STEM GAMES 2023

10. lipnja 2023.

1 Day 1 - Numerical warm up

1.1 Analysis and implementation of numerical algorithms

Your task for day 1 is to efficiently implement the numerical algorithms for the problems provided below. Try to implement the methods on such way that they can be used on general problems. For example, you may be asked to solve some ODE and write the program that solves it, but you should try to write the program for the general case that can solve the given ODE.
Your documentation should be clean and your code as readable as possible. You are allowed to use online solvers for checking your solution, but you are not allowed to use implemented solvers in libraries. You are allowed to use mathematical libraries that contain values of functions such as $\sin x$.
If any of your approaches are not self-explanatory, you are to provide the explanation. Theoretical part of solutions, proofs and answers should be written in LaTeX. If possible, you should also graph your numerical solutions with exact solutions for comparison.

1.2 Theory

For each of the functions provided in programming section, deduce the correct (exact) value of the problem (if the solution has a closed form). For every implementation of the numerical algorithms you provide, analyze the errors, prove that the method converges to the solution and explain the numerical procedures you've used. Graph the exact solutions (if you find them) and your solution. If you are unable to find the exact solution compare your solution with the one given by solver in the library you are using.

1.3 Programming

Programming Task 1.1 *Write a program that numerically calculates the value of the function*

$$f(x) = \int_0^x sin(t^2)dt$$

for any $x \in (0, 2]$.

Programming Task 1.2 *Write a program that numerically calculates the value of the function*

$$f(x) = \int_0^x \frac{1}{\sqrt{t}}dt$$

for any $x \in (0, \infty]$.

Programming Task 1.3 *Write a program that numerically calculates the value of the function*

$$f(x) = \int_0^x -ln(t)dt$$

for any $x \in (0, 1]$.

Programming Task 1.4 *Write a program that numerically calculates the value of the function*

$$f(x) = \int_0^x e^t cos(t)dt$$

for any $x \in (0, 5]$.

Programming Task 1.5 *Write a program that numerically calculates the value of the function*

$$f(x) = \int_0^x t^2 ch(t^2) dt$$

for any $x \in (0, 1]$.

Programming Task 1.6 *Write a program that numerically calculates the value of the function*

$$f(x) = \int_0^x sin(\frac{1}{t})\sqrt{t} + cos(x) dt$$

for any $x \in (0, 2]$.

Programming Task 1.7 (*) *Write a program that numerically calculates the value of the function*

$$f(x) = \int_0^x sin(t) sin(\frac{1}{t}) + cos(x) dt$$

for any $x \in (0, 2]$.

Programming Task 1.8 *Write a program that numerically solves the following differential equation*

$$y' = -y$$

with initial condition $y(0) = 0$.

Programming Task 1.9 *Write a program that numerically solves the following differential equation*

$$y' = -100(y - cos(x)) - sin(x)$$

with initial condition $y(0) = 0$.

Programming Task 1.10 *Write a program that numerically solves the following differential equation*

$$y' = 2y(1 - \frac{y}{1000})$$

with initial condition $y(0) = 100$.

2 Day 2 - It gets tricky

The task for day 2 is dealing with several theoretical and numerical problems involving a special kind of ODEs. While the theoretical part isn't too connected to day 3, the numerical part is. All of your solutions should be in general forms. For example, you may be asked to solve an equation, so you should write your code in such way that it can solve any equation of the given type. There is no limit to the amount of methods you can submit. You also may submit more than one method for sovling some problem, but only the best one will be considered as a solution for day 2 problems. If possible, you should also graph your numerical solutions with exact solutions for comparison.

IMPORTANT: After day 2, your code will be **locked** for day 3. You will not be allowed to modify your code, but you will still be allowed to view it and run it. Due to this restriction, it is crucial that the code submitted for day 2 is independent of its input, i.e., your code may be solving a linear system, but the program has to be able to accept the input of the system without modifying the code. Not all programming tasks need to be solved for day 3, but having more programming tasks solved increases your chance of success on day 3.

Eigenfunction problems for ODEs

In this task you will study the solution(s) of equations of the inhomogeneous form

$$Ly(x) = f(x),$$

where $f(x)$ is a given function in one variable and L represents a given linear differential operator acting upon the function $y(x)$. The general solution to this problem is the most general function $y(x)$ that satisfies the equation. Furthermore, boundary conditions for $y(x)$ need to be satisfied, for example at the limits $x = a$ and $x = b$. Problems of this type have interesting historical and everyday importance.[1] In general, unless $f(x)$ is both specific and simple, finding the solution can be difficult. The approach you will be practicing uses the fact that L is linear and builds the required solution $y(x)$ as a superposition of some functions $y_i(x)$ that each individually satisfy boundary conditions. In particular, the functions y_i you will be looking for will satisfy the equations

$$Ly_i(x) = \lambda_i y_i(x),$$

where λ_i is a constant. We call λ_i an eigenvalue and $y_i(x)$ an eigenfunction of the operator L. From this point on, consider that the abstract problem described above is given in the Hilbert space of functions. If you are not familiar with terms related to Hilbert spaces, now is a good time to check them out. Pay attention to:

- Hilbert space
- orthogonal basis
- dot product
- Hermitian operator
- harmonic oscillator (as an example).

[1] `https://en.wikipedia.org/wiki/Sturm_Liouville_theory`

One of the most important applications of Hermitian operators is the study of *Sturm-Liouville equations*. Their general form is

$$p(x)\frac{d^2y}{dx^2} + r(x)\frac{dy}{dx} + q(x)y + \lambda\rho(x)y = 0,$$

where

$$r(x) = \frac{dp(x)}{dx}$$

and q, p and r are real functions of x. Notice that in the terms of the operator theory discussed above, we can rewrite the equation as:

$$Ly = \lambda\rho(x)y.$$

By using the fact that $r(x) = p'(x)$ we can rewrite the entire equation as:

$$(py')' + qy + \lambda\rho y = 0,$$

while the Sturm-Liouville operator can be written as

$$L \equiv -\left[\frac{d}{dx}\left(p(x)\frac{d}{dx}\right) + q(x)\right].$$

The following theoretical tasks are considering mostly the Sturm-Liouville equations.

2.1 Theory

Task 2.1 *Show that the linear operator* $L = \frac{d^2}{dt^2}$ *is self-adjoint. Then, determine the required boundary conditions for the given operator to be Hermitian over some interval* $[t, t+\tau]$.

Task 2.2 *Let* $[a, b]$ *be some interval in* $\mathbb{R}$. *Show that the eigenfunctions of an Hermitian operator form a complete basis set over this interval. Furthermore, show that the all eigenvalues of Hermitian operator are real.*

Task 2.3 *Find the conditions under which the Sturm-Liouville operator is Hermitian over some interval* $[a, b]$.

Task 2.4 *Consider the Sturm-Liouville problem:*

$$y'' + q(x)y = \lambda y$$

with boundary conditions $y(0) = y(k) = 0$ *on some interval* $[0, k]$. *Prove that the zeros of a linear combination of the first* n *eigenfunctions for this problem divide the interval* $[0, k]$ *into at most* n *parts.*

Task 2.5 *Consider some set of functions* $\{h(x)\}$ *on* $\mathbb{R}$ *such that* $h(x) \to 0$ *as quickly as* $f(x) = \frac{1}{x}$ *when* $x \to \pm\infty$. *Determine whether each of the following operators is Hermitian when acting upon* $\{h(x)\}$

- $\frac{d}{dx} + x$
- $-i\frac{d}{dx} + x^2$
- $\frac{d^2}{dx^2} + \sin(x)$

Task 2.6 *Transform the following equations into the Sturm-Liouville form:*

1. $xy'' + (1-x)y' + ay = 0$
2. $y'' + \dfrac{\sin^2 x}{\cos^4 x + \cos x \sin^3 x} y' + e^{3x^2+1} y + 3\ln(x) y = 0$

Task 2.7 *Assume once again that you have been given the nonhomogeneous equation:*

$$Ly(x) = f(x).$$

If L is some general Hermitian operator, prove that the solution of the equation can be found as:

$$y(x) = \int_a^b \left\{ \sum_{n=0}^{\infty} \left[\frac{1}{\lambda_n} \hat{y}_n(x) \hat{y}_n^*(z) \right] \right\} f(z)\, dz = \int_a^b G(x,z) f(z)\, dz.$$

Here $\hat{y_n}$ represents the nth normalized eigenfunction of L. Function G is usually denoted as Green's function. HINT: See task 2.

Task 2.8 *For the following equations with the given boundary conditions, find the Green functions. Then, solve the equation for the given right side. (The second equation gives extra points).*

- $y'' + 0.25y = f(x)$ *with boundary conditions* $y(0) = y(\pi) = 0$.
- $y'' + 4y' + 4y = q(x)$ *with boundary conditions* $y(0) = 3$ *and* $y(\pi) = 0$.

Solve the first equation for $f(x) = \sin(3x)$ and $f(x) = x$. Then solve the second equation for $q(x) = \tan x$ and $q(x) = \frac{x^2}{2}$. Note that you can find the Green functions before solving the equations.

Task 2.9 *Let $[a,b]$ be some interval and let $y_n(x)$ be real eigenfunctions of Sturm-Liouville equation:*

$$(py')' + qy + \lambda \rho y = 0.$$

Furthermore, let p, q and ρ be continuously differentiable and assume that p does not change its sign on$[a,b]$. Let $a \le x_1 \le x_2 \le b$. Show that:

$$(\lambda_n - \lambda_m) \int_{x_1}^{x_2} \rho y_n y_m \, dx = [y_n p y_m' - y_m p y_n'] \Big|_{x_1}^{x_2}.$$

Next, deduce that if $\lambda_n > \lambda_m$ then $y_n(x)$ must change sign between two successive zeros of $y_m(x)$. Hint: Sketch the first few eigenfunctions for the system:

$$y'' + \lambda y = 0$$

with boundary conditions $y(0) = y(\pi) = 0$ and Legendre polynomials $P_n(x)$ for $n = 2, 3, 4, 5$.

Task 2.10 *By using Legendre polynomials and their properties, find the solution of:*

$$(1-x^2)y'' - 2xy' + by = f(x)$$

valid on $[-1,1]$ and finite at the origin. Express the solution in terms of Legendre polynomials. Furthermore, if $b = 2023$ and $f(x) = 4x^3$ find the explicit solution and verify it.

Task 2.11 *Assume you are observing the particle in a parabolic potential well. Its natural andular frequency has oscilation of* $\frac{1}{2}$*. At the moment* $t = 0$ *the particle passes through origin with velocity* v*. It is then subjected to an additional acceleration on the following way:*

- *the accleration is equal to* 1 *for* $0 \leq t \leq \pi/2$
- *the acceleration is equal to* -1 *for* $\pi/2 < t \leq \pi$*.*

After this period of time ends, the particle is once again at the origin. Show that:

$$v = -\frac{8}{\pi}\sum_{m=0}^{\infty}\frac{1}{(4m+2)^2 - 0.25} \approx -0.81$$

Task 2.12 *Consider the differential operator defined as below:*

$$Ly = -\frac{d}{dx}(e^x\frac{dy}{dx}) - \frac{1}{4}e^x y.$$

Determine the eigenvalues λ_n *of the problem:*

$$Ly_n = \lambda_n e^x y_n$$

on the interval $[0, 1]$ *with boundary conditions* $y(0) = 0$ *and* $y'(1) + 0.5y(1) = 0$*. Find the corresponding unnormalised eigenfunctions* y_n *and weight function* $\rho(x)$ *with respect to which* y_n *are orthogonal. (You can think of this part as selecting suitable normalization for* y_n*).*

Task 2.13 *Repeat the task 12 for the equation:*

$$Ly_n = \lambda_n \sinh(x)$$

Task 2.14 *Linear operator:*

$$L \equiv \frac{1}{4}(1+x^2)^2\frac{d^2}{dx^2} + \frac{1}{2}x(1+x^2)\frac{d}{dx} + a$$

acts upon functions in space $C^2([0,1])$ *which vanish at the end points of the interval.*

- *Show that* L *is Hermitian with respect to the weight function* $(1+x^2)^{-1}$*. Can you find another weight function that still preserves Hermitian property of the operator?*
- *Find two eigenfunctions of the equation* $Ly = \lambda y$*.*

2.2 Programming

Programming Task 2.15 *Solve the equations from task 2.6. on segment* $[a, b]$ *with boundary conditions* $y(a) = y(b) = 0$*. Then solve the equations for* $y'(a) = y'(b) = 0$*. You can vary* a *and* b *as long as* $a \neq b$*. How different are the solutions?*

Programming Task 2.16 *Solve both variants of the first equation in task 2.8 (for* $f(x) = \sin(3x)$ *and* $f(x) = x$*). Then, implement the Green function solution and compare your numerical approximation with Green function solutions. How many terms of Green function solutions you require for fixed precision? Estimate the errors when finding the solution numerically via Green function.*

Programming Task 2.17 *Consider a linear differential operator* $L = \frac{d}{dx} + \alpha$ *where* α *is some real constant. Implement the program that solves the eigenvalue problem for such operator on some given segment* $[a, b]$.

Programming Task 2.18 *Consider a linear differential operator* $L = \frac{d^2}{dx^2} + \alpha$ *where* α *is some real constant. Implement 3 programs that solve Dirichlet, Neumann and mixed boundary conditions for such problem.*

Programming Task 2.19 *Using the Green function solution formula, solve the differential equation:*

$$y'' + 9y = \frac{1}{\sqrt{x^2 + 12x + 40}}$$

with boundary conditions $y(0) = y'(0)$ *and* $y(3) = y'(3)$.

Programming Task 2.20 *Solve the task 2.11 numerically and implement the programm that can vary the initial and boundary conditions.*

Programming Task 2.21 (*) *Create a program that solves the general Sturm-Liouville equation with given boundary conditions. Test the program on some equation with known exact solution and analyze the error within your solutions.*

Optional tasks:
Eigenvalue problems for differential equations can be extremely time consuming. Typically, we are only interested in lowest or highest eigenvalue of the Sturm-Liouville equation. If possible, find and implement a way to estimate highest/lowest eigenvalues of Sturm-Liouville equations you are dealing with.

3 Day 3 - Fatal flaws

Construct a nontrivial example(s) for which every piece of your code from day 2 fails to do its job. While doing it, keep in mind the following:

- The code still has to compile.
- Changing your code is not allowed.
- Trivial answers such as division by zero, bug exploits, language specific exploits or memory leaks will not be considered.
- Provide the theoretical explanation on why the code fails. The more general example you can give the better.
- Analyze errors in your example.
- Avoid the examples that are too complicated only for the sake of failure (i.e. extremely large numbers as coefficients).
- Extra points will be awarded for any graphic explaining of where the problem occurs.
- You **don't** need to use every piece of locked code from day 2, but the more pieces you use the better.