KUTAISI INTERNATIONAL UNIVERSITY

Final Project

Sturm-Liouville problem

Noe Lomidze

Kutaisi International University

January 9, 2025

Contents

Tasks

Description

- Formulate algorithm, explain your approach in written.
- Describe properties of numerical methods written.
- Develop test cases and demonstrate validity of your results.
- Upload all necessary files, including
  1. Presentation file
  2. Code
  3. Test data and their description
- Using shooting method and ball motion equation is compulsory

Sturm-Liouville problem

Components

- Input: Sturm-Liouville problem
- Task: find first 8 eigenvalues and eigenfunctions
- Approach: approximate vanishing or singular coefficients
- Output: visualisation of eigenvalues and eigenfaunctions
- Test: test case description
- Methodology: should contain problem formulation, including equation with initial and boundary condition, method of solution, algorithm

Examples, Sturm-Liouville Problem

Theory

In mathematics and its applications, a **Sturm-Liouville** problem is a second-order linear ordinary differential equation of the form:

$$\frac{d}{\mathrm{d}x}\left[p(x)\frac{\mathrm{d}y}{\mathrm{d}x}\right] + q(x)y = -\lambda w(x)y \tag{1}$$

for the given functions $p(x), q(x)$ and $w(x)$ together with some Boundary Conditions at extreme values of x. The goals are:

- To find the λ (eigenvalue) for which there exists a non-trivial solution to the problem.
- To find the corresponding solution $y = y(x)$ of the problem, such functions are eigenfunctions

Main results

The main results in Sturm-Liouville theory apply to a Sturm-Liouville problem: $\frac{d}{\mathrm{d}x}\left[p(x)\frac{\mathrm{d}y}{\mathrm{d}x}\right] + q(x)y = -\lambda w(x)y$

on a finite interval $[a,b]$ that is "regular". The problem is regular if:

- the coefficient functions p, q, w and derivative p' are all continuous on $[a, b]$;
- $p(x) > 0$ and $w(x) > 0$ for all $x \in [a, b]$;
- the problem has separated boundary conditions of the form

$$\alpha_1 y(a) + \alpha_2 y'(a) = 0 \quad \alpha_1, \alpha_2 \text{ not both } 0, \tag{2}$$

$$\beta_1 y(b) + \beta_2 y'(b) = 0 \quad \beta_1, \beta_2 \text{ not both } 0, \tag{3}$$

The function $w = w(x)$ is called the *weight* or *density* function.

Reduction to Sturm–Liouville form

The differential Equation 1 is said to be in **Sturm-Liouville** or **self-adjoint form**

Bessel equation

$$x^2y'' + xy' + (x^2 - \nu^2)y = 0$$

which can be written in Sturm-Liouville form (first by dividing through by x, then by collapsing the first two terms on the left into one term) as:

$$(xy')' + \left(x - \frac{\nu^2}{x}\right)y = 0$$

Simple example

For $\lambda \in \mathbb{R}$, solve: $$y'' + \lambda y = 0 \quad y(0) = 0, \quad y'(\pi) = 0$$

Case 1. Let $\lambda < 0$. Then $\lambda = -\mu^2,\ \mu \in \mathbb{R} \setminus \{0\}$. Solution of ODE is $y(x) = Ae^{\mu x} + Be^{-\mu x}$

This y satisfies boundary conditions iff $A = B = 0 \Longrightarrow y \equiv 0$. So there are no negative eigenvalues..

Case 2. Let $\lambda = 0$. In this case, it easily follows that trivial solution is the only solution of

$$y'' = 0, \quad y(0) = 0, \quad y'(\pi) = 0. \text{ Thus, 0 is not an eigenvalue.}$$

Case 3. Let $\lambda > 0$. Then $\lambda = \mu^2$, where $\mu \in R \setminus \{0\}$. The general solution of ODE is given by

$$y(x) = A\cos(\mu x) + B\sin(\mu x)$$

We need: $A = 0$ and $B\cos(\mu\pi) = 0$. But $B\cos(\mu\pi) = 0$ iff either $B = 0$ or $\cos(\mu\pi) = 0$.

If $A = 0$ and $B = 0 \Rightarrow y \equiv 0$, Thus $\cos(\mu\pi) = 0$ should hold, the last equation has solutions given by $\mu = \left(\frac{1}{2} + n\right)$, for $n = 0, \pm 1, \pm 2, \ldots$ Thus the eigenvalues are $\lambda_n = \left(\frac{1}{2} + n\right)^2, \quad n = 0, 1, 2, \ldots$ and corresponding eigenfunctions are given by $\phi_n = B\sin\left(\left(\frac{1}{2} + n\right)x\right)$

Note: All the eigenvalues are positive. The eigenfunctions corresponding to each eigenvalue form a one dimensional vector space and so the eigenfunctions are unique upto a constant multiple.

Regular SL-BVP properties

Eigenvalues of regular *SL-BVP* are real. $\mathcal{L}[y] \equiv \frac{d}{\mathrm{d}x}\left[p(x)\frac{\mathrm{d}y}{\mathrm{d}x}\right] + q(x)y, \quad \mathcal{L}[y] + \lambda r(x)y = 0$

Proof

Suppose $\lambda \in \mathbb{C}$ is an eigenvalue and y be the corresponding eigenfunction. That is,
$\mathcal{L}[y] + \lambda r(x)y = 0,\ a_1 y(a) + a_2 p(a)y'(a) = 0,\ b_1 y(b) + a_2 p(b)y'(b) = 0$, Taking compl conj
$\mathcal{L}[\overline{y}] + \lambda r(x)\overline{y} = 0,\ a_1 \overline{y}(a) + a_2 p(a)\overline{y}'(a) = 0,\ b_1 \overline{y}(b) + a_2 p(b)\overline{y}'(b) = 0$
Multiply the first ODE with $\overline{y}$ and multiply that with y, subtracting one from another yields:
$[p(y'\overline{y} - \overline{y}'y)]' + \left(\lambda - \overline{\lambda}\right) ry\overline{y} = 0$, Integrating the last equation yields:

$$[p(y'\overline{y} - \overline{y}'y)]\ |_a^b = -\left(\lambda - \overline{\lambda}\right) \int_a^b r(x)\ |y(x)|^2\,\mathrm{d}x.$$

But LHS is zero, since we have both boundary conditions, also we know that $b_1^2 + b_2^2 \neq 0$ Thus

$$\left(\lambda - \overline{\lambda}\right) \int_a^b r\ |y|^2\,\mathrm{d}y = 0$$

Since y being an eigenfaunction $y \neq 0$, also $r > 0$, only possibility is that $\lambda = \overline{\lambda}$ which means that λ is real. Done.

Regular SL-BVP properties

Eigenfunctions of the distinct eigenvalues, of a regular SL-BVP are othogonal:

$$\int_a^b r(x)u(x)v(x) = 0 \tag{4}$$

Proof

As in the previous proof, writing down the equations satisfied by u and v, and multiplying the equation for u with v and vice versa, finally substracting we get:

$$[p(u'v - v'u)] + (\lambda - \mu)ruv = 0$$

Integrating the last equality yields:

$$[p(u'v - v'u)] \,|_a^b = -(\lambda - \mu) \int_a^b r(x)u(x)v(x)\,\mathrm{d}x$$

Reasoning exactly as in the previous proof, LHS is zero, since $\lambda \neq \mu$, proof is done.

Numerical Methods for ODEs

There are many methods to solve $\frac{d}{\mathrm{d}y} = f(t, \boldsymbol{y})$, but lets consider two:

Euler's method: $y_{j+1} = y_j + kf_j, \quad \text{which is } O(k)$

</> **Code**

```
def euler_method(f, t0, y0, h, t_end):
    t_values = [t0]
    y_values = [y0]
    while t_values[-1] < t_end:
        t_new = t_values[-1] + h
        y_new = y_values[-1] +
        h * f(t_values[-1], y_values[-1])
        t_values.append(t_new)
        y_values.append(y_new)

    return np.array(t_values), np.array(y_values)
```

Numerical Methods for ODEs

Classical Runge-Kutta Method:

$$y_{j+1} = y_j + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad O(k^4) \tag{5}$$

Where:

$$
\begin{aligned}
k_1 &= kf_j \\
k_2 &= kf\left(t_j + \tfrac{k}{2}, y_j + \tfrac{1}{2}k_1\right) \\
k_3 &= kf\left(t_j + \tfrac{k}{2}, y_j + \tfrac{1}{2}k_2\right) \\
k_4 &= kf\left(t_{j+1}, y_j + k_3\right)
\end{aligned}
$$

Numerical Methods for ODEs

Code

```
def rk4_method(f, t0, y0, h, t_end):
    t_values = [t0]
    y_values = [y0]
    while t_values[-1] < t_end:
        t = t_values[-1]
        y = y_values[-1]
        k1 = h * f(t, y)
        k2 = h * f(t + h/2, y + k1/2)
        k3 = h * f(t + h/2, y + k2/2)
        k4 = h * f(t + h, y + k3)
        y_new = y + (1/6) * (k1 + 2*k2 + 2*k3 + k4)
        t_new = t + h
        t_values.append(t_new)
        y_values.append(y_new)
    return np.array(t_values), np.array(y_values)
```

Shooting method for BVPs

In numerical analysis, the **shooting method** is a method for solving a boundary value problem by recuding it to an initial value problem.
Example:

$$w''(t) = \frac{3}{2}w^2(t), \quad w(0) = 4, \quad w(1) = 1, \text{to the initial value problem} \tag{6}$$

$$w''(t) = \frac{3}{2}w^2(t), \quad w(0) = 4, \quad w'(0) = s \tag{7}$$

After solving using different methods for s, we get

$$w'(0) = -8 \text{ and } w'(0) = -35.9 \text{ (approximately)} \tag{8}$$

Shooting method for BVPs

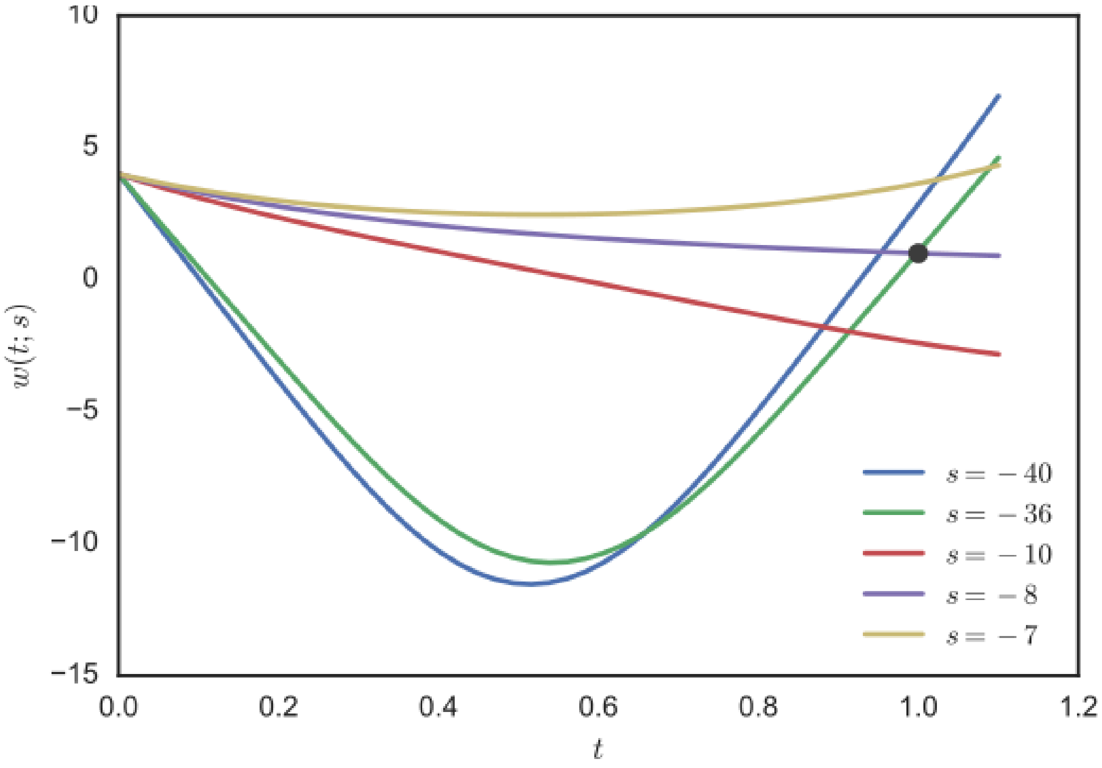


Figure 1: Trajectories $w(t;s)$ for $s = w'(0)$ equal to $-7, -8, -10, -36$ and -40

Shooting method for BVPs

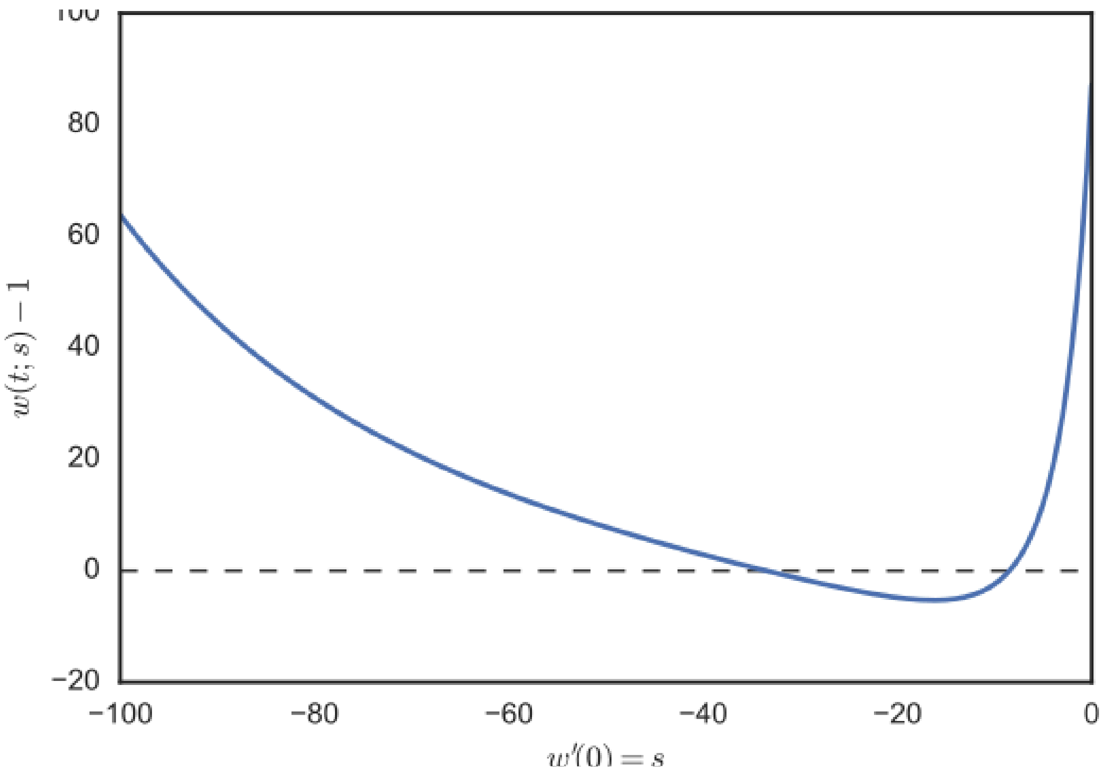


Figure 2: The function $F(s) = w(1; s) - 1$

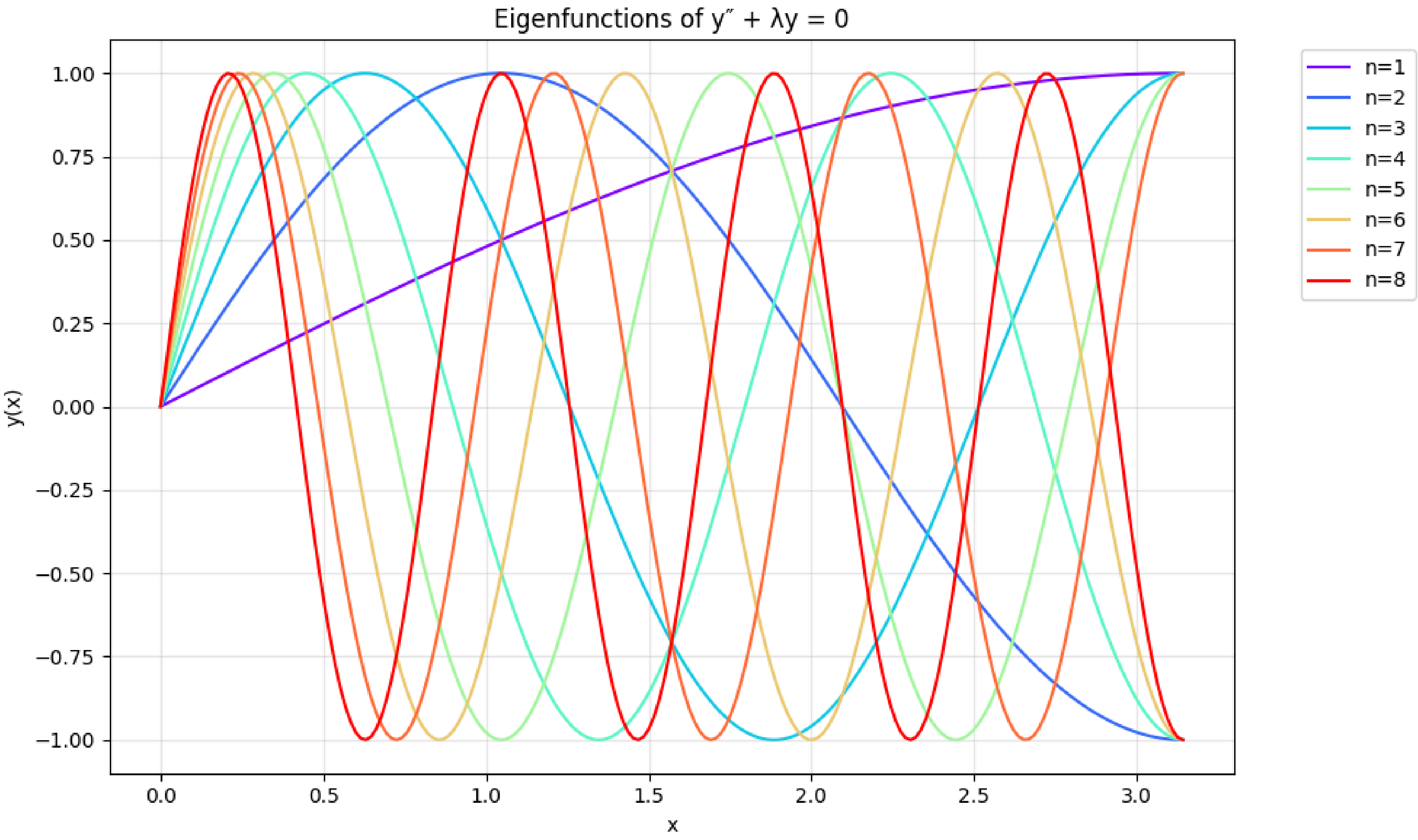
Previous example solved: Problem 3

By hand we got $\lambda_n = \left(\frac{1}{2} + n\right)^2, \ \phi_n(x) = \sin\left(\left(\frac{1}{2} + n\right)x\right)$
In python file: `SimpleSL_example.py` I used **shooting method** with **RK4** to do the same as before, with bisection method, I was able to find the eigenvalues iteratively taking midpoints of interval $[\lambda_{\min}, \lambda_{\max}]$, checking the sign of $y'(\pi)$ and so on..

</> **Code**

```python
def runge_kutta_4(f, y0, t, h):
def shooting_method(lambda_val, x, h):
def find_eigenvalues(n_eigenvalues, x_points):
def compute_eigenfunction(lambda_val, x, h):
```

Previous example solved: Problem 3



Previous example solved: Problem 3

Eigenvalues vs Analytical Values:

```
Eigenvalues vs Analytical Values:
n   Numerical   Analytical   Error (%)
--------------------------------------------------
1   0.250000    0.250000     0.000048
2   2.250000    2.250000     0.000001
3   6.250001    6.250000     0.000016
4   12.250000   12.250000    0.000002
5   20.250011   20.250000    0.000055
6   30.250031   30.250000    0.000103
7   42.250075   42.250000    0.000176
8   56.250180   56.250000    0.000319

Process finished with exit code 0
```

Another simple example

Almost the same equation, but different boundary points:

$$y'' + \lambda y = 0, y(0) = 0, y(1) = 0 \tag{9}$$

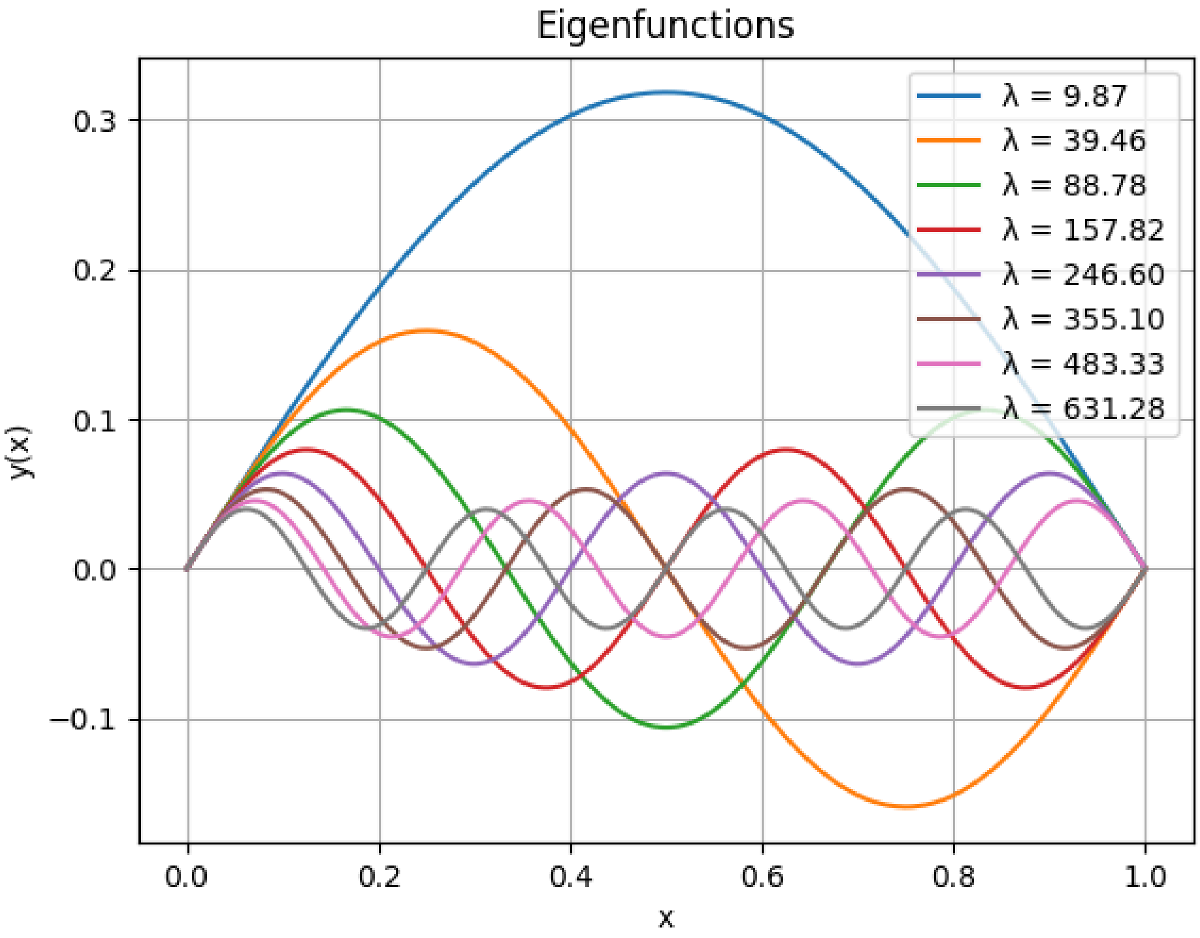
Solution by inspection is:

$$A\cos\left(\sqrt{\lambda}x\right) + B\sin\left(\sqrt{\lambda}x\right) \tag{10}$$

But because of boundary conditions we get: $\sin\left(\sqrt{\lambda}x\right) = 0$ so

$$\lambda_n = \pi^2 n^2,\ \ y_n(x) = \sin(\pi n x) \tag{11}$$

Another simple example



Orthogonality

Proof

1. $\int_0^1 \sin(m\pi x) * \sin(n\pi x) * 1 \, \mathrm{d}x = 0 \quad \text{if } m \neq n$
2. $\sin(A)\sin(B) = \frac{1}{2}[\cos(A - B) - \cos(A + B)]$ applying it $\Rightarrow$
3. $\left(\frac{1}{2}\right) \int_0^1 [\cos((m - n)\pi x) - \cos((m + n)\pi x)] =$

$$= \left(\frac{1}{2}\right) \left[\frac{\sin((m-n)\pi x)}{(m-n)\pi} - \frac{\sin((m+n)\pi x)}{(m+n)\pi}\right]_0^1$$

Since m and n are integers and $m \neq n \Longrightarrow$

$$\sin((m - n)\pi) = \sin((m + n)\pi) = 0, \text{ and we're done.} \tag{12}$$

Nice example:

$$y'' + 3y' + 2' + \lambda y = 0, \; y(0) = 0, \; y(1) = 0. \tag{13}$$

Solution The characteristic equation of that is:

$$r^2 + 3r + 2 + \lambda, \quad \text{with zeros } r_{1,2} = \frac{-3 \pm \sqrt{1 - 4\lambda}}{2} \tag{14}$$

Case 1: If $\lambda < \frac{1}{4}$ then r_1 and r_2 are real and distinct, so the general solution is

$$y = c_1 e^{r_1 t} + c_2 e^{r_2 t} \tag{15}$$

The boundary conditions require that: $c_1 + c_2 = 0 \;\wedge\; c_1 e^{r_1} + c_2 e^{r_2} = 0$

Since the determinant of this system is $e^{r_2} - e^{r_1} \neq 0$, the system has only the trivial solution. Therefore λ is not an eigenvalue of equation 13

Nice example:

Case 2: If $\lambda = \frac{1}{4}$ then $r_1 = r_2 = -\frac{3}{2}$ so the general solution of equation 13 is

$$y = e^{\frac{-3x}{2}}(c_1 + c_2 x) \tag{16}$$

The boundary condition $y(0) = 0$ requires that $c_1 = 0$, so $y = c_2 x e^{\frac{-3x}{2}}$ and the boundary condition $y(1) = 0$ requires that $c_2 = 0$. Therefore $\lambda = \frac{1}{4}$ is not an eigenvalue.

Case 3: If $\lambda > \frac{1}{4}$ then:

$$r_{1,2} = -\frac{3}{2} \pm iw \quad \text{with} \tag{17}$$

$$w = \frac{\sqrt{4\lambda - 1}}{2} \text{ or equivalently, } \quad \lambda = \frac{1 + 4w^2}{4} \tag{18}$$

Nice example:

Case 3(Continued): In this case the general solution of equation 13 is

$$y = e^{\frac{-3x}{2}}(c_1 \cos wx + c_2 \sin wx) \tag{19}$$

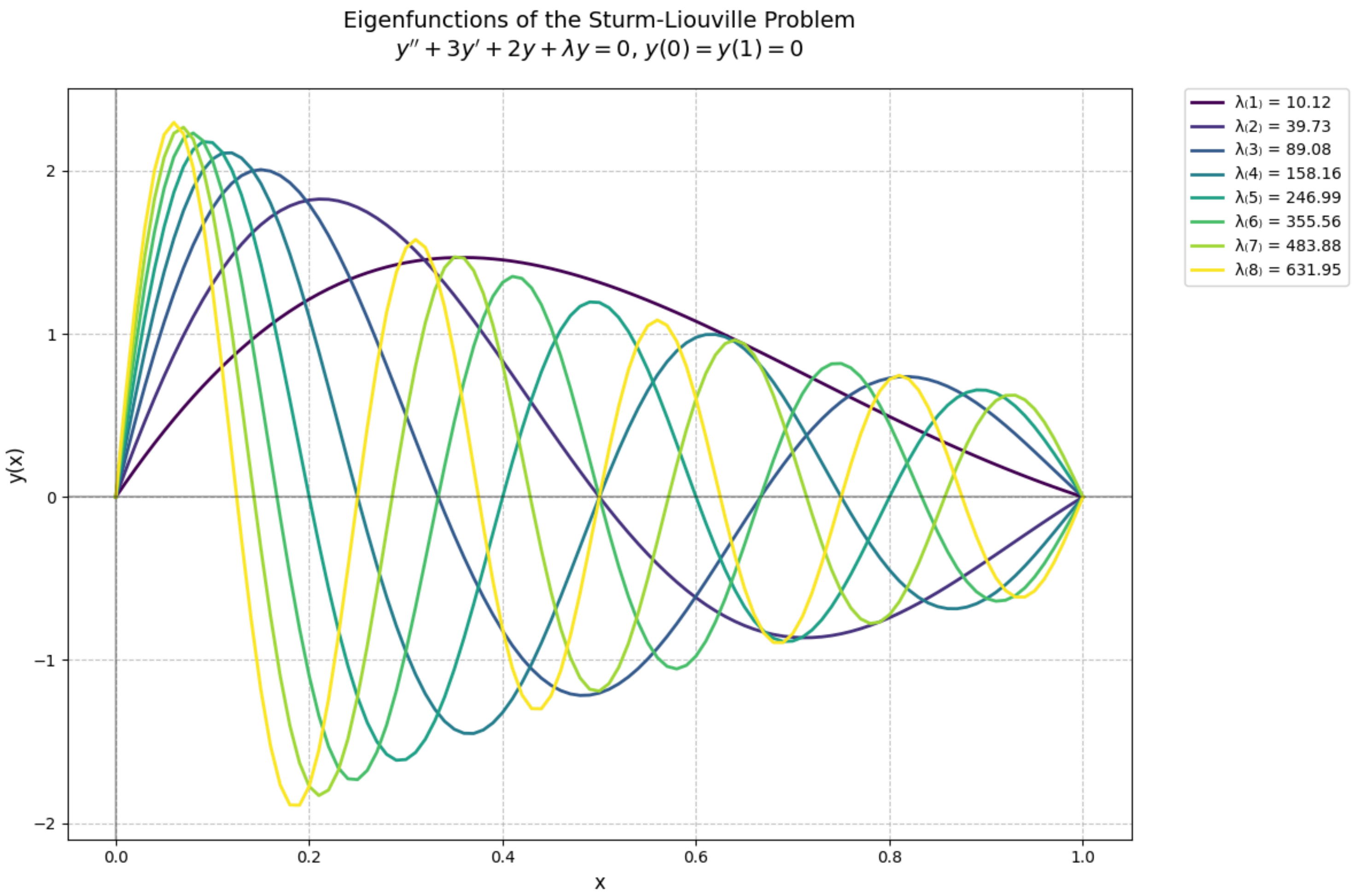
Boundary condition $y(0) = 0$ requires that $c_1 = 0$, so $y = c_2 e^{\frac{-3x}{2}} \sin wx$ which holds with $c_2 \neq 0$ iff $w = n\pi$ where n is an integer.
So the eigenvalues are

$$\lambda_n = \frac{1 + 4n^2\pi^2}{4}, \tag{20}$$

with associated eigenfunctions

$$y_n = e^{\frac{-3x}{2}} \sin n\pi x, \quad n = 1, 2, 3, \ldots. \tag{21}$$

Nice example:



Numerical Solution

Consider the Sturm-Liouville problem:

$$y'' + 3y' + 2y + \lambda y = 0,\ \ y(0) = 0,\ \ y(1) = 0 \tag{22}$$

Shooting Method Approach:

1. Convert to first-order system:

$$\begin{pmatrix} {y_1}' \\ {y_2}' \end{pmatrix} = \begin{pmatrix} y_2 \\ -3y_2 - 2y_1 - \lambda y_1 \end{pmatrix} \tag{23}$$

where $y_1 = y$ and $y_2 = y'$

2. For each λ, solve IVP with initial conditions:

$$y_1(0) = 0,\ \ y_2(0) = 1 \tag{24}$$

Numerical Solution

using RK4 method with step size h:

$$
\begin{aligned}
k_1 &= f(t_n, y_n) \\
k_2 &= f\left(t_n + \frac{h}{2}, y_n + \left(\frac{h}{2}\right)k_1\right) \\
k_3 &= f\left(t_n + \frac{h}{2}, y_n + \left(\frac{h}{2}\right)k_2\right) \\
k_4 &= f(t_n + h, y_n + hk_3) \\
y_{n+1} &= y_n + \left(\frac{h}{6}\right)(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}
\tag{25}
$$

3. Define shooting function:

Numerical Solution

$$F(\lambda) = y_1(1) \tag{26}$$

Eigenvalues occur when $F(\lambda) = 0$

4. Find eigenvalues using bisection:

For interval $[\lambda_L, \lambda_R]$, if $F(\lambda_L)F(\lambda_M) < 0$ where $\lambda_M = \frac{\lambda_L + \lambda_R}{2}$, then eigenvalue exists in $[\lambda_L, \lambda_M]$

5. Initial guesses based on analytical solution:

$$\lambda_n \approx \frac{1 + 4n^2\pi^2}{4} \tag{27}$$

Another Good Example

Legendre equation

$$-(1-x^2)y'' + 2xy' + \lambda y = 0 \text{ on } [-1,1] \tag{28}$$

with boundary conditions $y(-1) = y(1) = 0$

$$\frac{d}{\mathrm{d}x}\left[(1-x^2)\frac{\mathrm{d}y}{\mathrm{d}x}\right] + l(l+1)y = 0 \tag{29}$$

The above form is a special case of the so-called associated Legendre differential equation corresponding to the case $m = 0$. The Legendre differential equation has regular singular points at $-1, 1$, and ∞

Another Good Example

To solve this using the shooting method, we first transform the second-order ODE into a system of first-order ODEs:
Let $v = y'$, then:

$$\begin{aligned} y' &= v \\ v' &= \frac{2xv + \lambda y}{1 - x^2} \end{aligned} \tag{30}$$

This gives us the system:

$$\frac{d}{\mathrm{d}x}[y] = [v] \qquad \frac{d}{\mathrm{d}x}[v] = \left[\frac{2xv + \lambda y}{1 - x^2}\right] \tag{31}$$

Another Good Example

The shooting method converts our boundary value problem into an initial value problem:

- At $x = -1$:
  - We know $y(-1) = 0$ (given boundary condition)
  - We guess $y'(-1) = 1$ (arbitrary non-zero value)

For a given eigenvalue guess λ:

1. Set initial conditions $y(-1) = 0$, $y'(-1) = 1$
2. Integrate the system from $x = -1$ to $x = 1$
3. Check the value of $y(1)$

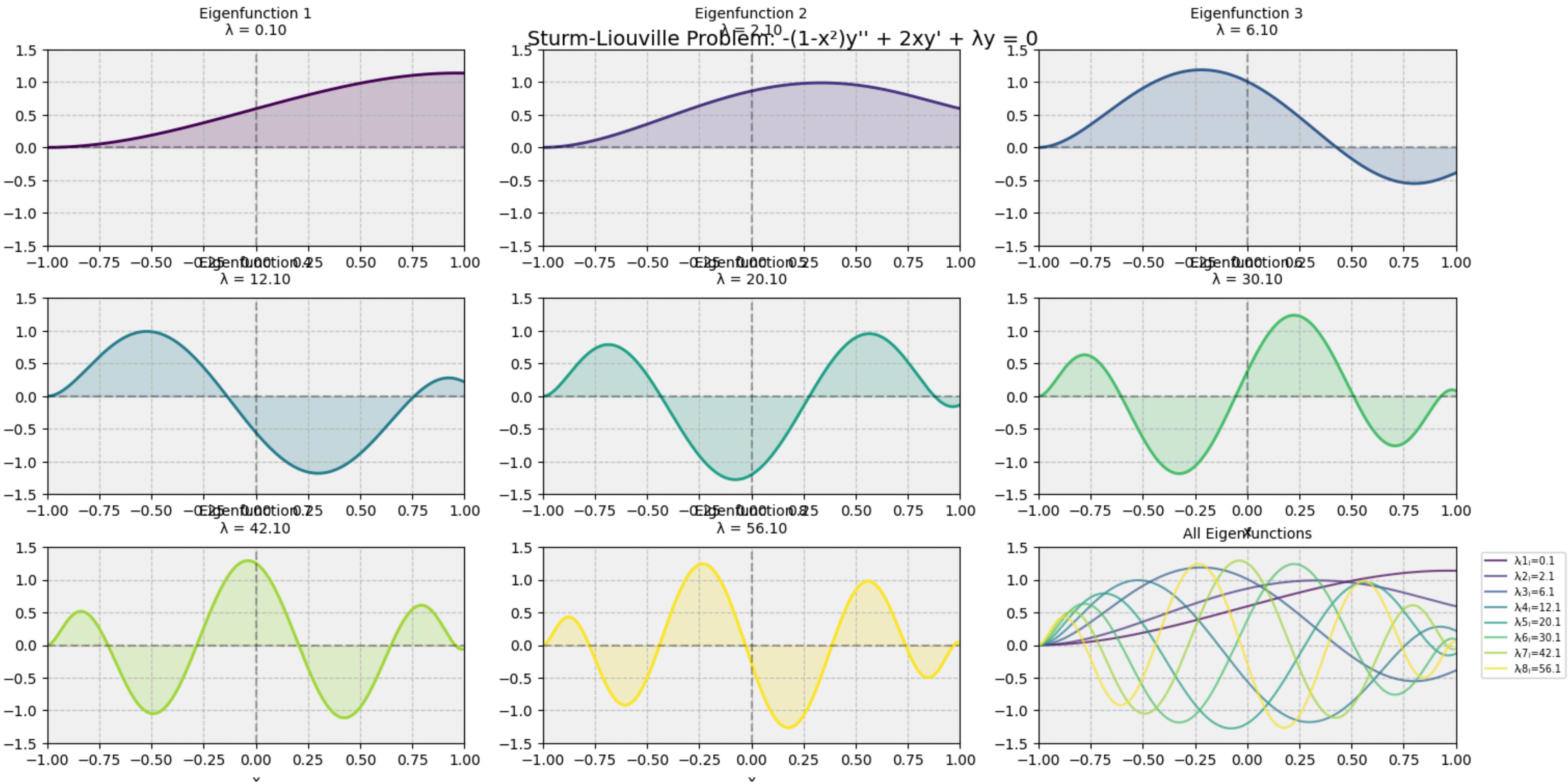
Eigenvalue Search, Define the shooting function: $F(\lambda) = y(1)$ The eigenvalues are the values of λ where $F(\lambda) = 0$, then runge kutta, and then bisection method for finding eigenvalues.

Another Good Example

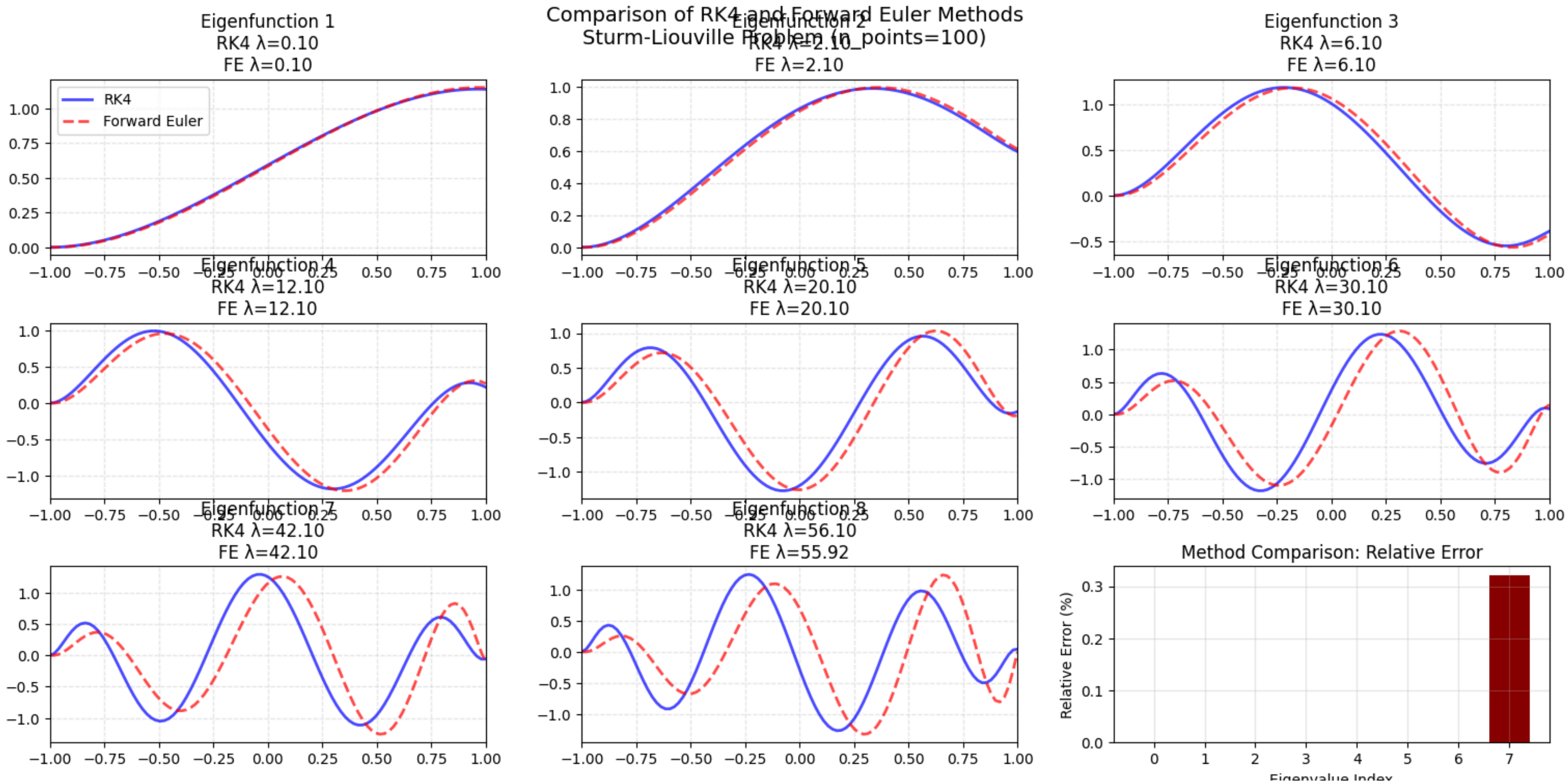
</> Methods

```
def runge_kutta_4(f, x, y, h):

def forward_euler(f, x, y, h):

def compare_methods(n_eigenvalues=8,
n_points=100):

def bisection_eigenvalue(lambda_left,
lambda_right, x_start, x_end, n_points, tol=1e-6,
max_iter=50, shoot_func=shoot):
```

Another Good Example

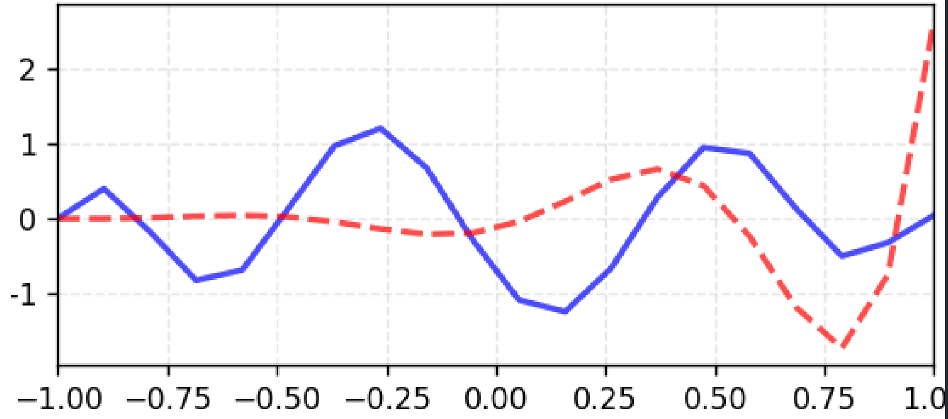


Another Good Example



Another Good Example

Lower the points get, better the difference between methods is:




```
Eigenvalue Analysis:
------------------------------------------------------------
  n      Computed λ    Theoretical λ    Error %
------------------------------------------------------------
  0        0.100000        0.000000   0.100000
  1        2.100000        2.000000    5.0000%
  2        6.100000        6.000000    1.6667%
  3       12.100000       12.000000    0.8333%
  4       20.100000       20.000000    0.5000%
  5       30.100000       30.000000    0.3333%
  6       42.100000       42.000000    0.2381%
  7       56.100000       56.000000    0.1786%

Method Comparison (RK4 vs Forward Euler) with 100 points:
Step size h = 0.020202
----------------------------------------------------------------------
  n             RK4 λ            FE λ        Abs Diff     Error %
----------------------------------------------------------------------
  0        0.100000        0.100000        0.000000      0.0000
  1        2.100000        2.100000        0.000000      0.0000
  2        6.100000        6.100000        0.000000      0.0000
  3       12.100000       12.100000        0.000000      0.0000
  4       20.100000       20.100000        0.000000      0.0000
  5       30.100000       30.100000        0.000000      0.0000
  6       42.100000       42.100000        0.000000      0.0000
  7       56.100000       55.919338        0.180662      0.3220
```

Thanks for your attention

Refs:

The-Shooting-Method - berkeley.edu

Numerical_Study_on_the_Boundary_Value

Problem_by_Using_a_Shooting_Method https://www.math.iitb.ac.in/~siva/

ma41707/ode7.pdf