express our root m in the form of $a \pm ib$ as

$$-2 \pm i\sqrt{\lambda - 1}$$

where $a = -2, b = \sqrt{\lambda - 1}$. Thus, obtaining the derivative of y for this case,

$$y = e^{ax}(c_1 \cos(bx) + c_2 \sin(bx))$$
$$y' = c_1[ae^{ax}\cos(bx) - e^{ax} b \sin(bx)] + c_2[ae^{ax}\sin(bx) + be^{ax}\cos(bx)]$$

Evaluating at the boundary conditions,

$$y'(0) = c_1[a\cos(0) - eb\sin(0)] + c_2[a\sin(0) + b\cos(0)] = 0$$
$$\Rightarrow c_1 a + c_2 b = 0 \Rightarrow c_2 = -c_1 \frac{a}{b}$$

$$y'(1) = c_1[ae^a\cos(b) - be^a\sin(b)] + c_2[ae^a\sin(b) + be^a\cos(b)] = 0$$
$$c_1[ae^a\cos(b) - be^a\sin(b)] - c_1\frac{a}{b}[ae^a\sin(b) + be^a\cos(b)] = 0$$
$$c_1 b[ae^a\cos(b) - be^a\sin(b)] - c_1 a[ae^a\sin(b) + be^a\cos(b)] = 0$$
$$c_1 b[a\cos(b) - b\sin(b)] - c_1 a[a\sin(b) + b\cos(b)] = 0$$
$$c_1 ab\cos(b) - c_1 b^2\sin(b) - c_1 a^2\sin(b) - c_1 ab\cos(b) = 0$$
$$-c_1 b^2\sin(b) - c_1 a^2\sin(b) = 0$$
$$c_1 a^2\sin(b) + c_1 b^2\sin(b) = 0$$
$$\Rightarrow c_1\sin(b)(a^2 + b^2) = 0$$

Since $a^2 + b^2 \neq 0$ and we want $c_1 \neq 0$ to yield a nontrivial solution, then $\sin(b) = 0 \Rightarrow b = n\pi$ for some integer n. Evaluating λ,

$$b = n\pi = \sqrt{\lambda - 1} \Rightarrow n^2\pi^2 = \lambda - 1$$
$$\boxed{\lambda = n^2\pi^2 + 1}$$

Since $1 - \lambda < 0 \Rightarrow 1 < \lambda$, then $n \in \mathbb{N}$.

2.6. Nontrivial Eigenfunctions and Corresponding Eigenvalues

For the nontrivial eigenvalue,

$$\lambda = n^2\pi^2 + 1 \text{ for } n \in \mathbb{N}$$

With $a = -2, b = \sqrt{\lambda - 1} = \sqrt{n^2\pi^2} = n\pi$ and $c_2 = -c_1\frac{a}{b}$, we derive the eigenfunction,

$$y = e^{ax}(c_1\cos(bx) + c_2\sin(bx))$$
$$\Rightarrow e^{-2x}\left(c_1\cos(n\pi x) - c_1\frac{a}{b}\sin(n\pi x)\right)$$
$$e^{-2x}\left(c_1\cos(n\pi x) + c_1\frac{2}{n\pi}\sin(n\pi x)\right)$$
$$c_1e^{-2x}\left(\cos(n\pi x) + \frac{2}{n\pi}\sin(n\pi x)\right)$$
$$\Rightarrow \boxed{y = e^{-2x}\left(\cos(n\pi x) + \frac{2}{n\pi}\sin(n\pi x)\right)}$$

In general, the solution is given by

$$y = \sum_{n=1}^{\infty} c_n e^{-2x}\left(\cos(n\pi x) + \frac{2}{n\pi}\sin(n\pi x)\right)$$

2.7. Does this Exist?

To determine if the eigenfunction exists for $y(0.5) = 138$, we need to use the eigenfunction to obtain some $n \in \mathbb{N}$. For $x = 0.5$,

$$y(0.5) = e^{-1}\left(\cos\left(n\frac{\pi}{2}\right) + \frac{2}{n\pi}\sin\left(n\frac{\pi}{2}\right)\right) = 138$$
$$\Rightarrow \cos\left(n\frac{\pi}{2}\right) + \frac{2}{n\pi}\sin\left(n\frac{\pi}{2}\right) = 138e$$

Wow, that's an ugly looking transcendental equation. If we use some numerical method to solve for n, we would get that $n = -1.20598 * 10^{-18} \pm 4.1237i \notin \mathbb{N}$. Thus, the eigenfunction does not exist.

2.8. Pig-ger and Better

2.8.1. What did you think of Problem Set 2?

Melo: The workload is lighter compared to Problem Set 1, so that is good to begin with. Other than that, it was a good experience. The Lecture problems were good practice for the upcoming Long Exam, and the Lab problems on `Julia` was a fascinating experience.

Art: Definitely lighter compared to Problem Set 1. The focus on `Julia` was interesting too compared to just coding everything in Python.

2.8.2. What did you think about the difficulty of the problems?

Melo: Some of the problems that I answered were still a bit challenging, but it wasn't that bad. Also, most of the challenge with the Lab problems is the learning curve with `Julia`. Though, it was an easy language to learn anyway.

Art: The problems weren't as tedious to do as Problem 1. I agree that a lot of the difficulty around the Lab problems was with learning how to use `Julia` in the first place.

2.8.3. Which problems were your favorites? Which problems were your least favorites? Why?

Melo: My favorite problems were the Lab problems, including the `PS2++`. It's because we are actually applying some of these methods in practice. Plus, coding is at least a bit more engaging for me than just doing pure math. The least favorite problem is Blue Bird Bonanza, since it's the only other problem I actually answered for this Problem Set. Even then, it was an okay problem.

Art: I probably liked Level 1-1 more than the Lab, mostly because it was a bit easier. That makes the Lab problems my lesser favorites, but that doesn't mean that I didn't find the challenge of solving them fun.

2.8.4. Do you think the problems will help you with the upcoming exam?

Melo: I think so. The problems were pretty standard and I can see how some of it might be asked in the exam. Though, there are still some conceptual items that might come out during the exam which I will not be prepared for.

Art: They probably will. I'm assuming the problems will come up in the exam in some form (it would be a waste otherwise).

2.8.5. Is there anything you would suggest changing or improving on regarding any aspect of the course so far?

Melo: At least for this part, we need more sample problems. Sir JBS' pandemic video lectures alone were not sufficient for practicing the problems 💀

Art: I agree that sample problems are probably the way to go. I usually found that solving problems repeatedly were effective in getting high scores in exams, and this class is probably one of the same too.

2.8.6. Anything else you want to say?

Melo: Thank you for allowing open handwritten notes…